Zbornik radova 15 (23)

SELECTED TOPICS ON IMAGE PROCESSING AND CRYPTOLOGY

Editor:
Miodrag Mihaljević

Matematički institut SANU

Издавач: Математички институт САНУ, Београд
серија: Зборник радова, књига 15 (23)
За издавача: Богољуб Станковић, главни уредник серије
Уредник свеске: Миодраг Михаљевић
Технички уредник: Драган Благојевић
Штампа: “Академска издања”, Земун
Штампање завршено децембра 2012.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

003.26(082)
004.056.55(082)
SELECTED Topics on Image Processing And Criptology / editor Miodrag Mihaljević. - Beograd : Matematički institut SANU, 2012 (Zemun : Akademska izdanja). - 186 str. : ilistr. ; 24 cm. - (Zbornik radova / Matematički institut SANU ; knj. 15 (23))

Tiraž 350. - Str. 3: Preface / Miodrag Mihaljević. - Bibliografija uz svaki rad.

ISBN 978-86-80593-47-0
1. Mihaljević, Miodrag, 1955– [уредник] [аутор додатног текста]
а) Криптологија – Зборници b) Криптографија – Зборници
COBISS.SR-ID 195232268

PREFACE

This volume is dedicated to a number of issues on applications of the mathematics to the area of information-communications technologies (ICT) and addresses one of the hot topics within ICT – information processing, and particularly image processing and cryptology. Also note that image processing is a background of biometric authentication for information security where cryptology provides the basic components for developing security mechanisms. The purpose of this volume is to draw attention to certain methods relevant as a background or as particular elements for design of information security mechanisms.

The chapters of this volume illustrate employments of different mathematical topics for certain real-life issues. as well as point out to novel mathematical challenges originating from information processing problems.

This volume entitled "Selected Topics on Image Processing and Cryptology" consists of the following three chapters: (i) "Shape Descriptors for Image Analysis" by Joviša Žunić, (ii) "The coverage model and its use in image processing" by Nataša Sladoje and Joakim Lindblad, and (iii) "On Certain Approaches for Analysis and Design of Cryptographic Techniques for Symmetric Encryption and Key Management" by Miodrag Mihaljević.

The presentations given in this volume also originate from over 100 top-level references of the authors (which have been cited over 1000 times up to now) accumulated through a number of research projects within Mathematical Institute, Serbian Academy of Sciences and Arts (SANU) where the authors of the chapters were very active participants. Also, it is important to point out that these results are also closely related to extensive international collaboration with a number of leading universities and institutes in U.K., Sweden and Japan.

We believe that this book yields a useful overview of a number of techniques and could serve as a reference background for further research activities, as well as an introduction to the addressed topics and an encouragement of beginners interested in image processing or cryptology.

Miodrag Mihaljević

Contents

Joviša Žunić *

SHAPE DESCRIPTORS FOR IMAGE ANALYSIS

Abstract. We give an overview of the shape based techniques used in object matching, object identification and object classification tasks. We distinguish between the area based methods, which use all the shape points, and boundary based methods, which use boundary information only. We also discuss a recent 'multi-component shape' approach. This approach considers a group of objects as a single but compound object. The idea is already shown to be very efficient in a wide spectrum of applications.

Illustrative examples are provided, including those related to personal signature identification and outliers detections, which have, pretty much, obvious and straightforward applications in security and crime prevention related tasks.

Mathematics Subject Classification (2010): Primary: 68U10, 94A08; Secondary 62H30.

Keywords: image processing, computer vision, shape analysis, object identification, data security.

* *University of Exeter, College of Engineering, Mathematics and Physical Sciences, Exeter EX4 4QF, U. K.*
and
Mathematical Institue SANU, Kneza Mihaila 36, Belgrade, Serbia

Contents

1. Motivation and Problem Description

Image technologies have developed rapidly. A huge amount of images and image related data are available in different domains: medicine, biology, industry, geology, astronomy, crime prevention, security, etc. Different objects appear on images and they should be recognized, classified, or identified. Working in object space, i.e. compar ing object pairwise, is shown to be inaccurate and computationally expensive. It has turned out that a better idea is to map objects of interest onto a set of numbers (a vector in R^d) and then perform searching in this space (e.g., in a subset of R^d). For such mapping we need some object characteristic which can be reasonably easily and efficiently quantified by numbers. One of such characteristics can be the color of the object or its texture, for example. The shape is another object characteristic, which allows a spectrum of numerical characterizations. Also, the shape, as an object characteristics, has a big discrimination capacity. I.e., objects of different kind, very often, can be distinguished by their shapes. In Figure 1(a) an original diatom image is given. For a further processing a preprocessing is needed. In Figure 1(b) a shape of diatom is presented, while boundary shape and interior details are in Figure 1(c).

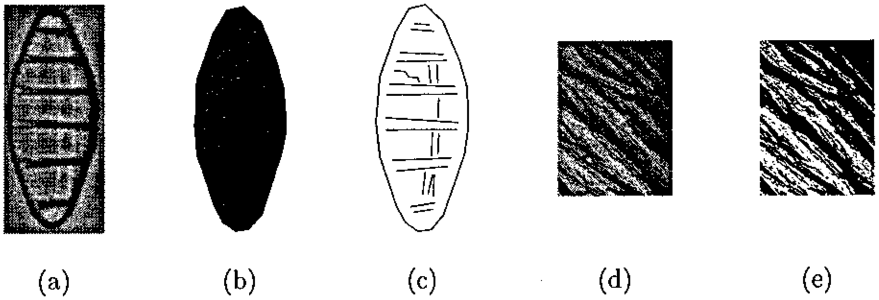


FIGURE 1. Diatom images: (a) original image; (b) whole object shape; (c) boundary shape and shape of internal contours.
Texture image: (d) original image, (e) original image thresholded.

Figure 1(d) presents a texture. A corresponding black-white image is in Figure 1(e). There is no a clear shape of such extracted texture.

We illustrate the basic idea by simple shape examples in Figure 2. Shapes in Figure 2(a)-(c) are rotationally symmetric and by this property they can be distinguished from the shapes in Figure 2(d,e). The question is: *How to define the function* $\mathbf{D}_1(S)$ *which maps planar shapes into the interval* $[0,1]$, *such that it assigns the value* 1 *to rotationally symmetric shapes, and assigns lower values to the shapes in Figure* 2(d) *(let say* 0.8*) and in Figure* 2(e) *(let say* 0.1*)?*

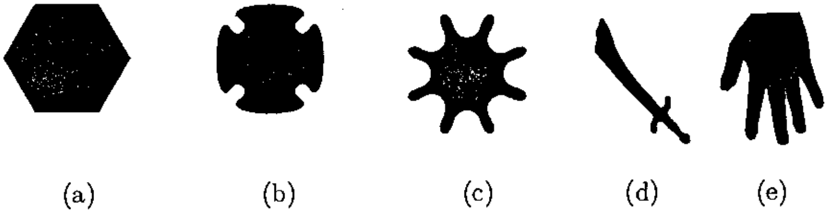


FIGURE 2. Shape examples.

Such a defined function $\mathbf{D}_1(S)$ could be called a 'symmetricity' measure. Further, because $\mathbf{D}_1(S)$ does not distinguish among the shapes in Figure 2(a-c), another descriptor should be considered. Since the shape in Figure 2(a) is convex, and since the shape in Figure 2(b) is 'more convex' than shape in Figure 2(c), a 'convexity' measure (i.e., another function) $\mathbf{D}_2(S)$, will do. $\mathbf{D}_2(S)$ is expected to assign 1 to convex shapes and smaller values for 'less convex' shapes (e.g., 0.92 to the shape in Figure 2(b) and 0.75 to the shape in Figure 2(c)). Such defined function, very likely, would distinguish between the shapes in Figure 2(d) and Figure 2(e) but is not clear would it distinguish among shapes in Figure 2(b) and Figure 2(d). It is difficult do judge which of them is 'more convex'. To overcome such problem, another descriptor, e.g., shape 'linearity', could be involved. A linearity measure

$D_3(S)$ should assign a high value to the shape in Figure 2(d) (let say 0.9) and a small and similar values for the rest of shapes (e.g. all close to 0.1).

This is a basic idea how to use shape descriptors and corresponding measures to be able to distinguish between shapes/objects. Of course, in most cases a single descriptor and a single measure are not enough, and several of them should be combined. For example, the shape in Figure 2(e) can be separated from the others as a shape with low $D_1(S)$ and $D_3(S)$ values (e.g. with low both symmetricity and linearity measures).

2. Introduction

Shape descriptors are a powerful tool used in wide spectrum of computer vision and image processing tasks such as object matching, classification, recognition and identification. Many approaches have been developed [40]. There are a number of generic shape descriptors that are capable of providing a high dimensionality feature vector that accurately describes specific shapes (for example, Fourier descriptors and moment invariants). Alternatively, other descriptors describe some single characteristic that is present over a variety of shapes, such as circularity [30], ellipticity, rectangularity, triangularity [33], rectilinearity [50], complexity [29], mean curvature [21], symmetry [48], etc. Even for a single characteristic of shapes there often exist many alternative measures which are sensitive to different aspects of the shape. Very likely, the shape convexity is a shape property with the largest number of different methods defined for its evaluation–see [3, 20, 31, 36, 37, 41, 51]. The need for alternative measures is caused by the fact that there is no a single shape descriptor which is expected to perform efficiently in all possible applications.

Generally speaking, there are two approaches to analyze shapes: boundary based (which use the information from boundary points only) and area based ones (which use all the shape points). It could be said that, in the past, more attention has been given to the area based methods. The area based methods are more robust (e.g. with respect to noise). Although not mentioned often, an additional reason for a larger number of methods that are based on 'interior' shape points, rather than methods based on boundary points, is that area based methods are usually simpler to compute. For example, to estimate accurately the area of a given shape, it is sufficient to enumerate the number of pixels inside the shape [17], while the perimeter estimation is not a straightforward task. Depending on particular situation and conditions assumed different methods have to be used [7, 38].

Another example would be geometric (area) moment invariants [16]; these are easily and accurately computable from the corresponding object image, while their boundary based analogues involve computation of path integrals, which are not simple to be estimated from discrete data, which are mainly used in image processing and computer vision applications. On the other side, the boundary based methods are more suitable for a high precision computer vision and image processing tasks (person identification, for example). They are able to cope much easier with objects with partially extracted boundaries or with partially occluded objects. Robustness is a very desirable property when we work with low quality data (e.g., noisy images or low resolution images), but recently, due to progress in image technology, high quality data can be provided, and the use of boundary based methods becomes highly acceptable in many applications. In addition, boundary based methods could have a much lower time complexity because shape boundaries are represented by a significantly smaller number of pixels than complete shapes are. Of course, there are methods which cannot be classified either as boundary based or volume (area) based ones. For example, a very popular shape measure, the shape compactness

$$\mathbf{C}_{st}(S) = \frac{4 \cdot \pi \cdot \text{Area\_of\_}S}{(\text{Perimeter\_of\_}S)^2}$$

obviously uses both boundary and interior information. This quantity indicates how much a given shape differs from a perfect circular disc, which is understood as the most compact shape. Accordingly, the highest possible compactness (equal to 1) is assigned to circular disc. Finally, there are methods which use only information from specific points (shape corners, for example) or specific boundary parts (e.g., parts belonging to the convex hull of the shape considered).

Here, we focus on shape analysis techniques based on the use of a set of suitably selected shape descriptors/measures. Generally speaking, a shape measure is a quantity which relates to a particular shape characteristic. More formally, a certain shape measure $\mathbf{D}(S)$ (related to a certain shape descriptor) maps a given planar shape S into a real number. In order to be applicable in object classification, recognition or identification task, any shape measure is expected to be invariant with respect to similarity transformations (translation, rotation, and scaling). Also, shape measures are preferred to be given in a normalized form. An easiest way to achieve a normalized form is to apply a scaling transformation which would preserve that $\mathbf{D}(S)$ varies through the interval $[0,1]$ (or even better through the interval $(0,1]$) while S varies through the set of bounded compact planar regions.

Thus, common desirable properties of a given shape measure $\mathbf{D}(S)$ are:

(a) $\mathbf{D}(S) \in [0,1]$
(b) $\mathbf{D}(S) = 1$
emphif and only if S satisfies a certain property (here called a shape descriptor) for which, actually, the shape measure $\mathbf{D}(S)$ is designed.
(c) $\mathbf{D}(S)$ is invariant with respect to the similarity transformations.
(d) For any $\delta > 0$ there is a shape S such that $\mathbf{D}(S) < \delta$
(e.g., 0 is the best possible lower bound for $\mathbf{D}(S)$.)

The paper is organized as follows. In the next section we consider area based shape descriptors. Section 4 is related to boundary based shape descriptors, while Section 5 relates to the recent concept of multi-component shapes. Concluding remarks are in Section 6.

3. Area Based Shape Descriptors

As mentioned, area based shape analysis methods (including shape descriptors based approaches) are, so far, mostly studied in literature. These methods are expected to be robust (e.g., with respect to noise or with respect to narrow boundary intrusions), and, because of that, they are very suitable when working with low quality data or with a low resolution images. In this section we discuss some of area based descriptors which are in a frequent use.

3.1. Geometric Moments and Moment Invariants. Being theoretically well founded and well understood, moments based techniques are very popular and very useful in the image processing and computer vision tasks. A number of methods were developed. We proceed with a short overview.

For a given planar shape S its geometric (area) (p,q)-moment $m_{p,q}(S)$ is defined as

$$m_{p,q}(S) = \iint_S x^p y^q dx\, dy. \tag{3.1}$$

The order of $m_{p,q}(S)$ is $p+q$. Trivially, $m_{0,0}(S)$ equals the area of S. When work in discrete space, i.e., when a real shape S is represented with its digitization $dig(S)$, then $m_{p,q}(S)$ is approximated as

$$m_{p,q}(S) = \iint_S x^p y^q dx\, dy \approx \sum_{\text{pixel } (i,j) \text{ belongs to } \text{dig}(S)} i^p \cdot j^q$$

if the pixel size is assumed to be 1 × 1.

Obviously, $\sum_{\text{pixel } (i,j) \text{ belongs to dig}(S)} i^p \cdot j^q$ is very simple to compute (only summations and multiplications are needed) and the approximation in (3.1) is very accurate [22]. There are also methods for a fast computation of such an approximation–e.g., [19,24]. These are reasons why moments are used to define very common features in image processing and computer vision applications. For example, one of the basic shape features, as it is the shape position, is usually expressed in terms of moments. Precisely, the position of a given shape S is described by the shape centroid $(x_c(S), y_c(S))$ which is defined as

$$(x_c(S), y_c(S)) = \left(\frac{m_{1,0}(S)}{m_{0,0}(S)}, \frac{m_{0,1}(S)}{m_{0,0}(S)}\right). \tag{3.2}$$

Since moments $m_{p,q}(S)$ are not translation invariant (e.g., if S moves the corresponding moments change) it is suitable to consider the central moments $\overline{m}_{p,q}(S)$ which are defined as

$$\overline{m}_{p,q}(S) = \iint_S (x - x_c(S))^p\, (y - y_c(S))^q\, dx\, dy$$

and which are translation invariant by definition.

Further, because isometric objects could appear on an image as object of different size (depending on their position with respect to the camera) it is suitable to have object features which are scaling invariant. Since $\overline{m}_{p,q}(S)$ are not scaling invariant, it is convenient to involve, so called, normalised moments. A normalized moment

$\mu_{p,q}(S)$ is defined as

$$\mu_{p,q}(S) = \frac{\overline{m}_{p,q}(S)}{m_{0,0}(S)^{1+\frac{p+q}{2}}}.$$

It is easy to verify that the normalized moments $\mu_{p,q}(S)$ do not change if a given shape S is scaled for a factor $\mathbf{r}$. In other words, if S is replaced with $\mathbf{r} \cdot S = \{(\mathbf{r} \cdot x, \mathbf{r} \cdot y) \mid (x,y) \in S\}$, then $\mu_{p,q}(S) = \mu_{p,q}(\mathbf{r} \cdot S)$.

Finally, in many object classification tasks, identical (or very similar) objects have to be grouped together. Since the objects presented on an image may be placed arbitrarily, descriptors which do not depend on the object position and object orientation are needed for such a grouping. This means that, apart from being translation and scaling invariant, we need shape descriptors which are rotationally invariant, as well. In his seminal work [16], Hu has introduced a set of, so called, algebraic invariants. These invariants are listed below:

$$\begin{aligned}
I_1 &= \mu_{2,0} + \mu_{0,2}\\
I_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4 \cdot (\mu_{1,1})^2\\
I_3 &= (\mu_{3,0} - 3 \cdot \mu_{1,2})^2 + (3 \cdot \mu_{2,1} - \mu_{0,3})^2\\
I_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2\\
I_5 &= (\mu_{3,0} - 3 \cdot \mu_{1,2}) \cdot (\mu_{3,0} + \mu_{1,2}) \cdot [(\mu_{3,0} + \mu_{1,2})^2 - 3 \cdot (\mu_{2,1} + \mu_{0,3})^2]\\
&\quad + (3 \cdot \mu_{2,1} - \mu_{0,3}) \cdot (\mu_{2,1} + \mu_{0,3}) \cdot [3 \cdot (\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2]\\
I_6 &= (\mu_{2,0} - \mu_{0,2}) \cdot [(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2]\\
&\quad + 4 \cdot \mu_{1,1} \cdot (\mu_{3,0} + \mu_{1,2}) \cdot (\mu_{2,1} + \mu_{0,3})\\
I_7 &= (3 \cdot \mu_{2,1} - \mu_{0,3}) \cdot (\mu_{3,0} + \mu_{1,2}) \cdot [(\mu_{3,0} + \mu_{1,2})^2 - 3 \cdot (\mu_{2,1} + \mu_{0,3})^2]\\
&\quad + (\mu_{3,0} - 3 \cdot \mu_{1,2}) \cdot (\mu_{2,1} + \mu_{0,3}) \cdot [3 \cdot (\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2].
\end{aligned}$$

Because the normalized moments $\mu_{p,q}$ were used, the quantities $I_1, I_2, \ldots, I_7$ are translation and scaling invariant by definition. It also can be verified that $I_1, I_2, \ldots, I_7$ are invariant with respect to rotations. For more details and recent developments we refer the reader to [12, 13].

3.2. Shape Orientation. Geometric moments are also used to determine the shape orientation, which is, together with shape position (usually defined by the shape centroid (3.2)), a necessary part of an image normalization procedure. The most standard method for the computation of the shape orientation is based on the, so called, axis of the least second moment of inertia [18, 40]. The axis of the least second moment of inertia is the line which minimizes the integral of the squared distances of the points (belonging to the shape) to the line. The integral which should be minimized is

$$I(\alpha, S, \rho) = \iint_S r^2(x, y, \alpha, \rho)\, dx\, dy$$

where $r(x, y, \alpha, \rho)$ is the perpendicular distance from the point $(x, y) \in S$ to the line given in the form

$$X \cdot \sin\alpha - Y \cdot \cos\alpha = \rho.$$

It is easy to check the axis of the least second moment of inertia passes through the shape centroid $(x_c(S), y_c(S))$ (see (3.2)). So, if the shape S' is the translation of S by the vector

$$-\overrightarrow{\left(\frac{m_{1,0}(S)}{m_{0,0}(S)}, \frac{m_{0,1}(S)}{m_{0,0}(S)}\right)} = -\overrightarrow{(x_c(S), y_c(S))}$$

then the centroid of S' coincides with the origin. This allows us to set $\rho = 0$ and proceed with the minimization of $I(\alpha, S', \rho = 0)$ instead of the minimization of $I(\alpha, S, \rho)$.

The squared distance $r^2(x, y, \alpha, \rho = 0)$ of a point (x, y) to the line $X \cdot \sin\alpha - Y \cdot \cos\alpha = 0$ is

$$(x \cdot \sin\alpha - y \cdot \cos\alpha)^2,$$

and, if for a shorten notation $F(\alpha, S) = I(\alpha, S', \rho = 0)$, the minimizing function can be expressed as follows

$$\begin{aligned}
(3.3)\quad F(\alpha, S) &= \iint_S \big((x - x_c(S)) \cdot \sin\alpha - (y - y_c(S)) \cdot \cos\alpha\big)^2 dx\, dy \\
&= \sin^2\alpha \cdot \iint_S (x - x_c(S))^2\, dx\, dy + \cos^2\alpha \cdot \iint_S (y - y_c(S))^2 dx\, dy \\
&\quad - \sin(2\alpha) \cdot \iint_S (x - x_c(S))\,(y - y_c(S))\, dx\, dy \\
&= \sin^2\alpha \cdot \overline{m}_{2,0}(S) + \cos^2\alpha \cdot \overline{m}_{0,2}(S) - \sin(2\alpha) \cdot \overline{m}_{1,1}(S).
\end{aligned}$$

The angle α for which the function $F(\alpha, S)$ (i.e., the integrals $I(\alpha, S', \rho = 0)$ and $I(\alpha, S, \rho)$) reaches its minimum defines the orientation of the shape S. We give a formal definition.

Definition 3.1. The orientation of a given shape S is determined by the angle α where the function $F(\alpha, S)$ reaches its minimum.

Shape orientation, as given by Definition 3.1, is easy to compute and can be expressed in terms of moments. Indeed, since the points (angles) where $F(\alpha, S)$ riches its maxima and minima are angles (points) where the first derivative $dF(\alpha, S)/d\alpha$ vanishes, i.e., where

$$\frac{dF(\alpha, S)}{d\alpha} = \overline{m}_{2,0}(S) \cdot \sin(2\alpha) - \overline{m}_{0,2}(S) \cdot \sin(2\alpha) - 2\overline{m}_{1,1}(S) \cdot \cos(2\alpha) = 0$$

we obtain immediately that the angle α which defines the orientation of S satisfies the following equation:

$$(3.4)\qquad \frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2 \cdot \overline{m}_{1,1}(S)}{\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S)}.$$

Although the standard method is naturally defined, straightforward and efficient to compute it breaks down in some circumstances. For example, problems arise when working with symmetric shapes [45, 49], but the method does not tell what the orientation should be even for some irregular shapes. If we consider the equality (3.3) we can see easily that $F(\alpha, S)$ becomes a constant function if

$$\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S) = 0 \quad \text{and} \quad \overline{m}_{1,1}(S) = 0. \tag{3.5}$$

Naturally, if $F(\alpha, S)$ is a constant function (for a given shape S), none of directions α could be pointed out as the shape orientation, and thus, the standard method fails.

This has caused the development of other methods, e.g., [6, 14, 18, 39, 45] and many more, for the computation of the shape orientation. Suitability of those methods strongly depends on particular application. It is not possible to say which of them is the best one or to establish a strict ranking among them as they each have their relative strengths and weaknesses (e.g., relating to robustness to noise, classes of shape that can be oriented, computational efficiency). A method dominant at one of applications could fail at another.

Notice that difficulties in the computation of the shape orientation can be caused by the nature of certain shapes. While for many shapes their orientations are intuitively clear and can be computed relatively easily, the orientation of some other shapes may be ambiguous or ill defined. Problems related to the estimation of the degree to which a shape has a distinct orientation are considered in [53].

3.3. Shape Elongation. Observations related to the computation of the shape orientation by the axis of the second least moment of inertia, easily lead to the definition of a new shape descriptor, named the shape elongation. It is naturally to predict that a given shape is said to be elongated (in a natural meaning of the word 'elongation') if it has a distinct orientation. I.e., the minima and maxima of the optimizing integral $I(S, \alpha, \rho)$ should differ essentially for more elongated shapes. Both, minima and maxima of $F(\alpha, S)$ are easy to compute. The minimum of the integral $I(S, \alpha, \rho)$ (also the minimum of $F(\alpha, S)$) is

$$\min_{\substack{\rho \geqslant 0 \\ \alpha \in [0, 2\pi]}} I(S, \alpha, \rho) = \frac{\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S) - \sqrt{4 \cdot (\overline{m}_{1,1}(S))^2 + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}}{2}$$

and is reached for $\rho = 0$ and α satisfying (3.4). This is in accordance with the fact that the axis of least second moment of inertia passes through the origin.

The maximum of the integral $I(S, \alpha, \rho)$, if $\rho \neq 0$ is allowed, obviously does not exist (i.e., the maximum is ∞). However, if $\rho = 0$ is assumed then

$$\max_{\substack{\rho = 0 \\ \alpha \in [0, 2\pi]}} I(S, \alpha, \rho) = \frac{\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S) + \sqrt{4 \cdot (\overline{m}_{1,1}(S))^2 + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}}{2}.$$

Now, we define the ratio between

$$\max_{\substack{\rho = 0 \\ \alpha \in [0, 2\pi]}} I(S, \alpha, \rho) = \max_{\alpha \in [0, \pi)} F(\alpha, S) \quad \text{and} \quad \min_{\substack{\rho \geqslant 0 \\ \alpha \in [0, 2\pi]}} I(S, \alpha, \rho) = \min_{\alpha \in [0, \pi)} F(\alpha, S)$$

as a measure for the elongation of S.

Definition 3.2. Let a given shape S. Then the elongation $\mathbf{E}_{st}(S)$ of S is defined as

$$\mathbf{E}_{st}(S) = \frac{\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S) + \sqrt{4 \cdot (\overline{m}_{1,1}(S))^2 + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}}{\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S) - \sqrt{4 \cdot (\overline{m}_{1,1}(S))^2 + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}}. \tag{3.6}$$

Shape elongation measure, as defined by the Definition 3.6, has several desirable properties. It reaches the minimal possible value of 1 for a circle. This matches our perception that a circle has the lowest possible elongation. Also, if we consider the rectangle $R(t)$ whose edge lengths are 1 and t, then the elongation $\mathbf{E}_{st}(R(t))$ tends to ∞, as $t \to \infty$. This is also in accordance with our perception. Let us mention that $\mathbf{E}_{st}(S)$ varies through $[1, \infty)$ and from the traditional reason is not normalized to be ranging in the interval $[0, 1]$, as preferred and mentioned in the introduction. A disadvantage of the standard elongation measure is that all the rotationally symmetric shapes but also some irregular shapes, have the elongation $\mathbf{E}_{st}(S)$ equal to 1. Those shape satisfy the conditions given in (3.5). In order to avoid such problems, some generalization of $\mathbf{E}_{st}(S)$ are suggested in [49].

Let us mention that there are also some naive methods to measure the shape elongation. For example, the shape elongation can be measured as the ratio of the edges of the minimum area rectangle which encloses the measured shape. It is worth mentioning that such bounding rectangles are easy to compute [10].

3.4. Shape Circularity. Hu invariants were introduced almost 50 years ago [16]. Many related aspects have been investigated, but there is still an ongoing interest. Recently, [47] considers geometric moment invariants and shows that the Hu invariants are particular case of geometric invariants. A new related problem was, first time, considered in another recent paper [56], where the authors consider shapes which optimize certain invariants. The following theorem, which shows that the first Hu invariant $I_1 = \mu_{2,0}(S) + \mu_{0,2}(S)$ is optimized by a circle, has been proved.

Theorem 3.1. *Let S be a given planar compact shape. Then* (a) $I_1(S) \geqslant \frac{1}{2\pi}$ *and* (b) $I_1(S) = \frac{1}{2\pi} \Leftrightarrow S$ *is a circle.*

Such a nice result, which says that $I_1(S) = \mu_{2,0}(S) + \mu_{0,2}(S)$ reaches its minimum $1/(2\pi)$ if and only if S is a circle, suggests the following definition of a shape circularity measure.

Definition 3.3. The circularity measure $\mathbf{C}(S)$, of a given shape S, is defined as

$$\mathbf{C}(S) = \frac{1}{2\pi \cdot (\mu_{2,0}(S) + \mu_{0,2}(S))} = \frac{1}{2\pi \cdot I_1(S)}.$$

The circularity measure $\mathbf{C}(S)$, defined as above, has several desirable properties, as summarized in the following theorem.

Theorem 3.2. *The circularity measure* $\mathbf{C}(S)$ *satisfies:*

(a) $\mathbf{C}(S) \in (0, 1]$, *for all shapes* S.
(b) $\mathbf{C}(S) = 1 \Leftrightarrow S$ *is a circle.*

(c) $\mathrm{C}(S)$ *is an invariant w.r.t. similarity transformations.*
(d) *For each* $\delta > 0$ *there is a shape* S *such that* $0 < \mathrm{C}(S) < \delta$.

Of course, the circularity, as one basic shape characteristics/descriptors, has already been considered in the literature. There are several measures. The most standard one considers the relation between the shape area and the shape perimeter [40]. Exploiting the fact that the circle has the largest area among all the shapes with the same perimeter, the most standard method defines the shape circularity $\mathrm{C}_{st}(S)$ in the following way

$$\mathrm{C}_{st}(S) = \frac{4 \cdot \pi \cdot \mathrm{Area\_of\_}S}{(\mathrm{Perimeter\_of\_}S)^2}. \tag{3.7}$$

The measure $\mathrm{C}_{st}(S)$ satisfies the properties (a)–(d) listed in Theorem 3.2. The proof is easy and straightforward. Notice that $\mathrm{C}_{st}(S)$ cannot be classified neither as area based nor boundary based because it uses both interior points (the shape area is needed) and boundary points (the shape perimeter is used for the computation).

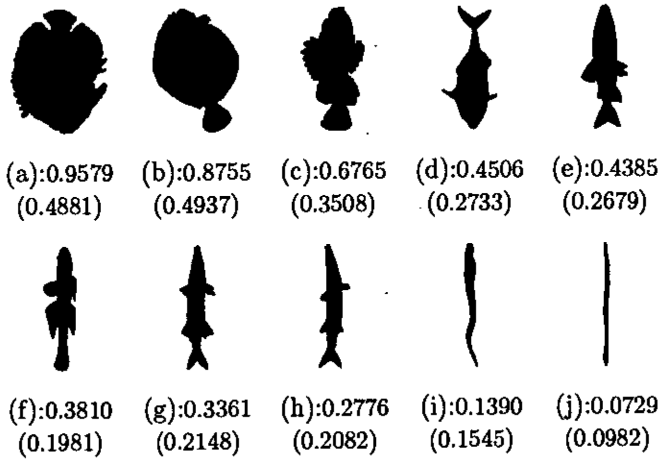
Now we give several examples to illustrate the behavior of these two, $\mathrm{C}(S)$ and $\mathrm{C}_{st}(S)$, circularity measures.

The first example is in Figure 3. Ten fish shapes are ranked with respect to their measured $\mathrm{C}(S)$ circularity (the numbers given immediately below the shapes). The obtained ranking $(a)(b)(c)(d)(e)(f)(g)(h)(i)(j)$ is pretty much in accordance with our perception.

If the same shapes are ranked with respect to $\mathrm{C}_{st}(S)$, a different ranking $(b)(a)(c)$ $(d)(e)(g)(h)(f)(i)(j)$ is obtained. Such a different ranking is expected, but also preferred, because the different rankings obtained suggest that a use of both measures could increase the classification efficiency. The standard circularity measure $\mathrm{C}_{st}(S)$ penalizes deep intrusions into the shape, because such intrusions lead to an essential perimeter increase. Consequently, deep intrusions imply a lower $\mathrm{C}_{st}(S)$ circularity. The measure $\mathrm{C}(S)$ is area based and does not penalizes such intrusions. This explains why the shape in Figure 3(a) has a higher measured $\mathrm{C}(S)$ circularity than the shape in Figure 3(b). On the other hand, the measure $\mathrm{C}_{st}(S)$ penalizes intrusions into the shape in Figure 3(a) and assigns a higher measured circularity $\mathrm{C}_{st}(S)$ to the shape in Figure 3(b). The eight position of the shape in Figure 3(f), if ranked by $\mathrm{C}_{st}(S)$, can be explained on a similar way.

The second example is in Figure 4. The selected shapes illustrate the robustness of $\mathrm{C}(S)$ and the sensitivity $\mathrm{C}_{st}(S)$. All four presented shapes have very similar $\mathrm{C}(S)$ circularity even though the fourth shape (in Figure 4(d)) has a very high noise level. Such obtained measures are caused by the fact that $\mathrm{C}(S)$ is an area based measure and, because of that, is very robust. On the other hand, $\mathrm{C}_{st}(S)$ can only cope with small levels of noise because it uses the shape perimeter for the computation. Indeed, the shape in Figure 4(a) has more than 2 times higher $\mathrm{C}_{st}(S)$ circularity than the shape in Figure 4(d).

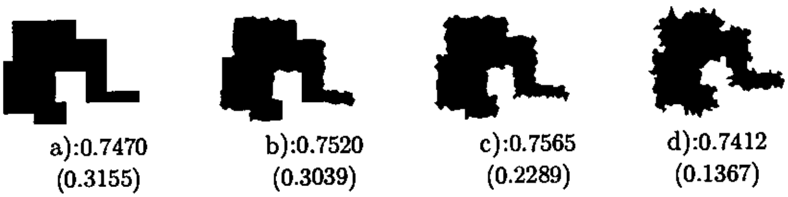
The third example is in Figure 5. A big advantage of $\mathrm{C}(S)$ over $\mathrm{C}_{st}(S)$ is demonstrated by using simple synthetic shapes. All three (compound) shapes displayed consist of three isometric circular discs–see Figure 5(a)–(c). In all three cases the same the standard circularity measure $\mathrm{C}_{st}(S)$ (equal to 1/3) is assigned. This is



(a):0.9579 (0.4881) (b):0.8755 (0.4937) (c):0.6765 (0.3508) (d):0.4506 (0.2733) (e):0.4385 (0.2679)

(f):0.3810 (0.1981) (g):0.3361 (0.2148) (h):0.2776 (0.2082) (i):0.1390 (0.1545) (j):0.0729 (0.0982)

FIGURE 3. Fish shapes are ranked with respect to their C(S) circularities (numbers immediately below the shapes). $\mathrm{C}_{st}(S)$ values are in brackets.

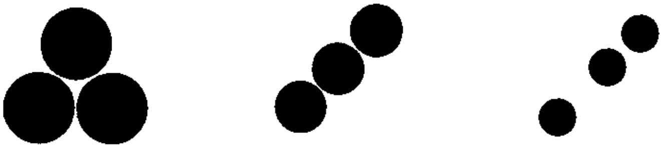


a):0.7470 (0.3155) b):0.7520 (0.3039) c):0.7565 (0.2289) d):0.7412 (0.1367)

FIGURE 4. C(S) circularities of shapes with added noise are given immediately below the shapes. The corresponding $\mathrm{C}_{st}(S)$ circularities are in brackets.

in accordance with the definition, see (3.7), because all three compound shapes have the same area and the same perimeter (the sum of perimeters of the shape components). On the other hand, C(S) assigns different circularities C(S) to the shapes in Figure 5(a)–(c). The computed circularities C(S) depend on the mutual position of the discs inside the shape, what is our preference.

3.5. Family of Circularity Measures. The method used to define circularity measure C(S) allows an extension to a family of circularity measures [56]. It has been shown that the measures from the family behave differently, implying that some of them can be combined in order to increase the classification performance. The key statements used for the extension of the C(S) measure to a family of circularity measures are given by the following two lemmas. Proofs can be found in [56].



(a): 0.7909 (b): 0.4540 (c): 0.2091

FIGURE 5. Measured circularity $\mathbf{C}(S)$ of a compound shape S depends on the mutual position of the components of S. All three compound shapes have $\mathbf{C}_{st}(S)$ circularity equal to $1/3$.

Lemma 3.1. *Let S be a planar compact shape whose centroid coincides with the origin, and let a constant $\beta > 0$. Then the following statements hold*

$$\frac{1}{(\mu_{0,0}(S))^{\beta+1}} \iint\limits_S (x^2+y^2)^\beta dx\,dy \geqslant \frac{1}{\pi^\beta \cdot (\beta+1)}$$

$$\frac{1}{(\mu_{0,0}(S))^{\beta+1}} \iint\limits_S (x^2+y^2)^\beta dx\,dy = \frac{1}{\pi^\beta (\beta+1)} \Leftrightarrow S \text{ is a circle.}$$

Lemma 3.2. *Let S be a planar compact shape whose centroid coincides with the origin and let β be a constant $-1 < \beta < 0$. Then*

$$\frac{1}{(\mu_{0,0}(S))^{\beta+1}} \iint\limits_S (x^2+y^2)^\beta dx\,dy \leqslant \frac{1}{\pi^\beta \cdot (\beta+1)}$$

$$\frac{1}{(\mu_{0,0}(S))^{\beta+1}} \iint\limits_S (x^2+y^2)^\beta dx\,dy = \frac{1}{(\beta+1)\pi^\beta} \Leftrightarrow S \text{ is a circle.}$$

The following definition comes naturally from the arguments from the previous two lemmas.

Definition 3.4. Let S be a shape whose centroid coincides with the origin and let a real β such that $-1 < \beta$ and $\beta \neq 0$. Then the circularity measure $\mathbf{C}_\beta(S)$ is defined as

$$\mathbf{C}_\beta(S) = \begin{cases} \dfrac{\mu_{0,0}(S)^{\beta+1}}{(\beta+1)\cdot\pi^\beta\cdot\iint_S (x^2+y^2)^\beta dx\,dy}, & \beta > 0 \\[2ex] \dfrac{(\beta+1)\cdot\pi^\beta\cdot\iint_S (x^2+y^2)^\beta dx\,dy}{\mu_{0,0}(S)^{\beta+1}}, & \beta \in (-1,0). \end{cases}$$

It is worth mentioning that the measures $\mathbf{C}_\beta(S)$, $\beta \in (-1,0)\cup(0,\infty)$, satisfy the following properties:

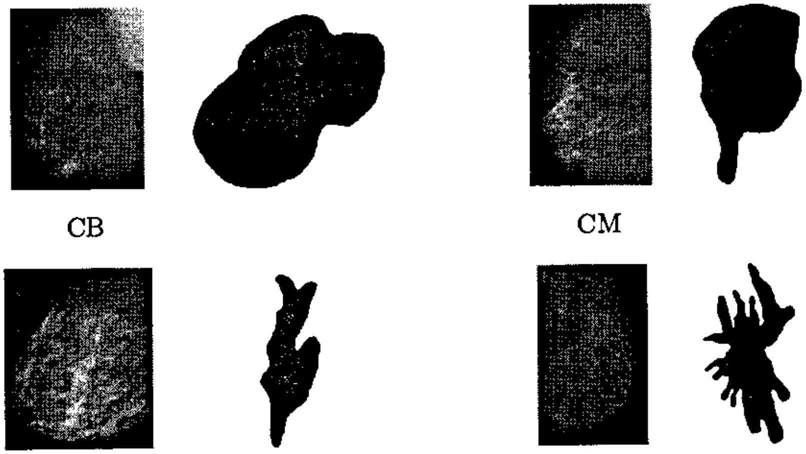
(a) $\mathbf{C}_\beta(S) \in (0,1]$ for all planar shapes S.
(b) $\mathbf{C}_\beta(S) = 1 \Leftrightarrow S$ is a circle.
(c) $\mathbf{C}_\beta(S)$ is invariant with respect to similarity transformations.
(d) For each $\delta > 0$ there is a shape S such that $0 < \mathbf{C}_\beta(S) < \delta$.

For proof details we refer to [56].

Now we give some examples. More details are in [56]. Circularities were measured for the set of 54 masses from mammograms, combining images from the MIAS and Screen Test databases [32], see Figure 6. Rangayyan et al. [32] assessed the measures by classifying them as circumscribed/spiculated, benign/malignant, and CB/CM/SB/SM, in two group and four group classification experiments. Their best shape measure results for the three classification tasks were:

1. Circumscribed versus spiculated: 88.9% achieved by both $C_{st}(S)$ and a Fourier based shape factor.
2. Benign versus malignant: 75.9% achieved by the Fourier based shape factor.
3. Four-way discrimination: 64.8% achieved by both $C_{st}(S)$ and the Fourier based shape factor.

From Table 1 we see that the best results from using $C_{\beta}(S)$ occurred for $\beta = 32$ and were respectively better, worse, and equal to Rangayyan et al.'s. Circularity measures from [15, 30] did not perform as well as $C_{\beta}(S)$.



CB

CM

SB

SM

FIGURE 6. Examples of the four classes of mammographic masses: circumscribed benign (CB), circumscribed malignant (CM), spiculated benign (SB), spiculated malignant (SM). The masses were extracted from the mammograms on the left, and have been drawn rescaled.

4. Boundary Based Shape Descriptors

Boundary based methods become more popular in the recent days. That is caused mainly by a strong demand for a higher precision in image processing and

circularity	mammography		
measure	circ./spic.	mal./ben.	4 groups
$\mathbf{C}_{\beta=1/8}(S)$	83.33	66.67	51.85
$\mathbf{C}_{\beta=1/4}(S)$	85.19	64.81	51.85
$\mathbf{C}_{\beta=1/2}(S)$	75.93	57.41	42.59
$\mathbf{C}_{\beta=1}(S)$	68.52	68.52	51.85
$\mathbf{C}_{\beta=2}(S)$	75.93	68.52	53.70
$\mathbf{C}_{\beta=4}(S)$	72.22	46.30	33.33
$\mathbf{C}_{\beta=8}(S)$	79.63	59.26	50.00
$\mathbf{C}_{\beta=16}(S)$	87.04	57.41	51.85
$\mathbf{C}_{\beta=32}(S)$	**90.74**	**70.37**	**64.81**
$\mathbf{C}_{st}(S)$ pix.	87.04	59.26	57.41
$\mathbf{C}_{st}(S)$ pol.	85.19	59.26	57.41
Haralick [15]	68.52	46.30	37.04
Proffitt [30]	51.85	42.59	25.93

TABLE 1. Applications of the circularity measures to classification of mammographic masses. The second, third and fourth columns report classification accuracies. Results for the best performing measure for each task is highlighted in bold.

computer vision tasks. Another reason is that, due to the permanent development in the image technology, a higher quality data can be provided. Despite boundary based approaches are less robust and very often theoretically more complicated, they have some obvious advantages. Apart from a higher precision, it is worth mentioning that boundary based methods can cope with particularly extracted boundaries and with objects which are linear in their nature (signatures, for example). Obviously, the later objects cannot be treated by area based methods. In addition, boundary based methods are usually faster to compute (the boundary consists a smaller number of pixels than the whole shape does).

Notice that sometimes there is an easy (at least theoretical) extension of the area based methods to their boundary analogues, or vice-versa. There are also situations where this is not a simple task. An example could be the rectilinearity measure [50] used to detect buildings on satellite images, whose area based analogue is not discovered yet.

4.1. Line Moments.

There is an obvious analogue for geometric (area) moments introduced by (3.1). If a curve γ is given in an arc-length parametrization,

$$\gamma: \quad x = x(s), \;\; y = y(s), \;\; s \in [0, \tau]$$

then the line moment $\eta_{p,q}(\gamma)$ is defined as $\eta_{p,q}(\gamma) = \int_\gamma x(s)^p y(s)^q ds$. Obviously, $\eta_{0,0}(\gamma) = \int_\gamma ds = \tau$ equals the length of the curve γ. Of course, γ can be the boundary ∂S of a bounded planar region S, but also it can be an open curve.

Central moments $\overline{\eta}_{p,q}(\gamma)$ are defined as

$$\overline{\eta}_{p,q}(\gamma)=\int_{\gamma}\left(x(s)-\frac{\eta_{1,0}(S)}{\tau}\right)^{p}\cdot\left(y(s)-\frac{\eta_{0,1}(S)}{\tau}\right)^{q}ds,$$

while the normalized moments $\zeta_{p,q}(\gamma)$ are defined as

$$\zeta_{p,q}(\gamma)=\frac{\overline{\eta}_{p,q}(\gamma)}{(\eta_{0,0}(\gamma))^{1+p+q}}.$$

Many of statements and methods based on a use of geometric (area) moments have a straightforward extension to analogue statements based on a use of line moments. Such an example are Hu invariants $I_1, I_2, \dots, I_7$ listed in Section 3.1. It is enough to replace the integrals $\iint_S x^p y^q dx\,dy$, appearing in $I_1, I_2, \dots, I_7$, with their analogue path/line integrals $\int_{\partial S} x(s)^p y(s)^q ds$, (where the boundary ∂S is given in an arc-length parametrization form: $x = x(s)$, $y = y(s)$) and all seven invariants remain valid. Notice that it is crucial that the boundary ∂S is given in an arc-length parametrized form in order to preserve the invariance.

4.2. Boundary based shape orientation. The standard method for the computation of the shape orientation, based on the axis of the least second moment of inertia, has its boundary based analogue. Informally speaking, we can define the shape orientation by the line which minimizes the line integral of the squared distance of the boundary points to the line, or more formally by the line which minimizes the integral

$$I(\alpha,\partial S,\rho)=\int_{\partial S}p^2(x,y,\alpha,\rho)\,ds \tag{4.1}$$

where $p^2(x, y, \alpha, \rho)$ is the distance from the point $(x, y) \in \partial S$ to the line given in the form $X \cdot \sin\alpha - Y \cdot \cos\alpha = \rho$. The boundary ∂S has to be is given in an arc-length parametrization. Following the same formalism, as in the case of the the standard method, we can deduce that the angle α which defines such a defined shape orientation satisfies the following equation

$$\frac{\sin(2\alpha)}{\cos(2\alpha)}=\frac{2\cdot\overline{\eta}_{1,1}(\partial S)}{\overline{\eta}_{2,0}(\partial S)-\overline{\eta}_{0,2}(\partial S)}. \tag{4.2}$$

Exploiting boundary points, to define the shape orientation, gives additional possibilities for new shape orientation methods. Some specific boundary features can be involved. Notice that boundary details may not play any essential role when work with area based methods. That is because changes in boundary details could lead to very small changes in the area of shape and in related features.

The recent paper [27] defines the shape orientation by the angle α which maximizes the integral of the weighted squared lengths $|\mathrm{pr}_\alpha\overrightarrow{(x'(s),y'(s))}|^2$ of the projections $\mathrm{pr}_\alpha\overrightarrow{(x'(s),y'(s))}$ of all the tangent vectors $\overrightarrow{(x'(s),y'(s))}$, to the shape boundary ∂S, onto a line having the slope α (see Figure 7). The definition is natural and well motivated. The weights, dependent on the curvature at the boundary points, allow us to tune the method behavior. For example, by a suitable choice of the weights, it is possible to give a higher impact to the straight sections on the boundary, or to the sections with a high curvature. A formal definition follows.

Definition 4.1. Let a shape S be given. Let $x = x(s)$, $y = y(s)$, $s \in [0,1]$ be an arc-length parametrization of the boundary ∂S. Also, let $f(\kappa(s))$ be a function dependent on $\kappa(s)$ which is the curvature of ∂S at the point $(x(s), y(s))$. The orientation $O_f(S)$ of the shape S is defined by the slope α that maximizes the integral

$$I_f(\alpha, \partial S) = \int_{\partial S} f(\kappa(s)) \cdot |\mathbf{pr}_\alpha \overrightarrow{(x'(s), y'(s))}|^2 ds$$

of the squared lengths of the projections of all the tangent vectors of ∂S on this line, weighted by the curvature at the boundary ∂S points.

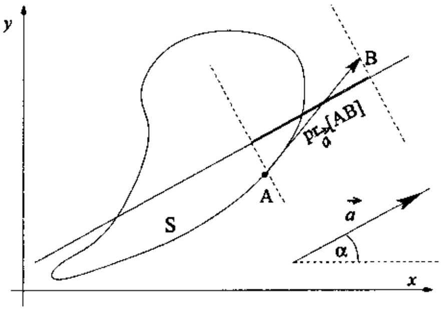


FIGURE 7. Projection of a tangent vector $(x'(s)\vec{,}y'(s)) = \vec{AB}$ onto a line having the slope α.

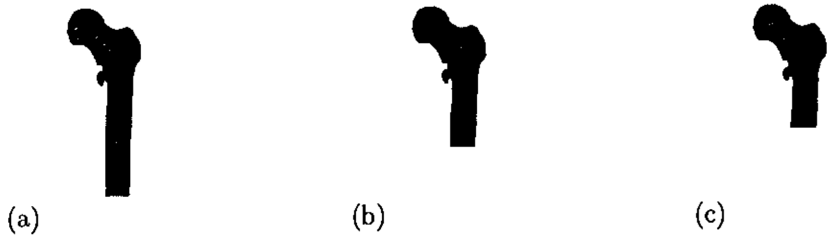
An obvious advantage of the new measure is that it has tuning possibilities. Also, Definition 4.1 enables a closed formula for the computation of the shape orientation. This is shown by the following theorem (for a proof see [27]).

Theorem 4.1. *Assume a given shape S and a function $f(\kappa)$. Then the orientation $O_f(S)$ satisfies*

$$\frac{\sin(2O_f(S))}{\cos(2O_f(S))} = \frac{2\int_{\partial S} f(\kappa(s))\, x'(s)\, y'(s)\, ds}{\int_{\partial S} f(\kappa(s))(x'(s)^2 - y'(s)^2)\, ds} \tag{4.3}$$

where $x = x(s)$, $y = y(s)$, $s \in [0,1]$ is an arc-length parametrization of the boundary ∂S and $\kappa(s)$ is the curvature of ∂S at the point $(x(s), y(s))$.

Examples in Figure 8 illustrate how a suitable choice of the weighting function could lead to the preferred method behavior. The same bone has been framed differently in each sub-picture. The orientations computed by the standard method (see (3.4)) gives inconsistent orientations: 107.0°, 120.4° and 131.0°. Let us notice that due to the nature of the object presented, the vertical orientation or, at least, a nearly vertical orientation is preferred. Equally weighting every boundary point ($f(\kappa(s)) = 1$) still produces rather different orientations for each frame (91.7°, 95.0° and 97.4°).



	standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = e^{-0.01\cdot\kappa(s)}$	$f(\kappa(s)) = e^{-0.05\cdot\kappa(s)}$
(a)	107.0°	91.7°	91.7°	91.7°
(b)	120.4°	95.0°	95.0°	91.7°
(c)	131.0°	97.4°	97.4°	91.7°

FIGURE 8. The same bone is captured in different frames and orientations of each frame are computed by the standard method and by using different weighting functions in (4.3).

However by using a weighting function $f(\kappa(s)) = e^{-0.05\cdot\kappa(s)}$, higher weights are given to boundary points with a small curvature (i.e., to the points on the straight sections of the boundary), and the same orientation of 91.7° is computed for each frame. In addition, the computed orientations are nearly vertical, as preferred.

The following discussion points out advantages of the measure given by Definition 4.1 over both standard method (3.4) and its analogue computed by (4.2).

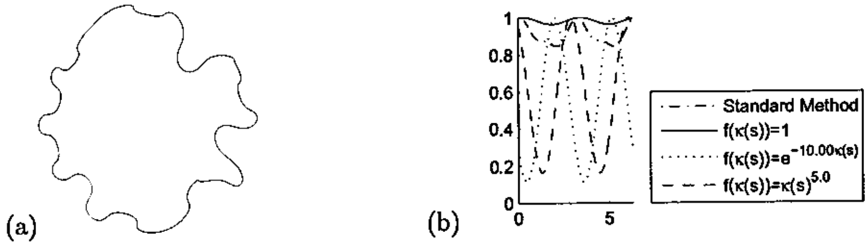
First advantage relates to the situations where some of methods considered fail. As already mentioned, due to the diversity of shapes, it is natural to expect that there are always situations where the method used fails. The standard method for the computation of shape orientation fails if the conditions in (3.5) are satisfied. The method which is an analogue of the standard method, also fails if the corresponding optimizing integral $I(\alpha, \partial S, \rho)$ (see (4.1)) is a constant function, i.e., when

$$\overline{\eta}_{2,0}(\partial S) \; - \; \overline{\eta}_{0,2}(\partial S) = 0 \quad \text{and} \quad \overline{\eta}_{1,1}(\partial S) = 0.$$

A simple idea to overcome such problems was to consider a higher exponent $2N$ in the optimizing integral (for more details see [45, 49]). The problem is that such modified optimizing functions (integrals) do not allow a closed form solution. So, higher exponents involved might be computationally very expensive.

On the other side, Definition 4.1 allows an additional option to overcome situations when the method does not work. Indeed, if the method fails for a certain choice of weighting function, it could work for another choice of them. I.e., if the weighting function $f(\kappa(s))$ is replaced with another weighting function $g(\kappa(s))$, the situation could be changed (for shapes which are not rotationally symmetric), meaning that the new optimizing function $I_g(\alpha, \partial S)$ becomes a nonconstant function, with a well-defined maximum. Then, the optimizing function $I_g(\alpha, \partial S)$ can be used to define a reliable orientation of S.

But a constant optimizing function is not the only problem. The problem is also if the optimizing function has no distinct extreme values. This would imply that the computed orientation is strongly dependent on noise or on the digitization process applied. In such situations a change of the weighting function $f(\kappa(s))$ could lead to more stable and more reliable orientations being computed. This is illustrated by the example in Figure 9. The shape in in Figure 9(a) has no intuitively clear orientation.



standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = e^{-10.0\kappa(s)}$	$f(\kappa(s)) = \kappa(s)^{5.0}$
123.0°	17.3°	116.5°	166.8°

FIGURE 9. Graphs of the optimization functions used to orient the shape in (a) are scaled such that their maximum becomes equal to 1, as displayed in (b).

The graphs displayed in Figure 9(b) correspond to the different waiting functions used to compute the orientations. Naturally, if the minima and maxima of the optimizing functions are more distinct then the shape is 'more orientable' (the shape orientability problem is discussed in [53]). I.e., the computed orientation is more reliable. In view of the presented example it can be said that the choice of $f(\kappa(s)) = \kappa(s)^5$ or $f(\kappa(s)) = \kappa(s)^{-10}$ leads to the computed orientations which are more reliable than the orientations computed by the standard method or if the weighting functions $f(\kappa(s)) = 1$ is used. Even that the orientation computed by using $f(\kappa(s)) = \kappa(s)^5$ and $f(\kappa(s)) = \kappa(s)^{-10}$ (166.8° and 116.5° respectively) differ essentially, they are both understood as very reliable because they correspond to the distinct maxima of the corresponding optimizing functions.

As mentioned, the shape in Figure 9(a) has not an intuitively clear orientation but notice that, for certain image processing tasks, this is not a problem. What is crucial is that the computed orientation is reliable. E.g., in a robot manipulation task, all copies of a given product would be positioned consistently if the computed orientation is reliable, independently does the computed orientation match human perception or not.

Another benefit from having a tunable method (i.e., weighting functions, in this case) is illustrated by a turkey shape in Figure 10. For this shape the standard method gives an orientation of 83.6°, but this orientation is not reliable. The optimization function varies between 0.0059 and 0.0065 and the corresponding graph is rather flat (see the graphs in Figure 10). Since the standard method does not give a distinct orientation, small deviations on the shape boundary (caused by the noise or by the digitization process applied) could lead to an essential deviation of the computed orientation. Indeed, when some noise is added to the shape in Figure 10(a) the computed orientation (by the standard method) changes to 68.8° (Figure 10(b)) and to 97.4° (Figure 10(c)), depending on the noise level. Thus, the standard method is not suitable to be applied to the shape in Figure 10(a) and its "noise" appearances in Figure 10(b) and Figure 10(c). That is in accordance with the values in the table below.

	$\lvert\mu_{1,1}(\partial S)\rvert$	$\lvert\mu_{2,0}(\partial S) - \mu_{0,2}(\partial S)\rvert$
Figure 10(a)	0.000063	0.000658
Figure 10(b)	0.000311	0.000727
Figure 10(c)	0.000086	0.000664

The values of $|\mu_{2,0}(\partial S) - \mu_{0,2}(\partial S)|$ and $|\mu_{1,1}(\partial S)|$ are almost zero, and, consequently, the optimizing function $F(\alpha, S)$ is almost constant. So, it is not possible to distinguish the extreme values of $F(\alpha, S)$ accurately.

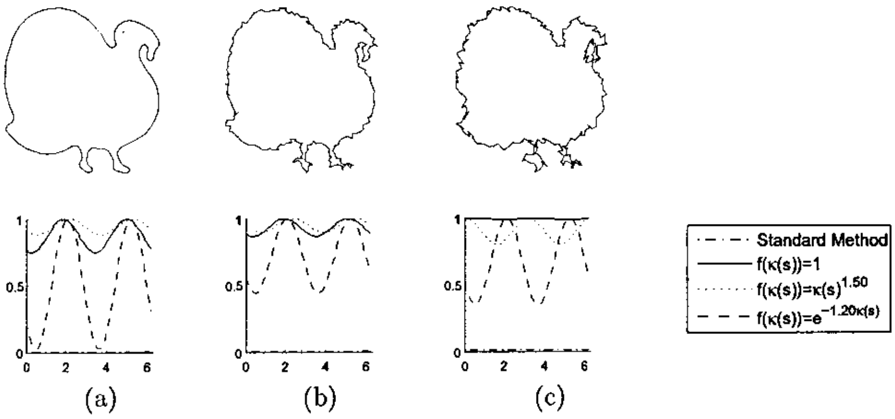
However, a reliable orientation of the shape in Figure 10(a) can be computed by the method given by Definition 4.1. Indeed, if the weighting function is $f(\kappa(s)) = e^{-1.5\kappa(s)}$ then the shapes in Figure 10(a)-(c) are oriented consistently. The reached consistency among the orientations computed is expected because the weighting function $f(\kappa(s)) = e^{-1.5\kappa(s)}$ gives a small weights to high curvature points. In this way the noise effects are minimized.

On the other side, if the weighting functions $f(\kappa(s)) = 1$ and $f(\kappa(s)) = \kappa(s)^{1.5}$ are used, then the computed orientations are highly dependent on the noise–see the table in Figure 10. This is because these weighting functions give reasonably high weights to a high curvature points.

4.3. Convexity Measure. One of the mostly employed shape descriptors is the shape convexity. Over the years many convexity measures have been developed (e.g., [3, 20, 26, 31, 36, 37, 51, 52]) and have been applied to tasks such as image segmentation, object classification, objects recognition, etc. Here we present a boundary based convexity measure developed in [52]. The measure can be applied to both closed and open curves. We start with the following definition of convex curves.

Definition 4.2. A curve γ is convex if and only if for each two points A and B on the curve γ the open line segment (AB) does not intersect the curve γ (i.e., $(AB) \cap \gamma = \emptyset$) or (AB) completely belongs to the curve γ (i.e., $(AB) \subset \gamma$).

It is easy to see that the Definition 4.2 is equivalent to a very common definition of a convex curve which says that a given curve γ is convex if and only if for each point $A \in \gamma$ there is a line l such that $A \in l$ and the curve γ completely lies in one of the closed half planes determined by the line l. Based on Definition 4.2 we are



	Standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = \kappa(s)^{1.5}$	$f(\kappa(s)) = e^{-1.5\kappa(s)}$
(a)	83.6°	106.5°	123.7°	117.9
(b)	68.8°	112.2°	135.1°	117.9
(c)	97.4°	26.3°	3.4°	120.3

FIGURE 10. Orientation of the shape in (a) with the different level of noise added in (b) and (c). The graphs corresponding to the new method (for different weighting functions) are scaled such that their maximum is 1, while the graph of the optimizing function used in the standard method is given on its natural scale. The table includes the computed orientations.

able to define a new convexity measure for single curves, but also for disconnected curves consisting of several arcs.

Definition 4.3. Let $\gamma = \gamma_1 \cup \cdots \cup \gamma_n$ be a curve that consists of $n \geqslant 1$ curve segments, and let A and B be two randomly selected points from γ. The convexity measure $\mathrm{M}(\gamma)$ is defined as the probability that one of the following two events occur:

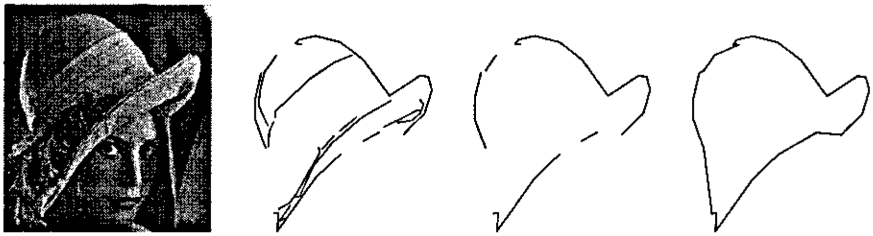
- the open straight line segment (AB) does not intersect γ (i.e., $(AB) \cap \gamma = \emptyset$), or
- the open straight line segment (AB) completely belongs to γ (i.e., $(AB) \subset \gamma$).

The measure $\mathrm{M}(\gamma)$ has the following desirable properties ($\gamma = \gamma_1 \cup \cdots \cup \gamma_n$ is not a necessarily connected curve):

(a) $\mathrm{M}(\gamma) \in (0, 1]$.
(b) $\mathrm{M}(\gamma) = 1$ if and only if there is a convex curve ρ such that $\gamma \subset \rho$.
(c) $\mathrm{M}(\gamma)$ is invariant under similarity transformations.
(d) for any $\varepsilon > 0$ there is a curve γ such that $\mathrm{M}(\gamma) < \varepsilon$.

A draw-back of $\mathrm{M}(\gamma)$ is that a closed formula for the computation of $\mathrm{M}(\gamma)$ can be computed reasonably easily only in particular cases. More over in most image processing tasks the equation of γ or the equations of γ-segments remain unknown. In such cases it is only possible to estimate $\mathrm{M}(\gamma)$. There are straightforward methods to do this quickly and accurately.

Now, we give several examples to illustrate how $\mathrm{M}(\gamma)$ acts. The first example in Figure 11 demonstrates how $\mathrm{M}(\gamma)$ acts in situations if there is some overlap between objects or there is no a clear difference between foreground and background pixels. In such cases only fragments of the boundary can be extracted; nevertheless, it would still be worth computing shape information from the available data and $\mathrm{M}(\gamma)$ can provide some.



(a) (b) $\mathrm{M} = 0.309$ (c) $\mathrm{M} = 0.767$ (d) $\mathrm{M} = 0.708$

FIGURE 11. (a) Extracting the boundary of Lena's hat is difficult due to poor contrast in places as well as clutter. (b) After edge detection and linking, the edge segments relating to the hat have been manually selected. (c) The outer boundary curves. (c) Gaps between the outer boundary curves completed by straight line segments. The new convexity measure $\mathrm{M}(S)$ can cope with all those situations.

The second example is in Figure 12. Handwritten digits are shown. The ranking the digits according to the convexity measure $\mathrm{M}(\gamma)$ demonstrates a good application potential. Digits "1", "4", "5" and "8" are separated correctly, even that there are substantial natural variations, not only in their overall shape, but also in topology. For instance, one of each of the "0" and "2" digits one example is self intersecting while the other is not. Even small, the example presented indicates that the convexity measure would be a useful property for classification of the digits.

The third example illustrates how $\mathrm{M}(\gamma)$ can be used for the signature recognition tasks. The signatures presented are treated as multiple curve segments–see (see Figure 13). Again noticeable variability is evident within each individual. Considerably lower convexity values are obtained because these curves (i.e., signatures) are more complex than the individual digits in Figure 12. It can be seen from the ranking by $\mathrm{M}(\gamma)$ that convexity is a sufficient descriptor for classification in this small example.

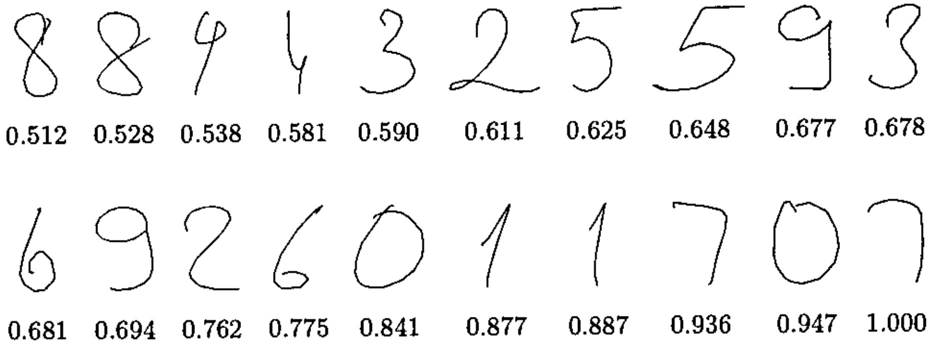


FIGURE 12. Handwritten digits ordered by their $M(\gamma)$ convexity values.

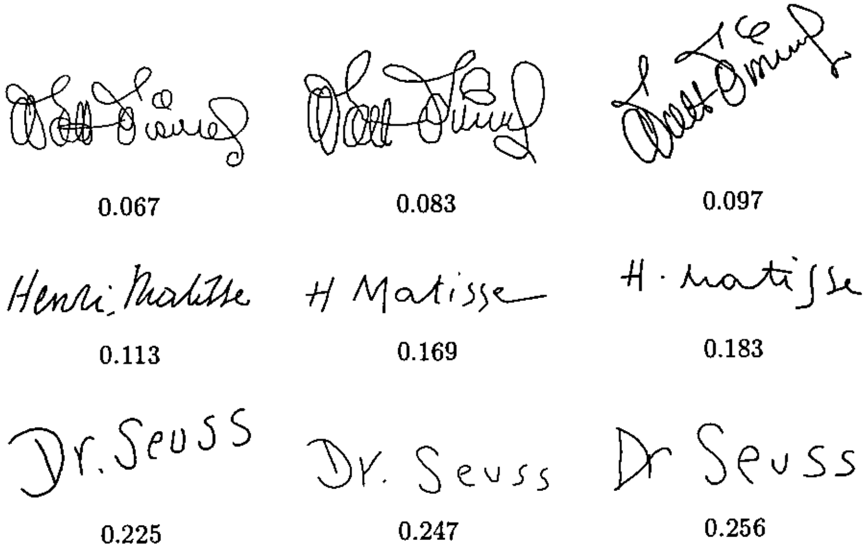


FIGURE 13. Signatures of Walt Disney, Henri Matisse and Dr. Seuss ordered by their $M(\gamma)$ convexity values.

5. Multi-component Shape Approach

As mentioned, there is no method for the computation of shape orientation which is dominant in all situations. That is a reason why many other methods, different from standard one, are developed. New applications cause new demands for particular method performances. The recent paper [55] has introduced a new method for the computation of shape orientation with a particular request that method should be applicable to multi-component shapes. The method is described as follows:

- Let S be a given shape, and consider all the line segments $[AB]$ where $A, B \in S$.
- Let $\overrightarrow{a} = (\cos\alpha, \sin\alpha)$ denote the unit vector in the direction α.
- Also, let $\mathbf{pr}_{\overrightarrow{a}}[AB]$ be the projection of the line segment $[AB]$ onto a line parallel to $\overrightarrow{a}$, while $|\mathbf{pr}_{\overrightarrow{a}}[AB]|$ denotes the length of such a projection (for the notations see Figure 14).

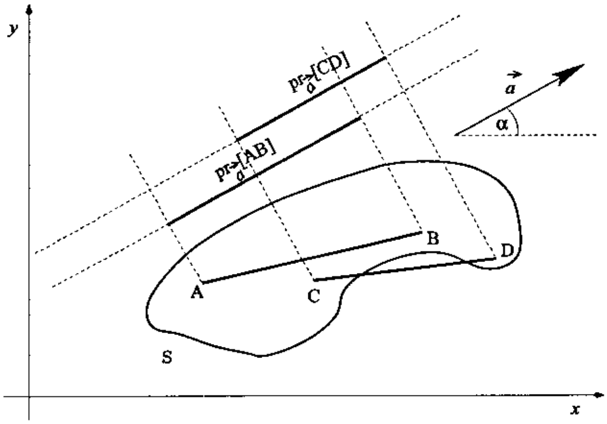

y
$\mathrm{pr}_{\overrightarrow{a}}[CD]$
$\mathrm{pr}_{\overrightarrow{a}}[AB]$
$\overrightarrow{a}$
α
B
D
A
C
S
x

FIGURE 14. Projections of all the line segments whose endpoints lie in S are considered, irrespective of whether the line segment intersects the boundary of S (e.g., the line segment $[CD]$) or not (e.g., $[AB]$).

Then, we define the orientation of the shape S by the direction α that maximizes the integral of the squared lengths of the projections $\mathbf{pr}_{\overrightarrow{a}}[AB]$ onto a line having this direction. A formal definition follows.

Definition 5.1. The orientation of a given shape S is defined by the angle α where the function

$$G(\alpha, S) = \iiiint\limits_{\substack{A=(x,y)\in S \\ B=(u,v)\in S}} |\mathbf{pr}_{\overrightarrow{a}}[AB]|^2 dx\, dy\, du\, dv \tag{5.1}$$

reaches its maximum.

Informally speaking, Definition 5.1 defines the orientation of a given shape S by the slope of a line that maximizes the total sum of squared lengths of projections of all straight line segments whose end points belong to S (see Figure 14).

Interestingly, even though Definition 5.1 and the Definition 3.1 come from different motivations, and even the optimizing functions $F(\alpha, S)$ and $G(\alpha, S)$ are different, it has been shown that the orientations computed by these two methods

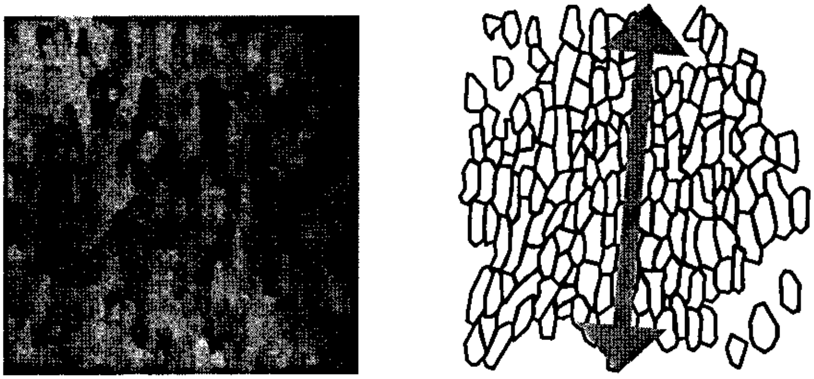


FIGURE 15. An embryonic tissue image (on the left) is presented as a multi-component shape (on the right). The desired computed orientation is presented by the arrow.

are the same. Indeed, Theorem 5.1 (for a proof see [55]) shows that for a fixed shape S the quantity $G(\alpha, S) + 2 \cdot m_{0,0}(S) \cdot F(\alpha, S)$ is the same for all $\alpha \in [0, 2\pi)$. Furthermore, $G(\alpha, S) + 2 \cdot m_{0,0}(S) \cdot F(\alpha, S) =$ constant implies that the maximum of $G(\alpha, S)$ and minimum of $F(\alpha, S)$ are reached at the same points, i.e., the angle where $G(\alpha, S)$ reaches the maximum must be the angle where $F(\alpha, S)$ reaches its minimum. So, the shape orientations computed by Definition 3.1 and Definition 5.1 coincide.

Theorem 5.1. *Let a shape S be given. Then*

$$G(\alpha, S) + 2 \cdot m_{0,0}(S) \cdot F(\alpha, S) = 2 \cdot m_{0,0}(S) \cdot (\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S))$$

is true for all $\alpha \in [0, 2\pi]$.

Definition 5.1 has an essential advantage over Definition 3.1 because it has a natural extension to compound (i.e., multi-component) shapes. Just to mention that the dealing with multi-component objects is of a great importance because in many situations, several single objects usually appear as a group and act together (e.g., blood cells, vehicles on the road, fish shoal or group of people (as in Figure 16), etc). Also, in many situations, it is suitable to consider a single object as a multi-component one, consisting of many components defined with respect to some natural criteria (as an embryonic tissue displayed in Figure 15, or materials micro structures, wood textures, etc). Further, a sequence of the same object appearing on a sequence of frames (e.g. a walking human in Figure 18) can be understood as as a multicomponent shape.

Surprisingly, the orientation of multi-component shapes has not been intensively studied in literature yet. Recently, [55] has introduced a method for computation of the orientation of multi-component shapes, presented in 2D images. The method is theoretically well founded, and thus, its behavior can be well understood, but also predicted to some extent, which is always an advantage.

It is worth mentioning that there is no an easy and straightforward way to compute the orientation of a multi-component shape from the orientations (computed by some of the existing methods for single component shapes) of its components. Indeed, a very natural idea would be to compute the orientation of a multi-component shape from the orientations assigned to its components, probably weighted by some coefficients computed from the 'component importance' (e.g. the component size or its relative position inside the shape as whole). However, the problem is that a huge majority of methods define the shape orientation by a line (not by a vector). This implies an ambiguity of 180 degrees (e.g. the computed orientations α and $\alpha + 180^\circ$ are assumed to be the same). So, if $S_1, S_2, \dots, S_n$ are components of a multi-component shape S, then most of the existing methods would compute their orientations as $\varphi_1 + a_1 \cdot 180^\circ$, $\varphi_2 + a_2 \cdot 180^\circ, \dots, \varphi_n + a_n \cdot 180^\circ$, where the numbers $a_1, a_2, \dots, a_n$ are arbitrarily chosen from $\{0, 1\}$. Thus if, in the simplest variant, the orientation of multi-component shape $S = S_1 \cup S_2 \cup \dots \cup S_n$ is computed as the average value of the orientations assigned to its components, then the orientation of S would be computed as

$$\frac{(\varphi_1 + a_1 \cdot 180^\circ) + \ldots + (\varphi_n + a_n \cdot 180^\circ)}{n} = \frac{\varphi_1 + \ldots + \varphi_n}{n} + \frac{(a_1 + \ldots + a_n) \cdot 180^\circ}{n}$$

and obviously, for different choices of $a_1, a_2, \dots, a_n$, the computed orientations are inconsistent (i.e., they could differ for an arbitrary multiple of the fraction $180^\circ/n$). This is obviously unacceptable.

As mentioned, Definition 5.1 allows a straightforward extension to the multi-component shapes, as given by the following definition.

Definition 5.2. Let S be a compound shape which consists of m disjoint shapes $S_1, S_2, \dots, S_m$. Then the orientation of S is defined by the angle that maximizes the function $G_{comp}(\alpha, S)$ defined by

$$G_{comp}(\alpha, S) = \sum_{i=1}^{m} \iiiint\limits_{\substack{A=(x,y)\in S_i \\ B=(u,v)\in S_i}} |\mathrm{pr}_{\vec{a}}[AB]|^2 dx\, dy\, du\, dv. \tag{5.2}$$

The previous definition enables an easy computation of the orientation of compound objects, as shown by the following theorem (see [55] for the proof details).

Theorem 5.2. *Let a compound shape S, consisting of m disjoint shapes $S_1, S_2, \dots$, S_m, be given, and let the function $G_{comp}(\alpha, S)$ be defined as in* (5.2).

The angle α where the function $G_{comp}(\alpha, S)$ reaches its maximum satisfies the following equation

$$\frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2 \cdot \sum_{i=1}^{m} \overline{m}_{1,1}(S_i) \cdot m_{0,0}(S_i)}{\sum_{i=1}^{m} (\overline{m}_{2,0}(S_i) - \overline{m}_{0,2}(S_i)) \cdot m_{0,0}(S_i)}. \tag{5.3}$$

The new method has some specific properties which appear to be very desirable when computing the orientation of multi-component shapes. These properties do not hold if the 'single component' methods are applied to multi-component shapes. Some of such properties are listed below as separate remarks.

Remark 5.1. Any component S_i, of a compound shape $S = S_1 \cup \cdots \cup S_m$, which cannot be oriented by optimizing $G(\alpha, S_i)$ (i.e., $G(\alpha, S_i) = \text{constant}$) will not contribute to (5.3), and is therefore ignored in the computation of $G_{comp}(\alpha, S)$. That is because $G(\alpha, S_i) = \text{constant}$ implies $\overline{m}_{1,1}(S_i) = 0$ and $\overline{m}_{2,0}(S_i) = \overline{m}_{0,2}(S_i)$.

Remark 5.2. If all components of S have identical orientation according to $G(\alpha, S)$ then this same orientation is also computed by $G_{comp}(\alpha, S)$.

The weighting given to the shape components in (5.3) causes a big influence of the larger components to the computed orientation. For instance, let a compound shape S which consists of shapes S_1 and S_2' such that the shape S_2' is the dilation of a shape S_2 by a factor $\mathbf{r}$, i.e., $S_2' = \mathbf{r} \cdot S_2 = \{(\mathbf{r} \cdot x, \mathbf{r} \cdot y) \mid (x, y) \in S_2\}$. Then,

$$m_{0,0}(S_2') = \mathbf{r}^2 \cdot m_{0,0}(S_2),\ \overline{m}_{1,1}(S_2') = \mathbf{r}^4 \cdot \overline{m}_{1,1}(S_2),$$
$$\overline{m}_{2,0}(S_2') = \mathbf{r}^4 \cdot \overline{m}_{2,0}(S_2),\ \overline{m}_{0,2}(S_2') = \mathbf{r}^4 \cdot \overline{m}_{0,2}(S_2).$$

Entering these estimates into (5.3) we obtain that the orientation α of the compound shape $S = S_1 \cup S_2'$ should be computed from

$$\begin{aligned}(5.4)\quad \frac{\sin(2\alpha)}{\cos(2\alpha)} &= \frac{2 \cdot \overline{m}_{1,1}(S_1) \cdot m_{0,0}(S_1) + 2 \cdot \overline{m}_{1,1}(S_2') \cdot m_{0,0}(S_2')}{(\overline{m}_{2,0}(S_1) - \overline{m}_{0,2}(S_1)) \cdot m_{0,0}(S_1) + (\overline{m}_{2,0}(S_2') - \overline{m}_{0,2}(S_2')) \cdot m_{0,0}(S_2')} \\ &= \frac{2 \cdot \overline{m}_{1,1}(S_1) \cdot m_{0,0}(S_1) + 2 \cdot \mathbf{r}^6 \cdot \overline{m}_{1,1}(S_2) \cdot m_{0,0}(S_2)}{(\overline{m}_{2,0}(S_1) - \overline{m}_{0,2}(S_1)) \cdot m_{0,0}(S_1) + \mathbf{r}^6 \cdot (\overline{m}_{2,0}(S_2) - \overline{m}_{0,2}(S_2)) \cdot m_{0,0}(S_2)}.\end{aligned}$$

Obviously, the influence of S_2' to the computed orientation of S could be very big, if the dilation factor $\mathbf{r}$ is much bigger than 1. This suggests a modification of (5.3) to enforce different weighting (as a function of the components area), assigned to the shape components

$$(5.5)\qquad \frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2 \cdot \sum_{i=1}^{m} \overline{m}_{1,1}(S_i)/m_{0,0}(S_i)}{\sum_{i=1}^{m} (\overline{m}_{2,0}(S_i) - \overline{m}_{0,2}(S_i))/m_{0,0}(S_i)}.$$

If the orientation α of $S = S_1 \cup S_2' = S_1 \cup \mathbf{r} \cdot S_2$ is computed by (5.5) then it satisfies

$$\frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2 \cdot \overline{m}_{1,1}(S_1)/m_{0,0}(S_1) + 2 \cdot \mathbf{r}^2 \cdot \overline{m}_{1,1}(S_2)/m_{0,0}(S_2)}{(\overline{m}_{2,0}(S_1) - \overline{m}_{0,2}(S_1))/m_{0,0}(S_1) + \mathbf{r}^2 \cdot (\overline{m}_{2,0}(S_2) - \overline{m}_{0,2}(S_2))/m_{0,0}(S_2)}.$$

So, the impact of an increase of $\mathbf{r}$ to both nominator and denominator is smaller than if (5.3) is applied directly (see (5.4)).

Further, it is not difficult to imagine situations where the size of components should have no effect on the computed shape orientation. For instance, objects (i.e., components of a compound object) may be of the same size in nature, but appear as objects of a different size in the image due to varying distances from the camera. If we would like to avoid any impact of the size of the components to the computed orientation, then we can use the following formula

$$\frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2\cdot\sum_{i=1}^{m}\overline{m}_{1,1}(S_i)/(m_{0,0}(S_i))^2}{\sum_{i=1}^{m}(\overline{m}_{2,0}(S_i)-\overline{m}_{0,2}(S_i))/(m_{0,0}(S_i))^2}. \tag{5.6}$$

In the view of the previous simple example, the computed orientation of $S = S_1 \cup S_2' = S_1 \cup \mathbf{r}\cdot S_2$ is the same for each $\mathbf{r} > 0$. Indeed, if the last formula is applied then the computed orientation α satisfies

$$\frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{2\cdot\overline{m}_{1,1}(S_1)/(m_{0,0}(S_1))^2 + 2\cdot\overline{m}_{1,1}(S_2)/(m_{0,0}(S_2))^2}{(\overline{m}_{2,0}(S_1)-\overline{m}_{0,2}(S_1))/(m_{0,0}(S_1))^2 + (\overline{m}_{2,0}(S_2)-\overline{m}_{0,2}(S_2))/(m_{0,0}(S_2))^2}.$$

Thus, $\mathbf{r}$ does not have any impact to the computed orientation.

The behaviour of the shape orientation method, based on the new multi-component approach, introduced in [55] is demonstrated on several examples.

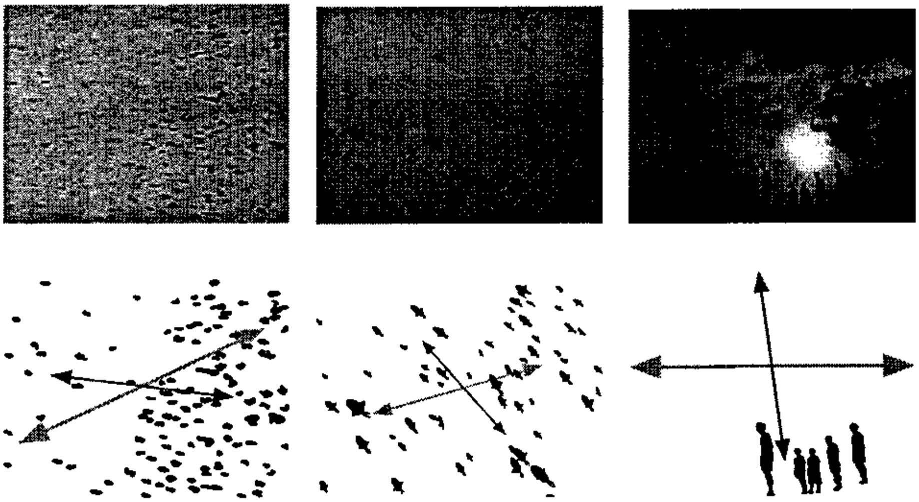
First, we consider examples in Figure 16. Obvious difficulties are apparent in Figure 16(a) in which most of the components are at best only moderately oriented, while many have no distinct orientation, leading to considerable variability in individual orientation estimates. Nevertheless, the overall orientation (formula (5.5) is used) is correctly determined.

In Figure 16(b) a shoal is presented. The orientation of most of fish in the shoal is well defined. Fish orientations (in the most cases) are coincident with the direction of shoal motion. The same orientation is obtained if the formula (5.5) is applied to the shoal as a compound object. Notice that the standard method gives an unacceptable shoal orientation.

Similar discussion holds for the silhouettes of men presented in Figure 16(c).

Images in Figure 17 demonstrates an interesting and very desirable property of the multi-component approach. The central pair of arrows are the orientations shown previously in Figure 16. In addition, the each image was split into two parts by a vertical line, and the orientations were calculated separately for each sub-set of components. Whereas the the new method produces a consistent orientation in both cases, the standard method's orientation varies considerably and does not much our perception.

The third example is in Figure 18 and is related to the, so called, outlier detection problem. More precisely, the application is gait recognition and the binary data is taken from the NLPR Gait Database [46]. The binary masks were generated and provided by Wang et al. [46] using background subtraction but, as noted by the authors, many segmentation errors remain. This causes that the silhouettes are often fragmented into multiple components. Although most of these can be readily corrected using standard morphological operations there remain larger errors that would need to be identified and processed separately. To apply the multi-component shape approach here, the set of blobs in each image frame is considered as a single component. So far, we considered multiple components distributed spatially within a single image, in this application the multiple components are distributed



(a) blood cells (b) shoal (c) people

FIGURE 16. Presented objects are treated as components of a multi-component shape. Orientations were computed by (5.5)–shorter black arrows, and by the standard method–long gray arrows.

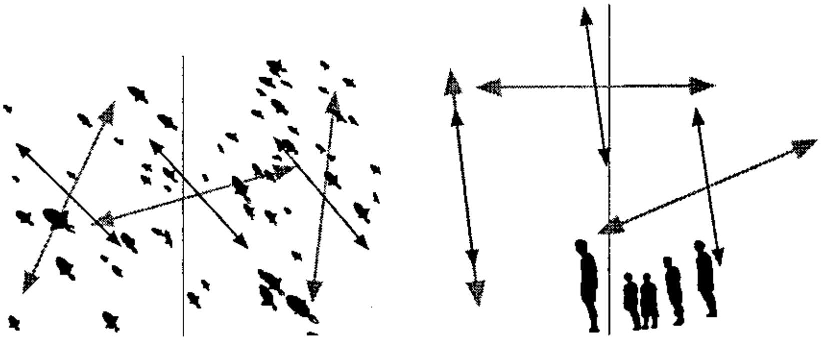


FIGURE 17. Orientations computed separately for the left and right halves of the images, and also for the complete images. Orientations computed by (5.5) are given by shorter dark arrows; orientations computed by the standard method are given by long light arrows.

temporally across the sequence of images. Since we assume that all frames in the sequence have the same importance, the weighting for each component is set to be independent of size, and therefore (5.6) is used to compute orientation. Also, a natural expectation is that if the components are fairly consistently oriented, then faulty segmentation is likely to result in atypical component orientations. Two examples are given in Figure 18. The difference in orientation caused by removing the least consistent component (i.e., image frame) is computed for up to half the number of components, and the frames are replotted with darkness proportional to their difference values. The remaining half of the frames are considered as inliers and their differences are ignored. It can be seen that in Figure 18(a) there has been some kind of error in the original processing chain that produced the binary images, and the person's leading leg has been displaced. In Figure 18(b) the quality of change detection at the beginning of the sequence is poor, which is more visible from Figure 18(c) and Figure 18(d). In both cases these segmentation errors have been identified as orientation outliers (indicated by the darker frames below). In Figure 18(b) there is a second instance of poor change detection two thirds of the way through the sequence, which has not been clearly identified as containing outliers.

6. Conclusion

In this article we have given an overview of some of standard shape based technique used in object classification, object recognition and object identification tasks. Also, some recent developments were discussed, mainly those introduced by the author and his collaborators. We have started with area based shape descriptors, which were most popular in the past because of their robustness and simplicity. Moments, Hu moment invariants, shape elongation, and the standard method for the computation of shape orientation are overviewed. A new circularity measure, derived from the first Hu moment invariant is also studied and its extension to a family of circularity measures is introduced. This illustrates that the classification efficiency can be improved by using several measures devoted to estimate the same shape property (in this case, the shape circularity). Illustrative experiments are provided in order to explain how the methods presented work.

The next, attention has been paid to the boundary based shape descriptors. Boundary based shape descriptors become more popular, in the recent days, because they are more sensitive and can be used in high precision tasks. Particularly, they are suitable for object and person identification tasks, which is very important due to a strong demand for high precision identification tools (e.g., in crime prevention applications). Some area based shape descriptors can be easily extended to their boundary based analogues (e.g., Hu moment invariants [9]), but this is not always possible. It is worth mentioning that boundary based shape descriptors have additional advantages over area based ones. For example, boundary based descriptors are able to cope with objects with particularly extracted boundaries, they are usually faster to compute, and they can be applied to the objects which are linear in their nature (digits, signatures, face details, etc).

Very often, different applications require different method performances. Thus, tuning possibilities are always welcomed when a shape descriptor/measure is created. It has been described here how to define the shape orientation by using the curvature (at the shape boundary points) as a possible tuning parameter. Several

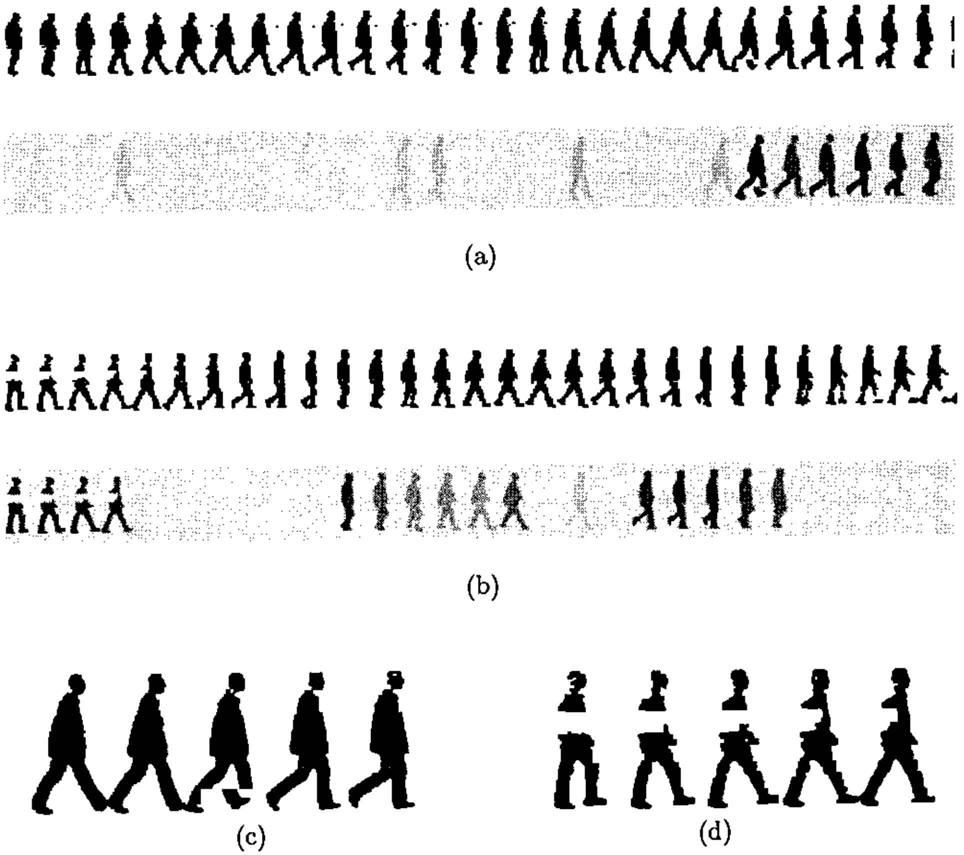


FIGURE 18. Two gait sequences. (a),(b) For each sequence the extracted silhouettes are displayed and underneath is an intensity coding of each silhouette to show its degree of being an outlier (dark means high likelihood). (c),(d) a magnified view of the most outlying silhouette from each sequence and its neighbours.

benefits are obtained. E.g., the behavior of a particular measure can be controlled, more distinct shape orientations can be computed, shapes which are not orientable by certain methods become orientable when use a suitable choice of the tuning parameters, etc. In addition, we have discussed one of convexity measures which can be applied to open curve segments. Simple examples which illustrate the applicability of such a convexity measure to the digits recognition and signature classification tasks are given.

Finally, a recent multi-component shape analysis approach is discussed. Namely, very often it is better to consider a group of objects as a single multi-component object (fish shoal, vehicles on a road, etc). Also, sometimes is more convenient to treat a single object, as a multi-component one, consisting of naturally defined components (e.g., cellular materials like bones, artificially tailored materials, words or signatures decomposed onto characters, etc). Another possibility is to consider an appearance of the same object in a frame sequence as a multi-component object. An application to the detecting outliers, in a sequence of images, is given. In a similar way, an unusual behavior of a person can be detected (could be of an interest in the crime prevention).

To close this overview article, the author believes that challenges and possibilities for further performance improvements, in object classification, recognition and identifications applications, lie in new boundary based approaches, rather than in the area based ones.

Due to the luck of space, $3D$ shape descriptors are not discussed. Just to mention that the recent developments in image technologies made $3D$ data widely available. Also, methods for the reconstruction of $3D$ objects from the corresponding $2D$ images are already well established. Notice that shape descriptor techniques are particularly suitable when working in $3D$ space. This is because shape descriptors, as global $3D$ object features [5, 25, 28], require much less time for the processing than local based features do (simply, there can be too many of them when working in $3D$ space).

Acknowledgement. This work is partially supported by the Serbian Ministry of Education and Science, the projects ON174008, "Advanced Techniques of Cryptology, Image Processing and Computational Topology for Information Security", and III44006, "Development of new information and communication technologies, based on advanced mathematical methods, with applications in medicine, telecommunications, power systems, protection of national heritage and education".

References

[1] N. Alajlan, M. S. Kamel, G. H. Freeman, *Geometry-based image retrieval in binary image databases*, IEEE Trans. Patt. Anal. Mach. Intell. **30** (2008), 1003–1013.

[2] E. T. Bowman, K. Soga, T. Drummond, *Particle shape characterization using Fourier analysis*, Geotechnique **51** (2001), 545–554.

[3] L. Boxer, *Computing deviations from convexity in polygons*, Pattern Recognition Letters **14** (1993), 163–167.

[4] J. M. H. du Buf, M. M. Bayer (eds), *Automatic Diatom Identification*, World Scientific, 2002.

[5] E. Bustos, D. A. Keim, D. Saupe, T. Schreck, D. V. Vranic, *Feature-based similarity search in object databases*, ACM Comput. Surv. **37** (2005), 345–387.

[6] J. Cortadellas, J. Amat, F. De la Torre, *Robust normalization of silhouettes for recognition applications*, Pattern Recognition Letters **25** (2004), 591–601.

[7] D. Coeurjolly, R. Klette, *A comparative evaluation of length estimators of digital curves*, IEEE Trans. Patt. Anal. Mach. Intell. **26** (2004), 252–257.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, *Active shape models-their training and application*, Computer Vision and Image Understanding **61** (1995), 38–59.

[9] C.-C. Chen, *Improved moment invariants for shape discrimination*, Pattern Recognition **6** (1993), 683-686.

[10] H. Freeman, R. Shapira, *Determining the minimum-area encasing rectangle for an arbitrary closed curve*, Comm. ACM **18** (1975), 409–413.

[11] G. H. Granlund, *Fourier preprocessing for hand print character recognition*, IEEE Trans. Computers **21** (1972), 195–201.

[12] J. Flusser, T. Suk, *Rotation moment invariants for recognition of symmetric objects*, IEEE Transactions Image Processing, textbf15 (2006), 3784–3790.

[13] J. Flusser, J. Kautsky, F. Sroubek, *Implicit moment invariants*, International J. of Computer Vision **86** (2010), 72–86.

[14] V. H. S. Ha, J. M. F. Moura, *Affine-permutation invariance of 2-D shapes*, IEEE Transactions Image Processing **14** (2005), 1687–1700.

[15] R. M. Haralick, *A measure for circularity of digital figures*, IEEE Transactions Systems, Man and Cybernetics **4** (1974), 394–396.

[16] M. Hu, *Visual pattern recognition by moment invariants*, IRE Trans. Inf. Theory **8** (1962), 179–187.

[17] M. N. Huxley, *Exponential sums and lattice points. III*, Proc. London Math. Soc. **87** (2003), 591–609.

[18] A. K. Jain, A. Vailaya, *Shape-based retrieval: A case-study with trademark image databases*, Pattern Recognition **31** (1998), 1369–1390.

[19] X. Y. Jiang, H. Bunke, *Simple and fast computation of moments*, Pattern Recognition **24** (1991), 801–806.

[20] R. Kakarala, *Testing for convexity with Fourier descriptors*, Electronics Letters **34** (1998), 1392–1393.

[21] R. Klette, A. Rosenfeld,*Digital Geometry*, Morgan Kaufmann, San Francisco, 2004.

[22] R. Klette, J. Žunić, *On Discrete Moments of Unbounded Order*, LNCS **4245** (2006), 367–378.

[23] L. J. Latecki, R. Lakämper, U. Eckhardt. *Shape descriptors for non-rigid shapes with a single closed contour*, In Proc. Conf. Computer Vision Pattern Recognition, pp. 1424–1429, 2000.

[24] J.-G. Leu, *Computing a shape's moments from its frontier*, Pattern Recognition **24** (1991), 949–957.

[25] Z. Lian, P. L. Rosin, X. Sun, *Rectilinearity of 3D meshes*, Int. J. Comput. Vis. **89** (2010), 130-151.

[26] R. R. Martin, P. L. Rosin, *Turning shape decision problems into measures*, Int. J. Shape Modelling **10** (2004), 83-113.

[27] C. Martinez-Ortiz, J. Žunić, *Curvature weighted gradient based shape orientation*, Pattern Recognition **43** (2010), 3035–3041.

[28] C. Martinez-Ortiz, J. Žunić, *Tunable cubeness measures for 3D shapes*, Pattern Recognition Letters **32** (2001), 1871–1881.

[29] D. L. Page, A. Koschan, S. R. Sukumar, B. Roui-Abidi, M. A. Abidi, *Shape analysis algorithm based on information theory*, In Int. Conf. Image Processing, volume 1, pp. 229–232, 2003.

[30] D. Proffitt, *The measurement of circularity and ellipticity on a digital grid*, Pattern Recognition **15** (1982), 383–387.

[31] E. Rahtu, M. Salo, J. Heikkilä, *A new convexity measure based on a probabilistic interpretation of images*, IEEE Trans. Patt. Anal. Mach. Intell. **28** (2006), 1501–1512.

[32] R. M. Rangayyan, N. M. Elfaramawy, J. E. L. Desautels, O. A. Alim, *Measures of acutance and shape for classification of breast-tumors*, IEEE Transactions Medical Imaging 16 (1997), 799-810.

[33] P. L. Rosin, *Measuring shape: Ellipticity, rectangularity, and triangularity*, Machine Vision and Applications **14** (2003), 172–184.

[34] P. L. Rosin, *Computing global shape measures*, In C. H. Chen and P. S. P. Wang, eds, *Handbook of Pattern Recognition and Computer Vision*, pp. 177–196, World Scientific, 2005.

[35] P. L. Rosin, *A two-component rectilinearity measure*, Computer Vision and Image Understanding **109** (2008), 176–185.

[36] P. L. Rosin, C. L. Mumford. *A symmetric convexity measure*, Computer Vision and Image Understanding **103** (2006), 101–111.

[37] P. L. Rosin, J. Žunić, *Probabilistic convexity measure*, IET Image Processing **1** (2007), 182–188.

[38] N. Sladoje, J. Lindblad, *High precision boundary length estimation by utilizing gray-level information*, IEEE Trans. Patt. Anal. Mach. Intell. **31** (2009), 357–363.

[39] D. Shen, H. H. S. Ip, *Optimal axes for defining the orientations of shapes*, Electronic Letters **32** (1996), 1873–1874.

[40] M. Sonka, V. Hlavac, R. Boyle, *Image processing, analysis, and machine vision,* Thomson-Engineering, 2007.

[41] H. I. Stern, *Polygonal entropy: a convexity measure*, Pattern Recognition Letters **10** (1989), 229–235.

[42] M. Stojmenović, J. Žunić, *Measuring linearity of planar point sets*, Pattern Recognition **41** (2008), 2503–2511.

[43] M. Stojmenović, J. Žunić. *Measuring elongation from shape boundary*, J. of Mathematical Imaging and Vision **30** (2008), 73–85.

[44] H. Süsse, F. Ditrich, *Robust determination of rotation-angles for closed regions using moments*, In Int. Conf. Image Processing, volume 1, pp. 337–340, 2005.

[45] W. H. Tsai, S. L. Chou, *Detection of generalized principal axes in rotationally symetric shapes*, Pattern Recognition **24** (1991), 95–104.

[46] L. Wang, T. N. Tan, W. M. Hu, H. Z. Ning, *Automatic gait recognition based on statistical shape analysis*, IEEE Trans. Image Processing **12** (2003), 1120–1131.

[47] D. Xu, H. Li, *Geometric moment invariants*, Pattern Recognition **41** (2008), 240–249.

[48] H. Zabrodsky, S. Peleg, D. Avnir, *Symmetry as a continuous feature*, IEEE Trans. Patt. Anal. Mach. Intell. **17** (1995), 1154–1166.

[49] J. Žunić, L. Kopanja, J. E. Fieldsend, *Notes on shape orientation where the standard method does not work*, Pattern Recognition **39** (2006), 856–865.

[50] J. Žunić, P. L. Rosin, *Rectilinearity measurements for polygons*, IEEE Trans. Patt. Anal. Mach. Intell. **25** (2003), 1193–1200.

[51] J. Žunić, P. L. Rosin, *A new convexity measurement for polygons*, IEEE Trans. Patt. Anal. Mach. Intell. **26** (2004), 923–934.

[52] J. Žunić, P. L. Rosin, *Convexity measure for shapes with partially extracted boundaries*, Electronics Letters **43** (2007), 380–382.

[53] J. Žunić, P. L. Rosin, L. Kopanja, *On the orientability of shapes*, IEEE Trans. Image Processing **15** (2006), 3478–3487.

[54] J. Žunić, M. Stojmenović, *Boundary based shape orientation*, Pattern Recognition **41** (2008), 1785–1798.

[55] J. Žunić, P. L. Rosin, *An alternative approach to computing shape orientation with an application to compound shapes*, Int. J. Comput. Vision **81** (2009), 138–154.

[56] J. Žunić, K. Hirota, P. L. Rosin, *A Hu invariant as a shape circularity measure*, Pattern Recognition **43** (2010), 47–57.

Nataša Sladoje* and Joakim Lindblad*

THE COVERAGE MODEL AND ITS USE IN IMAGE PROCESSING

Abstract. The coverage model provides a framework for representing continuous objects present in digital images as spatial fuzzy subsets. Assigned membership values indicate to what extent image elements are covered by the imaged objects. We present the basic definitions and properties of this model and show how it can be used to improve information extraction from digital images and to reduce problems originating from limited spatial resolution. We describe a number of image segmentation methods that result in coverage representations. We present methods for estimating geometric moments and object perimeter from coverage representations and derive the corresponding maximal estimation errors as functions of sampling density and number of quantization levels. Compared to a classic binary approach the coverage model provides greatly increased precision. We show how to generate an appropriate binary representation from a coverage one, and also how to use the information rich coverage representation to reconstruct a binary representation at an increased resolution. Empirical studies as well as presented image analysis applications demonstrate the practical advantages of the coverage model and the superior performance of the described methods.

Mathematics Subject Classification (2010): 68U10, 65D18

Keywords: digital image analysis, object representation, image segmentation, feature extraction, error estimation, fuzzy sets

**Faculty of Technical Sciences, University of Novi Sad, Serbia*

Contents

1. Introduction

The interdisciplinary field of imaging science, including image processing, image analysis, image understanding and visualization, is undergoing a very rapid development. Closely tied to advancements in technology, digital imaging and digital image processing, have grown to, not only become very important parts of the scientific world of information processing, but also to become important parts of our everyday lives, in the sense that imaging and image processing are integrated in both society and science.

Together with the development of science and technology, imaging has become a rather general term, related not only to, as traditionally understood, capturing of light reflected of a two-dimensional (2D) surface, but also the measuring many other physical properties of objects of interest, in two or more dimensions. Such different types of imaging, naturally create rather different types of images. These images have in common the acquisition of, in some sense, spatially distributed measurements and they, in general, constitute a valuable source of information about the observed objects. "Visible" is not always in focus any longer; the challenge became to capture images of objects we cannot see (distant stars, atomic-size objects, parts of a living human body, unborn babies, blood flow), or images of unconventional properties of objects (heat, density, water content, etc). To be able to understand and interpret such images, the observer has to know what physical property is expressed and how that property relates to the intensity levels expressed in the image.

Essentially all imaging techniques provide some kind of geometric information about the object: some provide information about anatomy and/or function (e.g., magnetic resonance angiography–MRA, positron emission tomography–PET), some show topographic properties of an object (radar, ultrasound), others may provide very detailed spectral or temporal information (hyperspectral or high speed cameras). The acquired data are, in general, organized in a way that preserves some spatial structure of the object of interest, even if that is not necessarily the main observed property; this analogy with the traditional concept of an image is why the process of creating such data structures is called imaging, and the data themselves–images.

Digital images. In most application areas of imaging sciences, information about some objects of interest, captured in images, needs to be extracted, visualized, manipulated and analysed. When addressing such tasks, we more and more rely on the power of computers. Computers can handle huge amounts of data and accomplish many tasks, primarily those defined in terms of processing large sets of numerical values, much faster, and more reliably than humans can. Connected with digital computers inability to represent continuous information, the imaging process is generally assumed to, instead of capturing a continuum of an observed piece of space containing objects of interest, only observe a sample of points. In such a way, the image domain is discretized and mapped onto a discrete set of points.

The image sample points are often regularly distributed in a grid, and are, in many cases, addressed by integer coordinates. Each such grid point, in some sense, represents a portion of the observed continuous space (often a Voronoi region of the point, i.e., the part of the image space which is closer to the observed grid point than to any other grid point). Every image element (called pixel in 2D, voxel in 3D, or spel–spatial element–in the general case) is, in the imaging process, assigned a value corresponding to the intensity of the physical property observed in that piece of continuous space.

Despite us referring to the "value" or "intensity" of an image element, the image function may not be a scalar function and these words may well refer to a vector of values instead. For example, in a typical colour image, every image point is assigned, not one, but three intensities: red, green, and blue, or RGB for short.

The observed physical property can essentially never be measured perfectly and without error. In addition, the obtained values are stored in limited memory space in a computer. Therefore, the range of the image function is usually restricted to a set of integers (or fractions with common denominator). This process is called quantization. Discretization (sampling) and quantization, applied together to an, initially continuous, image function (theoretically continuous), lead to what is called a digital image, where both the domain and the range are discrete and limited. The number of sample points per unit (density of sampling) is often referred to as image resolution (spatial, spectral, time), where higher image resolution in general provides more information about the imaged objects, and most often better subsequent analysis results. (It should be noted that this is not really a strict usage of the notion of *resolution*, since it does not say what we actually can resolve in the image; the latter is dependent on the physics of the imaging device and not on the number of pixels in an image.) Unfortunately, to increase image resolution deliberately is seldom possible; resolution is imposed by the imaging conditions. Therefore, a challenge within the field of image analysis is to develop creative methods that are capable to utilize and extract as much as possible from the data that is available. Our work summarized in this paper, is in line with this challenge of overcoming limitations of a given spatial resolution and to increase the quality of image analysis results by utilizing the available information as well as possible.

Segmentation. No matter what physical property is imaged, it is practically never exhibited so that in creates well defined and homogeneous regions. Imprecision is a result of imaging conditions, like noise or limited resolution, but also of the properties of the imaged objects. This makes it difficult to clearly separate and outline different objects appearing in the image. Image segmentation aims at defining the extents of the different objects in the image by partitioning the image into a number of regions characterized by a certain intra-component homogeneity and inter-component discontinuity. This is generally considered to be both the most important and the most challenging task in image processing. A decision if a point belongs to the object of interest, or not, is crucial for the quality of all following analysis steps and is often very difficult to make.

In an ideal case, a one-to-one correspondence between the set of image intensities and the set of image components exists and a partitioning can be based on a straightforward classification of pixel intensities. However, even in such an ideal case, discretization of the continuous image space leads to ambiguous situations where one pixel may be partly covered by more than one object in the image. The intensity assigned to such a pixel is a mixture of the intensities associated with the corresponding "pure" components.

However, segmentation is traditionally performed in a crisp way, where each image element is given only one label, i.e., a pixel is completely associated to one single image component. This type of crisp segmentation does not allow partial belongingness of a pixel to an object, and a hard decision of the belongingness has to be made. Intuitively, some kind of thresholding is applied and the crisp classification of a "mixed" pixel as belonging to only one of the image components is performed. In less ideal and more general cases, the presence of noise excludes the possibility of a straightforward classification based on individual pixel intensities only, and more complex, sophisticated, and task dependent segmentation methods are applied, often utilizing spatial information and/or some type of a priori knowledge in addition. Different segmentation methods deal with (different types of) noise in different, more or less successful, ways; however, the issue of mixed border pixels remains, being caused by discretization itself.

Even for the simplest case, where the image only contains one object and the segmentation task reduces to that of defining what is foreground (object) and what is background, it starts to be clear that a segmentation which leads to a binary (two-valued) image as a result, where object points are mapped to one, or "white", and nonobject points (background) to zero, or "black", cannot handle uncertainties and heterogeneity of object properties very well. Despite the ability of the human visual system to provide an intuitive perception of an object as a whole, also in the presence of vague borders and "variability" in the image, it is observed when looking at small regions of an image, that humans can no longer make clear statements whether elements belong to an object or not. Our perception seems to define belongingness of image elements to an object not in a binary (crisp), but more in a graded, or fuzzy, manner. This observation can be successfully transferred and utilized in the field of digital image processing; to handle uncertainties and heterogeneity of object properties appropriately, the suggested methods should be fuzzy, as well, [1, 62]. More precisely, it is, in general, beneficial to perform a *fuzzy segmentation* of an image. Such an approach allows image elements to belong to an object to some extent, and therefore crisp decisions at this early analysis step are avoided. In this way, the risk of making early wrong decisions about object belongingness is reduced, and a larger amount of information is preserved and can be used later in the process.

The result of a fuzzy segmentation of an image containing a single object is a grey-level image of the object of interest, where object points are "white", background is "black", and grey-levels in between correspond to partial belonging of the points to the object, determined according to intensity, geometric, or other information available from the image.

To fully exploit the fuzzy framework, appropriate mathematical theories and algorithms for handling fuzzy discrete data are needed, not only for image segmentation, but in all steps of the image analysis process. There are many challenges to address and many questions to answer on the way of developing such. To list just a few: How are objects to be mathematically defined in fuzzy digital setting, to best address graded composition and hanging-togetherness of the image elements? How are fuzzy boundaries to be defined satisfying a Jordan boundary property? What are the appropriate algorithms to extract these entities from scenes in such a way to satisfy relevant definitions? After a discrete fuzzy spatial set (object) is extracted, how to proceed with the analysis and what analysis tools to use? How to, in the end, reach crisp nonambiguous results from the fuzzy data?

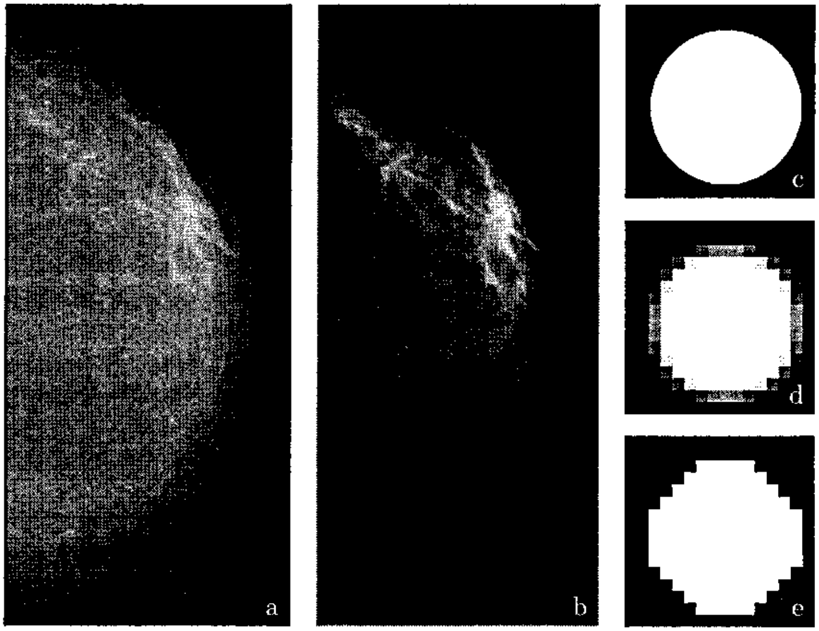


FIGURE 1. Examples of different object representations: (a) grey-level image showing a digitized X-ray mammogram; (b) fuzzy segmentation of a fibroglandular region in (a); (c) high resolution crisp representation of a disk; (d) low resolution coverage representation; and (e) low resolution crisp representation of the same disk.

Coverage representation. In our attempt to contribute to the development of this emerging image processing framework, we have focused our interest to one specific type of fuzzy discrete object representations. These are representations where membership function values correspond to *pixel coverage* (or, in higher dimensional images, spel coverage). Pixel values assigned in this model are equal to the relative area of a pixel covered by the imaged (presumably crisp continuous) object. For such images, pixel values (or, coverage values) range from 0 (assigned to pixels having empty intersection with the object) to 1 (pixels completely covered by the object) and the pixel values strictly between 0 and 1 appear only on the border of an object.

Starting from the idea of such a type of object representations, *coverage representations*, we are working on formulating and developing a general image processing framework that utilizes the benefits that come from appropriately treating the coverage information, while still respecting the discrete nature of digital images. We have conducted a number of studies which show many advantages of the proposed type of coverage representations, compared to crisp (binary) digital image representations. We have developed different feature estimators which utilize the coverage information to improve the estimation precision and accuracy, [50, 52], and have proved that a possible lack of precision resulting from limited spatial resolution may be overcome by properly utilizing grey-level information contained in the images when estimating relevant features of the objects. The encouraging results, proven both theoretically and by empirical studies on synthetic objects, directed our interest to applications of the developed estimators on real images. The first step required for such use was the development of appropriate image segmentation methods that result in a coverage representation. We have suggested several such methods, appropriate for different applications. We further have proposed to utilize the high precision feature values obtained from the coverage information to generate high resolution reconstruction of the observed discrete object, and thereby to "improve" its visual appearance (in crisp representations), too.

In the following sections we describe in more detail our results related to the development of the *coverage model* and its applications in image processing. These results are based on the work presented in a number of publications, where additional details about the individual parts can be found, [28, 30, 33, 34, 50–55]. We will briefly mention some of the applications of the proposed methods, as well [29, 30, 50, 52, 53, 60]. Additionally, we will try to envision some of the possible future research and application directions. We believe they are numerous, since methods that provide results with sub-pixel precision are of highest importance in many fields where precision is a key factor. In addition, analysis of images at low (or simply insufficient) resolution is constantly a hot research topic; with the resent progress in imaging techniques, allowing imaging to reach nanometer scales, a previously inaccessible world of structures of sizes all the way down to molecular scale, opens up. Modern technology, together with humans' curiosity and vision, constantly challenge science to push its limits ever further. Our wish is to be a part of this journey of exciting research.

2. Background and related work

Our work on coverage models is related to several research tracks within the field of image processing. This section, where we list and briefly introduce some of these tracks, aims at providing the reader a wider context for our research.

Initial studies, showing the usefulness of utilizing grey-level information, when analysing black and white 2D images obtained by a scanner, were presented in the early 1990s. Originating from that work, different methods for sub-pixel segmentation evolved. Due to often direct utilization of the image intensities in the algorithms, developed methods usually have strong ties to a specific method of image acquisition. Two sub-fields of image processing, where particularly refined methods for utilizing the intensity information in segmentation have emerged, are remote sensing and tomographic medical imaging.

The notions of *fractional pixels* and *partial volume (tissue fraction) effect* are often mentioned in remote sensing and in tomographic imaging, respectively. Both these notions relate to two distinct phenomena that influence intensity values in images in an undesired way. The first phenomenon which causes inconsistency between ideal and achieved capturing of a signal is the blurring that is introduced by the limited resolving power of the imaging system, leading to a "leakage" of the signal from its actual source to the neighbouring regions in the image. The signal appears weaker, but is also spread over a wider area (the notion of *point spread function* is introduced). The second cause of the above mentioned phenomenon is image sampling. The signal from the imaged object is sampled on a discrete grid, but the contours of the image elements do not match the actual contours of the imaged intensity distribution. A number of spels therefore cover multiple image objects. This second effect is present in any digital image; no matter how high spatial resolution is used, discretization will always lead to that (some) image elements are covered by more than one object. If precision (particularly when it comes to measurements) is required, fractional/partial coverage has to be handled carefully. This challenge is exactly the one we address in our work.

Not only the tasks of image analysis, but also those related to visualisation and image generation impose the need for increased (i.e., sub-pixel) precision in image segmentation and careful handling of image intensities. In film-making, the technique to combine two or more images into a single one, referred to as *image compositing*, or *image matting*, dates back to the Lumière brothers. Since the mid-1980s, when advancements in computer graphics allowed matte painters to work directly in the digital realm, this technique has become less associated with double exposures and painted glass, and more with pixels and alpha channels. In chroma key compositing, commonly used for weather forecast broadcasts, wherein the presenter appears to be standing in front of a large map, which in the studio is actually a monochrome blue background, careful treatment of partially covered image elements, to avoid creation of a bluish aura around the presenter, is required. The field of computer graphics also includes a significant amount of work related to anti-aliasing, aiming to reduce the visual disturbance caused by representing smooth objects by square pixels on a screen. Also in this task, a careful treatment of partially covered pixels is most important. Both these techniques have connections to the work presented in this paper.

In our work on development of image analysis methods, our intention is to propose algorithms and approaches which are generally applicable in a range of situations and applications. Therefore, we try to avoid connecting our proposed methods to any particular way of image creation/acquisition. The foundation of the developed framework is, very suitably for the intended generality, in the fuzzy set theory, which provides a both flexible and powerful framework to represent and describe the methods at an abstract level, without ties to applications, or, if desired, even without ties to digital images. This text will, however, mainly stay close to conventional digital images.

In the following of this section, we give some more details about the background within the field of image processing, and, at the end, a brief overview of the concepts and notions of the fuzzy set theory, used in our work.

2.1. Grey-scale information. The relation between the spatial resolution, the grey-level quantization (grey-level resolution), and the achievable reconstruction accuracy for certain types of images/objects, with a similar motivation as for our later research, is studied in [20]. It is shown that objects with straight edges can be reconstructed without error if grey-levels are not quantized, even though the spatial resolution of an image is limited. In other words, it is concluded that low-resolution grey-scale images of polygonal silhouettes induce less ambiguity than high-resolution bi-level images. This has served as an important inspiration for our work.

To improve the accuracy and precision of local estimators, estimation methods that utilize grey-level information in images have been suggested. In [11], an arc length estimation method that uses normal vectors computed from intensity values, at a number of pixels sampled along the boundary of an object, is presented; arc length is estimated as a cumulative sum of the length of short line segments, derived from the normal directions. A local step may in that way be assigned a variety of normal directions, instead of the very limited set of normal directions available for (Freeman style) local estimators on binary images. Our work on perimeter estimation (see Section 5.3) is based on the same idea that increased precision of normal directions estimation leads to increased precision of perimeter estimates. However, a thorough analysis of the performance of the method presented in [11] is not provided and no optimization of the local lengths is performed. Another approach to increase the precision of measurements is presented in [63]. The method is based on transformation of object boundaries in grey-level images into corresponding volumes, where the length estimation problem is converted into a (simpler) problem of volume estimation. The method relies on sampling theory and discrete approximations of analogue filters. The results are encouraging, but the evaluation is unfortunately only performed on discs of increasing radii, thereby somewhat limiting the possible conclusions. The method also includes some "practical choices" without full theoretical justification. Even though our approach to the same problem differs from the one in [63], it is important to notice that a possibility to increase precision of image analysis results by utilization of grey-levels has attracted attention of a number of researchers during last couple of decades. It still does.

2.2. Remote sensing–fractional pixels. It is not surprising that the issue of mixed pixels is thoroughly addressed in remote sensing applications. Pixels in remotely sensed images are of sizes ranging from a couple of meters to a couple of kilometers, which very often leads to individual pixels being covered by different classes/objects imaged on the ground. To assign the whole pixel to one class (even if that is the class mostly covering the pixel) leads to imprecision which is often intolerable. Estimation of partial coverage (also known as *soft classification*) of a pixel by all individual classes is preferred. Most often used approaches for such sub-pixel proportion estimation are linear mixture models, due to their simplicity. In more complex cases, e.g., due to multiple scattering leading to nonlinear mixtures, or requirements for advanced corrections of atmospheric distortions, rather involved and specialized methods for estimating partial pixel coverage values have been required. Most popular among them are based on neural networks [19].

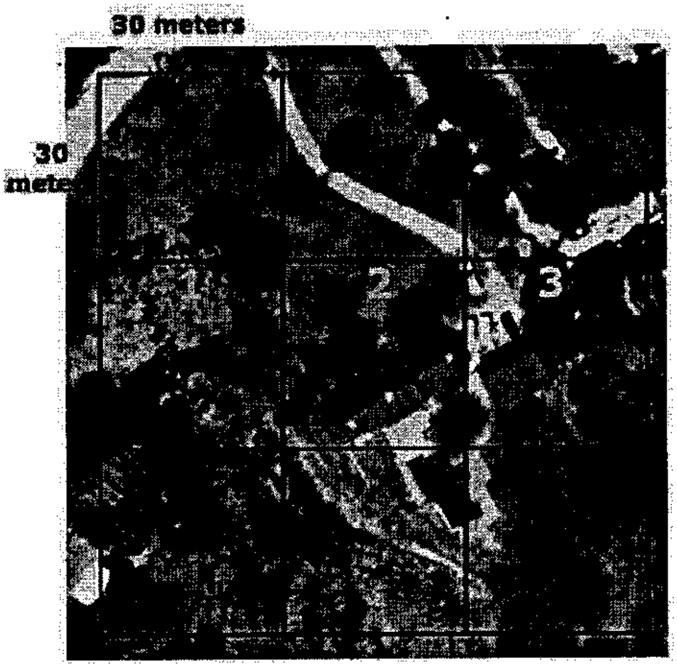


FIGURE 2. A high resolution aerial photograph, with a superimposed 30 meter resolution grid, illustrating the pixel size of a Landsat 7 satellite image. Pixel #1 is a fairly homogenous pixel, almost completely covered with trees, whereas pixels #2 and #3 are mixed pixels, partially covered by vegetation, buildings, and road.

An important characteristics of up-to-date remote sensing is utilization of spectral imaging systems. Spectral imaging for remote sensing of terrestrial features and objects arose as an alternative to high-spatial-resolution large-aperture satellite imaging systems. This type of imaging has evolved to include, instead of just one (grey-scale) band or a few colour bands, several hundred or more bands, encompassing not only the visible spectrum, but also parts of the surrounding electromagnetic spectrum, as well. Data coming from many wavelengths can provide very useful information about the materials in a scene, however extraction of such data usually requires sophisticated processing methods. This is, therefore, an important research direction in remote sensing. It does not fully coincide with our research interest, which is more focused on extraction of information from spatial, rather than from (multi)spectral data. The aim is, however, the same in both cases: to precisely determine the content of a pixel, at sub-pixel precision.

One observation, made in [13], adds additional connection between our work and research interests in remote sensing; it is emphasized that knowledge about the class composition of every pixel still does not provide any information about the spatial distribution of the classes within the pixel. This information can be important

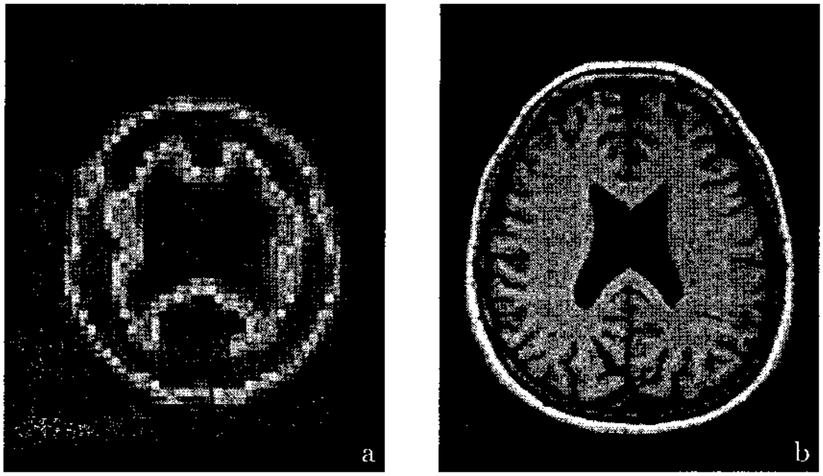


FIGURE 3. A 2D slice from a SPECT image (a), and an MR image (b), showing the same region of a brain.

and one way to acquire it is seen in a multiscale approach, i.e., in utilization of the possibility to observe the environment at a range of scales. This idea is to some extent explored in our work on object reconstruction (see Section 6), where a multiscale approach is taken in the task of high resolution object reconstruction.

2.3. Tomographic images–partial volume effect. Partially covered image elements attract significant attention in medical imaging, not only in cases of low spatial resolution (such as, e.g., SPECT or PET), but also in higher-resolution imaging, such as MRI or CT. Figure 3 illustrates difference in visual appearance due to, among other reasons, difference in spatial resolution, between (two 2D slices of) SPECT and MR images. To appropriately address this issue, in tomographic imaging known as *partial volume effect* (PVE), is particularly important when accurate measurements are required from the acquired images. The significance of this problem, and the need for sub-voxel precision, is well illustrated in [36], where it is shown that consistently misplacing the tissue borders in a brain volume having voxels of size 1 mm^3 by only a single voxel in each slice, resulted in volume errors of approximately 30%, 40%, and 60% for white matter, grey matter and cerebrospinal fluid (CSF), respectively. Negative effects of PVE on tumour detection and monitoring, and on therapy control based on PET images (an imaging modality ideal for this purpose) are thoroughly described in [56].

The complexity of the shape and structure of the human brain, specificities of imaging techniques, and demand for high quality visualization and high precision of (primarily volume) measurements have resulted in a significant number of studies and publications introducing a number of methods for partial volume effect correction in 3D medical imaging. First approaches were not focused on the PVE at a pixel level, but rather on improved estimation of total volume of each tissue in the whole image, [46]. Further work led to approaches aiming at assigning, to each voxel in an image, an estimated portion of each of the tissues that is contained in it. Often used for that purpose are methods based on expectation-maximization, e.g. [25], scale-space approaches [65], wavelets [3], Markov random fields [5], fuzzy techniques, e.g. [57], etc. Different assumptions can be made, which leads to unmixing models of different complexities; a unifying framework for partial volume segmentation of brain MR images, presented in [25], gives a nice overview.

As opposed to remote sensing, medical tomographic imaging does not rely on a range of spectral bands, but more on the spatial distribution of grey-scale intensities, often in combination with a priori anatomical knowledge. In that sense research conducted to address PVE corresponds more to our main interests. However, our developed methods are more general and less tied to particular imaging situations than what is common for methods developed for handling PVE in medical imaging.

2.4. Fuzzy set theory in image processing. A fuzzy set is a collection of elements with a continuum of grades of membership; it is characterized by a membership function, which assigns a membership value between zero and one to each element. A fuzzy set is a generalization of a crisp set; while a crisp set either contains a given element, or it does not, which is described by the membership values one and zero, respectively (as given by the characteristic function of a set), belongingness of an element to a fuzzy set can be partial, and is therefore described by any value between zero and one. When introduced by Zadeh [68], the notion of a fuzzy set was intended to provide a starting point for the building of a conceptual framework, to exist in parallel with the framework of crisp ("ordinary") sets, but to be more general and potentially provide increased applicability in different fields; image analysis became one of them. The framework provided a natural way of dealing with problems in which the source of imprecision is in the absence of sharply defined criteria for class membership.

Having on mind the difficulties in image segmentation, mainly caused by the existence of nonsharp boundaries between the objects in an image, it is not surprising that the comfortability of fuzzy sets, not forcing us to make hard (and possible wrong) decisions about object belongingness, became appreciated and well accepted in image analysis; for an overview of several applications, see [42].

A fuzzy membership function is defined as a mapping from an arbitrary set, the reference set, to, usually, the interval of real numbers $[0,1]$. More formally, a *fuzzy subset* S of a reference set X is a set of ordered pairs $S = \{(x, \mu_S(x)) \mid x \in X\}$, where $\mu_S : X \to [0,1]$ is the *membership function* of S in X [68].

The crisp set of points having strictly positive memberships to the set S is called the *support* of S, while the *core* of a fuzzy set S contains the points with memberships to S equal to 1 (it is sometimes referred to as the *kernel*). When defined on a discrete domain, the membership function is a discrete function, and a corresponding set is a discrete fuzzy set.

To represent an object in an image, we usually consider a fuzzy set defined on $\mathbb{Z}^2$ or $\mathbb{Z}^3$, being typical spaces of discrete images. Such a set is called a *discrete spatial* *fuzzy set* [1]. When represented in a computer, the number of different membership values is finite; integer values are often used to represent memberships, to increase the speed of computations. In this way, the range of a digital fuzzy function is not the interval $[0,1]$, but rather the set $\{0,1,\dots,\ell\}$. The value ℓ is often equal to 255, or 65 535, which corresponds to 8-, or 16-bit pixel depth (number of bits used to represent a pixel value).

Fuzzy set theory is nowadays rather rich and well developed. However, most of the theoretical results are derived for continuous, analytically defined, membership functions and often strongly rely on the properties and analytical expressions of these functions. On the other hand, membership values of image elements are derived from grey-levels, assigned to the image points during an imaging process, and sometimes, additionally, from a set of criteria designed to capture geometric, structural, and other properties of the imaged object. This makes the membership functions on an image (defining fuzzy objects observed) highly complex and practically never analytically defined. As a consequence, many of the well defined and thoroughly explored notions, relations, and properties of (analytically defined) continuous fuzzy sets become nonapplicable to the discrete fuzzy sets, which are most common in image processing. Therefore, it is often required to design new methods, which are more appropriate for the analysis of discrete fuzzy sets, and to develop mathematical theories and algorithms for handling fuzzy discrete data appearing in digital images.

A representation of a fuzzy set, which is often used as an alternative to representation by a membership function, is the one based on α-cuts. For a fuzzy set F, defined on a reference set X, the following two representations are equivalent [10]:

- a membership function $\mu_F : X \to [0,1]$ which assigns to each $x \in X$ its membership grade $\mu_F(x)$ to the fuzzy set F;
- the set of α-cuts $\{F_\alpha \mid \alpha \in (0,1]\}$ of the set F, where $F_\alpha = \{x \in X \mid \mu_F(x) \geqslant \alpha\}$.

The connection between the membership function and the stack of α-cuts provides a common approach for extending functions defined on crisp sets, to functions defined on fuzzy sets. The so called fuzzification principle, based on one of the following equations:

$$f(S) = \int_0^1 \hat{f}(S_\alpha)\, d\alpha, \tag{2.1}$$

$$f(S) = \sup_{\alpha\in(0,1]} [\alpha \hat{f}(S_\alpha)] \tag{2.2}$$

can be used to extend a function $\hat{f}$ to the domain of fuzzy sets. In this way, various properties defined for crisp sets (here, α-cuts) can be generalized to fuzzy sets, including the membership function itself; if the characteristic functions of the α-cuts are observed, the membership function of the corresponding fuzzy set can be obtained by either their integration (2.1), or by taking the supremum of their weighted values, over the "height" of the stack (2.2). This approach, and in particular equation (2.1), is often used in our work, when some relevant features of objects represented by a coverage model are defined and extracted. Notably, area and perimeter of a spatial fuzzy set in 2D, can be defined based on (2.1) [41, 43].

One particular concept that is extended to the set of fuzzy sets, which we will use in the following, is that of a partition. Classically, a partition of a set S is a family of disjoint nonempty subsets of X whose union is equal to X. An often used definition of a fuzzy partition is the following one [45]: A *fuzzy partition* of a set X is a finite family $P = \{P_1, P_2, \ldots, P_n\}$ of nonempty fuzzy subsets of X such that $\sum_{k=1}^{n} \mu_{P_i}(x) = 1$ for all $x \in X$.

3. The coverage model

The approach that we take to handle partially covered image elements and to best utilize intensity information in order to reach sub-pixel precision of estimates, differs quite significantly from most previously presented work with similar goals. As noticed in the background section, a lot of related work is based on more or less direct usage of the image intensities, leading to strong ties between developed methods and the specific imaging conditions. The path we take is, instead, to start from a well defined abstract theoretical model, with no connections with any particular application. For the proposed theoretical framework, we have developed feature estimators and derived exact results regarding their performance. The connection to specific imaging conditions is handled through a separate segmentation step, which serves the purpose of transforming the application dependent image information into an application independent form. This clear separation between the different parts of the presented approach is what provides generality of the developed image processing tools, where application specific information can be fully utilized in the segmentation step, while still not interfering negatively with the later steps.

The foundation of the proposed model lies in the fuzzy set theory, which provides a framework that has shown to be both powerful and flexible. Within the field of image processing, methods utilizing the concept of fuzzy/graded memberships have found a number of applications, and discrete fuzzy sets have shown to be a very good tool for representing image objects. The ability provided by fuzzy sets to represent uncertain and vague data facilitates development of robust methods which successfully handle noise and image artefacts. At the same time, these methods can be designed to enable high precision of measurements, overcoming well known problems originating from the discretization of the continuous space observed, unavoidably imposed by essential properties of computers and imaging devices involved.

The fuzzy set theory is very general and provides a lot of flexibility; the interpretation of the membership values can be adjusted to any need, property, or application. A great power of this large freedom lies in the possibility to, by adding well chosen restrictions, shape it so that it best fits a particular problem observed. Quite clearly, keeping full generality available in every applications is, in addition to being difficult, also hardly practically useful. Not surprisingly, the large freedom makes it difficult to derive well defined and strong statements about specific properties of the observed (general) fuzzy sets. We found that, by suitably restricting the interpretation of membership values, strong theoretical results are more readily available.

Our research focuses on mathematical tools for representing and analyzing continuous crisp objects in digital n-dimensional images. For that specific task we have appropriately restricted the fuzzy membership function used, and have defined a particular type of fuzzy sets for representation of imaged objects. We refer to such representation of digital objects as *coverage representations.* We have shown that using such a coverage representation has many advantages compared to a traditional crisp representation.

3.1. Basic definitions. We start with a general definition of a coverage representation. We will then restrict it to better fit the application we have on mind, i.e., representing crisp objects in digital images.

Definition 3.1. Given a partition $\Sigma = \{\sigma_i\}_{i\in I}$ of a reference set X, a coverage representation of a set $S \subset X$ on Σ is a fuzzy subset $\{(\sigma_i, \alpha(\sigma_i)) \mid \sigma_i \in \Sigma\}$, such that $\alpha(\sigma_i) = |\sigma_i \cap S|/|\sigma_i|$.

In the context of digital image processing, we assume that the reference set X is the Euclidean space $\mathbb{R}^n$ and that Σ is the Voronoi tessellation of $\mathbb{R}^n$ defined by the set of integer points $\mathbb{Z}^n$. We refer to the Voronoi region of a grid point $x = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n$ as the *spel* at x (short for spatial element) and denote it with $\sigma(x)$. That is, $\sigma(x)$ contains the points of $\mathbb{R}^n$ which are closer to x, in terms of Euclidean distance, than to any other point in $\mathbb{Z}^n$ (for points at equal distance we round upwards, i.e., the lower/left edge in each dimension is included in the spel). In other words, the set Σ^n of nD spels of an integer grid, consists of translations of the right open n-dimensional unit origin-centred cube by vectors $x \in \mathbb{Z}^n$:

$$\Sigma^n = \{\sigma(x) \,|\, x \in \mathbb{Z}^n\}\,, \quad \sigma(x) = \left[-\tfrac{1}{2}, \tfrac{1}{2}\right)^n + x\,.$$

Based on the above we define the following digitization model:

Definition 3.2. For a given continuous object $S \subset \mathbb{R}^n$, inscribed into a grid $\mathbb{Z}^n$, the *coverage digitization* of S is

$$\mathcal{D}_c(S) = \{(x, \alpha(x)) \mid x \in \mathbb{Z}^n\}\,, \quad \alpha(x) = \frac{|\sigma(x) \cap S|}{|\sigma(x)|}\,,$$

where $|X|$ here denotes the area/volume/Lebesgues measure of a set X.

Remark 3.1. To simplify notation, we utilize the one-to-one correspondence between elements $x \in \mathbb{Z}^n$ and their respective spels (Voronoi regions) $\sigma(x) \in \Sigma^n$, and consider a coverage digitization to be a set of ordered pairs from $\mathbb{Z}^n \times [0,1]$, and not from $\Sigma^n \times [0,1]$.

Definition 3.2 assumes assignment of nonquantized real coverage values α to the spels of the grid. However, when using digital approaches (computers) to represent, store, and analyze images, we are limited to a finite number of grey-levels to represent coverage of an individual spel. This leads to the following quantized version of coverage digitization:

Definition 3.3. For a given continuous object $S \subset \mathbb{R}^n$, inscribed into a grid $\mathbb{Z}^n$, the *ℓ-level quantized coverage digitization* of S is

$$\mathcal{D}_c^{\ell}(S) = \left\{(\boldsymbol{x}, \alpha^{\ell}(\boldsymbol{x})) \;\middle|\; \boldsymbol{x} \in \mathbb{Z}^n\right\}, \quad \alpha^{\ell}(\boldsymbol{x}) = \frac{1}{\ell}\left\lfloor \ell\frac{|\sigma(\boldsymbol{x}) \cap S|}{|\sigma(\boldsymbol{x})|} + \frac{1}{2}\right\rfloor,$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than x.

Clearly $|\alpha(\boldsymbol{x}) - \alpha^{\ell}(\boldsymbol{x})| \leqslant \frac{1}{2\ell}$. We denote the set of possible coverage values $\alpha^{\ell}(\boldsymbol{x})$ in ℓ-level quantized coverage digitization by $\mathcal{Q}_{\ell} = \left\{0, \frac{1}{\ell}, \frac{2}{\ell}, \dots, \frac{\ell}{\ell} = 1\right\}$. This set corresponds to the set of grey-levels available; e.g., $\ell = 1$ for a binary image, while $\ell = 255$ provides the set of grey-levels for an 8-bit representation. Similarly as using spatial resolution to denote the spatial sampling density, we let *coverage resolution* denote the number of (meaningful) coverage levels.

The coverage digitization model stands in contrast to the more common Gauss digitization model, where an object is represented by the set of integer grid points within the objects (or, mainly being a matter of notation, by the corresponding set of spels). More formally:

Definition 3.4. The *Gauss digitization* of $S \subset \mathbb{R}^n$ is $\mathcal{D}_g(S) = \{S \cap \mathbb{Z}^n\}$.

In our work, coverage of a spel is sometimes approximated by a super-sampling approach where a spel is split into several sub-spels, and a sample is taken from the centre of each. This facilitates easy approximation of coverage values for more complex synthetic objects, where true coverage may be difficult to compute analytically. Let the r-sampled spel $\hat{\sigma}^r(\boldsymbol{x})$ be the following set of r^n points within $\sigma(\boldsymbol{x})$:

$$\hat{\sigma}^r(\boldsymbol{x}) = \left\{\sigma(\boldsymbol{x}) \cap \left(\frac{\boldsymbol{y} - \delta(r)}{r}\right) \;\middle|\; \boldsymbol{y} \in \mathbb{Z}^n\right\},$$

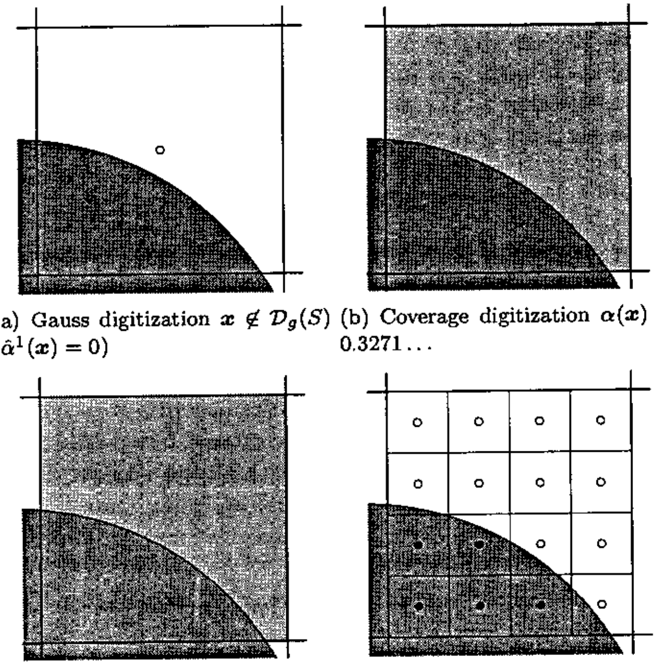
where $\delta(r)$ is the vector $\left(\frac{r-1}{2}, \frac{r-1}{2}, \dots, \frac{r-1}{2}\right)$.

Definition 3.5. For a given continuous object $S \subset \mathbb{R}^n$, inscribed into a grid $\mathbb{Z}^n$, the *r-sampled coverage digitization* of S is

$$\hat{\mathcal{D}}_c^{r}(S) = \left\{(\boldsymbol{x}, \hat{\alpha}^{r}(\boldsymbol{x})) \;\middle|\; \boldsymbol{x} \in \mathbb{Z}^n\right\}, \quad \hat{\alpha}^{r}(\boldsymbol{x}) = \frac{|\hat{\sigma}^r(\boldsymbol{x}) \cap S|}{|\hat{\sigma}^r(\boldsymbol{x})|}.$$

Remark 3.2. Coverage values of a 1-sampled coverage digitization, $\hat{\alpha}^1(\boldsymbol{x})$, correspond to the characteristic function of a Gauss digitization.

Figure 4 illustrates the different digitization approaches and the output for one pixel, partly covered by a disk shaped imaged object S. The observed pixel is not included in the Gauss digitization (Fig. 4(a)), since the centre of the pixel is not covered. For the coverage digitization (Fig. 4(b)), the pixel is associated with a real number $\alpha(\boldsymbol{x}) \in [0, 1]$ indicating how large area of the pixel is covered by the object. In the ℓ-level quantized coverage digitization (Fig. 4(c)), the real coverage is approximated by its closest number in $\mathcal{Q}_{\ell}$. For the r-sampled coverage digitization (Fig. 4(d)), the pixel is divided into sub-pixels, and the coverage is approximated by the number of covered sub-pixel centres, divided by the number of sub-pixels.



(a) Gauss digitization $x \notin \mathcal{D}_g(S)$ ($\hat{\alpha}^1(x) = 0$)

(b) Coverage digitization $\alpha(x) = 0.3271\ldots$

(c) 10-level quantized coverage digitization $\alpha^{10}(x) = 0.3$

(d) 4-sampled coverage digitization $\hat{\alpha}^4(=)\frac{5}{16}$

FIGURE 4. Different digitization approaches. Shown is one pixel $\sigma(x)$, partly covered by a disk shaped object S.

The r-sampled coverage digitization takes values from the same set $\mathcal{Q}_\ell$ as an ℓ-level quantized coverage digitization with $\ell = r^n$. However, the r-sampled digitization has lower accuracy in approximating the true coverage digitization values. For a spel x, intersected by a straight edge of an image object, it holds that $|\alpha(x) - \hat{\alpha}^r(x)| \leqslant \frac{1}{2r}$. Another important property to notice is that the quantized coverage digitization $\mathcal{D}_c^\ell(S)$, with $\ell = r^n$, is never a worse approximation of the true coverage digitization $\mathcal{D}_c(S)$ than the r-sampled coverage digitization $\hat{\mathcal{D}}_c^r(S)$. This holds by definition, since $\mathcal{D}_c^\ell(S)$ and $\hat{\mathcal{D}}_c^r(S)$ take values from the same set, and $\alpha^\ell(x) = \arg\min_{\beta \in \mathcal{Q}_\ell} |\beta - \alpha(x)|$.

3.2. Properties of coverage digitizations. A coverage representation of a crisp real object with a well defined continuous border is, ideally, characterized by the presence of homogeneous connected regions of "pure" spels, completely covered by either object or background. Each two such regions are separated by a thin layer of "mixed" spels, i.e., those partially covered by both object and background. Pure spels are assigned coverage values 0 (background) or 1 (object), while mixed spels are assigned values between 0 and 1, in accordance to their respective coverage by the image object.

Following the terminology of fuzzy sets, we define the *core* and the *support* of a coverage representation as the crisp set of spels with coverage values 1 and the crisp set of spels with nonzero coverage values, respectively:

$$\operatorname{core}(C) = \{x \mid \alpha(x) = 1\}, \quad \operatorname{supp}(C) = \{x \mid \alpha(x) > 0\}.$$

We note that there is a close connection between $\operatorname{supp}(\mathcal{D}_c(S))$ and the *outer Jordan* or *super-cover digitization* of the set S (the union of all spels having a nonempty intersection with S), where a difference between the two notions appears only when the nonempty intersection of a spel and the set S has a measure zero. Such a spel is included in the outer Jordan digitization but not in $\operatorname{supp}(\mathcal{D}_c(S))$. Similarly, for $\operatorname{core}(\mathcal{D}_c(S))$ and the *inner Jordan digitization* of S, where the two notions differ only for spels with a nonempty intersection with the background of measure zero. Such a spel is excluded from the inner Jordan digitization but is not excluded from $\operatorname{core}(\mathcal{D}_c(S))$. These differences, being mainly theoretical in the case of a real valued coverage digitization, become more prominent in the case of quantized or sampled coverage digitizations.

We call the set of "mixed" spels of a coverage representation C, with coverage values strictly between 0 and 1, the *fuzzy border* of C. This set is equal to the closed difference set between $\operatorname{supp}(C)$ and $\operatorname{core}(C)$. Formally:

Definition 3.6. The *fuzzy border* of a coverage representation $C = \{(x, \alpha(x))\}$ is

$$\partial_f C = \{x \mid \alpha(x) \in (0,1)\}.$$

If the crisp set S has a reasonably smooth boundary (i.e., S is Jordan measurable) and digitized at a high enough resolution, then the fuzzy border $\partial_f \mathcal{D}_c(S)$, is not more than one spel thick.

An important property of coverage digitization is that it preserves partitions of the digitized space. Let $\{S_k \mid |S_k| \neq 0, k = 1,2,\ldots,m\}$ be a partition of a reference set X. Then $\{\mathcal{D}_c(S_k) \mid k = 1,2,\ldots,m\}$, is a fuzzy partition [45], i.e., $\forall k : \mathcal{D}_c(S_k) \neq \emptyset$, and $\forall x \in X : \sum_{k=1}^{m} \alpha_k(x) = 1$.

Proof. Follows directly from the additivity of the Lebesgue measure on $\mathbb{R}^n$: for disjoint sets $A, B \in \mathbb{R}^n$, $|A \cup B| = |A| + |B|$. □

We call such a fuzzy partition, where membership values correspond to spel coverage, a *coverage partition* of X.

Remark 3.3. Note that a similar family of ℓ-level quantized coverage digitizations is not necessarily a coverage partition. For example, a 10-level quantized coverage digitization of a spel $\sigma(x)$ equally covered by three objects will lead to $\sum_{k=1}^{3} \alpha_k(x) = 0.9 \neq 1$. A sampled coverage digitization of a partition is, however, always a fuzzy partition.

4. Coverage segmentation methods

The definition of coverage digitization provides a way to compute coverage representations of relatively simple, or analytically defined, continuous subsets of $\mathbb{R}^n$ such as, e.g., simple geometric objects. However, to extract similar information about more realistic and more complex objects in digital images, we instead need segmentation algorithms. To address the task of extracting coverage information about objects in images, we have proposed a number of *coverage segmentation* methods.

As already mentioned, image segmentation is commonly considered to be the process of assigning a label to every spel in an image, so that spels with the same label share certain visual characteristics. Traditionally, such an assignment is done in a crisp fashion, where a spel can only be associated with one single component of the image. More generally, a segmentation may be performed in a fuzzy manner, in which case each spel is associated with a membership, in the range zero to one, to each of the image components. The membership values assigned to one spel do not have to sum up to one, although this is assumed for many methods. For the coverage model it seems reasonable to assign memberships (coverage values) of a spel to different crisp nonoverlapping objects in $\mathbb{R}^n$, so that they do sum up to one.

Let $\mathbb{A}_m$ denote the set of m-component (fuzzy) *segmentation vectors*

$$\mathbb{A}_m = \left\{\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) \in [0,1]^m \;\middle|\; \sum_{k=1}^{m} \alpha_k = 1\right\},$$

and let $\mathcal{A}_m \subset \mathbb{A}_m$ be the corresponding set of crisp *segmentation vectors*

$$\mathcal{A}_m = \left\{\alpha \in \{0,1\}^m \;\middle|\; \sum_{k=1}^{m} \alpha_k = 1\right\}.$$

A *coverage segmentation* of an image I into m components is a set of ordered pairs

$$\mathcal{S}_c(I) = \{(x, \alpha(x)) \mid x \in I_D, \alpha(x) \in \mathbb{A}_m\}, \quad \alpha_k \approx \frac{|\sigma(x) \cap S_k|}{|\sigma(x)|},$$

where $S_k \subset \mathbb{R}^n$ is the extent of the k-th (out of m) image component and $I_D \subsetneq \mathbb{Z}^n$ is the discrete image domain. The continuous sets S_k are, in general, not known, and the values α_k therefore have to be estimated from the image data. We may also refer to a coverage segmentation as a coverage partition of I_D, i.e., a collection of nonoverlapping sets $\mathcal{S}_c(I)_k = \{(x, \alpha_k(x)) \mid x \in I_D\}, k = 1, 2, \dots, m$.

In many imaging situation, acquired image intensities correspond almost directly to coverage values. One such situation is, e.g., the integration of photons over finite sized sensor elements, as present in a digital camera. In the absence of object texture and large illumination variations, a suitable mapping of the image intensities may provide good enough coverage values. In Section 4.1 we present a method for automatically defining such a mapping by a double thresholding scheme. Direct mapping of intensity values does not work well in the presence of large scale intensity variations, however. In Sections 4.2 and 4.3 we present more elaborate methods for performing coverage segmentation.

4.1. Coverage segmentation based on double thresholding. Segmentation by thresholding is the most intuitive and simplest segmentation method. Successful application of any thresholding based segmentation method requires that the intensity distributions of the object and the background are well separable. If a separation between the image components is to be performed in a crisp manner, one threshold value is selected and two disjunct sets of intensities are determined; the corresponding spels are then accordingly classified into two classes.

Assume that we are imaging a bright object on a black background. This situation will, under reasonable conditions, result in two well separated sets of intensity values, and classification (segmentation) can be performed by thresholding. If the imaging device has a linear response, and the optical blurring is small compared to the pixel size, the partly covered pixels along the boundary of the object exhibit grey values between those of the background and the foreground, where the grey value assigned to a pixel is proportional to its coverage by the object. This observation makes it natural to try to develop a segmentation method that, to a high extent, utilizes the grey-levels of the graded transition between the two observed classes (object and background) for estimating coverage values. We notice that such a direct relation between grey-levels and spel coverage is a reasonable model for images where resolution is decided based on limited means for handling of the data rather than on the optical system (e.g. in high speed video), or when detector elements are intentionally grouped together (binned) to reduce photon/Poisson noise (e.g. in low dose CT).

In order to estimate coverage values from the image spel intensities, we need to first estimate the grey-levels of the completely covered foreground and background spels, respectively. Denote with f and b the intensity of the imaged foreground (object) and background, respectively. We then model the intensity $I(x)$ of a partly covered pixel $\sigma(x)$, with coverage $\alpha(x)$, as a convex combination of f and b:

$$I(x) = \alpha(x) \cdot f + (1 - \alpha(x)) \cdot b \quad \Rightarrow \quad \alpha(x) = \frac{I(x) - b}{f - b} \tag{4.1}$$

In [52] we presented a method which is based on the above model and which automatically finds two threshold values, f and b, that define the minimum intensity of the high intensity component (foreground) and the maximum intensity of the low intensity component (background). The threshold selection is based on the observation that coverage representations of crisp continuous objects are characterised by having fuzzy boundaries which are not more than one spel thick (see Section 3.2). This is, however, not enough to uniquely define the threshold; we additionally require the contrast between foreground and background to be as large as possible, giving a border with as rich intensity variations as possible. The initial reason to perform a coverage segmentation was to preserve the information given by the grey-levels; the more we keep and use, the better results of subsequent analysis.

To summarize, given a grey-scale image, we seek a threshold couple, b and f, where spels darker than b are considered to belong completely to the background, while spels brighter than f are considered to belong completely to the foreground, such that the spels in between form a one spel thick separating layer. In addition, we want the contrast between foreground and background, i.e., the difference $\ell = f - b$, to be as large as possible.

The algorithm loops over all background thresholds b'; each of them defines a possible support of the object. Instead of looping over all foreground thresholds, f' is found as the minimum grey-level within the core of the potential object. For an assumed support, the core should be not more than at one spel distance, which is conveniently expressed by mathematical morphology. For each potential support, the corresponding minimal core is found by eroding the support with a $3 \times 3 \times \cdots \times 3$ binary structuring element. The foreground threshold f' is then found as the minimal grey-level in that potential core. However, we noticed that individual dark noise points make f' unnecessarily low. We therefore first perform a morphological closing before selecting the level f'. The difference $f' - b'$ is computed, and a new background threshold is tested. The thresholds leading to the largest intensity range $\ell = f' - b'$ is selected as the best one and, based on that, partial coverage values are computed according to (4.1). We also include an opening of the background, to avoid isolated bright noise points in the background appearing as object points. The processing is invariant w.r.t. intensity inversion; looping over the foreground thresholds and finding b' as the maximum value of the opened background, leads to the same result.

Instead of first performing thresholding, and then morphological operations, we speed up the processing by using grey-scale morphology (where erosion and dilation become min and max filters). The more time consuming morphological operations can then be done once for the whole process, and only threshold and min operations remain inside the loop.

Algorithm 1 summarizes the described steps (see also [52]). We denote the grey-scale erosion $I \ominus B$ by εI and the grey-scale dilation $I \oplus B$ by δI. The opening and closing of I by B are denoted $\delta\varepsilon I$ and $\varepsilon\delta I$, respectively.

An example of the use of this coverage segmentation method is presented in Section 7.1.

4.2. Coverage segmentation by local unmixing. Image segmentation is, as already stated, a difficult problem, which has been addressed more times than any other problem in image processing. As a consequence, very many segmentation algorithms have been proposed. A variety of methods, utilizing different theoretical concepts, and being more or less general, i.e., more or less tied to a particular application, are published. With an increasing complexity of the imaged scene, more information than bare grey-scale intensities of individual spels is required to perform successful (meaningful) separation of image components, which makes thresholding based methods nonapplicable. Spatial (geometric) information about the objects in the image is very often utilized, but other types of a priori knowledge about the imaged objects, if available, may be incorporated in a segmentation method, as well. Segmentation can be performed based on intensity homogeneity preservation (region based methods), or discontinuity detection (boundary based approaches); more and more often, a combination of these approaches is proposed,

Algorithm 1.

Input: A grey-scale image I with a bright object on a dark background.
Output: An approximate ℓ-level coverage segmentation C of the object in I.

```
$b = 0$; $f = 0$
for each grey-level $b'$
  $F' = \{x \mid [\varepsilon I](x) > b'\}$ /* Foreground */
  if $F' \neq \emptyset$
    $f' = \min_{x \in F'} [\varepsilon\delta I](x)$
    if $f' - b' > f - b$ /* Better than previous */
      $f = f'$; $b = b'$
    endif
  endif
endfor
$\ell = f - b$
```

$$\alpha(x) = \begin{cases} 0 & , \quad [\delta\varepsilon I](x) \leqslant b, \\ 1 & , \quad [\varepsilon\delta I](x) \geqslant f, \\ \frac{I(x)-b}{\ell} & , \quad \text{otherwise.} \end{cases}$$

to answer better to the high complexity of the tasks. If available, colour information is precious. Typically, three channels are used in colour imaging, which adds a lot of information compared to monochrome (grey-scale) images. The number of channels may, in some imaging techniques, be even higher (reaching hundreds), leading to so-called spectral images, typical in, e.g., remote sensing (see Section 2.2). Segmentation methods design to extract such type of information are accordingly developed.

Noticing this rich variety of different segmentation methods, more or less refined and adjusted to various imaging conditions, we propose to not "re-invent" a whole range of segmentation methods that should provide good coverage segmentation for a variety of tasks, but instead appropriately adjust already existing crisp segmentation methods. After all, our aim to perform segmentation so that it leads to a coverage representation of objects, is not far from the aim of "traditional" segmentation methods, providing a crisp segmentation. For all image spels which are completely covered by a single object class, there is no difference between a coverage segmentation and a crisp segmentation; for those spels there is no reason for a different output than what is achieved by any appropriately chosen "traditional" segmentation method.

In [53] we presented a method that, based on *any* existing crisp segmentation, enhances it to a coverage segmentation by identifying boundary spels and suitable re-evaluating their coverage values. By this approach we reached two important goals: (i) the segmentation results in a coverage representation of an object; (ii) all the advantages of well chosen crisp segmentation methods for a particular task are preserved and utilized. The methods is briefly described in the following.

To obtain a coverage segmentation, we propose a method composed of four steps:

- application of a crisp segmentation method, appropriately chosen for the particular task;
- selection of spels to be assigned partial coverage;
- application of a local linear mixture model for "un-mixing" of partially covered spels and assignment of corresponding coverage values;
- ordered thinning of the set of partly covered spels to provide one spel thin fuzzy borders (see Section 3.2) of mixed spels.

The first step in the proposed method is expected to provide correct assignment of class belongingness to pure spels. We suggest to utilize any appropriate existing segmentation method, and assume that the resulting segmentation provides a trustworthy result for all but boundary spels. Each spel $\sigma(x)$, inner (i.e., not neighbouring a spel of a different component) for the component k, is assigned crisp segmentation vectors $\boldsymbol{\alpha}(\boldsymbol{x})$ such that $\alpha_k = 1$ and $\alpha_{l\neq k} = 0$.

In the second step of the suggested segmentation method, spels possibly being intersected by the boundaries of continuous imaged objects are to be detected. Such spels are possibly mixed, with partial coverage by two or more image components. We define the set B to consist of all (nD) spels sharing an $(n-1)$-dimensional hyper-surface with a spel assigned a different segmentation label. In the sense of Definition 3.6, these spels are candidate mixed spels, and as such, they will be processed in the next steps of the algorithm. If continuous crisp objects are imaged at a reasonably high resolution, and the segmentation performed in step one correctly labels inner, completely covered spels, then the set of mixed spels will be a subset of the set B. Even though it is clear that, in the presence of noise, inner region spels are not of accurate reference intensity of a pure class, but are often exhibiting properties of mixed spels, the idea is to have confidence in the used crisp segmentation method up to the dichotomization into inner/pure and border spels. The spels detected as inner will, therefore not be revisited, or reassigned.

The third step in the coverage segmentation process is computation of partial coverage values of the (potentially mixed) spels of the set B. We suggest to use a linear model, due to its simplicity, and the fact that it corresponds to the ideal (noise-free) spel coverage assignment that arises when integrating spatially distinct signals over finite sized detector elements (e.g. in a digital camera). This model assumes that the value of a mixed spel is a convex combination of the values corresponding to the pure classes c_k covering the observed spel, where the coefficients in the combination correspond to the proportions of the pure classes appearing. Note that for imperfect imaging devices, the assumption that the value of a spel depends only on the content of that particular piece of the image domain, may not hold. Given a particular imaging situation, it is recommended to verify this assumption and possibly act accordingly, e.g. by incorporating a deconvolution step into the process.

In general, the intensity values of the pure classes are not known, but have to be estimated from the image data. We suggest to use a local approach when estimating the intensities characterizing a class k. For each spel observed in the process of partial coverage assignment, the local pure class representation $c_k(x)$ is estimated as the mean value of the image intensities in a local neighbourhood of a suitable size, which are classified, according to the two first steps, as completely belonging to the observed class k. This approach, in our opinion, has two main advantages: 1) only the relevant classes –existing in the neighbourhood of the observed spel are considered for a mixture in that spel, and 2) sensitivity of the pure class description to intensity variations over the image is decreased; in general, the local within class variation is significantly smaller than the global one.

The image intensity values $I(x) = (I_1, I_2, \dots, I_b)$ of a mixed spel $\sigma(x)$ (b being the number of channels (bands) of the image) are assumed, in a noise-free environment, to be a convex combination of the (locally estimated) m existing pure classes $c_k(x)$:

$$I(x) = \sum_{k=1}^{m} \alpha_k c_k(x)\,, \quad \sum_{i=k}^{m} \alpha_k = 1\,, \quad \alpha_k \geqslant 0\,, \tag{4.2}$$

where each coefficient α_k corresponds to the coverage of the spel $\sigma(x)$ by an object of a class k. In a noise-free environment, and if the number m of classes (variables) is not bigger than the number $(b+1)$ of equations (including the equation $\sum_{k=1}^{m} \alpha_k = 1$), the problem of partial coverage is solved as a system of linear equations.

In real imaging conditions noise has to be considered. However, in the presence of noise, it is not certain that there exists a (convex) solution to the linear system (4.2). Therefore we reformulate the problem to the following minimization problem:

Find a vector I^* of the form $I^* = \sum_{k=1}^{m} \alpha_k^* c_k(x)$, such that I^* is a convex *combination of $c_k(x)$ and the distance $d(I(x), I^*)$ is minimal.*

The distance measure can be selected to appropriately fit the settings, e.g., a locally estimated Mahalanobis distance. For simplicity, we use the Euclidean distance in the following.

We solve the constrained optimization problem by using Lagrange multipliers method (leading to a least squares type of problem), and we minimize the function

$$F(\alpha_1, \dots, \alpha_m, \lambda) = \left\| I(x) - \sum_{k=1}^{m} \alpha_k c_k(x) \right\|_2^2 + \lambda \left(\sum_{k=1}^{m} \alpha_k - 1 \right)$$

over all $\alpha_k \geqslant 0$, for given intensity values of a spel $I(x)$ and local class intensities $c_1(x), \dots, c_m(x)$. The obtained solution $\alpha(x)^* = (\alpha_1^*, \dots, \alpha_m^*)$ provides estimated partial coverage of the spel $\sigma(x)$ by each of the observed classes $k \in \{1, 2, \dots, m\}$.

Coverage values, $\alpha(x)$, are computed for all spels in the set B. However, since B is, in general, not a one spel thick set, it may happen that some of its elements, which should be pure, are assigned partial coverage due to presence of noise. To reduce the impact of noise, we, in the fourth step of the algorithm, perform thinning of the set of mixed spels. We iteratively assign back the simply connected elements of B which are at a smallest distance to one of the crisp class vectors. This continues until the resulting set of spels constitute a thin boundary of a coverage representation, in the sense of Definition 3.6.

By this, a coverage segmentation of the observed image is obtained. Performance of this method is illustrated in Section 5.3.

4.3. Graph based coverage segmentation. Several efficient methods for image segmentation have been formulated in the framework of edge weighted graphs. The graph theoretic approach to image processing naturally leads to methods that are applicable to images of any dimension, and images sampled on non-Cartesian or spatially variant grids [16,58]. An image is often associated with a graph by identifying each spel with a vertex in the graph, and defining edges of the graph so that they represent local adjacency between spels. Each edge in the graph may also be associated with a (real-valued) weight, reflecting the image content [15]. A segmentation of a graph is formulated either as a mapping from the vertices of the graph to some set of object categories, or in terms of graph cuts. Informally, a graph cut is a set of edges such that, if they are removed, the graph is separated into two or more components. The two representations–classification of vertices and separation by cuts–are closely related, and the choice of one representation over the other is largely a matter of preference. In any of the cases, the graph structure utilized in the task of image segmentation provides generality and wide applicability of the designed methods.

We were interested in developing a graph based segmentation method which results in coverage representation, or at least in its approximation; a main interest is to enable subsequent precise feature estimation. Commonly, a segmentation of a graph is only defined on the vertices of the graph, and it is traditionally crisp. Our approach presented in [33,34] is to interpret the edges of the graph as paths between the vertices, and to assign membership labels also to the points along the edges of the graph to one or more object classes. Thereby, we obtain an *edge segmentation* of the graph. In relation to this, we have also introduced the concept of *located cuts*, which are graph cuts defined with sub-edge precision. Via the concept of *induced edge segmentation*, located cuts provide a convenient way of extending a segmentation defined on the vertices of the graph to all points along the edges of the graph. Finally, we have defined *vertex coverage segmentation* as a graph theoretic equivalent of coverage segmentation, and have presented a method for its approximate computation.

In the following we describe this idea in more details. Further information can be found in [33,34]

A framework for sub-pixel segmentation on graphs. A *graph* is defined as an ordered pair $G = (V, E)$, consisting of vertices $v \in V$ and edges $e \in E \subseteq V \times V$. An edge spanning two vertices v_i and v_j is denoted by e_{ij}. Edges can be assigned weights, in which case we refer to a graph as an *edge weighted graph*. If $e_{ij} \in E$, the vertices v_i and v_j are *adjacent*. The set of vertices adjacent to a vertex v is denoted by $\mathcal{N}(v)$. For undirected graphs, an edge is an unordered pair $\{v_i, v_j\}$, i.e., $e_{ij} \equiv e_{ji}$.

A *path* is an ordered sequence of vertices $\pi = \langle v_1, v_2, \ldots, v_k \rangle$ such that $v_{i+1} \in \mathcal{N}(v_i)$ for all $i \in [1, k-1]$. Two vertices v and u are *linked* in G if there exists a path π in G that starts at v and ends at u; we write $v \sim_{G'} u$. If all pairs of vertices in G are linked, then G is *connected*, otherwise it is *disconnected*.

Let $G = (V, E)$, $S \subseteq E$, and $G' = (V, E \setminus S)$. If, for all $e_{ij} \in S$, it holds that $v_i \nsim_{G'} v_j$, then S is a *(graph) cut* on G. For any cut $S \neq \emptyset$, the graph $(V, E \setminus S)$ is disconnected, i.e., it consists of two or more components.

In order to introduce more formally the main concepts of the framework, we start with definitions of vertex and edge segmentations.

Definition 4.1. A *vertex segmentation* $\mathcal{V}$ of a graph $G = (V, E)$ into m components is a mapping $\mathcal{V} : V \to \mathbb{A}_m$.

In the general case, this is a fuzzy segmentation and each vector component $\mathcal{V}(v)_k$ in $\mathcal{V}(v)$ represents the degree to which the vertex v belongs to the corresponding class k.

Vertex segmentations and graph cuts are closely related. If the *boundary*, $\partial\mathcal{V}$, of a vertex segmentation $\mathcal{V}$ is defined as the set of edges $\partial\mathcal{V} = \{e_{ij} \in E \mid \mathcal{V}(v_i) \neq \mathcal{V}(v_j)\}$, then the boundary of a vertex segmentation determines a cut on G.

We interpret edges as connected paths between the vertices. Let a point on an edge e_{ij} be specified by a parameter $t \in (0, 1)$, and let the vertices v_i and v_j be associated with $t = 0$, and $t = 1$, respectively (for undirected graphs, we assume that the vertices are indexed, and use the convention to associate $t = 0$ with the vertex having lower index). If every $v \in V$ is included in E at least once (i.e., there are no isolated vertices), then every point on a graph can be specified by a pair (e, t), where $e \in E$ and $t \in [0, 1]$. In particular, points corresponding to vertices are of the form $(e, 0)$ or $(e, 1)$.

Definition 4.2. An *edge segmentation* $\mathcal{E}$ of a graph $G = (V, E)$ is a mapping $\mathcal{E} : E \times [0, 1] \to \mathbb{A}_m$.

An edge segmentation $\mathcal{E}$ is said to be *consistent* if all segmentation vectors associated with a vertex point (by its different edges) are equal. A vertex segmentation $\mathcal{V}$ and an edge segmentation $\mathcal{E}$ are said to be *consistent* if $\mathcal{E}(e_{ij}, 0) = \mathcal{V}(v_i)$ and $\mathcal{E}(e_{ij}, 1) = \mathcal{V}(v_j)$ for all $e_{ij} \in E$. If $\mathcal{V}$ is a vertex segmentation and $\mathcal{E}$ is an edge segmentation such that $\mathcal{V}$ and $\mathcal{E}$ are consistent, then we may view $\mathcal{E}$ as an extension of $\mathcal{V}$ from the set of vertices to the points along the edges of the graph.

An important concept of the framework is the one of *located cuts*. The idea is to increase the precision of the separation of the objects in the graph by specifying a point along each edge of a cut, indicating where the transition between the different objects occurs. We refer to such a "precise" cut as a *located cut*. Such a cut can define a segmentation where also the points along the edges of the cut are assigned to the separated components, as opposed to a classical graph cut, where the cut edges are left unassigned. We denote a located cut on an edge e_{ij} (of the classical cut) by $\mathcal{T}(e_{ij})$ and, using the introduced edge parametrisation, the location of the cut (the point of transition between the components) is conveniently expressed as a real value in $[0, 1]$.

Located cuts provide a natural way to define a particular type of edge segmentation, consistent with a given vertex segmentation, via the concept of *induced edge segmentation*.

Definition 4.3. Given a vertex segmentation $\mathcal{V}$, and location $\mathcal{T}$ such that $(\partial\mathcal{V},\mathcal{T})$ is a located cut, the *induced edge segmentation* $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ is

$$\mathcal{I}_{\mathcal{V},\mathcal{T}}(e_{ij},t)=\begin{cases}\mathcal{V}(v_i) & \text{if } e_{ij}\in\partial\mathcal{V} \text{ and } t<\mathcal{T}(e_{ij})\\ \frac{1}{2}(\mathcal{V}(v_i)+\mathcal{V}(v_j)) & \text{if } e_{ij}\in\partial\mathcal{V} \text{ and } t=\mathcal{T}(e_{ij})\\ \mathcal{V}(v_j) & \text{otherwise.}\end{cases}$$

Essentially, edges that belong to the cut are "divided" into two parts, as determined by the location of the cut, and the parts are then assigned to the two components as determined by the vertices at the ends of the cut.

An edge segmentation contains, in general, more information than a vertex segmentation. Our interest is to utilize this additional information to obtain precise feature measurements of segmented objects on the graph. However, there are two issues that we have to consider: (i) existing feature estimators are defined for vertex segmentations only, and therefore extraction of relevant information from edge segmentation requires appropriate adjustments; (b) depending on criteria used to define initial (fuzzy) vertex segmentation, this segmentation may, or may not, be appropriate for extraction of geometric features. An appropriate model for this purpose is, as already shown, the coverage based one. It is therefore of high practical interest to convert an edge segmentation, which does contain information related to geometrical properties of the object, to an appropriate vertex segmentation that can be used within the existing framework for feature extraction. We have, in [34] introduced the concept of *vertex coverage segmentation*, a graph theoretic equivalent of the concept of coverage segmentation, which is highly appropriate for precise feature extraction. Finally we have proposed an approach for computing a vertex coverage segmentation, that corresponds to a given edge segmentation.

Assuming no isolated vertices in the graph (a reasonable assumption for the envisioned applications in image processing), we define the *domain* of a vertex v_i as the set of points on the "half-edges" adjacent to the vertex (this can be seen as a graph-theoretical counterpart of a spel). Let $\mathcal{E}$ be an edge segmentation of G into m-components. The vertex coverage segmentation $\mathcal{C}_{\mathcal{E}}$ of a vertex v_i is a vector of $\mathbb{A}_m$ defined as

$$\mathcal{C}_{\mathcal{E}}(v_i)=\frac{1}{|\mathcal{N}(v_i)|}\sum_{j,v_j\in\mathcal{N}(v_i)}2\int_0^{\frac{1}{2}}\mathcal{E}(e_{ij},t)\,dt, \tag{4.3}$$

for all $v_i\in V$. For an induced edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$, the integral in the numerator of Eq. (4.3) can be written in closed form:

$$2\int_0^{\frac{1}{2}}\mathcal{I}_{\mathcal{V},\mathcal{T}}(e_{ij},t)\,dt=\begin{cases}2\mathcal{T}(e_{ij})\mathcal{V}(v_i)+(1-2\mathcal{T}(e_{ij}))\mathcal{V}(v_j) & \text{if } \mathcal{T}(e_{ij})<\frac{1}{2},\\ \mathcal{V}(v_i) & \text{otherwise.}\end{cases} \tag{4.4}$$

Equations (4.3) and (4.4) provide the final required items for the following proposed processing chain: starting from a given (fuzzy) vertex segmentation, compute

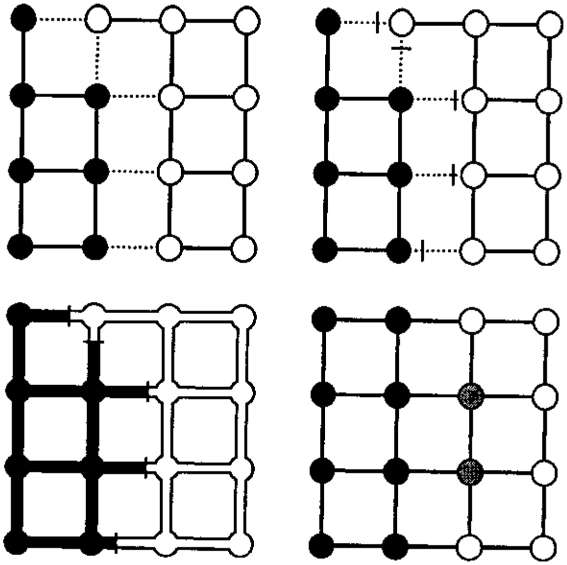


FIGURE 5. Notions of the proposed framework. (Top left) A crisp vertex segmentation $\mathcal{V}$ of a graph. The boundary, $\partial\mathcal{V}$, of the segmentation is shown as dashed lines. (Top right) One possible located cut. (Bottom left) The edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ induced by $\mathcal{V}$ and $\mathcal{T}$. (Bottom right) One component of the corresponding vertex coverage segmentation $\mathcal{C}_{\mathcal{V},\mathcal{T}}$.

a located cut, extend the segmentation to an induced edge segmentation, from that compute a vertex coverage segmentation. Figure 5 illustrates different notions of the framework. The motivation for introducing these steps are, first of all, to reach a vertex coverage segmentation, providing highly improved feature estimates of imaged objects, but also, thanks to the division of the processing chain into individual and separately defined parts, to facilitate easy exchange of one step for another, providing flexibility of the approach and simplifying adjustments to fulfil task specific requirements.

A question that is not addressed here is how to compute located cuts, essential for defining an appropriate induced edge segmentation, and therefore for the final vertex coverage segmentation. One particular method for located cuts computation, applicable to *any* starting fuzzy segmentation, is suggested in [34]. It is based on an appropriate "reduction" of fuzziness of the initial vertex segmentation, so that only information relevant for the subsequent steps is preserved (i.e., precise location of the cut/object boundaries).

Evaluation of the proposed method contains results of area estimation of 2D synthetic objects obtained by the proposed segmentation method; area of an object is estimated as the sum of values assigned to vertices in the vertex coverage segmentation. Theoretical and empirical analysis of the results shows that, even

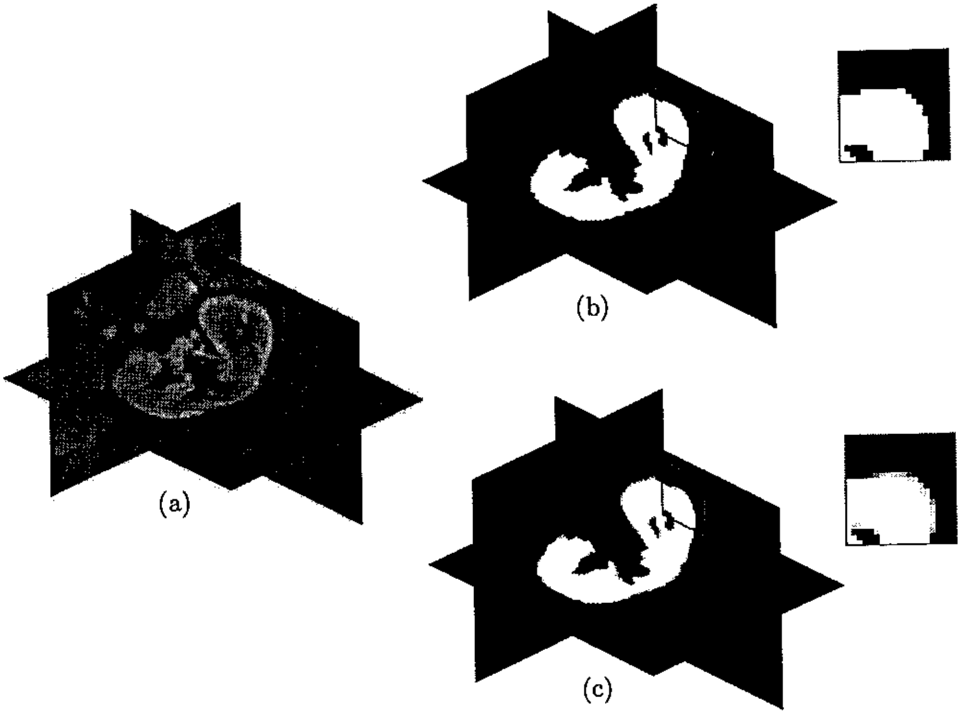


FIGURE 6. A kidney, segmented from an MR volume image of a human abdomen using the Relaxed IFT method [35]. (a) Original grey-scale image volume. (b) Crisp segmentation. (c) Vertex coverage segmentation.

though the convergence rate of the estimate is the same as in the crisp case, the area estimation error obtained by the proposed method is, for every given resolution, significantly smaller than the error obtained from a crisp representation. The method has also been used for segmentation of real medical images; an example of a kidney segmentation is shown in Figure 6, highlighting the difference between the classical crisp segmentation, and the proposed method.

Clearly, the true coverage model for object representation provides higher precision of geometric feature estimates, where convergence rate is proved to be higher than for the crisp case. The proposed graph-theoretical based method is, however, only an approximation of the coverage model, since information about fuzzy memberships over a finite number of one-dimensional sets of points (the edges) is used instead of information about memberships over an nD set of points (the spel). However, generality and applicability of the method are its appealing advantages, compensating for its somewhat lower accuracy.

5. Feature extraction

A coverage segmentation preserves more information about crisp original objects than a corresponding crisp segmentation. This additional information may be highly beneficial in subsequent processing steps, e.g., when estimating features of continuous imaged objects. It is reasonable to expect, however, that methods for feature extraction have to be more or less adjusted to be suited for application to coverage representations of objects.

It is important to notice that digital image analysis aims at measuring features of continuous (real world) objects on the basis of their digital images. Consequently, such measurements, derived from digital shapes, can only be estimates, since the interest is seldom in the features of the digital object, but rather in the original object that has been digitized. Consequently, an important task when designing an estimator is to evaluate its performance, preferably by providing some relevant error bounds.

The possibility to increase precision of estimates of various properties of a continuous original shape by utilizing the information available in a fuzzy representation was studied first for representations based on rather general fuzzy membership functions. Methods derived for estimation of perimeter, area, surface area, volume, geometric moments, signature of a shape, from a fuzzy representation, are presented in [4, 49, 55]. These publications contain statistical studies demonstrating improvement in precision of the estimates, as compared to estimates from a crisp representation. However, the generality of fuzzy membership functions considered, is prohibiting derivation of stronger theoretical statements about the developed methods. We therefore introduced the proposed constraints to the fuzzy membership assignment and the strict interpretation of membership values of the coverage model. This approach, which corresponds well to the outcome of many imaging devices, was, in our opinion, natural to be used. Most importantly, the imposed restriction enabled theoretical derivation of error bounds for a number of feature estimates.

Since a coverage representation is a special case of a fuzzy representation, the previously developed methods for feature estimation from a fuzzy representation are still applicable. For the case of estimation of geometric moments, the proposed method for general fuzzy sets works excellently also for the special case of coverage representations, and we were able to prove that the error decreases to zero at a rate faster than for the crisp case [50]. However, for the proposed perimeter and surface area estimators, despite providing on average much improved estimates, a general faster convergence to the true error was not possible to prove. Instead of starting from the definition of perimeter of a fuzzy set, but rather fully utilizing the knowledge that membership values correspond to pixel coverage, we have derived a method that does provide faster convergence to the true value [52].

We have analysed the accuracy of the estimation of geometric moments, when they are calculated from different representations of a shape. We show that the order of the error can be reduced if the estimation is based on a coverage representation; use of such a representation can therefore be an alternative to increasing the spatial resolution of the image. Geometric moments of objects provide information about area, (hyper-)volume, centroid, principal axes, and a number of other features of the shape. In addition, we were interested in estimation of perimeter and surface area. The results obtained for perimeter estimation are published in [52]; surface area estimation remains our future work. (Note, however, that the method developed in [55] for general fuzzy sets, provides empirically very good surface estimates when applied to a coverage representation.)

Our main results, presented in [50,52], and given in a fairly detailed description in the following subsections, are upper bounds for the estimation error as functions of spatial and coverage resolution. We have confirmed that inter-relations between these two types of resolutions affect the precision of estimation, and that one of the resolutions can, to some extent, be used to compensate for the other. It is usually the case that spatial resolution is given by the imaging device and cannot be changed, whereas improved intensity information, or simply better utilization of grey-levels, already at hand, may be much more easily accessible. Our main message is therefore that *appropriate utilization of intensity information available provides an excellent way to increase estimation precision*, and we specially notice that this applies also for fixed spatial resolution. We find this to be a very useful, applicable, and important result.

Before giving more detailed presentation of methods derived for estimation of geometric moments and perimeter from coverage representation of shapes, we give a brief introduction to main notions and tasks of shape description and analysis.

5.1. Shape analysis. The shape of an object is a representation of its geometric extent. It can be thought of as a silhouette of the object. It is often referred to as a region. The shape of an object is invariant to geometric transformations such as translation, rotation, (uniform) scaling, and reflection. Therefore, shape can be understood as an equivalence class in the set of objects; two objects are equivalent (i.e., have the same shape) if there exists a series of translations, rotations, scalings, and reflections that maps one of them to the other. There are many situations where image analysis can be reduced to the analysis of shapes, which gives high importance to the field of shape analysis.

There exist different classifications of shape analysis techniques, see, e.g., [31] for an overview. Depending whether only the shape boundary points are used for the description, or alternatively, the whole interior of a shape is used, the two resulting classes of algorithms are known as boundary-based (external) and region-based (internal), respectively. Examples of the former class are algorithms which parse the shape boundary and various Fourier transforms of the boundary. Their main advantages are reduction of data and that they may offer a compact description of complex forms. Region-based methods include, e.g., the medial axis transform, moment-based approaches, and methods of shape decomposition into the primitive parts. Their main advantages are easier characterisation and stability in practical applications, where there is unavoidable noise.

A *description* of a shape is data representing it in a way suitable for further computer processing. Such data can be low-dimensional (perimeter and moments), or high-dimensional (medial axis and primitive parts). The first type of data is suitable for, e.g., shape classification, while the second, often called shape *representation*, provides good visual interpretation and compression.

The goal of a shape description is to uniquely characterize the shape. Desired properties of a shape description scheme are invariance to translation, scale, and rotation; these three transformations, by definition, do not change the shape of an object, and consequently should not change its descriptor. However, it should be noted that in the discrete case such invariance exists only up to discretization effects, and special care must often be taken in order to fulfil it.

Additional desired properties of a good shape description method are [31]:

- accessibility–How easy is it to compute a descriptor in terms of memory requirements and computational time; are the operations local or global?
- scope–How wide is the class of shapes that can be described by the method?
- uniqueness–Is the representation uniquely determined for a given shape?
- information preservation–Is it possible to recover the shape from its descriptor?
- stability and sensitivity–How sensitive is a shape descriptor to small changes of a shape?

Descriptors usually perform well regarding some of the listed properties, while failing regarding some others. Therefore, a common approach is to combine them in some appropriate way, to achieve a description that fulfils the requirements of a given task.

5.2. Geometric moments. The two-dimensional Cartesian moment, $m_{p,q}$ of a function $f(x,y)$ is defined as

$$m_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)x^p y^q dx\,dy,$$

for integers $p,q \geqslant 0$. The moment $m_{p,q}(S)$ has the order $p+q$. Cartesian moments are often referred to as *geometric moments.* The geometric moment $m_{p,q}$ can be seen as the projection of $f(x,y)$ to the monomial basis set $x^p y^q$. A *complete moment set* of order n consists of all moments $m_{p,q}$ such that $p+q \leqslant n$.

Moments were made popular in image analysis by Hu, [18]. Hu's *Uniqueness Theorem* states that for a piece-wise continuous function $f = f(x,y)$, nonzero only within a bounded set in $\mathbb{R}^2$, the moments of all orders exist. Moreover, the set of moments of a function f is uniquely determined by f, and conversely, the set of all moments of f uniquely determines f. In order to utilize geometric moments as shape descriptors, their behaviour under scaling, translation, rotation, and reflection has been studied. To provide shape descriptors which are invariant to scale, translation, and rotation of a shape, Hu defined seven nonlinear combinations of geometric moments up to order three, which are known as *absolute moment invariants.*

When used in image analysis, moments are calculated for discrete functions on discrete bounded domains. The definition of a geometric moment $m_{p,q}$ of a digital image $f(x, y)$ is $m_{p,q} = \sum_i \sum_j f(i,j) i^p j^q$, where (i, j) are points in the (integer) sampling grid.

An image is always bounded and an image function is piece-wise continuous, which ensures that the Hu's statement holds for digital images. However, potentially very large number of moments (as many as there are pixels in the image itself) may be required for a unique representation and reconstruction of a digital image. To make the description practically feasible, a smaller set of moments has to be used, and consequently, only an approximate reconstruction can be provided. An important question is how to make an appropriate selection of moments, such that sufficient information is provided for a good enough characterization of the image.

Even though the nonorthogonality of the basis monomials $x^p y^q$ causes some undesired properties of geometric moments, which has initiated several alternative approaches to moment based shape description (e.g., Legendre and Zernike polynomials are defined on orthogonal basis sets, which provides more stable and simpler reconstruction), the Cartesian geometric moments are still well accepted shape descriptors, due to their simple definition, their uniqueness for a given shape, the possibility to derive descriptors invariant to rotation, translation, and scaling, and to express them as integers, their linearity, and the possibility to reconstruct a number of features of a shape from an appropriately chosen set of its moments. In addition, it is possible to express all other types of moments in terms of geometric moments.

The disadvantage of sensitivity to noise mostly applies to high-order moments. Their use in object description can be avoided (or, at least, reduced) by, e.g., first decomposing complex shapes into simpler and more regular parts, which can, then, be represented by a smaller set of lower order moments. Still, precision of estimated moments are of highest importance; decomposition of objects into smaller parts leads to moments computed for smaller objects, and, correspondingly, estimations based on fewer spels. If a special care is not taken, discretization errors may accumulate, cancelling the positive effect of the decomposition.

Our focus is on analysis of the errors that result from estimation of moments of a continuous shape from the corresponding moments of its crisp and different fuzzy digitizations. In particular, we have studied objects represented according to the pixel coverage model, with an aim to explore relations between coverage and spatial resolution and their influence on precision of object representation.

Moments estimated from a Gauss digitization.

Definition 5.1. The $p_1, p_2, \ldots, p_n$*-moment* of a crisp bounded set S in the n-dimensional Euclidean space equipped with the Cartesian coordinate system is

$$m_{p_1,p_2,\ldots,p_n}(S) = \iint\limits_S \cdots \int \prod_{i=1}^{n} x_i^{p_i} \, dx_1 dx_2 \ldots dx_n,$$

for integers $p_1, p_2, \ldots, p_n \geqslant 0$. The moment $m_{p_1,p_2,\ldots,p_n}(S)$ has the order $q = \sum_{i=1}^{n} p_i$.

If the set S is inscribed into an integer grid and digitized, instead of the moments $m_{p_1,p_2,\dots,p_n}(S)$, the moments of the Gauss digitization $\mathcal{D}_g(S)$ are available.

Definition 5.2. The *discrete moment* $\tilde{m}_{p_1,p_2,\dots,p_n}(S)$ of a crisp set S is

$$\tilde{m}_{p_1,p_2,\dots,p_n}(S) = \sum_{x\in\mathcal{D}_g(S)} \prod_{d=i}^{n} x_i^{p_i} ,$$

where $x = (x_1, x_2, \dots, x_n)$, and $\mathcal{D}_g(S)$ is given by Definition 3.4.

Several features of a shape can be calculated from a sufficient number of its moments. In fact, a shape can be recovered from an appropriately chosen set of its moments. If continuous moments are replaced by their discrete counterparts, more or less good estimates of the observed features of a continuous shape can be obtained. An upper bound of the error introduced when approximating continuous moments by their crisp counterparts, can be derived from a theorem by Davenport [7].

To avoid problematic cases, the following restrictions are imposed on the set S, which is assumed to be an n-dimensional closed and bounded set of points.

I Any line parallel to one of the n coordinate axes intersects S in a set of points which, if not empty, consists of at most h intervals.

II The same is true (with m in place of n) for any of the m-dimensional regions obtained by projecting S on one of the coordinate spaces defined by equating a selection of $n-m$ of the coordinates to zero; and this condition is satisfied for all m from 1 to $n-1$.

Theorem 5.1. *[7] If S satisfies the conditions* I *and* II, *then*

$$\big||\mathcal{D}_g(S)| - |S|\big| \leqslant \sum_{m=0}^{n-1} h^{n-m} S_m ,$$

where S_m is the sum of the m-dimensional volumes of the projections of S on the various coordinate spaces by equating any $n-m$ coordinates to zero, and $S_0 = 1$ by convention.

It is desirable to know how the accuracy of approximation changes with a change of image resolution. Instead of increasing the resolution of the digitization grid, we keep the integer grid, and instead scale the set S. Let rS denote a scaling of the continuous set S about the origin by the factor r: $rS = \{(rx, ry) \mid (x, y) \in S\}$. Since h does not change with scale, and $V_m = \mathcal{O}(r^m)$, it is easy to see that $|S| = r^{-n}|\mathcal{D}_g(rS)| + \mathcal{O}(1/r)$, or, expressed in terms of moments, that

$$m_{0,0,\dots,0}(S) = \frac{1}{r^n}\tilde{m}_{0,0,\dots,0}(rS) + \mathcal{O}\Big(\frac{1}{r}\Big). \tag{5.1}$$

Observing that a first order moment of a continuous n-dimensional set S, can be expressed as the zero-order moment of a set S' in an $(n+1)$-dimensional space, Davenport's theorem can, by induction, be generalized to moments of arbitrary order $q \in \mathbb{N}$.

Theorem 5.2. *If the closed and bounded set S satisfies the conditions* I *and* II, *then*

$$m_{p_1,p_2,\dots,p_n}(S) = \frac{1}{r^{n+q}}\,\tilde{m}_{p_1,p_2,\dots,p_n}(rS) + \mathcal{O}\Big(\frac{1}{r}\Big)\,, \tag{5.2}$$

where $q = \sum_{i=1}^{n} p_i, q \in \mathbb{N}$.

Proof by induction. The base case, for zero order moments $q = 0$, is given by (5.1). Inductive step: Assume that (5.2) holds for an arbitrary moment of order q. We then show that (5.2) also holds for a moment of order $q+1$, were we increase the exponent of the kth coordinate, p_k, by one.

$$\begin{aligned} m_{p_1,p_2,\dots,p_k+1,\dots,p_n}(S) &= m_{p_1,p_2,\dots,p_n}(S') \\ &= \frac{1}{r^{n+q+1}}\,\tilde{m}_{p_1,p_2,\dots,p_n}(rS') + \mathcal{O}\Big(\frac{1}{r}\Big) \\ &= \frac{1}{r^{n+q+1}}\,\tilde{m}_{p_1,p_2,\dots,p_k+1,\dots,p_n}(rS) + \mathcal{O}\Big(\frac{1}{r}\Big), \end{aligned}$$

where

$$S' = \big\{(x_1,x_2,\dots,x_n,x_{n+1}) \,\big|\, (x_1,x_2,\dots,x_n) \in S, x_{n+1} \in [0,x_k)\big\}\,.$$

The first equality holds from the definition of moments, the second is given by the assumption, and the third holds by noticing that for every integer point in rS, there are $rx_k + \mathcal{O}(1)$ integer points in rS'. □

For a class of 2D shapes, a stronger statement related to error bounds is derived in [22].

Theorem 5.3. *[22] The moments of a planar 3-smooth convex set S, digitized in a grid with resolution r (the number of grid points per unit), can be estimated by*

$$m_{p_1,p_2}(S) = \frac{1}{r^{p_1+p_2+2}}\,\tilde{m}_{p_1,p_2}(rS) + \mathcal{O}\left(\frac{1}{r^{\frac{15}{11}-\varepsilon}}\right)$$

for $p_1 + p_2 \leqslant 2$

Remark 5.1. A planar 3-smooth convex set is a convex set in the Euclidean plane whose boundary consists of a finite number of arcs having continuous third order derivatives and a positive curvature at every point, except the end points of the arcs. These conditions exclude the existence of straight boundary segments.

Remark 5.2. Despite being given for convex sets, Theorem 5.3 also holds for finite unions and intersections of convex sets.

Moments estimated from a coverage digitization. Fuzzy moments and the centre of gravity of a fuzzy set are among the first defined fuzzy concepts.

Definition 5.3. The $p_1,p_2,\dots,p_n$-moment of a fuzzy subset S of a reference set $X \subset \mathbb{R}^n$ is

$$m_{p_1,p_2,\dots,p_n}(S) = \underset{X}{\iint\!\dots\!\int} \mu_S(x) \prod_{i=1}^{n} x_i^{p_i}\; dx_1 dx_2 \dots dx_n,$$

where $\mu_S(x)$ is the membership of the point x to the set S.

The membership function μ, defining a fuzzy set, can be any mapping $X \to [0,1]$. We further study the special case where the membership of a point $\boldsymbol{x}$ is defined by the coverage $\alpha(\boldsymbol{x})$ of the corresponding spel $\sigma(\boldsymbol{x})$.

Replacing the Gauss digitization in Definition 5.2 with a coverage digitization, and replacing membership μ in Definition 5.3 with coverage values α, the following three definitions follow naturally.

Definition 5.4. The *discrete coverage moment* $\mathcal{M}_{p_1,p_2,\dots,p_n}(S)$ of a crisp set $S \in \mathbb{R}^n$ is

$$\mathcal{M}_{p_1,p_2,\dots,p_n}(S) = \sum_{(\boldsymbol{x},\alpha(\boldsymbol{x}))\in\mathcal{D}_c(S)} \alpha(\boldsymbol{x}) \prod_{i=1}^{n} x_i^{p_i} ,$$

where $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$, and $\mathcal{D}_c(S)$ is given by Definition 3.2.

Definition 5.5. The *ℓ-level quantized discrete coverage moment* $\mathcal{M}^{\ell}_{p_1,p_2,\dots,p_n}(S)$ of a crisp set $S \in \mathbb{R}^n$ is

$$\mathcal{M}^{\ell}_{p_1,p_2,\dots,p_n}(S) = \sum_{(\boldsymbol{x},\alpha^{\ell}(\boldsymbol{x}))\in\mathcal{D}^{\ell}_c(S)} \alpha^{\ell}(\boldsymbol{x}) \prod_{i=1}^{n} x_i^{p_i} ,$$

where $\mathcal{D}^{\ell}_c(S)$ is given by Definition 3.3.

Definition 5.6. The *r-sampled discrete coverage moment* $\hat{\mathcal{M}}^{r}_{p_1,p_2,\dots,p_n}(S)$ of a crisp set $S \in \mathbb{R}^n$ is

$$\hat{\mathcal{M}}^{r}_{p_1,p_2,\dots,p_n}(S) = \sum_{(\boldsymbol{x},\hat{\alpha}^{r}(\boldsymbol{x}))\in\hat{\mathcal{D}}^{\ell}_c(S)} \hat{\alpha}^{r}(\boldsymbol{x}) \prod_{i=1}^{n} x_i^{p_i} ,$$

where $\hat{\mathcal{D}}^{\ell}_c(S)$ is given by Definition 3.5.

In the following we derive error bounds for estimation of moments of a continuous shape from its discrete coverage moments. Theorems are formulated for the n-dimensional case, extending the results presented in [50].

Given a closed and bounded set $S \subset \mathbb{R}^n$, satisfying conditions I and II, we compare the r_f-sampled coverage moment $\hat{\mathcal{M}}^{r_f}(r_s S)$ of an r_s times dilated set S, with the crisp moment $\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S)$ of an r_f times further dilated set (corresponding to r_f times higher image resolution).

Theorem 5.4. *The discrete moments (Definition 5.2) of a set $r_s r_f S \subset \mathbb{R}^n$, can be estimated by the r_f-sampled coverage moments of a set $r_s S$ by*

$$\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S) = r_f^{q+n} \hat{\mathcal{M}}^{r_f}_{p_1,p_2,\dots,p_n}(r_s S) + \mathcal{O}\big(r_s^{q+n-2} r_f^{q+n}\big),$$

where $q = p_1 + p_2 + \ldots + p_n$ is the order of the moment.

Combining this result with Theorems 5.2 and 5.3 leads to the following corollary (extending [50]):

Corollary 5.1. *The moments of a closed and bounded set $S \subset \mathbb{R}^n$, satisfying conditions* I *and* II, *can be estimated by*

$$m_{p_1,p_2,\dots,p_n}(S) = \frac{1}{r_s^{q+n}} \hat{\mathcal{M}}^{r_f}_{p_1,p_2,\dots,p_n}(r_s S) + \mathcal{O}\Big(\frac{1}{r_s^2}\Big) + \mathcal{O}\Big(\frac{1}{r_s r_f}\Big), \tag{5.3}$$

where $q = \sum_{i=1}^{n} p_i, q \in \mathbb{N}$. The moments of a planar 3-smooth convex 2D shape S can, for $p_1 + p_2 \leqslant 2$, be estimated by

$$m_{p_1,p_2}(S) = \frac{1}{r_s^{p_1+p_2+2}} \hat{\mathcal{M}}^{\,r_f}_{p_1,p_2}(r_s S) + \mathcal{O}\Big(\frac{1}{r_s^2}\Big) + \mathcal{O}\Big(\frac{1}{(r_s r_f)^{\frac{15}{11}-\varepsilon}}\Big). \tag{5.4}$$

We note that Theorem 5.4 and Corollary 5.1 also hold for estimations based on ℓ-level quantized discrete coverage moments $\mathcal{M}^{\,\ell}_{p_1,p_2,\dots,p_n}(r_s S)$, with $\ell = r_f^n$. This follows from the fact that the coverage values of an ℓ-level quantized coverage digitization, with $\ell = r_f^n$, do not differ from the values assigned by a real coverage digitization more than the (corresponding) values assigned by an r_f-sampled coverage digitization (see Section 3.1). We conclude that once the spatial resolution is high enough to fully "exploit" the coverage values of spels, i.e., when $r_s > C r_f$, where C is a constant derived from the asymptotic expression for the error bound, using r_f^n coverage values provides the same accuracy of moment estimation as increasing the (crisp) spatial resolution of the image r_f times.

Proof of Theorem 5.4. Without loss of generality, we assume that S is fully contained within $[0,1]^n$. The $r_s r_f$ times dilated shape $r_s r_f S$, then fits in $I_D = [0, r_s r_f]^n$. Let us partition this region (image domain) into r_s^n nonoverlapping blocks of size r_f^n. Each such part of the image space can be expressed as an r_f times dilated spel, such that $\bigcup_{x \in I_d} r_f \sigma(x) = I_D$, where $I_d = \{\frac{1}{2}, \frac{3}{2}, \dots, \frac{2r_s-1}{2}\}^n$ is the set of half-integer points of the r_f times smaller domain $[0, r_s]^n$. The moment $\tilde{m}$ can be computed as a sum of the moments of all such parts of the image space.

$$\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S) = \sum_{x \in I_d} \tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S \cap r_f \sigma(x)) \tag{5.5}$$

Assume that for a block $r_f \sigma(x), x \in I_d$, there are k out of the r_f^n spels which have their centroids within the continuous crisp shape $r_s r_f S$. The moment of such a block is

$$\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S \cap r_f \sigma(x)) = \sum_{j=1}^{k} \prod_{i=1}^{n} (r_f x_i + r_f \Delta_{i,j})^{p_i} = r_f^q \sum_{j=1}^{k} \prod_{i=1}^{n} (x_i + \Delta_{i,j})^{p_i}\,,$$

where $r_f \Delta_{i,j}$, denotes the ith coordinate of the displacement of the jth covered spel with respect to the centre of the block, $r_f x$. $\Delta_{i,j}$ takes values in the range $[-\frac{1}{2}, \frac{1}{2}]$.

We consider two different cases: (i) the block $r_f \sigma(x)$ is completely covered by the set $r_s r_f S$ and $k = r_f^n$; (ii) the block is partly covered and $1 \leqslant k \leqslant r_f^n$. The moment of an empty block, $k = 0$, is correctly estimated as zero, and does not contribute to any estimation error.

Case (i): $k = r_f^n$. Since the spels are symmetrically distributed around x, then for each spel j, there is a corresponding spel j' such that $\Delta_{i,j'} = -\Delta_{i,j}$, for all i. We count both contributions each time and divide by two.

$$\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S \cap r_f \sigma(x)) = \frac{1}{2} r_f^q \sum_{j=1}^{r_f^n} \Big(\prod_{i=1}^{n} (x_i - \Delta_{i,j})^{p_i} + \prod_{i=1}^{n} (x_i + \Delta_{i,j})^{p_i} \Big)$$

For $x_i = \mathcal{O}(r_s)$, and $a_i \in \mathbb{R}$, such that $x_i \gg a_i$, it holds that

$$\begin{aligned}
&\prod_{i=1}^{n}(x_i - a_i)^{p_i} + \prod_{i=1}^{n}(x_i + a_i)^{p_i} \\
&= \prod_{i=1}^{n}\left(x_i^{p_i} - a_i x_i^{p_i-1} + \mathcal{O}\left(r_s^{p_i-2}\right)\right) + \prod_{i=1}^{n}\left(x_i^{p_i} + a_i x_i^{p_i-1} + \mathcal{O}\left(r_s^{p_i-2}\right)\right) \\
&= \prod_{i=1}^{n} x_i^{p_i} - \sum_{j=1}^{n} a_j x_j^{p_j-1} \prod_{\substack{i=1\\ i\neq j}}^{n} x_i^{p_i} + \mathcal{O}\left(r_s^{q-2}\right) + \prod_{i=1}^{n} x_i^{p_i} + \sum_{j=1}^{n} a_j x_j^{p_j-1} \prod_{\substack{i=1\\ i\neq j}}^{n} x_i^{p_i} + \mathcal{O}\left(r_s^{q-2}\right) \\
&= 2\prod_{i=1}^{n} x_i^{p_i} + \mathcal{O}\left(r_s^{q-2}\right)
\end{aligned}$$

And, therefore,

$$\begin{aligned}
\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S \cap r_f \sigma(x)) &= r_f^q \sum_{j=1}^{r_f^n}\left(\prod_{i=1}^{n} x_i^{p_i} + \mathcal{O}\left(r_s^{q-2}\right)\right) \\
&= r_f^{q+n} \prod_{i=1}^{n} x_i^{p_i} + \mathcal{O}\big(r_s^{q-2} r_f^{q+n}\big) \\
&= r_f^{q+n} \hat{\mathcal{M}}^{\,r_f}_{p_1,p_2,\dots,p_n}(r_s S \cap \sigma(x)) + \mathcal{O}\big(r_s^{q-2} r_f^{q+n}\big)
\end{aligned}$$

Case (ii): $k = \mathcal{O}\big(r_f^n\big)$. To cover the worst case, we assume that all covered spels are at maximal distance from the centre of the block; $\Delta_{i,j} = \frac{1}{2}$.

$$\begin{aligned}
\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S \cap r_f \sigma(x)) &\approx r_f^q \sum_{j=1}^{k} \prod_{i=1}^{n}\left(x_i + \frac{1}{2}\right)^{p_i} \\
&\underset{x_i \gg \frac{1}{2}}{=} r_f^q \cdot k \prod_{i=1}^{n} x^{p_i} + \mathcal{O}\big(r_s^{q-1} r_f^{q+n}\big) \\
&= r_f^{q+n} \hat{\mathcal{M}}^{\,r_f}_{p_1,p_2,\dots,p_n}(r_s S \cap \sigma(x)) + \mathcal{O}\big(r_s^{q-1} r_f^{q+n}\big)
\end{aligned}$$

For a closed and bounded set S, satisfying conditions I and II, there are $\mathcal{O}(r_s^n)$ blocks of type (i) and $\mathcal{O}\left(r_s^{n-1}\right)$ blocks of type (ii) in the sum of Eq. (5.5), leading to the final result:

$$\tilde{m}_{p_1,p_2,\dots,p_n}(r_s r_f S) = r_f^{q+n} \hat{\mathcal{M}}^{\,r_f}_{p_1,p_2,\dots,p_n}(r_s S) + \mathcal{O}\big(r_s^{q+n-2} r_f^{q+n}\big).$$

□

Statistical study on synthetic test images. We perform a statistical study to examine the properties of moments estimated at low resolutions. Multigrid resolution is expressed by dilations of the observed objects. Tests are performed on squares and disks of increasing size. For each of the observed real-valued side lengths a large number of randomly positioned squares (in various rotations) are considered. Similarly, for each of the observed real-valued radii, a large number of disks with random centre position, are observed. The continuous objects are digitized using

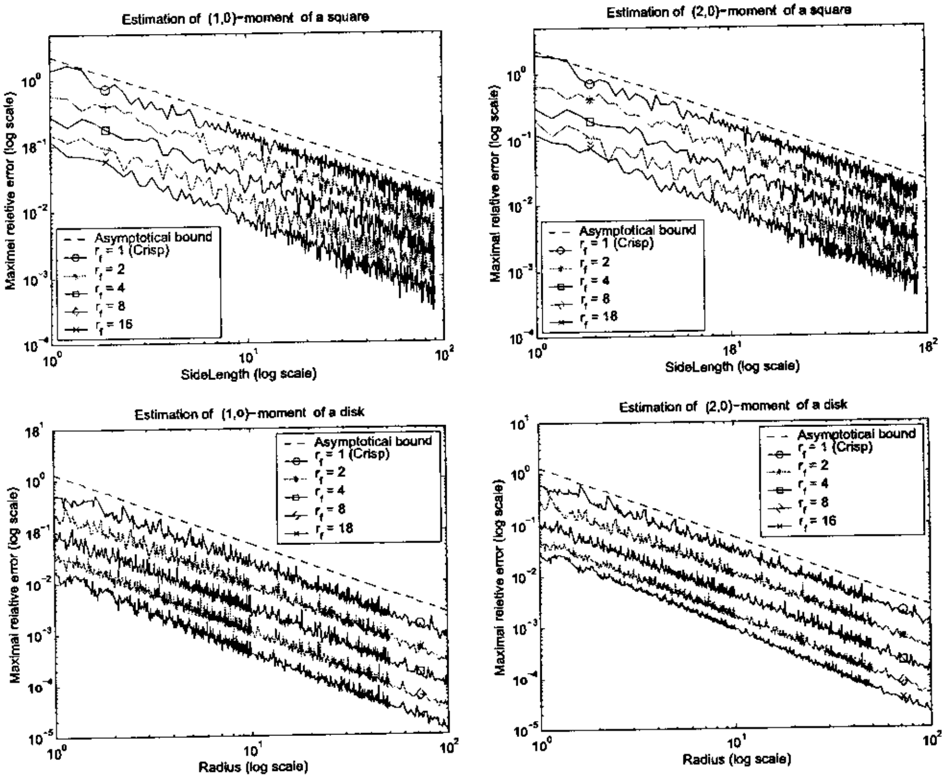


FIGURE 7. Plots of maximal observed errors for first and second order moments estimation for different spatial and coverage resolutions. *Top:* Moment estimation of a square. *Bottom:* Moment estimation of a disk.

r_f-sampled coverage digitization, with different super-sampling levels. Note that $r_f = 1$ corresponds to crisp segmentation, and that $r_f = 16$ approximates the upper limit for the coverage resolution of 8-bit pixel depth.

For each size of an object, we determine the maximal relative estimation error for moments up to the order two. We present the results for $m_{1,0}$ and $m_{2,0}$ moments estimation, both for squares and for disks, in Figure 7. The estimation errors for squares show asymptotic behaviour in accordance with expression (5.3). Disks are 3-smooth convex objects, and the corresponding estimation bounds agree with expression (5.4).

Plots are presented in a logarithmic scale so that the "slopes" of the curves correspond to the order of estimation error, and can be compared with the plotted straight line which has a slope equal to the theoretically derived order of error (-1 for squares and $-\frac{15}{11}$ for disks). Note that the plots show accordance with the asymptotic bounds also at low spatial resolutions. The relative positions of the curves clearly show how the estimation error becomes smaller both with the increase of spatial and coverage resolution.

5.3. Perimeter. The length of the boundary (perimeter) of an object in a two dimensional (2D) image is an essential object features in image analysis. Despite its apparent simplicity, it is a feature that is very difficult to accurately compute from the information provided in a digital image. Accordingly, a large number of publications have addressed the issue of achieving accurate and precise estimates of object perimeter.

Most methods presented in the literature deal with binary images, where pixels either have a value one, being assigned to the object, or zero, if they are assigned to the background; for an overview, see, e.g., [6] and [9]. The binary model corresponds well with the output of the Gauss centre point digitization scheme. It, however, discards a large amount of useful information, especially along the object boundary.

In this section we show how perimeter of a continuous object can be accurately estimated from its coverage digitization. For the case of a quantized coverage digitization, we derive optimal scale factors, minimizing the maximal estimation error for straight edge segments. Both proven theoretically, for straight edges, and observed empirically, for more general shapes, a significant improvement in the accuracy and precision of perimeter estimates is achieved by utilizing the coverage information.

Background. The length of the boundary of a digitized object can be estimated as the cumulative sum of the lengths of local steps along the border of the object. Such estimates are straightforward to accomplish by summing the distances between pixel centres as determined from the Freeman chain code [14]. Doing so, however, results in rather big overestimates; the (intuitive) local step weights, 1 for isothetic and $\sqrt{2}$ for diagonal steps, are not optimal when measuring digitized line segments, this is illustrated in Figure 8.

Starting from an assumption that the boundary of an object is locally planar, optimal weights for the local steps along the border have been derived, [23, 39], leading to improved perimeter estimates. Weights for the 2D case, optimized to provide an unbiased estimator with minimal mean square error for straight lines with length tending to infinity, have been proven to perform even better for curved contours [9]. This last property is important to notice since it provides much more general applicability of the estimator.

In addition to the local type of estimators mentioned above, different nonlocal perimeter estimators have been developed; see e.g [6] for an overview. By basing the estimate on information from larger regions of the image, nonlocal estimators can be made to ensure convergence toward the true value, as the spatial grid resolution increases [22]. Such estimators are often referred to as multigrid convergent estimators. A common approach for nonlocal perimeter estimation is to recognize straight boundary segments, and to perform a polygonalization of the object. In spite of the fact that local methods can not be made multigrid convergent in a general sense (see e.g., [59]), they are still often preferred to nonlocal ones, due to their several advantages. Local methods are relatively easy to implement, parallelizable

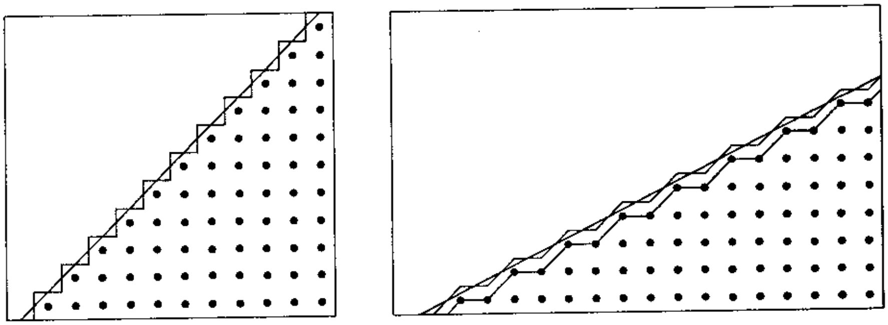


FIGURE 8. Estimation of the length of a straight edge using local steps, (left) using 4 directions, and (right) using 8 directions. Direct use of Euclidean lengths of the steps leads to estimates which are too large for certain directions, due to the shown staircase effect. Errors can be reduced by scaling the estimate with an appropriate factor. However, the estimate remains variant to rotation, and the maximal error (for the better, 8 direction version) is still almost 4%. This error does not decrease with increased image resolution.

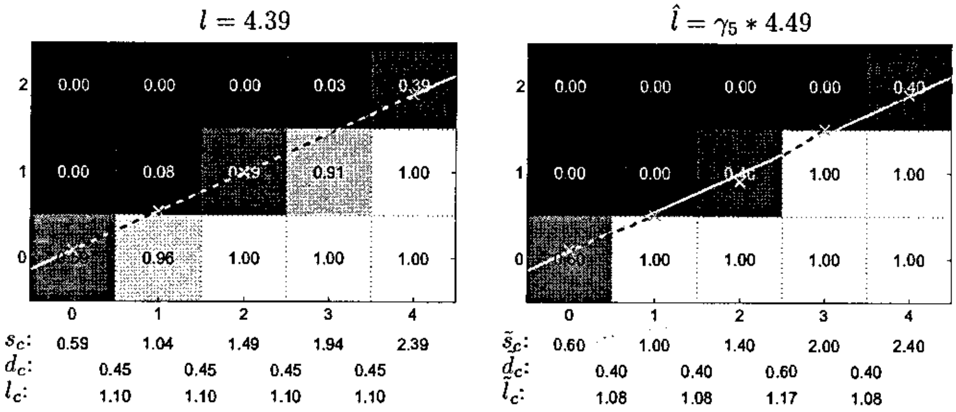
(enabling very fast implementations), and inherently stable (in a sense that a small change in an image causes only a small change of the estimate). These important properties do not in general hold for nonlocal methods.

In this section we present an algorithm, proposed in [67], for estimating the boundary length of a continuous object from its coverage digitization. The method uses only local data and a parallel implementation is straightforward. Effects of quantization of coverage values are considered and the optimal scale factor for the (Freeman-style) cumulative sum of local steps is derived, as a function of the number of coverage levels available. The maximal error (difference from the Euclidean length of the original continuous line segment) is minimized for digital straight segments with the length tending to infinity. The method is applied and evaluated on objects with nonstraight boundaries as well. The issue of a trade-off between spatial and coverage resolution for a good performance of the estimator is explored by observing the performance of the method on shapes digitized at increasing spatial resolution, for a range of coverage resolutions. The results show that the accuracy and precision of estimates rapidly increase with the increase of coverage resolution, once a "reasonable" spatial resolution is provided.

Edge length estimation based on difference of column sums.

Non-quantized case A well known formula for computing the arc length of a function $y = f(x)$ over an interval $[a, b]$ is $l = \int_a^b \sqrt{1 + [f'(x)]^2}dx$. Applied to a linear function, $y = kx + m$, with $k \in [0, 1]$, this formula gives the length l of a line segment for $x \in [0, N], N > 0$ as

$$l = N\sqrt{1 + k^2}\,. \tag{5.6}$$


$l = 4.39$
$\hat{l} = \gamma_5 * 4.49$
2 0.00 0.00 0.00 0.03 0.39
1 0.00 0.08 0.49 0.91 1.00
0 0.59 0.96 1.00 1.00 1.00
0 1 2 3 4
s_c: 0.59 1.04 1.49 1.94 2.39
d_c: 0.45 0.45 0.45 0.45
l_c: 1.10 1.10 1.10 1.10
2 0.00 0.00 0.00 0.00 0.40
1 0.00 0.00 0.40 1.00 1.00
0 0.60 1.00 1.00 1.00 1.00
0 1 2 3 4
$\tilde{s}_c$: 0.60 1.00 1.40 2.00 2.40
$\tilde{d}_c$: 0.40 0.40 0.60 0.40
$\tilde{l}_c$: 1.08 1.08 1.17 1.08

(a) Non-quantized (b) Quantized $\ell = 5$

FIGURE 9. Example illustrating edge length estimation based on the difference d_c of column sums s_c for a segment ($N = 4$) of a halfplane edge given by $y \leqslant 0.45x + 0.78$. The true halfplane edge is shown as a solid white line. The approximations of the edge segment using local steps of slope d_c and $\tilde{d}_c$, respectively, are shown as dashed lines with × marking the ends of each step.

Given a halfplane H defined by $y \leqslant kx+m$, $k \in [0,1]$, $m \geqslant 0$, we can use Eq. (5.6) to compute the length of the edge segment $y = kx+m$, $x \in [0,N]$. For the straight edge $y = kx+m$ of H, the slope k can be expressed as $k = \frac{y(x+\Delta x)-y(x)}{\Delta x} = y(x+1)-y(x)$. Observing integrated function values over a unit interval, $s_c = \int_{c-1/2}^{c+1/2} y(x)\,dx + \frac{1}{2}$, and denoting $d_c = s_{c+1} - s_c$, we conclude that $k = d_c$ for all $c \in \mathbb{R}$.

Assume $N \in \mathbb{Z}^+$. If we observe $c \in \{0,1,\ldots,N-1\}$, then each unit-wide interval used in the integration above defining s_c corresponds to one column of pixels in a digital grid. More precisely, given an image I of width N, being a coverage digitization of the halfplane H, $I = \mathcal{D}_c(H)$, the length l in Eq. (5.6) can be computed as

$$l(I) = \sum_{c=0}^{N-1} l_c\,, \quad \text{for} \quad l_c = \sqrt{1+d_c^2}\,, \tag{5.7}$$

where the value $d_c = s_{c+1} - s_c$ is the difference of two consecutive column sums of the pixel values of the image. This is illustrated in Figure 9(a). The corresponding results for $k \notin [0,1]$ follow by symmetries of the square grid.

Quantized case If the observed image I is, instead, an ℓ-level quantized pixel coverage digitization $I = \mathcal{D}_c^\ell(H)$, then the differences d_c are computed from quantized pixel coverage values. We denote such column differences, derived from a quantized coverage digitization, by $\tilde{d}_c$. These values are from the set $\mathcal{Q}_\ell$ and an edge with real valued slope $k \in [0,1]$ is thereby approximated using local steps with slopes from $\mathcal{Q}_\ell$. An illustration is given in Figure 9(b). Due to the quantization, and the edge line only being approximated, an error is unavoidable. With an aim to minimize the maximal error, we introduce a scale factor γ_ℓ, providing the following formula for the estimation of the length of the edge present in the image:

$$\hat{l}(I) = \sum_{c=0}^{N-1} \hat{l}_c\,, \quad \text{where} \quad \hat{l}_c = \gamma_\ell\sqrt{1+\tilde{d}_c^2}\,. \tag{5.8}$$

In the next section we derive a formula for the optimal value of the scale factor γ_ℓ as a function of the number of quantization levels ℓ so as to minimize the estimation error of (5.8) and we show that $\lim_{\ell\to\infty}\gamma_\ell = 1$; this is when formula (5.8) reduces to formula (5.7).

Binary case. Let us observe the special case of a 1-level quantized pixel digitization, that is, a binary image. The differences $\tilde{d}_c$ of column sums for a binary image of a given edge with $k \in [0,1]$ belong to the set $\mathcal{Q}_1 = \{0,1\}$; the difference value $\tilde{d}_c = 0$ corresponds to a horizontal step (with slope $k = 0$) and the value $\tilde{d}_c = 1$ corresponds to a diagonal step (with slope $k = 1$). In this way, any edge with a real valued slope $k \in [0,1]$ is approximated by a sequence of steps with slopes $k = 0$ or $k = 1$.

The estimation error for this situation is studied in e.g. [23, 39]. As already mentioned, to approximate the length l with $\tilde{l} = \sum_{c=0}^{N-1}\sqrt{1+\tilde{d}_c^2}$ leads to an overestimate of the true edge length, in all cases when the slope k of the edge is not equal to 0 or 1. By scaling the step length with a properly chosen factor γ_1, estimates with a minimal error are achieved. In [64] a value $\gamma_1 \approx 0.9604$ is shown to minimize the maximal error for the binary case.

Optimization to minimize maximal error. Let a set of linearly independent vectors $\boldsymbol{S} = \{\boldsymbol{S}_i = (1, \frac{i}{\ell}),\ i \in \{0,1,\dots,\ell\}\}$ be given. Their slopes are $\frac{i}{\ell} \in \mathcal{Q}_\ell$ and they correspond to the possible slopes of local steps, $\tilde{d}_c$, as derived in the previous section. The length S_i of the vector $\boldsymbol{S}_i$ is $S_i = \sqrt{1+(i/\ell)^2}$.

The edge segment $y = kx + m$, $k \in [0,1]$, on the interval $[0, N]$, represented by the vector $\boldsymbol{l} = (N, kN)$, can be expressed as a linear combination of two vectors $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ from the set $\boldsymbol{S}$, having slopes $\frac{i}{\ell}, \frac{j}{\ell} \in \mathcal{Q}_\ell$ such that $\frac{i}{\ell} \leqslant k \leqslant \frac{j}{\ell}$, as follows:

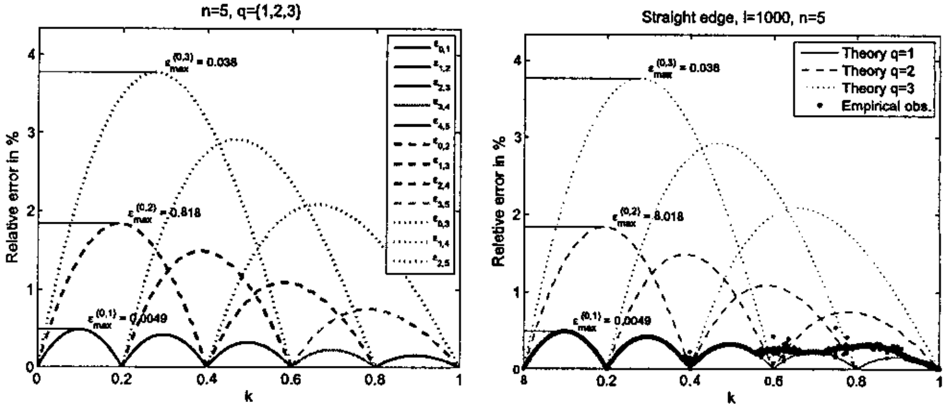
$$\boldsymbol{l} = \frac{(j-\ell k)N}{j-i}\boldsymbol{S}_i + \frac{(\ell k - i)N}{j-i}\boldsymbol{S}_j\,. \tag{5.9}$$

The length of $\boldsymbol{l}$ can be estimated by using Eq. (5.9) as

$$\hat{l} = \gamma_\ell^{(i,j)}\left(\frac{(j-\ell k)N}{j-i}S_i + \frac{(\ell k - i)N}{j-i}S_j\right). \tag{5.10}$$

This corresponds to an edge segment such that $\tilde{d}_c \in \{\frac{i}{\ell}, \frac{j}{\ell}\}$ for all c, for which Eq. (5.8) is equivalent to Equation (5.10).

We derive $\gamma_\ell^{(i,j)}$ as a function of ℓ to minimize the maximal error of estimation formula (5.10). In the given context, the coefficients $\frac{(j-\ell k)N}{j-i}$ and $\frac{(\ell k-i)N}{j-i}$ of Eq. (5.10) represent the nonnegative number of repetitions of each of the local steps



(a) Theoretical

(b) Empirical

FIGURE 10. Relative error $\varepsilon_\ell^{(i,j)}(k)$, for $k \in [0,1]$, $\gamma_\ell = 1$, $\ell = 5$. The values $\varepsilon_{\max}^{(i,j)}$ are indicated. (a) Theoretically derived behaviour of $\varepsilon_\ell^{(i,j)}(k)$, for $i \in \{0,1,2,3,4\}$, and $q = j - i \in \{1,2,3\}$. (b) Empirically observed values of $\varepsilon_\ell^{(i,j)}(k)$ for straight edges $y = kx + m$ of length $l = 1000$ for 10 000 values of k and random m, superimposed on the theoretical results shown in (a).

S_i and S_j in approximation of l and are therefore required to be integers. This condition is, however, rather difficult to impose in the general case. We avoid the problem of integer valued coefficients by deriving the theory for segments of infinite length ($N \to \infty$) (see also [64]).

The *signed relative error* of the length estimate $\hat{l}$ of an edge segment with slope k, such that $k \in [\frac{i}{\ell}, \frac{j}{\ell}]$, is given by the formula

$$\varepsilon_\ell^{(i,j)}(k) = \frac{\hat{l} - l}{l} = \gamma_\ell \frac{(j - \ell k)S_i + (\ell k - i)S_j}{(j-i)\sqrt{1+k^2}} - 1 \,. \tag{5.11}$$

To get a visual impression of the error function given by Eq. (5.11), we plot $\varepsilon_\ell^{(i,j)}(k)$ as a function of k in Figure 10(a), for the case $\gamma_\ell = 1$, $\ell = 5$ and for a number of combinations of i and j.

The best trade-off to minimize $\left|\varepsilon_\ell^{(i,j)}(k)\right|$ is found when

$$\max_{k \in [\frac{i}{\ell}, \frac{j}{\ell}]} \varepsilon_\ell^{(i,j)}(k) = - \min_{k \in [\frac{i}{\ell}, \frac{j}{\ell}]} \varepsilon_\ell^{(i,j)}(k)$$

which gives the following optimal value for $\gamma_\ell^{(i,j)}$

$$\gamma_\ell^{(i,j)} = \frac{2(j-i)\ell}{(j-i)\ell + \sqrt{\left(\sqrt{\ell^2+i^2}\sqrt{\ell^2+j^2} - (\ell^2+ij)\right)^2 + \ell^2(j-i)^2}} \,,$$

where, for two given vectors S_i and S_j, the maximal error is $\left|\varepsilon_\ell^{(i,j)}\right| = 1 - \gamma_\ell^{(i,j)}$. The derivation of this result is given in [51].

If we assume that $j - i = q$ is constant, i.e., that vectors $\boldsymbol{S_i}$ and $\boldsymbol{S_{i+q}}$ are used for the approximation of a given line, it can be derived, [51], that the error $\left|\varepsilon_\ell^{(i,i+q)}\right|$ is the largest for $i = 0$. To minimize the maximal error for $i \in \{0, 1, \ldots, \ell\}$, it is sufficient to observe $\gamma_\ell^{(0,q)}$:

$$\gamma_\ell^{(0,q)} = \frac{2q}{q + \sqrt{\left(\sqrt{\ell^2 + q^2} - \ell\right)^2 + q^2}} = \frac{2}{1 + \sqrt{\left(\sqrt{(\ell/q)^2 + 1} - \ell/q\right)^2 + 1}} \,. \tag{5.12}$$

The value $\gamma_\ell^{(0,q)}$ is denoted γ_ℓ and is used as optimal scale factor in the length estimation defined by Equation (5.10). The corresponding maximal estimation error, $|\varepsilon_\ell|$, is given by

$$|\varepsilon_\ell| = \left|\varepsilon_\ell^{(0,q)}\right| = 1 - \gamma_\ell^{(0,q)} \,.$$

Observing that $\sqrt{(\ell/q)^2 + 1} - \ell/q$ decreases as the ratio $\frac{\ell}{q}$ increases, we conclude that $\gamma_\ell^{(0,q)}$ increases, and consequently the length estimation error decreases, with $\frac{\ell}{q}$ increasing. In other words, by either increasing ℓ or decreasing q the maximal estimation error is reduced. This supports our main motivation for this work: using more coverage levels reduces the length estimation error.

It can be noticed that for $\ell = 1$, corresponding to a binary image, and with $q = 1$, being the only option for $q \leqslant \ell$, Eq. (5.12) provides that the scale factor γ_1 that minimizes the maximal length estimation error is

$$\gamma_1 = \gamma_1^{(0,1)} = \frac{2}{1 + \sqrt{\left(\sqrt{2} - 1\right)^2 + 1}} \approx 0.9604$$

which is a well known optimal result, [64]. The corresponding estimation error is $|\varepsilon_1| = 1 - \gamma_1 \leqslant 4\%$.

More generally, observing the estimation error corresponding to $\gamma_\ell = \gamma_\ell^{(0,q)}$ as a function of ℓ, as given by Eq. (5.12), we conclude that for any constant $q \ll \ell$

$$|\varepsilon_\ell| = \mathcal{O}(\ell^{-2}) \,, \tag{5.13}$$

which gives an asymptotic upper bound for the estimation error as $\ell \to \infty$. The derivation of this result is given in [51]. Empirical studies performed and presented in [67] are in agreement with this theoretical results.

The value of $q = j - i$, appearing in the factor γ_ℓ, deserves some more attention. It reflects the difference in slope of the vectors S_i and S_j used in the linear combination, Eq. (5.9). Their slopes $\frac{i}{\ell}$ and $\frac{j}{\ell}$ correspond to the column differences $\tilde{d}_c$ of the image $I = \mathcal{D}_c^\ell(H)$. A larger value of q leads to a larger error (according to Eq. (5.12)). For the purpose of minimization w.r.t. the maximal error, we observe the worst case situation, i.e., the value q should reflect the larges range of possible column differences $\tilde{d}_c$ for any ℓ-level digitization of a halfplane edge for any given slope $k \in [0, 1]$

We have proved [51] that $q = 3$ corresponds to the worst case situation. According to this observation and Equation (5.12), the optimal scale factor is:

$$\gamma_\ell^{(0,3)} = \frac{6}{3 + \sqrt{\left(\sqrt{\ell^2+9} - \ell\right)^2 + 9}}\,.$$

This leads to the maximal estimation error for the general case: $|\varepsilon_\ell| = 1 - \gamma_\ell^{(0,3)}$. Combining this result with the asymptotic convergence as $\ell \to \infty$, Equation (5.13), the quantization level convergence of the length estimate follows straightforwardly:

Theorem 5.5. *[52] Let $I = \mathcal{D}_c^\ell(H)$ be an ℓ-level quantized coverage digitization of a halfplane $H : y \leqslant kx + m$, $k \in [0,1]$. The length l of the straight edge segment $y = kx+m$ for $x \in [0, N]$, $N \in \mathbb{Z}^+$ can be estimated from I by using Formula* (5.8) *and it holds that $l = \hat{l} + 1/\ell^2$.*

However, for a coverage digitization with few coverage levels ℓ, the worst case situation with $q = 3$ does not appear. As already noticed, for $\ell = 1$ only two different slopes are available, and thus q cannot be greater than 1. We proved that q relates to ℓ as follows:

$$q \leqslant \begin{cases} 1, & \ell \leqslant 2, \\ 2, & 3 \leqslant \ell \leqslant 8, \\ 3, & \ell \geqslant 9 \end{cases} \tag{5.14}$$

We have also found examples where the upper bounds of q, in relation (5.14), are reached, i.e., for $\ell \geqslant 9$, $q = 3$ has to be considered for the (theoretical) maximal error bound estimation.

However, despite explicit examples that $q = 3$ is required for $\ell \geqslant 9$, empirical studies show that the situations where more than two different slopes appear in the estimation are very rare in practise. Observing the plot in Figure 10(b), we see that very few edges have an unscaled error greater than $\varepsilon_{\max}^{(0,1)}$, which supports the idea that computing γ_ℓ using $q = 1$ may be a better choice in practise, than using the theoretically derived worst case value $q = 3$. To use $\gamma_\ell^{(0,1)}$ rather than $\gamma_\ell^{(0,3)}$ is, therefore, recommended in general case, since it is observed as a better choice in our empirical test performed on other shapes, as well.

Local computation of length. Local computation of the edge length relies on local computation of the $\tilde{d}_c$ values of Equations (5.8). How to estimate the slope of the observed edge from a small neighbourhood is a very important question which was answered in details in [51, 52]. The answers addressed issues related to the effects of quantization on the estimation algorithm, as well as what step are required to obtain an algorithm generally applicable, i.e., handling cases when $k \notin [0, 1]$.

Here, we only briefly summarize the most important observations and steps of the algorithm.

First observation is that for lines of a slope $k \in [0, 1]$, each value $\tilde{d}_c$ depends on at most six pixels, located in a 3×2 rectangle; the remaining pixels in the column

pair do not contribute to the difference, neighbouring pixels in each row being the same. The difference $\tilde{d}_c$ of two columns can thus be computed using information only from one, appropriately selected, 3×2 region–a subset of the two observed columns. We denote such a 3×2 configuration $D_{(c,r)}$, where the left pixel of the middle row is located at (c, r), and with $\tilde{d}_{(c,r)}$ the difference of column sums within $D_{(c,r)}$. We then formulate a criterion to detect which $D_{(c,r)}$, for each observed pair of neighbouring columns, is intersected by the straight edge. Carefully treating the effects of quantization we prove that the following holds (see [51]):
For $\tilde{u}(c,r)$ defined as

$$\tilde{u}(c,r) = \tfrac{1}{2} \sum_{\tilde{p} \in D_{(c,r)}} \tilde{p} + \left(r - \tfrac{3}{2}\right),$$

then

$$\tilde{u}(c,r) \in \left[r - \tfrac{1}{2}, r + \tfrac{1}{2}\right] \quad \Rightarrow \quad \tilde{d}_c = \tilde{d}_{(c,r)}.$$

Based on this, we derive an estimation formula for the length of a line segment, with a slope $k \in [0, 1]$, from a local neighbourhood of a size 3×2.

Further, we observe that if the slope k of the observed line is greater than one, then the differences calculated to determine $\tilde{d}_{(c,r)}$ should be taken row-wise instead of column-wise, and a 2×3 region should therefore be used instead of a 3×2 region. To simplify application of the method, we suggest to not use two different region sizes for the two situations ($|k| \leqslant 1$ and $|k| > 1$), but instead to use a 3×3 region in all the cases. Assigning a local edge length contribution to the central pixel of each 3×3 configuration also provides a more appealing output of the algorithm, where edge length values are associated with pixels instead of with edges in between pixels. We denote a 3×3 configuration, with the central pixel located at (c, r), by $T_{(c,r)}$, with (quantized) pixel values $\tilde{p}_i$, $i = 1, 2, \ldots, 9$, indexed row-wise, from left to right and top to bottom. A 3×3 configuration $T_{(c,r)}$ contains (if we consider lines of slope $k \in [0, 1]$) two 3×2 sub-configurations, $D_{(c-1,r)}$ and $D_{(c,r)}$.

Based on the above, we derive the following local edge length $\hat{l}^T_{(c,r)}$, assigned to one 3×3 configuration $T_{(c,r)}$:

$$\hat{l}^T_{(c,r)} = \hat{l}_l + \hat{l}_r, \text{ where } \quad \begin{array}{l} \hat{l}_l = \begin{cases} \frac{\gamma_\ell}{2}\sqrt{1 + \tilde{d}^2_{(c-1,r)}}, & \tilde{u}_{c-1} \in \left[r - \frac{1}{2}, r + \frac{1}{2}\right) \\ 0, & \text{otherwise,} \end{cases} \\ \hat{l}_r = \begin{cases} \frac{\gamma_\ell}{2}\sqrt{1 + \tilde{d}^2_{(c,r)}}, & \tilde{u}_c \in \left(r - \frac{1}{2}, r + \frac{1}{2}\right] \\ 0, & \text{otherwise.} \end{cases} \end{array}$$

The proposed edge length estimate over the whole image I is

$$\hat{l}^T(I) = \sum_{T_{(c,r)} \subset I} \hat{l}^T_{(c,r)}\,. \tag{5.15}$$

Finally, it remains to observe a general situation–a 3×3 configuration $T_{(c,r)} \subset I$, where the image $I = \mathcal{D}^\ell_c(H)$ is an ℓ-level quantized coverage digitization of a halfplane $H : y \leqslant kx + m$ or $H : y \geqslant kx + m$, where the slope k is in $[-\infty, \infty]$. We present a method to isometrically transform every general configuration $T_{(c,r)}$ so that the transformed configuration $T'_{(c,r)}$ corresponds to that of a halfplane

$H' : y \leqslant k'x + m'$, where $k' \in [0,1]$. In that way we extend application of the estimation formula (5.15) to the general case. We have defined criteria for selection of the appropriate transformation to be applied to an observed configuration $T_{(c,r)}$, and proved their correctness also for the quantized case.

Combining all of the above, the following algorithm is presented to compute the edge length contribution $\hat{l}^T_{(c,r)}$ for a given 3×3 configuration. To compute the complete edge length $\hat{l}^T$, the algorithm is applied to all pixels (or, alternatively, only to those adjacent to the object edges, if information about edges of the object is available), and the total length is obtained as a sum of the local edge length contributions, according to formula (5.15). For quantized coverage digitizations of straight edges, this algorithm provides the accuracy guaranteed by Theorem 5.5.

Algorithm 2.

Input: Pixel coverage values $\tilde{p}_i$, $i = 1, \dots, 9$, from a 3×3 neighbourhood $T_{(c,r)}$.
Output: Local edge length $\hat{l}^T_{(c,r)}$ for the given 3×3 configuration.

```
if $\tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 < \tilde{p}_1 + \tilde{p}_2 + \tilde{p}_3$ /* $y \geqslant kx + m$ */
  swap($\tilde{p}_1$,$\tilde{p}_7$)
  swap($\tilde{p}_2$,$\tilde{p}_8$)
  swap($\tilde{p}_3$,$\tilde{p}_9$)
endif
if $\tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 < \tilde{p}_1 + \tilde{p}_4 + \tilde{p}_7$ /* $k < 0$ */
  swap($\tilde{p}_1$,$\tilde{p}_3$)
  swap($\tilde{p}_4$,$\tilde{p}_6$)
  swap($\tilde{p}_7$,$\tilde{p}_9$)
endif
if $\tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 < \tilde{p}_2 + \tilde{p}_3 + \tilde{p}_6$ /* $k > 1$ */
  swap($\tilde{p}_2$,$\tilde{p}_4$)
  swap($\tilde{p}_3$,$\tilde{p}_7$)
  swap($\tilde{p}_6$,$\tilde{p}_8$)
endif
$\tilde{s}_1 = \tilde{p}_1 + \tilde{p}_4 + \tilde{p}_7$
$\tilde{s}_2 = \tilde{p}_2 + \tilde{p}_5 + \tilde{p}_8$
$\tilde{s}_3 = \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9$
$\tilde{u}_l = (\tilde{s}_1 + \tilde{s}_2)/2$
$\tilde{u}_r = (\tilde{s}_2 + \tilde{s}_3)/2$
if $1 \leqslant \tilde{u}_l < 2$
  $\tilde{d}_l = \tilde{s}_2 - \tilde{s}_1$
  $\hat{l}_l = \frac{\gamma_\ell}{2}\sqrt{1 + \tilde{d}_l^2}$
else
  $\hat{l}_l = 0$
endif
if $1 < \tilde{u}_r \leqslant 2$
  $\tilde{d}_r = \tilde{s}_3 - \tilde{s}_2$
  $\hat{l}_r = \frac{\gamma_\ell}{2}\sqrt{1 + \tilde{d}_r^2}$
else
  $\hat{l}_r = 0$
endif
$\hat{l}^T_{(c,r)} = \hat{l}_l + \hat{l}_r$
```

Estimator performance on synthetic test images. To study the accuracy and stability of the method applied to (both convex and nonconvex) curves, we evaluate the presented algorithm with respect to the accuracy of length estimation on a set of synthetic objects digitized using coverage digitization. We use the set of test shapes proposed in [21] (also used in e.g. [6]), containing convex and nonconvex objects with known perimeter, see Figure 11. The test shapes are digitized at a

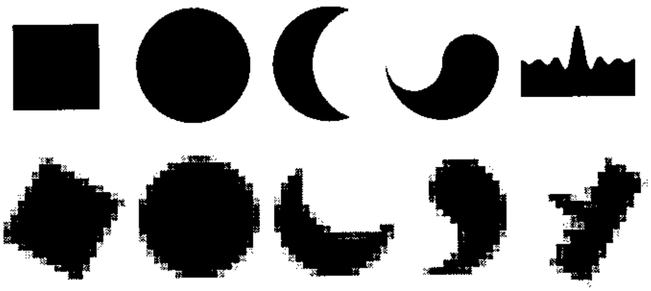


FIGURE 11. Pixel coverage digitizations of the test data set; one high resolution (grid resolution 512) (top) and one rotated low resolution (grid resolution 20) digitization (bottom).

range of resolutions, with random alignment in the digitization grid. Results of this evaluation are presented in Figure 12. We use $q = 1$ when computing the scale factor $\gamma_\ell = \gamma_\ell^{(0,q)}$, since that shows to provide empirically better results than using $q = 2$ or $q = 3$. For each test shape and for a number of resolutions the coverage digitizations for several different rotations and positions of the shape in the digital grid are computed. A number of quantization levels are observed and the nonquantized digitization, indicated with $\ell = \infty$, is also included. The average performance of the method is plotted as a function of resolution. The true pixel coverage digitizations of the test shapes are approximated by 256-sampled coverage digitizations, $\hat{\mathcal{D}}_c^{256}(S)$.

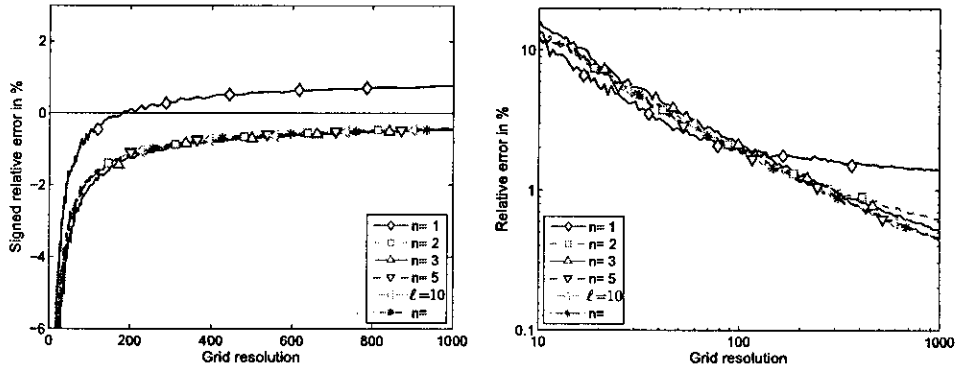
As noticed in [21], the thin elongated peak of the yin-yang curve slows down the convergence to the true value quite significantly. For complementary comparison we also show Figure 13, where the yin-yang shape is removed from the test material.

As an additional test (also performed in [6]), we estimate the perimeter of a rotated square and plot, in Figure 14, the estimate as a function of angle. As can be seen the rotational variation decreases rapidly with the increase of number of coverage levels, and is for $\ell \geqslant 3$ within ±1% of the mean estimate. The slight overall underestimate of the perimeter of the square (less than 0.5%) is attributed to its four corners.

In general, we observe that the presented *local* estimator for boundary length estimation performs very well in comparison with the nonlocal multigrid convergent estimators evaluated in [6]. Note that this holds also for estimates based on relatively few coverage levels.

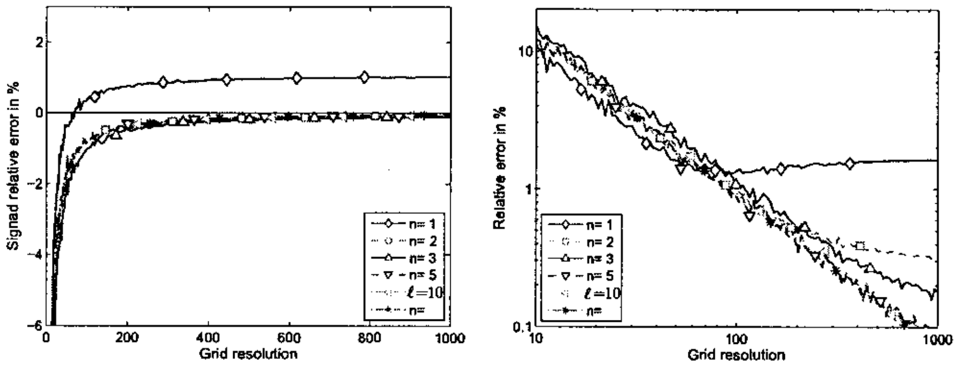
6. Defuzzification and high resolution reconstruction

As described in previous sections, there are several reasons to consider discrete fuzzy representations as a useful way to for represent objects in images. Among first mentioned were advantages of fuzzy representations in handling noise and intensity variations in images, as well as imprecision of various types. No matter what physical property is imaged (reflection of light, density of a material, intensity



(a) lin-lin scale (b) log-log scale

FIGURE 12. Relative errors (in percent) of perimeter estimates for the test shapes (shown in Figure 11) digitized at increasing resolution for 5 different degrees of quantization and for nonquantized ($\ell = \infty$) coverage digitization.



(a) lin-lin scale (b) log-log scale

FIGURE 13. Relative errors (in percent) of perimeter estimates when the yin-yang shape is removed from the set of test shapes.

of a flow, or amount of movement), and independently of if a crisp real object or a naturally fuzzy one (such as a cloud, or a flame, for example, but also properties like blood flow, or activity of cells) are represented in an image, an appropriately chosen membership function can always be chosen so that a corresponding fuzzy representation provides better preservation of information relevant for the imaged object and better treatment of appearing imprecision, than the crisp one.

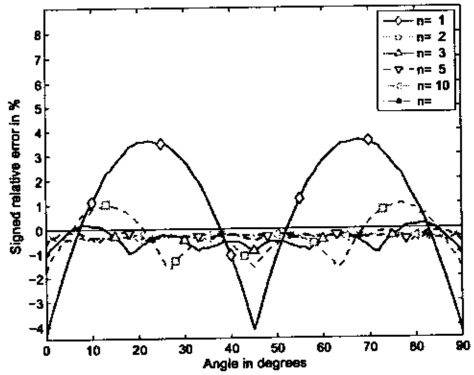


FIGURE 14. Relative errors (in percent) of perimeter estimates for a rotating square of size 128 × 128, from images with 5 different degrees of quantization and for the nonquantized ($\ell = \infty$) case.

Our focus has been on preservation of geometric features of the crisp continuous imaged objects; for that purpose we find the coverage representation most appropriate. Features such as perimeter, area, and other geometric moments are good examples for that.

In spite of mentioned advantages of utilizing fuzzy segmented images, a crisp representation of objects may still be needed. Reasons for that are, e.g., to facilitate easier visualization and interpretation. Even though it contains less information, a crisp representation is often easier to interpret, understand, and manipulate, especially if the spatial dimension of the image is higher than two. Moreover, analogues for many tools available for the analysis of binary images are still not developed for fuzzy images. This may force us to perform at least some steps in the analysis process by using a crisp representation of the objects, requiring the ability to "switch to" an appropriate crisp representation at any point in the process.

In our work, presented in [28, 30, 54], we explored possibilities to generate crisp representations of image objects, starting from fuzzy ones. The process of replacing a fuzzy set with an appropriately chosen crisp set is, in fuzzy set theory, referred to as defuzzification. It can be performed either as an inverse of fuzzification, [40], with the intention to recover a fuzzified crisp original, or as a process independent of any fuzzification, but based on some pre-defined conditions that should be fulfilled for a crisp set to be the representation of a given fuzzy set [24, 44]. In image analysis the fuzzification function is rarely known, and practically never analytically defined; as mentioned earlier, fuzzification of an image is a consequence of a combination of properties of the continuous original, discretization effects, and imaging conditions. Therefore, the inverse of a fuzzification function cannot, in general, be used to define defuzzification. Defuzzification is rather performed so that certain predefined criteria are respected in the process. It seems both natural and beneficial to, for this purpose, impose criteria that reflect properties of a (possible) continuous crisp original. By that, the two approaches to defuzzification are combined. We refer to such defuzzification, which is defined based on some set of imposed criteria, but with "awareness" of the crisp original, as *object reconstruction*.

It can be observed that defuzzification, following fuzzy segmentation, is an alternative to crisp segmentation. We found reasonable to expect that such an approach to crisp segmentation can be tuned so that it enables preservation of the most relevant information (features) of the observed object (and available in the given fuzzy representation) for the application in question. In other words, loss of information is inevitable when defuzzification is performed, but preservation of some features can be prioritized, if appropriate for the application.

The reconstruction that we propose is based on preservation of geometric properties of an object; they are estimated with high precision from a fuzzy object representation and imposed as defuzzification criteria. We determine a crisp representative of the given fuzzy set to be a crisp set which has the selected features as similar as possible to the corresponding features of a given fuzzy set. In this way, defuzzification is defined as an optimization process, where the distance between the given fuzzy set and its crisp reconstruction (defuzzification) should be as small as possible. Formal definition, together with further details related to choice of features to consider, choice of distance to minimize, and choice of optimization method to perform, are given below, in accordance with [28, 30, 54].

Optimal defuzzification. Let $\mathcal{F}(X)$ be the set of fuzzy subsets of a reference set X, and $\mathcal{P}(X)$ be the set of crisp subsets of X, also known as the *power set* of X.

Definition 6.1. Given a fuzzy set $A \in \mathcal{F}(X)$, an *optimal defuzzification* $\mathcal{R}(A)$ of A, with respect to the distance measure d, is

$$\mathcal{R}(A) = \operatorname*{arg\,min}_{B \in \mathcal{P}(X)} [d(A,B)] \,. \tag{6.1}$$

Distance Measure. For any injective mapping Φ from $\mathcal{F}(X)$ into a metric space H, we can define a metric on $\mathcal{F}(X)$ by requiring that Φ is an isometry. Assuming a mapping $\Phi : \mathcal{F}(X) \to H \subset \mathbb{R}^n$, where the vector $\Phi(A)$ contains different features of a fuzzy set A, we define a *feature distance* between fuzzy sets.

Definition 6.2. The *feature distance* $d_p^{\Phi}(A,B)$ between fuzzy spatial sets A and B, on the same reference set X, is the Minkowski distance d_p between the representations $\Phi(A)$ and $\Phi(B)$ of the sets A and B in the feature space $H \subset \mathbb{R}^n$:

$$d_p^{\Phi}(A,B) = d_p(\Phi(A), \Phi(B)). \tag{6.2}$$

For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the Minkowski distance is defined as $d_p(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$.

By suitably designing the mapping Φ, i.e., by considering suitable relevant features of the observed sets, the above distance measure can be tuned to provide defuzzifications where both shape characteristics and membership values are taken into account. This enables defuzzification that fits the individual problem well, and provides a powerful family of defuzzification methods.

Features. Preservation of geometric features of shapes in fuzzy representations gives good motivation to include them into the representation Φ of a fuzzy set. In the 2D case, area, perimeter, and geometric moments of a continuous shape are shown to be preserved with a high precision in a fuzzy representation. They are used in reconstruction as global features, providing information about global geometric properties of the object. Memberships of all the points should be considered as well; they are referred to as local features of the object. Gradient in each point is another relevant feature considered. In addition to the local and global features, a range of meso-scale features can be considered. Our work, presented in [30], introduces meso-scale area measurements into the defuzzification procedure.

Different combinations of features were included in representations of the observed sets and their influence on reconstruction is evaluated in [54]. Even though the selection of features is highly dependent on the requirements imposed by the task, and often on the type of objects, it is clear that inclusion of features of different scales (local, meso, and global) improves the reconstruction.

Tests performed in [54] gave insight in behaviour of Minkowski distances depending on values of p. In most cases, choices $p = 1$ and $p = 2$ showed to be best suited for our needs, and requirements of the observed tasks.

Optimization. In general, the optimization problem (6.1) cannot be solved analytically. In addition, the search space $\mathcal{P}(X)$ is too big to be exhaustively traversed. As a consequence, we are forced to rely on some numerical optimization method, to minimize the distance between the fuzzy set and its crisp counterpart. In [54], two methods, floating search and simulated annealing, are used to find an approximate solution for Eq. (6.1). Simulated annealing performs very well for the task. It is, however, nondeterministic, while at the same time the trade-off between computation efficiency and performance may lead to long computation time required. The optimization task is a well separated problem, so many other search methods can be used to approximately solve Eq. (6.1). We applied DC (Difference of Convex functions) in [32], and SPG (Spectral Projected Gradient) based optimization in [27]. These methods are deterministic and fast, however less flexible with respect to inclusion of features into the feature vector representations Φ.

Reconstruction by optimal α-cut. In general, we do not impose any topology related constraint to defuzzification, even though the proposed defuzzification algorithm allows inclusion of such constraints and control of, e.g., the number of connected components of the resulting reconstruction. Further, we do not, in general, impose criterion of preservation of monotonicity of membership values, i.e., we do not require that an obtained reconstruction must be an α-cut of the starting fuzzy set. The optimization method used in the process is allowed to "decide" about the most appropriate selection of points included in the crisp representation; it may therefore happen that for two points of a fuzzy set one with lower membership is included in defuzzification, while the one with higher membership is not, if that leads to overall better optimization result.

If appropriate, monotonicity preservation can be imposed. Such an approach leads to defuzzification by α-cutting; the defuzzified set is found by thresholding the fuzzy membership function at an appropriate level. This is an appealingly simple method, however, the selection of a threshold is to be determined in some way (in most cases depending of an application) an this is often a rather difficult task. We notice that the simplicity of the method somewhat restricts its performance

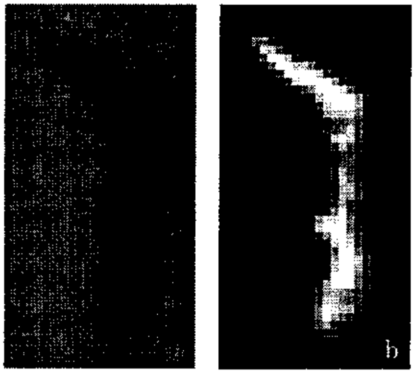


FIGURE 15. Example of fuzzy segmented 2D image. (a) A part of a microscopy image of a bone implant. (b) Fuzzy segmentation of a bone region in (a).

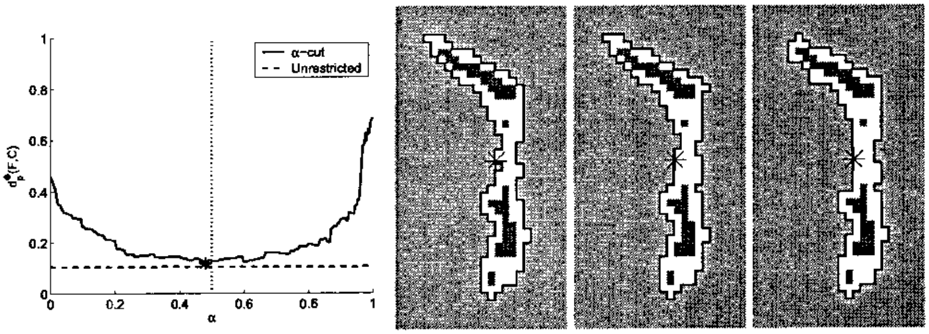
and applicability; this is a commonly known disadvantage of thresholding as a segmentation method in image processing. However, defuzzification by α-cutting fits well, as one specific case, within the proposed framework of defuzzification by feature distance minimization, and we have tested and compared its performance with other, less restricted, defuzzifications.

With the constraint to preserve monotonicity added, the task of minimization of differences between relevant selected (geometric) features of the fuzzy set and its defuzzification is restricted to the selection of *the optimal α-cut*, i.e., the α-cut at the smallest distance from the fuzzy set in terms of feature distance. The search space in this type of optimization is much smaller (there are as many α-cuts to explore as there are different membership levels in the fuzzy set), and exhaustive search can easily be performed to select the optimal α-cut.

Figure 16 presents examples of different defuzzifications; threshold at $\alpha = 0.5$, optimal α-cut, and defuzzification without monotonicity constraint are observed. Achieved minimal distances, as well as reconstructed sets are presented, for a part of a histological image shown in Figure 15.

Reconstruction at increased spatial resolution. A fuzzy representation, in general, contains a lot more information than a crisp representation at the same spatial resolution. If defuzzification is performed at a given spatial resolution, i.e., the crisp representative is generated at the same spatial resolution as the given fuzzy set, this additional information is, to a high extent, lost. It is reasonable to pose the question whether, instead, this information can be utilized to provide a crisp reconstruction at an increased spatial resolution. We explored this issue in [28,30].

If r_F is the spatial resolution of the given fuzzy set, and r_C is the spatial resolution of the crisp (defuzzified) set, then increase of a spatial resolution is expressed by a factor $r = \frac{r_C}{r_F}$. This factor is required to have an integer value. In that case,



(a)

(b) $d_1^\Phi = 0.1271$ (c) $d_1^\Phi = 0.1177$ (d) $d_1^\Phi = 0.1053$

FIGURE 16. Different defuzzification approaches, and their respective feature distances. (a) Plots of feature distance p_1^Φ as a function of α for defuzzification by α-cutting. The minimum is indicated with a star ($*$), while the vertical line is positioned at $\alpha = 0.5$. (b) Defuzzification by α-cutting at $\alpha = 0.5$. (c) Defuzzification by α-cutting at optimal α. (d) Defuzzification by simulated annealing, starting from the optimal α-cut.

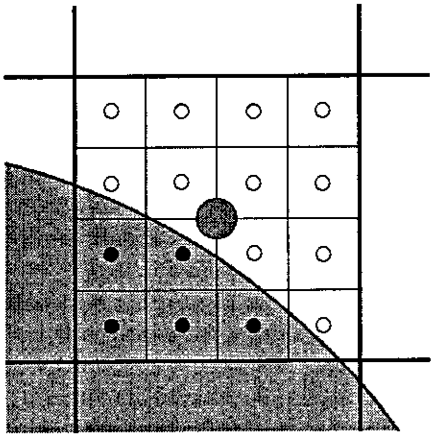


FIGURE 17. One pixel in a low resolution (fuzzy) image, and the corresponding block of 4 × 4 pixels of a 4 times higher resolution (crisp) reconstruction.

each spel in the low resolution representation corresponds to a block of $r \times r$ spels in the high resolution representation; a 2D illustration is shown in Figure 17.

As in Section 5.2, we recall that there are two approaches to perform multigrid studies: one is to observe the (r times) dilated object in the unchanged grid, whereas

TABLE 1. The contribution of the different features to the feature distance, and the total distance, without (Dist 1), and with (Dist 2), the meso-scale area features.

Figure	Perimeter	Area	Centroid	Membership	Meso-scale	Dist 1	Dist 2
18(b)	0.0000	0.0000	0.0000	0.0957	0.3828	0.0957	0.4785
18(c)	0.0015	0.0381	0.0000	0.0957	0.1758	0.1353	0.3111

the other is to observe the unchanged object inscribed in the (r times) refined grid. These two approaches are dual. We use the first one, which implies that the size of the spel is equal to 1 in all the observed grids, whereas the object features calculated in different grids are resolution-variant.

A main idea is to interpret the membership value of a (n-dimensional) spel as an additive property which is distributed over the whole (hyper)-volume of the spel. As such, membership can be summed over blocks of spels, or divided into parts if a spel is divided into sub-spels. This corresponds to the area/volume of a fuzzy set, which by definition is the sum of membership values of all elements of the set [41]. Therefore, instead of comparing pairs of corresponding spels in the fuzzy and the crisp image, we relate the membership value of a spel in the fuzzy image to the sum of membership values of the corresponding block of r^n spels (i.e., for the crisp set, the number of object covered (sub-)spels) in the high resolution representation. This approach can be further generalized so that blocks of spels can be observed in both fuzzy and crisp representations. In that way an additional range of features can be incorporated in defuzzification; local sums of membership values (i.e. area/volume/Lebesgue measure) computed for blocks of spels and interpreted as meso-scale features, ranging from the local (one spel size) to the global (whole object size). This scale space approach provides an appropriate treatment of details in images, where the details are usually relevant only in some range of scales. We addressed it in [30].

An illustrative example is given in Figure 18, where defuzzifications without, and with, meso-scale features are shown, for a synthetic example. The result of defuzzification of the object in Figure 18(a), using the proposed scale space approach, is shown in Figure 18(c). Even though the global features are perfectly matched in the solution presented in Figure 18(b) (obtained without meso-scale features), we consider the solution in Figure 18(c) to better preserve the properties of the original set. A problem which we refer to as "transportation of area" over the image appears if no meso-scale features are used (Figure 18(b)), and is avoided if such features are included in feature vector and considered in defuzzification (Figure 18(c)). The contributions of the different features to the overall distance are, for this example, given in Table 1. From the presented data, it is clear that selection of features considered in defuzzification has very high influence on the result. If understanding of the scene, or a priori knowledge about the object are available, this can be used to obtain the most appropriate reconstruction of an observed object.

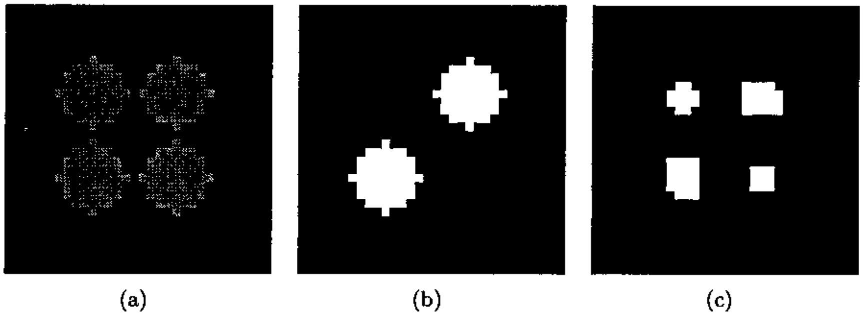


FIGURE 18. (a) Four discrete disks of radius 4 and membership 0.5. (b) Optimal defuzzification using feature distance without meso-scale area components. (c) Defuzzification using feature distance including meso-scale area components.

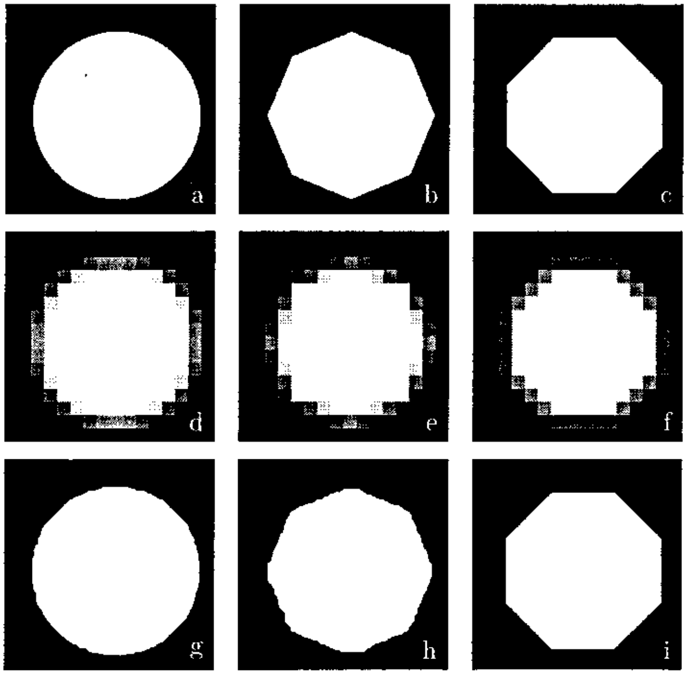


FIGURE 19. Top row: High resolution crisp representations of three crisp continuous shapes. Middle row: Fuzzy representations of the same crisp continuous shapes at relatively low resolution. Bottom row: Defuzzifications of the sets given in the middle row, at 16 times higher resolution.

High resolution reconstruction of a shape by the suggested method is illustrated in Figure 19. Defuzzifications of the fuzzy sets in the middle row of Figure 19, at 16 times higher resolution, are shown in the bottom row of Figure 19. The defuzzification performed is based on minimization of differences between area, perimeter, centroid and membership values of the fuzzy and crisp representations. The amount of information in the images shown in the top row, considered to constitute "ground truth" for this example, as the best possible crisp representations of the observed objects, is $256 \times 256 \times 1$ bit while the amount of information in the fuzzy images in the middle row, and consequently, in the high resolution reconstructions in the bottom row, is $16 \times 16 \times 8$ bit. This means that, without increasing the amount of information by a factor of 32 beyond initially available, an exact reconstruction (shown in the top row) is not possible. Some artefacts are therefore visible in the images in the bottom. However, considering the amount of available information, the visual appearance of the reconstruction result is, in our opinion, rather appealing and facilitates judgement on the original continuous crisp shape.

In order to simplify the notation, we describe the necessary steps in the process of scale-space high resolution reconstruction in the 2D case. Generalization to nD is straightforward, up to availability of appropriate feature estimates.

Weighting of Features. In order to provide that the effect of the total contribution of all measures of one (type of) feature, observed at one particular scale, is of approximately the same size as the effect of one global feature, features of multiplicity h are scaled with $\frac{1}{\sqrt[p]{h}}$, where p is the exponent of the Minkowski distance in Eq. (6.2).

To compare features calculated at different scales, measures also have to be rescaled with respect to the spatial resolution of the image and the dimensionality of the particular feature; e.g., perimeter of an object increases linearly with the spatial resolution, whereas area increases quadratically. To get resolution invariant global features we divide each feature with the feature value of the observed reference set X, that is, for an arbitrary feature F, we observed the resolution invariant feature $\tilde{F}(S) = \frac{F(S)}{F(X)}$.

Feature Vector Representation. For a given fuzzy set $S \in \mathcal{F}(X)$ of size $2^m \times 2^m$ pixels, we generate $(m+1)$ partitions of the set into square blocks of $2^{m-i} \times 2^{m-i}$ pixels, for $i = 0, \dots, m$. Each partition i consists of 2^{2i} blocks. Let B_j^i represent the jth block of $2^{k-i} \times 2^{k-i}$ pixels, where $j = 1, \dots, 2^{2i}$, $i = 0, \dots, m$. A feature of highest interest is area of a set, at all levels (block sizes). Block B_1^0 is equal to the set S and, correspondingly, $\tilde{F}(S) = \tilde{F}(B_1^0)$, for all the observed features F. The membership values of all the pixels are included in such a representation, being local areas of one-pixel-size blocks ($i = m$). In addition, the perimeter of the set S, as well as the coordinates of its centroid, are included in the feature representation.

This leads to the following form of the feature representation $\Phi_m(S)$ of S:

$$\begin{aligned}\Phi_m(S) = \Big(&\frac{1}{\sqrt[p]{2^{2m}}}\tilde{A}(B_1^m),\ldots,\frac{1}{\sqrt[p]{2^{2m}}}\tilde{A}(B_{2^{2m}}^m),\\ &\frac{1}{\sqrt[p]{2^{2(m-1)}}}\tilde{A}(B_1^{m-1}),\ldots,\frac{1}{\sqrt[p]{2^{2(m-1)}}}\tilde{A}(B_{2^{2(m-1)}}^{m-1}),\\ &\ldots\frac{1}{\sqrt[p]{2^0}}\tilde{A}(B_1^0),\tilde{P}(S),\tilde{C}_x(S),\tilde{C}_y(S)\Big).\end{aligned}$$

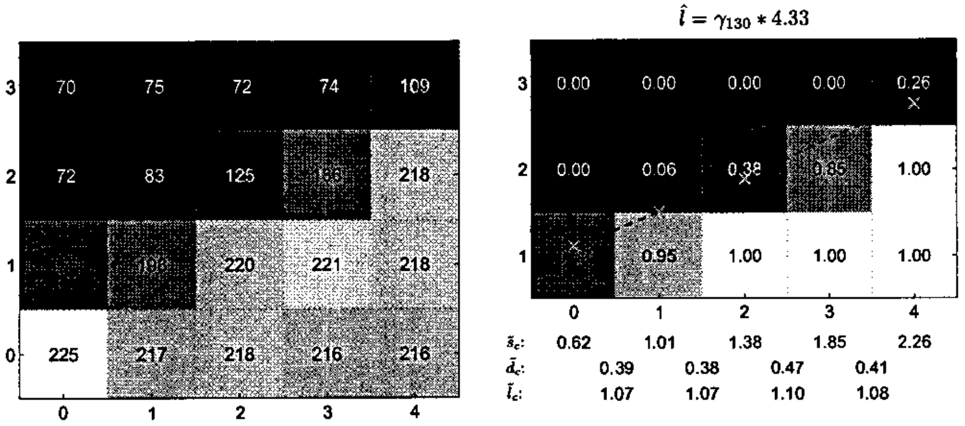
A convenient way to efficiently implement and utilize scale dependent features in defuzzification is to use a resolution pyramid. We use two resolution pyramids for storing the areas of the blocks B_j^i of the fuzzy original set, and of the crisp defuzzification. Pyramids are built by grouping 2×2 neighbouring (*children*) pixels in the image at the current resolution level, and create one (*parent*) pixel at the next, lower, resolution level, where the value of the parent pixel is assigned to be the sum of the values of the children pixels. The process is repeated at every newly created resolution level, until the lowest possible resolution.

Defuzzification. For a given fuzzy set S, containing $2^m \times 2^m$ pixels, a resolution pyramid representation with $m+1$ resolution levels is built. For reconstruction at $r = 2^k$ times increased resolution a resolution pyramid for the crisp set K, with $m+k+1$ resolution levels, is created and defuzzification is performed by minimizing the feature distance $d^\Phi(S,K) = d(\Phi_m(S),\Phi_m(K))$, where d is the Minkowski distance for appropriate choice of p.

Depending on the optimization method selected, a starting configuration may be an important issue; performance of simulated annealing, e.g., highly depends on the selection of the initial configuration. We suggested to use the optimal α-cut of S, i.e., the α-cut at minimal distance d^Φ to S, as the starting configuration for defuzzification. In order to obtain the initial configuration K at r times increased resolution, each pixel in the α-cut is subdivided into 2^r sub-pixels.

Scale space defuzzification of 3D fuzzy sets. The defuzzification method, suggested for 2D discrete spatial fuzzy sets is straightforwardly generalized to the 3D case. The features selected to be included in the feature distance are local, mesoscale, and global volumes, obtained by iterative grouping of blocks of $2\times 2\times 2$ voxels, and surface area and centroid, as additional global features.

Once when the feature representation is generated, the defuzzification process is exactly the same as in the 2D case. However, due to a rapid increase of data, compared to 2D images, some practical implementation related issues may become relevant. An important one is certainly the choice of optimization strategy; in our work presented in [27,32] we have addressed utilization of optimization methods (DC based, SPG based) applicable to large scale optimization problems to the task of defuzzification. Examples of application of a 3D high resolution reconstruction on real medical images are given in Section 7.4.


$\hat{l} = \gamma_{130} * 4.33$

(a) (b)

FIGURE 20. (a) Close up of the straight edge of a white paper imaged with a digital camera. (b) Segmentation output from Algorithm 1, utilizing 130 positive grey-levels. Approximating edge segments are superimposed (dashed lines) on the image (compare with Figure 9).

7. Performance of the presented approach

The coverage model, in our opinion, offers an appealing way to improve information preservation in image processing. With fast advancement in imaging techniques, which naturally comes with development of technology, it becomes less and less acceptable to lower the quality of image processing results and subsequent conclusions by suboptimal methods, not suited to handle the available information and achieved precision in imaging. The coverage model is one of the potential answers to the challenge of "keeping up with the technological development". However, a lot of work still remains, in order to have a complete tool-box for image processing with the coverage model. We have presented in this paper our results obtained along the way; they include segmentation methods, methods for feature extraction, and methods for object reconstruction. Some properties and theoretical evaluation of the proposed approaches are given, separately for each of the methods. Our main interest is in having a processing "chain" consisting of methods developed for the coverage model. We therefore evaluated some of the combinations of the described methods, applied on different real tasks. In some cases, we have used real images obtained in controlled environment, where ground truth is known and the performance evaluation of the method is direct, whereas in some cases we showed applicability of the method in real conditions, where no ground truth is available and conclusions related to performance are derived more implicitly.

7.1. Comparison of different perimeter estimation methods.

We a created controlled real environment for testing performance of the perimeter estimator presented in Section 5.3. This test is also presented in [52]. We took a number of photos of a straight edge of a white paper on black background at a number of angles, using a digital camera in grey-scale mode. All images were cropped to the same width and the slope of each edge was computed using moments [48]. The edge lengths were computed according to Eq. (5.6). We consider these estimates to be correct edge lengths and we use them as the ground truth.

We have evaluated the perimeter estimator presented in Section 5.3 on the task of edge length estimation for the described images, by comparing its performance with performances of several methods previously presented in the literature. In order to confirm that utilization of grey-levels indeed improves the performance, we compared the proposed estimator with most known and best performing estimators for bi-level (crisp) images; we considered the method in [64] (which is equivalent with the proposed one if binary case is observed), and the corner count method, [66]. In order to confirm superiority of the proposed method over previously existing ones applicable to grey-level images, we considered the method of Eberly and Lancaster [11]. Observing high noise-sensitivity of the original approach described in [11], where the gradient components are computed only based on the difference of two pixels, we complemented this method with an additional smoothing step. Two Gaussian filters of different sizes are used to smooth the images prior to length estimation. This improved the performance of the method [11] when applied to straight edges estimation. However, Gaussian filtering is expected to reduce performance on nonstraight edges.

Figure 20(a) shows a part of one image taken as described above, where an edge of a half-plane is presented. The correct value of the slope, $k = 0.42$, is computed. The original image presented in Figure 20(a) is segmented by the coverage segmentation method described in Section 4.1. Figure 20(b) shows the output of the coverage segmentation. Double thresholding is performed, and the number of grey-levels preserved in a one pixel thick border of the object is found to be between 90 and 140 (out of 255) for the different photos. Values of estimated slopes, $\bar{d}_c$, and lengths, $\bar{l}_c$, for local steps, are indicated, as well.

Binary segmentation, required for testing binary estimators, is performed by using Otsu's thresholding method [38], which works very well on the high contrast scene.

Evaluation results are presented in Figure 21. Relative estimation errors for six considered methods are shown, for digital straight segments with different slopes. The maximal errors for the observed methods are presented in Table 2. These results confirm superior performance of the proposed coverage model, and in particular, the proposed perimeter estimation method.

7.2. Coverage segmentation followed by feature estimation for noisy data.

The segmentation method presented in Section 4.2 provides exact coverage values if continuous pixel coverage values and a noise-free environment are ensured. These ideal conditions, however, never exist in practise; quantization errors and presence

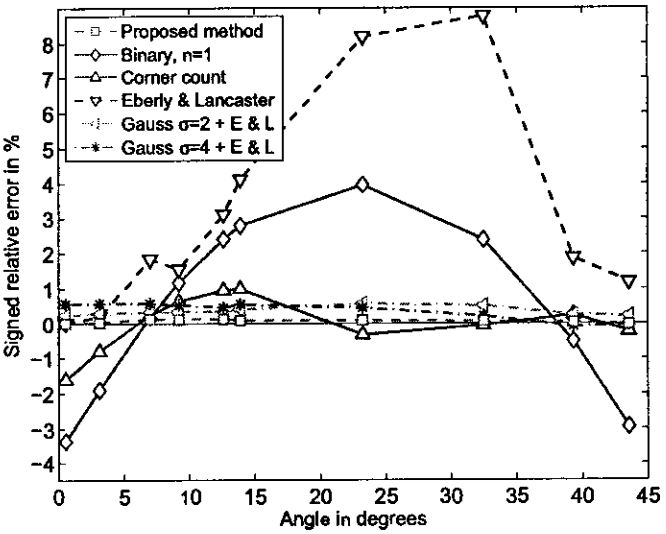


FIGURE 21. Relative errors for different methods when used to estimate the length of the edge of a white paper photographed at different angles with a digital camera.

TABLE 2. Maximal perimeter estimation error when using different approaches.

Method	Max error
Proposed coverage based method [52]	0.14%
Binary method [64]	3.95%
Corner count [66]	1.61%
Eberly & Lancaster, [11]	8.78%
Smoothing ($\sigma = 2$) & Eberly & Lancaster	0.57%
Smoothing ($\sigma = 4$) & Eberly & Lancaster	0.58%

of noise are unavoidable in real images. Performance tests of the method in such environment are certainly of interest.

Being interested in coverage segmentation primarily for its further use for precise and accurate feature estimation, we have, in addition to directly testing the performance of the proposed segmentation method, also evaluated feature extraction based on such segmentations. We have observed perimeter estimates computed by the method presented in [52] (see Section 5.3), and area estimates computed according to [50] (Section 5.2).

We have performed two types of tests, also presented in [53]. First, we have observed synthetic objects with known feature values, affected by simulated noise. We have evaluated both coverage values assignment in the segmentation method and subsequent feature extraction. Second test is performed on real histological colour images.

7.2.1. Synthetic noisy images. A synthetic object (Fig. 22(a)) of known dimensions is randomly placed (rotated and translated) at a number of different positions in the square grid and digitized using coverage digitization. A zoomed-in part of the resulting object, with a (one-pixel thick) partial coverage at its boundary, as well as its superimposed crisp discretization, are shown in Fig. 22(b). Each digitization is subsequently corrupted by increasing levels of additive uncorrelated Gaussian noise, which provides our observed set of test images.

Crisp digital representations of continuous objects are created by a Gauss centre point digitization. This digitization is considered to be equivalent to an ideal, error free, crisp segmentation, and is used as a starting crisp segmentation, required for our proposed coverage segmentation method described in Section 4.2. In that way, evaluation of the method is not dependent of the properties of any particular segmentation method, and is therefore more objective. The same (Gauss centre point) crisp digitization is also used as a reference in comparisons. The neighbourhood required for estimating pure class values is defined by using an appropriate 2D Gaussian mask.

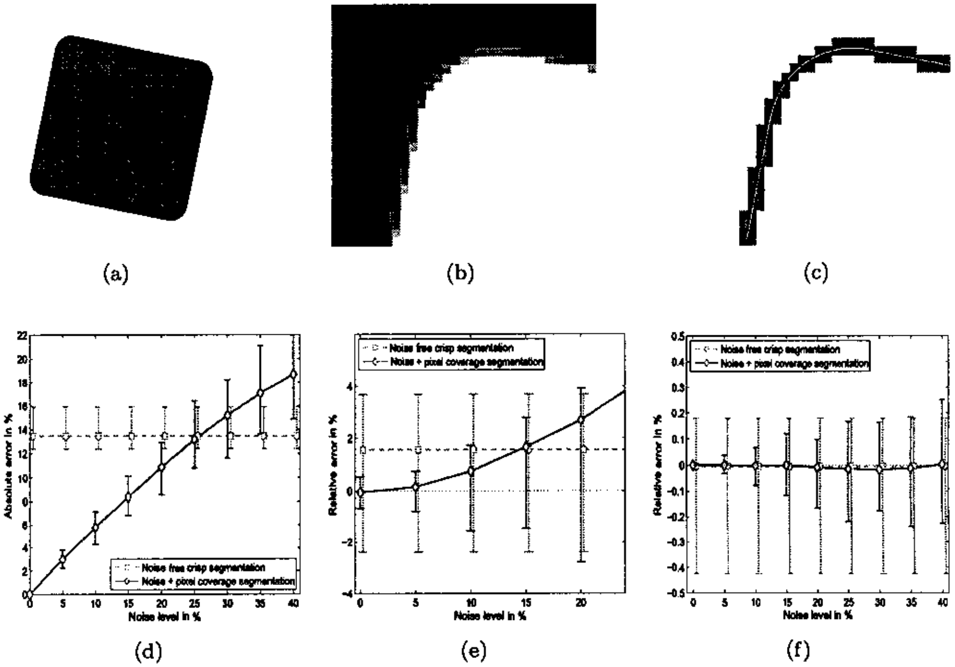
To evaluate the pixel coverage segmentation, the assigned coverage values are, per pixel, compared with the true ones, for increasing amounts of added noise. The average absolute error for coverage values of boundary pixels,

$$\varepsilon = \frac{1}{N} \sum_{p \in B} |\hat{\alpha}(p) - \alpha(p)|$$

where B is the set of evaluated boundary pixels, N is the cardinality of B, and $\hat{\alpha}(p)$ and $\alpha(p)$ are, respectively, assigned and true coverage for a pixel $p \in B$, is computed and presented in Fig. 22(d). A number of random displacements of the object are observed for each level of noise. In Fig. 22(e) we show the relative error of perimeter estimation for the observed synthetic object for increasing levels of noise, whereas Fig. 22(f) presents the area estimation on the same test object, under the same conditions.

As it is visible from the plots in Fig. 22(d-f), the improvement when using the suggested method, compared to the results obtained for an ideal (noise free) crisp segmentation, is significant when the standard deviation of the present noise does not exceed 20%. Above that level, the suggested method does not provide any improvement in terms of accuracy. It is, however, worth noting that the precision of the feature estimates (exhibited as low variation of the obtained results) is significantly higher for the proposed method, and in the case of area estimation provides improvement of the result for all the observed noise levels (i.e., for up to 40% of noise). These observations confirm applicability and excellent performance of the proposed methods–coverage segmentation and feature extraction–for analysis of noisy images.

7.2.2. Quantitative analysis of a histological image. After performing tests on synthetic images, we have tested applicability of the coverage segmentation



(a) (b) (c)

(d) (e) (f)

FIGURE 22. (a) Continuous synthetic test object. (b) Part of a pixel coverage segmentation of (a), with superimposed border of its crisp segmentation. (c) Part of the set B. Grey pixels are removed in the thinning step. Superimposed is the border of the original continuous object. (d) Average absolute error of pixel coverage values in B for increasing levels of noise, lines represent means of 100 observations and bars show max and min errors. (e) and (f) Relative error for the estimated perimeter and area of the object, respectively.

method followed by feature extraction methods on a real example. We use the coverage segmentation method presented in Section 4.2 to segment a microscope slide from a histomorphometrical study. Starting from the obtained coverage segmentation, we compute feature estimates. Comparison with results obtained by previously existing methods, and with results considered to be the ground truth, confirm that the proposed coverage model provides estimates with increased precision. Details of this work are presented in [53]. Here we give a brief summary.

The image shown in Fig. 23(a) is a part of material used in a histomorphometrical study described in [47]. It contains three regions: a screw-shaped implant, bone region, and soft tissue. Quantification is performed by measuring the length of the contact between the implant and the bone region, relative to the overall length of the implant border, and by measuring the percentage of bone area in the vicinity of

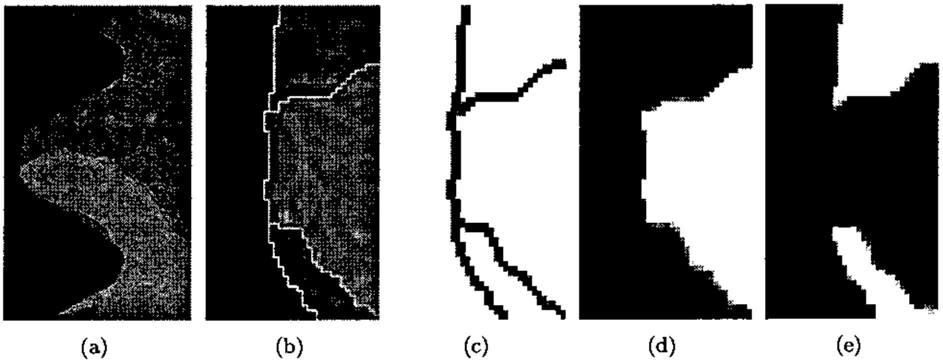


FIGURE 23. (a): The screw-shaped implant (black), bone (purple with a number of hollow spaces) and soft tissue (light blue). (b) Part of a crisp (manual) segmentation of (a) into the three regions. (c) The set B of (grey and black) pixels. Partial coverage values are assigned to the black pixels. (d) and (e) Pixel coverage segmentations of the soft tissue and the bone region, respectively.

the implant; for a detailed description see [47]. Measurements obtained manually, by an expert using integrated microscope software and with higher magnification available, are used as a ground truth.

We applied the proposed coverage segmentation method to segment the RGB image; this example is an illustration of applicability of the proposed method to multi-channel images. First step was to perform a crisp segmentation of the image. For this illustrative example, manual crisp segmentation is used, to get a good starting segmentation and to not mix errors from the crisp segmentation process with errors from the pixel coverage estimation. A part of the manual segmentation is presented in Fig. 23(b). The extracted set B of pixels to be re-evaluated in the process is shown in Fig. 23(c) as the union of black and grey pixels. The grey pixels are detected as pure in the thinning step and only the one-pixel thick 4-connected region in black is assigned partial coverage values. The result of the suggested pixel coverage segmentation method is presented in Fig. 23(d) and 23(e). The first presents segmented soft tissue, whereas the second shows segmented bone. Grey values, visible on the borders between the regions, correspond to partial coverage of pixels.

The aim of the study is to obtain bone-implant contact length estimates, as well as bone area estimates, which provide an improvement in terms of accuracy and precision, compared to those obtained in [47]. We apply the length estimation method presented in [52], and the area estimation method presented and analysed in [50], to the coverage segmentation. While the estimation of area is straightforward, some adjustments of the length estimation method are required. The method, as presented in [52], is applicable for estimation of the border between two classes, whereas in the observed example there may exist pixels which are partly covered by three classes. To adjust the method appropriately, we observe the border line as being between the two classes– implant and nonimplant (soft tissue and bone together)–where the existing method is directly applicable. After the border line within a pixel is estimated, it is distributed to the two nonimplant classes proportionally to their coverage of that pixel. This approach is attractive due to its simplicity, and is acceptably accurate. Due to very low number of pixels in the image which are covered by three classes, potentially introduced errors have minimal impact on the result. Pixels covered by more than two classes are presumably rare in most applications.

TABLE 3. Feature estimation results for manual (using integrated microscope software), crisp [47], and the herein suggested method. See [47] for notation details.

Method	Contact length	Bone area R	Bone area M
Manual	79%	48%	78%
Crisp	88%	50%	81%
Suggested	85%	49%	81%

The results of feature estimates (based on length and area measurements) are presented in Table 3. It is clear that the proposed method provides feature estimates closer to the manual measurements, compared to the previously used methods. This, again, confirms the high performance and applicability of the proposed pixel coverage approaches.

7.3. Estimation of affine deformations of shapes. Image registration is an important task of image processing. Its goal is to find the geometric correspondence between images. Many approaches have been proposed for a wide range of problems in the past decades [69]. Shape matching, that is object registration based on geometry alone, and not on radiometric information, is a viable model when the image intensities are only weakly related and the relation between intensities of two images is hard to model; this happens in, e.g., multimodal registration (between images acquired by rather different imaging modalities) or when the image intensities undergo strong nonlinear deformations, e.g., in case of X-ray imaging. Shape matching requires an initial segmentation step, where the same region is segmented in the two images to match. This segmentation can be performed in a crisp or in a fuzzy way. In the following we present shape matching with improved precision based on coverage representations of shapes. This approach is described in details in [60, 61]. It gives additional evidence about advantages of using the coverage model to improve quality of image processing steps.

Domokos et al. proposed an extension [8] to the parametric estimation method of Francos et al. [17] to handle affine matching of crisp shapes. Estimation methods of this type have the advantage of providing accurate and computationally simple solution, avoiding both the need for finding point correspondences in the images as well as the need for computationally demanding optimization. Appropriate (global) low-dimensional representations of a shape are instead utilized and correspondence between such representations is found by a direct computation. We have extended this approach to the case when the segmentation method is capable of producing a coverage segmentation instead of a classic crisp segmentation [60]. We know that the information preserved by using coverage representation may be successfully utilized to improve precision and accuracy of several shape descriptors. Precise moment estimation is essential for a successful application of the object registration method presented in [8] and the advantage of coverage representations is clearly noticeable in the represented study.

Observation that all individual correspondences between pairs of points of the two shapes, related by an affine transformation to be recovered, can be observed at the same time (instead of, e.g., only correspondences between some selected pairs), and can be integrated over the shape, is what the approach, developed for binary shape registration, and described in details in [8], relies on. Extending the established correspondences by applying appropriately chosen mappings to them, and integrating, a system of independent equations can be created. Unknowns of the system are the parameters of the applied unknown affine transform. For 2D shapes, six parameters are to be recovered, and therefore six equations are required in the system. It is observed that polynomial equations, for polynomials of at least second order, are simplest ones being at the same time linearly independent. The suggested system is therefore:

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k^n \, dx = \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} q_{k1}^{n-i} q_{k2}^{i-j} q_{k3}^{j} \int_{\mathcal{F}_o} y_1^{n-i} y_2^{i-j} \, dy, \tag{7.1}$$

where $k = 1, 2$; $n = 1, 2, 3$ and q_{ki} denote the unknown elements of the inverse transformation $\mathbf{A}^{-1}$ with Jacobian $|\mathbf{A}|$.

This polynomial system is derived in the continuous space. However, digital image space provides only limited precision for these derivations and the integral can only be *approximated* by a discrete sum over the pixels. In [8] the Gauss centre point digitization is used. We explored whether using a coverage digitization would improve the registration performance. The coefficients of the system of equations in Eq. (7.1) are the first, second and third order geometric moments of the *template* and *observation*. Replacing them with their corresponding discrete approximations, we expect that the increased precision achieved from a coverage representation [50] will lead to improved registration performance.

Following the definition of discrete moments, the approximating discrete system of polynomial equations corresponding to Eq. (7.1) can now be produced:

$$|\mathbf{A}| \sum_{x \in X_t} \mu_{\mathcal{F}_t}(x) p_k^n = \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} q_{k1}^{n-i} q_{k2}^{i-j} q_{k3}^{j} \sum_{x \in X_o} \mu_{\mathcal{F}_o}(x) p_1^{n-i} p_2^{i-j}.$$

Clearly, the spatial resolution of the images affects the precision of this approximation. We note that sufficient spatial resolution may be unavailable in real applications or may lead to too large amounts of data to be successfully processed within the time constraints. On the other hand, it was shown in [50] that increasing the number of grey-levels ℓ, representing pixel coverage, by a factor r^2 provides asymptotically the same increase in precision as an r times increase of spatial resolution. This makes the suggested approach, utilizing increased membership resolution, a very powerful way to compensate for insufficient spatial resolution, while still preserving desired precision of moments estimates.

Evaluation of the transformation estimation–synthetic tests. Evaluation tests are first performed on a database of synthetic binary shapes. We examine the effect of the number of quantization levels on the precision of registration and compare results with the binary case. Pairs of corresponding synthetic fuzzy shapes are obtained by applying known affine transformations and the presented registration results for synthetic images are neither dependent nor affected by a segmentation method. This also means that the ground truth is available.

The data set consists of a number of different shapes and their transformed versions, a total of 2000 images. The transformation parameters (including rotations, translations, shear, and scaling) were randomly selected from uniform distributions. The templates are binary images, i.e., pixels in them are assigned coverage values either 0 or 1 (this corresponds to 1-bit representation). The coverage representations of the observation images are quantized and represented by integer values using k-bit ($k = 1, \dots, 8$) representation. Some typical examples of these images and their registration accuracies are shown in Figure 24.

In order to quantitatively evaluate the results, we use two error measures. The first error measure (denoted by ϵ) is the average distance in spels between the true ($\mathbf{A}\boldsymbol{x}$), and recovered ($\widehat{\mathbf{A}}\boldsymbol{x}$) positions of the transformed spels over the template. This measure can be used for evaluation only if the true transformation is known; this is the case in the tests on synthetic images. Another error measure is the absolute difference (denoted by δ) between the *registered* template image and the *observation* image

$$\epsilon = \frac{1}{|T|}\sum_{\boldsymbol{x}\in T}\left\|\mathbf{A}\boldsymbol{x} - \widehat{\mathbf{A}}\boldsymbol{p}\right\|, \quad \text{and} \quad \delta = \frac{|R \triangle O|}{|R| + |O|},$$

where $|T|$ is the number of *template* spels, $\triangle$ denotes the symmetric difference, while R and O are the set of spels of the *registered* shape and the *observation* respectively. Before computing the errors, the images are binarized by taking the α-cut at $\alpha = 0.5$.

The medians of errors for both ϵ and δ are presented in Table 4 for different quantization levels. Experimental data confirm the theoretical results on increased precision of moments estimation based on coverage representation. Consequently, the registration results, compared to the binary case, are improved. It is important to notice that registration based on coverage representation may be applied for lower image resolutions, i.e. where the binary approach becomes unstable.

An important property of the proposed registration method is that, although based on solving a system of polynomial equations, it provides the result without any iterative optimization step. Its performance is based on the precision and

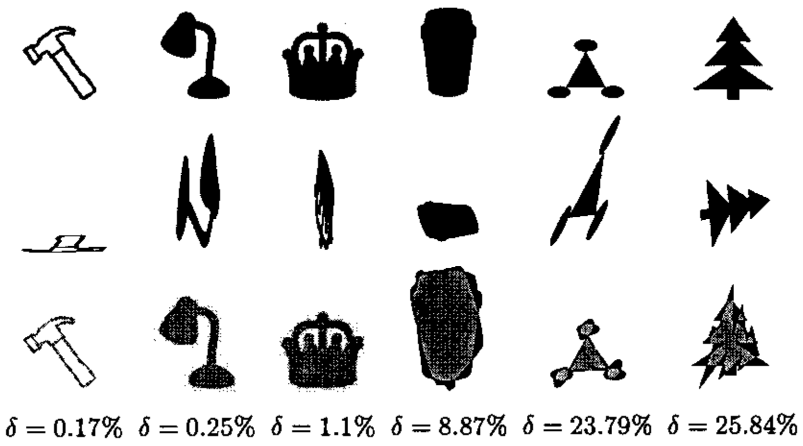
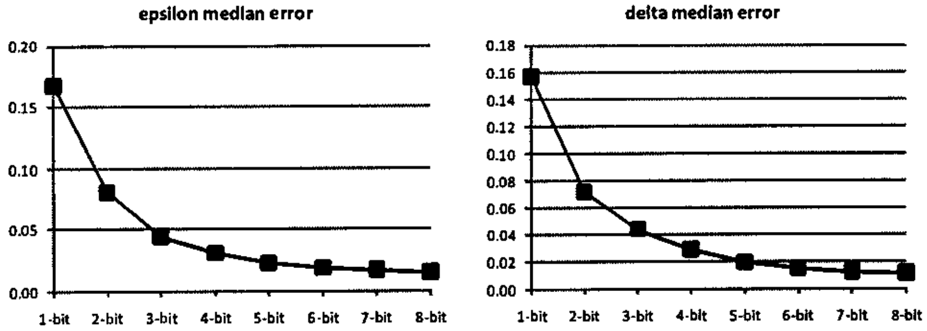


FIGURE 24. Examples of *templates* (top row) and *observations* (middle row) images. In the third row, grey pixels show where the registered images matched each other and black pixels show the positions of registration errors.

TABLE 4. Registration results of 2000 images using different quantization levels of the fuzzy boundaries.

	Fuzzy representation							
	1-bit	2-bit	3-bit	4-bit	5-bit	6-bit	7-bit	8-bit
ϵ median (pixels)	0.168	0.080	0.0443	0.0305	0.0225	0.0186	0.0169	0.0147
δ median (%)	0.157	0.072	0.0439	0.0292	0.0196	0.0151	0.0125	0.0116
Registered	1905	1919	1934	1943	1933	1929	1925	1919
Not registered	95	80	66	57	67	71	75	81



accuracy of moments estimates. The time complexity of the method is $\mathcal{O}(N)$, where N is the number of the pixels of the image, enabling real time registration of shapes.

Experiments on real X-ray images. Hip replacement is a surgical procedure in which the hip joint is replaced by a prosthetic implant. In the short post-operative time, infection is a major concern. An inflammatory process may cause bone resorption and subsequent loosening or fracture, often requiring revision surgery. In current practise, clinicians assess loosening by inspecting a number of post-operative X-ray images of the patient's hip joint, taken over a period of time. Obviously, such an analysis requires the registration of X-ray images. Even visual inspection benefit from registration, as clinically significant prosthesis movement can be very small.

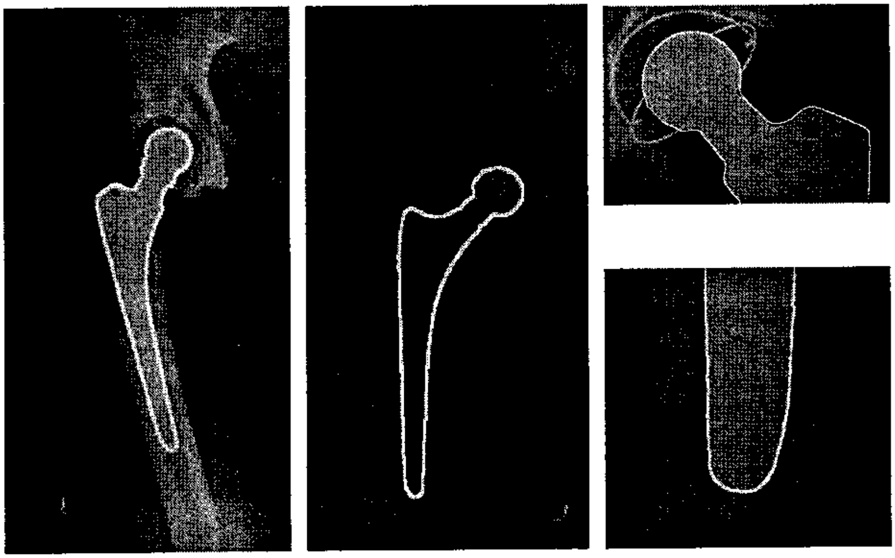
There are two main challenges in registering hip X-ray images: One is the highly nonlinear radiometric distortion [12] which makes any grey-level-based method unstable. Fortunately, the segmentation of the prosthetic implant is quite straightforward [37] so shape registration is a valid alternative here. Herein, we used an appropriate coverage segmentation method to segment the implant. The second problem is that the true transformation is a projective one which depends also on the position of the implant in 3D space. Indeed, there is a rigid-body transformation in 3D space between the implants, which becomes a projective mapping between the X-ray images. Fortunately, the affine assumption is a good approximation here, since the X-ray images are taken in a well defined *standard position* of the patient's leg.

For the diagnosis, the area around the implant (especially the bottom part of it) is the most important for the physician. It is where the registration must be the most precise. Fig. 25 shows some registration results. Since the best aligning transformation is not known, only the δ error measure can be evaluated. We also note, that in real applications the δ error value accumulates the registration error and the segmentation error.

The preliminary results obtained on real X-ray images of hip prosthetic implants taken during post-operative controls are, in our opinion, very encouraging; they show that our approach using coverage segmentation and subsequent registration as described above can be used in real applications. Further research on possible improvements, generalizations, and thorough evaluation of these initial studies is in progress.

7.4. High resolution reconstruction–two examples.

Some examples of the performance of our proposed high resolution reconstruction method are shown already in Section 6. 2D objects are observed, and one illustrations of influence of selection of features in feature representation to the reconstruction result is given. More elaborate evaluation of various relevant issues of defuzzification is presented in [54]. In this section we present two examples of application of the proposed method on 3D images. Both examples are within medical imaging, imaging modalities are different (CT and X-rays), sizes and complexity of data differ (first one is a solid object, of a rather simple structure, second is an object of a high complexity, but smaller in data size). Issues of interest are choice of an optimization method that gives a good result at a reasonable cost, selection of a starting position for the optimization, decision about topology preservation constraint, selection of (scales of)



$\delta = 2.17\%$ $\delta = 4.81\%$ $\delta = 1.2\%$

FIGURE 25. Real X-ray registration results. (a) and (b) show full X-ray observation images and the outlines of the registered template shapes. (c) shows a close up view of a third study around the top and bottom part of the implant.

features to be used in feature vector representation. Both examples show applicability and very good performance of the proposed method. Topological constraints, size of the data, complexity of the task are all addressed well by the presented method.

3D CT image of a bone implant. First example of high resolution reconstruction is performed on a 3D object presented in Figure 26. The data volume is a CT image of a bone implant (inserted in a leg of a rabbit). We applied the method to a part of the image (Figure 26(a) shows a slice through the volume) containing a connected piece of bone area (dark grey), surrounded by a nonbone area (light grey). Figure 26(b) shows a slice through a 3D fuzzy set representing the bone region.

All features are matched well in this example; there are no large regions of high fuzziness, and the global features do not provide any reason for "transportation" of volume as in the example in Section 6. Defuzzifications with or without meso-scale features are therefore practically identical. No topological constraints are required, since this object appears to be simple enough for the process to handle the topology automatically. Main challenge can be seen in the size of a data set (even though this particular example is made small enough); regarding this issue a selection of an optimization method is of rather high importance. The result presented here is obtained by simulated annealing, but increasing difficulties in optimization,

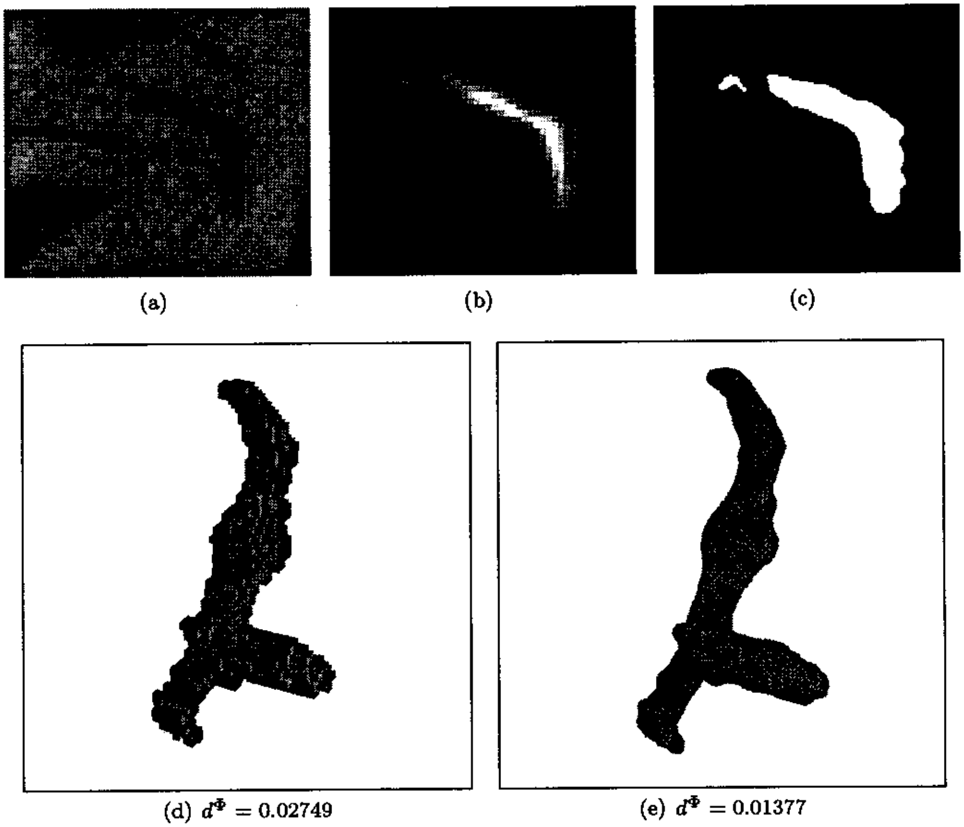


FIGURE 26. Defuzzification of a part of a 3D image of a bone implant. (a) Slice through the image volume. The dark grey area is bone, the light parts are nonbone areas. (b) Slice through a fuzzy segmentation of the bone region in the image volume. (c) Slice through a defuzzification, using meso-scale volume features, of the fuzzy segmented image volume. (d) 3D rendering of the α-cut at smallest feature distance to the fuzzy object. (e) 3D rendering of a high resolution defuzzification of the fuzzy segmented object. A four times scaled up version of the best α-cut (d) was used as starting set for the simulated annealing search.

caused by increasing size of data, motivated our research on alternative options for optimization strategy. Results are presented in [27, 32]. Starting position can influence result significantly; a good choice, used in the example shown in Figure 26, is an optimal α-cut.

High resolution reconstruction of X-ray image of vessels. Fuzzy representations, and coverage representation in particular, of image objects are especially useful when the spatial resolution is too low to provide a good crisp representation. One such situation can be seen in Figure 27(b), which displays a maximum intensity projection of a part of a rotational b-plane X-ray scan of the arteries of the right half of a human head (provided by Philips Research, Hamburg, Germany), shown in Figure 27(a). A contrast agent is injected into the blood and an aneurism is shown to be present. The intensity values of the image voxels correspond fairly well with partial volume coverage, and are therefore used directly as coverage membership values.

This example image violates the sampling theorem; the vessels imaged are not resolved since they are less than one voxel thick. This fact causes a number of problems related to information extraction. Using a priori knowledge about the image, it is still possible to obtain a reasonable high resolution reconstruction. One such a priori piece of information is the knowledge that the vessel tree is simply connected. Starting from one simply connected component, and preserving topology [2] throughout the search, it is provided that the obtained crisp representation (reconstruction) is also simply connected.

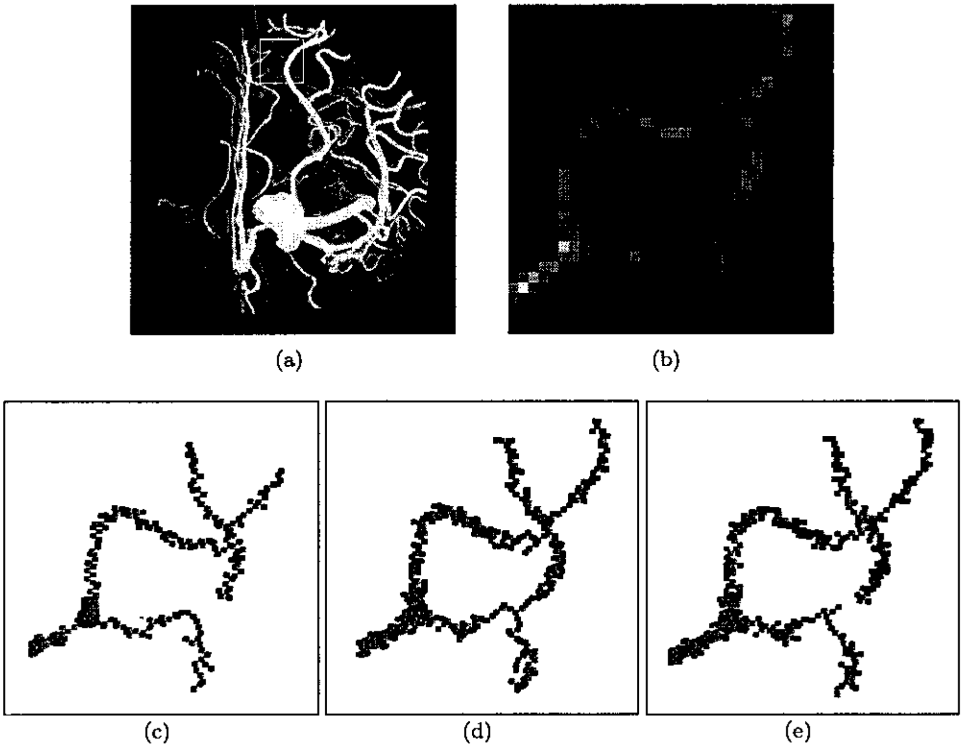
Centroid position is not an intuitive feature to use for defuzzification of a vessel tree. It may interfere in undesirable ways with the topology preservation during the search procedure, so we exclude centroid from the feature representation in this example.

It is clear that high resolution reconstruction is really needed here; any crisp representative at the same resolution as the original image would be a rather bad representation; to preserve the volume of the fuzzy image, many parts of the vessel tree would not be included in the crisp set.

Performing reconstruction at two times the original resolution, we get the result presented in Figure 27(c). The result is not visually appealing, due to severe underestimation of the surface area of a crisp thin (less than one voxel thick) structure by the surface area of the fuzzy set. This problem is not present for a crisp object whose fuzzy representation is obtained at sufficiently high resolution and contains points with memberships equal to one in the interior of the object. In the case presented in Figure 27, however, the reconstruction using the inaccurate surface area estimate fails to preserve the vessel structure.

It would be of high interest to have a better surface area estimate for the defuzzification; study about this feature estimate is certainly included in our future work. In the absence of such, we perform reconstruction without the surface area feature. Using only volume based information (at a range of scales) the high resolution reconstruction is fairly unconstrained, which leads to the rather jagged result of Figure 27(d). Dropping the meso-scale feature from the feature representation, we get the result presented in Figure 27(e).

We note that, although not visible in Figure 27, the topology is in deed preserved; all the resulting objects are simply connected. However, the vessels may not always be connected in a correct way, so some additional information on how vessels branch and bend may be required in this case.



(a) (b)

(c) (d) (e)

FIGURE 27. Defuzzification of a selected part of an angiography 3D image, showing the arteries of the right half of a human head. (a) Maximum intensity projection through the image volume; the white square in the upper part of the image indicates the location of the selected part of the volume that is defuzzified in this example. (b) Maximum intensity projection through the selected part the volume. (c) 3D rendering of a defuzzification at twice the resolution using volumes of all scales and surface area. (d) 3D rendering of a defuzzification at twice the resolution using only volumes of all scale. (e) 3D rendering of a defuzzification at twice the resolution using only global and local volumes.

8. Conclusions and future work

In the field of computerized image processing and analysis, increasing attention is lately given to methods that provide results with sub-pixel precision. Such methods are especially important for applications where precision is a key factor, for example in medicine. When working with 3D images (CT, MR), systematic errors of the size of a pixel may accumulate to unacceptably large overall errors in the final results. In addition, analysis of images at low resolution is constantly a hot research topic;

with the resent progress in imaging techniques, allowing imaging to reach nanometer scales, a previously inaccessible world of structures of sizes all the way down to molecular scale, opens up.

The work presented in this chapter is a summary of our research on development of image processing methods that provide results at sub-pixel precision; this research was conducted during last couple of years. In our opinion, it is of highest importance for the field of image processing to go on with this type of research and with further development of high precision image processing tools and algorithms; a call imposed by the technological development is already shown to be well addressed by this very promising research track with obvious high applicability.

Object representations that facilitate utilization of sub-pixel precision methods, and particularly those based on spel coverage by an object, were in our research focus throughout the recent years. We developed segmentation methods characterized by generality and wide applicability, as well as some important feature estimators. We explored crisp object reconstruction methods based on the derived feature estimators and we showed that information preserved in coverage representation can be successfully used to compensate for lacking spatial resolution, both in estimations, and in reconstructions. The following natural step in research is development of more methods for analysis of images segmented by some of the coverage segmentation methods, in an attempt to make a complete tool-box of processing methods for images providing sub-pixel information precision. Tasks related to development of methods for feature estimation and object description, primarily in 3D, will be first addressed. Our intention is to develop strong theoretical background for every method we suggest, and to test and prove their applicability on real world tasks and challenges.

Application fields for our developed methods, as well as those that are to be developed, are numerous. Medical applications have already been addressed, and increased precision of the methods has shown to be of high importance and benefit for them. We intend to continue with development of image registration methods; an additional research track will be improvement of distance measures so that they are applicable to objects represented at sub-pixel precision. Developed appropriate distances, together with new high precision features (descriptors) will find applications in content retrieval. Studies and utilization of sub-pixel methods in the field of optical character recognition (OCR), being of high interest in a wide range of fields, including, e.g., digitization of cultural heritage, are also envisioned in our future work.

Finally, an application field that is particularly in our interest for future work, is the field of biometrics. Biometrics, which in a broad sense refers to the science and technology of measuring and statistically analyzing biological data, and therefore already concerns the presented medical applications of our work, is attracting more and more attention also outside the fields of medicine and biomedicine, e.g., in (information) security, where it finds use for identification and authentication purposes. Biometrics, provides a variety of research challenges; increased precision of representations and analysis methods are certainly of highest importance for biometrics applications, and we feel that our approaches, involving coverage (sub-pixel precision) models, may provide improvements for methods used in biometrics tasks.

Biometrics considers methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioural properties/characteristics. In particular, biometrics is used as a form of identity access control, being of highest importance in (information) security related issues. Applications of biometrics recognition systems include computer systems security, secure electronic banking, mobile phones, credit cards, secure access to buildings, health and social services. As opposed to several traditionally used access control tools, such as passwords and ID cards, biometric can not be borrowed, stolen, or forgotten, which makes it increasingly popular in security systems. Nowadays, most often used biometrics characteristics and methods are fingerprint, face, DNA, palm print, hand geometry, and iris recognition.

Biometric recognition assumes representation of individuals by a feature vector derived from their physiological and/or behavioural characteristic; automatic recognition of these individuals becomes recognition of their representations. Feature vectors are usually stored in appropriate databases. Searching procedures and matching algorithms are typical tasks involved. Our so far developed methods involve feature extraction algorithms, feature selection and object representations, geometric matching of objects, as well as studies related to distance measures, [26]. Performance of biometrics systems strongly depends on distinctiveness of the representations, as well as their accuracy and precision. It should be possible to measure/estimate the selected features accurately every time when data is required and acquired, which demands high precision of estimation algorithms used: it is also important to have distance/similarity measures and matching algorithms offering high precision performance, to ensure full discriminative power of the system. All these requirements are met by our proposed concept of sub-pixel (coverage) model. We find very promising to adjust existing algorithms and propose new ones to address the tasks of this rapidly developing and highly important field.

Acknowledgement. This work is partially supported by the Serbian Ministry of Education and Science, the projects ON174008, "Advanced Techniques of Cryptology, Image Processing and Computational Topology for Information Security", and III44006, "Development of new information and communication technologies, based on advanced mathematical methods, with applications in medicine, telecommunications, power systems, protection of national heritage and education".

References

[1] I. Bloch, *Fuzzy spatial relationships for image processing and interpretation: A review*, Image Vis. Comput. **23** (2005), 89–110.

[2] G. Borgefors, I. Nyström, and G. Sanniti di Baja, *Connected components in 3D neighbourhoods*, in: *Proc. of Scand. Conf. on Image Analysis*, Pat. Rec. Soc. of Finland, 1997, 567–572.

[3] N. Bousion, M. Hatt, A. Reilhac, and D. Visvikis, *Fully automated partial volume correction in PET based on a wavelet approach without the use of anatomical information*, in: *Proc. of IEEE Nuclear Science Symp.*, IEEE, 2007, 2812–2816.

[4] J. Chanussot, I. Nyström, and N. Sladoje, *Shape signatures of fuzzy sets based on distance from the centroid*, Pattern Recognit. Lett. **26(6)** (2005), 735–746.

[5] H.S. Choi, D.R. Haynor, and Y. Kim, *Partial volume tissue classification of multichannel magnetic resonance images-a mixed model*, IEEE Trans. on Medical Imaging **10** (1991), no. 3, 395–407.

[6] D. Coeurjolly and R. Klette, *A comparative evaluation of length estimators of digital curves*, IEEE Trans. Pattern Anal. Mach. Intell. (2004), 252–258.

[7] H. Davenport, *On a principle of Lipschitz*, J. London Math. Soc. **s1-26** (1951), no. 3, 179–183.

[8] C. Domokos, Z. Kato, and J.M. Francos, *Parametric estimation of affine deformations of binary images*, in: *Proc. of Int. Conf. on Acoustics, Speech, and Signal Process.* (Las Vegas, USA), IEEE, April 2008, 889–892.

[9] L. Dorst and A.W.M. Smeulders, *Length estimators for digitized contours*, Comput. Vis. Graph. Image Process. **40** (1987), 311–333.

[10] D. Dubois and M.-C. Jaulent, *A general approach to parameter evaluation in fuzzy digital pictures*, Pattern Recognit. Lett. **6** (1987), 251–259.

[11] D. Eberly and J. Lancaster, *On gray scale image measurements: I. Arc length and area*, CVGIP: Graph. Models Image Process. **53** (1991), no. 6, 538–549.

[12] C. Florea, C. Vertan, and L. Florea, *Logarithmic model-based dynamic range enhancement of hip X-ray images*, in: *Proc. of Advanced Concepts for Intell. Vision Systems* (Delft, Netherlands), LNCS, vol. 4678, Springer-Verlag, 2007, 587–596.

[13] G.M. Foody, *Sharpening fuzzy classification output to refine the representation of sub-pixel land cover distribution*, Int. J. Remote Sensing **19** (1998), no. 13, 2593–2599.

[14] H. Freeman, *On the encoding of arbitrary geometric configurations*, IRE Trans. Electron. Comput. **EC-10** (1961), 260–268.

[15] L. Grady and M.-P. Jolly, *Weights and topology: A study of the effects of graph construction on 3D image segmentation*, in: *Proc. of Int. Conf. on Medical Image Computing and Comput. Assisted Intervention*, vol. 1, Springer-Verlag, 2008, 153–161.

[16] L. Grady, *Space-variant machine vision-a graph theoretic approach*, Ph.D. thesis, Boston University, 2004.

[17] R. Hagege and J.M. Francos, *Linear estimation of sequences of multi-dimensional affine transformations*, in: *Proc. of Int. Conf. on Acoustics, Speech and Signal Process.* (Toulouse, France), vol. 2, IEEE, 2006, 785–788.

[18] M.K. Hu, *Visual pattern recognition by moment invariants*, IRE Trans. Inform. Theory **8** (1962), 179–187.

[19] J. Jua, E.D. Kolaczykb, and S. Gopa, *Gaussian mixture discriminant analysis and sub-pixel land cover characterization in remote sensing*, Remote Sensing and Environment **84** (2003), no. 4, 550–560.

[20] N. Kiryati and A. Bruckstein, *Grey levels can improve the performance of binary image digitizers*, CVGIP: Graph. Models Image Process. **53** (1991), no. 1, 31–39.

[21] R. Klette, V. Kovalevsky, and B. Yip, *Length estimation of digital curves*, in: *Proc. of SPIE Vision Geometry VIII*, 1999, (expanded version available: Machine Graphics & Vision, vol. 9, pp. 673-703, 2000.), 117–129.

[22] R. Klette and J. Žunić, *Multigrid convergence of calculated features in image analysis*, J. Math. Imaging Vis. **13** (2000), 173–191.

[23] Z. Kulpa, *Area and perimeter measurement of blobs in discrete binary pictures*, Comput. Graphics and Image Process. **6** (1977), 434–454.

[24] W. Van Leekwijck and E.E. Kerre, *Defuzzification: Criteria and classification*, Fuzzy Sets Syst. **108** (1999), 159–178.

[25] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens, *A unifying framework for partial volume segmentation of brain MR images*, IEEE Trans. on Medical Imaging **22** (2003), no. 1, 105–119.

[26] J. Lindblad, V. Ćurić, and N. Sladoje, *On set distances and their application to image registration*, in: *Proc. of Int. Symp. on Image and Signal Process. and Analysis* (Salzburg, Austria), IEEE, 2009, 449–454.

[27] J. Lindblad, T. Lukić, and N. Sladoje, *Defuzzification by feature distance minimization based on DC programming*, in: *Proc. of Int. Symp. on Image and Signal Process. and Analysis* (Istanbul, Turkey), IEEE, 2007, 373–378.

[28] J. Lindblad and N. Sladoje, *Feature based defuzzification at increased spatial resolution*, in: *Proc. of Int. Workshop on Combinatorial Image Analysis* (Berlin, Germany), LNCS, vol. 4040, Springer-Verlag, 2006, 131–143.

[29] J. Lindblad, N. Sladoje, V. Ćurić, H. Sarve, C. B. Johansson, and G. Borgefors, *Improved quantification of bone remodelling by utilizing fuzzy based segmentation*, in: *Proc. of Scand. Conf. on Image Analysis* (Oslo, Norway), LNCS, vol. 5575, Springer-Verlag, 2009, 750–759.

[30] J. Lindblad, N. Sladoje, and T. Lukić, *Feature based defuzzification in Z^2 and Z^3 using a scale space approach*, in: *Proc. Int. Conf. on Discrete Geometry for Comput. Imagery* (Szeged, Hungary), LNCS, vol. 4245, Springer-Verlag, 2006, 379–390.

[31] S. Loncaric, *A survey of shape analysis techniques*, Pattern Recognition **31** (1998), no. 8, 983–1001.

[32] T. Lukić, N. Sladoje, and J. Lindblad, *Deterministic defuzzification based on spectral projected gradient optimization*, in: *Proc. of Symp. of the German Association for Pattern Recognition* (Munich, Germany), LNCS, vol. 5096, Springer-Verlag, 2008, 476–485.

[33] F. Malmberg, J. Lindblad, and I. Nyström, *Sub-pixel segmentation with the Image Foresting Transform*, in: *Proc. of Int. Workshop on Combinatorial Image Analysis* (Playa del Carmen, Mexico), LNCS, vol. 5852, Springer-Verlag, 2009, 201–211.

[34] F. Malmberg, J. Lindblad, N. Sladoje, and I. Nyström, *A graph-based framework for sub-pixel image segmentation*, Theor. Comput. Sci. **412** (2011), no. 15, 1338–1349.

[35] F. Malmberg, I. Nyström, A. Mehnert, C. Engstrom, and E. Bengtsson, *Relaxed Image Foresting Transforms for interactive volume image segmentation*, in: *Proc. of SPIE Medical Imaging*, vol. 7623, SPIE, 2010.

[36] W. J. Niessen, K. L. Vincken, J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever, *Multiscale segmentation of three-dimensional MR brain images*, Int. J. Comput. Vision **31** (1999), no. 2/3, 185–202.

[37] A. Oprea and C. Vertan, *A quantitative evaluation of the hip prosthesis segmentation quality in X-ray images*, in: *Proc. of Int. Symp. on Signals, Circuits and Systems* (Iasi, Romania), vol. 1, IEEE, 2007, 1–4.

[38] N. Otsu, *A threshold selection method from gray-level histograms*, IEEE Trans. on System Man and Cybernetics **9** (1979), no. 1, 62–66.

[39] D. Proffit and D. Rosen, *Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges*, Comput. Graphics and Image Process. **10** (1979), 318–332.

[40] L. Rondeau, R. Ruelas, L. Levrat, and M. Lamotte, *A defuzzification method respecting the fuzzification*, Fuzzy Sets Syst. **86** (1997), 311–320.

[41] A. Rosenfeld, *The fuzzy geometry of image subsets*, Pattern Recognit. Lett. **2** (1984), 311–317.

[42] A. Rosenfeld, *Fuzzy geometry: An updated overview*, Information Sciences **110** (1998), no. 3–4, 127–133.

[43] A. Rosenfeld and S. Haber, *The perimeter of a fuzzy subset*, Pattern Recognition **18** (1985), 125–130.

[44] E. Roventa and T. Spircu, *Averaging procedures in defuzzification processes*, Fuzzy Sets Syst. **136** (2003), 375–385.

[45] E. H. Ruspini, *A new approach to clustering*, Inf. Control **15** (1969), 22–32.

[46] P. Santago and H. D. Gage, *Quantification of MR brain images by mixture density and partial volume modeling*, IEEE Trans. on Medical Imaging **12** (1993), no. 3, 566–574.

[47] H. Sarve, J. Lindblad, C. B. Johansson, G. Borgefors, and V. F. Stenport, *Quantification of bone remodeling in the proximity of implants*, in: *Proc. of Int. Conf. on Comput. Analysis of Images and Patterns* (Vienna, Austria), LNCS, vol. 4673, Springer-Verlag, 2007, 253–260.

[48] N. Sladoje, *A straight line segment estimation by using discrete moments*, in: *Proc. of XIII Conf. on Applied Mathematics* (Igalo, Yugoslavia), 1998, 121–129.

[49] N. Sladoje, *On analysis of discrete spatial fuzzy sets in 2 and 3 dimensions*, Ph.D. thesis, Swedish University of Agricultural Sciences, Uppsala, 2005.

[50] N. Sladoje and J. Lindblad, *Estimation of moments of digitized objects with fuzzy borders*, in: *Proc. of Int. Conf. on Image Analysis and Process.* (Cagliari, Italy), LNCS, vol. 3617, Springer-Verlag, 2005, 188–195.

[51] N. Sladoje and J. Lindblad, *Perimeter estimation based on grey level object representation*, Internal Report 33, Centre for Image Analysis, Uppsala, Sweden, 2008, Available from the authors.

[52] N. Sladoje and J. Lindblad, *High-precision boundary length estimation by utilizing gray-level information*, IEEE Trans. Pattern Anal. Mach. Intell. **31** (2009), no. 2, 357–363.

[53] N. Sladoje and J. Lindblad, *Pixel coverage segmentation for improved feature estimation*, in: *Proc. of Int. Conf. on Image Analysis and Process.* (Vietri sul Mare, Italy), LNCS, vol. 5716, Springer-Verlag, 2009, 923–938.

[54] N. Sladoje, J. Lindblad, and I. Nyström, *Defuzzification of spatial fuzzy sets by feature distance minimization*, Image Vis. Comput. **29** (2011), no. 2-3, 127–141.

[55] N. Sladoje, I. Nyström, and P. K. Saha, *Measurements of digitized objects with fuzzy borders in 2D and 3D*, Image Vis. Comput. **23** (2005), 123–132.

[56] M. Soret, S. L. Bacharach, and I. Buvat, *Partial-volume effect in PET tumor imaging*, The J. Nuclear Medicine **48** (2007), no. 6, 223–235.

[57] A. Souza, J. K. Udupa, and P K. Saha, *Volume rendering in the presence of partial volume effects*, IEEE Trans. on Medical Imaging **24** (2005), no. 2, 223–235.

[58] R. Strand, *Distance functions and image processing on point-lattices*, Ph.D. thesis, Uppsala University, 2008.

[59] M. Tajine and A. Daurat, *On local definitions of length of digital curves*, in: *Proc. of Int. Conf. on Discrete Geometry for Comput. Imagery* (Naples, Italy), LNCS, vol. 2886, Springer-Verlag, Nov. 2003, 114–123.

[60] A. Tanács, C. Domokos, N. Sladoje, J. Lindblad, and Z. Kato, *Recovering affine deformations of fuzzy shapes*, in: *Proc. of Scand. Conf. on Image Analysis* (Oslo, Norway), LNCS, vol. 5575, Springer-Verlag, 2009, 735–744.

[61] A. Tanacs, J. Lindblad, N. Sladoje, and Z. Kato, *Estimation of linear deformations of 3D objects*, in: *Proc. of IEEE Int. Conf. on Image Process.* (Hong Kong), IEEE, 2010, 153–156.

[62] J. K. Udupa and G. J. Grevera, *Go digital, go fuzzy*, Pattern Recognit. Lett. **23** (2002), 743–754.

[63] P. W. Verbeek and L. J. van Vliet, *Estimators of 2D edge length and position, 3D surface area and position in sampled grey-valued images*, Bioimaging **1** (1993), 47–61.

[64] B. Verwer, *Local distances for distance transformations in two and three dimensions*, Pattern Recognit. Lett. **12** (1991), 671–682.

[65] K. L. Vincken, A. S. E. Koster, and M. A. Viergever, *Probabilistic segmentation of partial volume voxels*, Pattern Recognit. Lett. **15** (1994), 477–484.

[66] A. M. Vossepoel and A. W. M. Smeulders, *Vector code probability and metrication error in the representation of straight lines of finite length*, Comput. Graphics Image Process. **20** (1982), 347–364.

[67] J. Žunić and N. Sladoje, *Efficiency of characterizing ellipses and ellipsoides by discrete moments*, IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000), no. 4, 407–414.

[68] L. Zadeh, *Fuzzy sets*, Inf. Control **8** (1965), 338–353.

[69] B. Zitová and J. Flusser, *Image registration methods: A survey*, Image Vis. Comput. **21** (2003), no. 11, 977–1000.

Miodrag J. Mihaljević*

ON CERTAIN APPROACHES FOR ANALYSIS AND DESIGN OF CRYPTOGRAPHIC TECHNIQUES FOR SYMMETRIC ENCRYPTION AND KEY MANAGEMENT

Abstract. This chapter yields a review of certain mathematical approaches for analysis and design of the basic cryptographic elements for establishing information security in information-communication systems. The following two topics are addressed: selected issues on stream ciphers for encryption and key management based on broadcast encryption. Certain coding related issues for security evaluation and design of stream ciphers are considered. The discussed security evaluation techniques corresponding to decoding approaches include one-step and iterative decoding paradigms, and the addressed design issues involve homophonic coding for joint employment of pseudorandomness and randomness. Elements for cryptographic security evaluation and advanced design of the key managements based on broadcast encryption are pointed out.

Mathematics Subject Classification (2010): Primary: 94-02, 94A60; Secondary 11T71.

Keywords: cryptology, cryptography, cryptanalysis, pseudorandomness, randomness, coding theory.

**Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihaila 36, Belgrade*

CONTENTS

1. Introduction

Establishing data security in information-communication systems or more generally, establishing cyber-security is one of the most important issues in order to avoid that information-communication technologies become misused with potentially catastrophic impacts. Cryptology is a mathematical discipline which provides basic methods and techniques for establishing elements of mechanisms for information and communications security. Employing cryptology we can develop a large number of different elements for achieving the security goals including the following main ons: (i) secrecy and privacy; (ii) integrity control; (iii) authenticity control (including the non-repudiation). For achieving the previous goals, cryptology deals with design and security evaluation of certain cryptographic primitives. The cryptographic primitives are mathematical algorithms which mainly (but not always) involve certain secret data for achieving the addressed goals. These secret data are called cryptographic keys or keys (for short). Among the main cryptographic primitives are the ones for encryption and key management. Cryptographic primitives for encryption are basic elements for the secrecy protection, and the ones for key management are main elements of the necessary "infrastructure" management the secret data employed for encryption and in a number of other cryptographic primitives. It is out of the scope of this chapter to serve as an introduction to cryptology and regarding this issue an interested reader is advised to check some of the related text books like [98] (which is available at http://cacr.uwaterloo.ca/hac/).

This chapter is devoted to the following two cryptographic primitives: stream ciphers for encryption and key management based on broadcast encryption. Selection of the addressed topics and contents of this chapter originate from the results reported in [1]–[62].

Encryption Based on Stream Ciphers. Stream ciphers play an important role in information security and they are a well recognized topic within cryptology. A stream cipher encrypts one individual character of a plaintext message at a time, using an encryption transformation which varies with time. Such a cipher is typically implemented by the use of a pseudorandom number generator or a keystream generator which expands a short secret key into a long running key sequence. A keystream generator is equivalent to a finite state machine that, based on a secret key, generates a keystream for controlling an encryption transformation. Design of highly efficient and secure stream ciphers is still an important challenge.

This chapter addresses certain coding related issues for security evaluation and design of stream ciphers. The discussed security evaluation techniques corresponding to decoding approaches include one-step and iterative decoding paradigms.

Key Management Based on Broadcast Encryption. In order to perform symmetric encryption/decryption the secret session key should be shared between the encryption and decryption entities. Broadcast encryption is a technique for distribution, via a public communication channel, secret session keys employing the pre-shared secret keys which provide that only selected parties can learn the secret session key. This chapter provides elements for cryptographic security evaluation and advanced design of the key managements based on broadcast encryption.

2. Decoding Based Approach for Security Evaluation of Certain Stream Ciphers

A number of the published keystream generators are based on binary linear feedback shift registers (LFSRs) assuming that parts of the secret key are used to load the LFSRs initial states (see [98], for example).

Note that a binary LFSR generate recurrence sequences over GF(2), and under certain assumption these sequences have the maximum possible period (for the given recurrence order) and good properties of pseudorandomnes.

The unpredictability request, which is one of the main cryptographic requests, implies that the linearity inherent in LFSRs should not be "visible" in the generator output. One general technique for destroying the linearity is to use several LFSRs which run in parallel, and to generate the keystream as a nonlinear function of the outputs of the component LFSRs. Particularly, suitable Boolean functions can be employed for realization of the nonlinear mapping. Such keystream generators are called nonlinear combination generators (see [98], for example).

Accordingly, an output sequence from nonlinear combination generator can be considered as follows:

$$y_i = f(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(m)}), i = 1, 2, \dots,$$

and assuming the binary case,

$$f(\cdot) : \{0,1\}^m \rightarrow \{0,1\},$$

$$x_i^{(j)} = \bigoplus_{l=1}^{L} \alpha_l^{(j)} \cdot x_{j-l}^{(j)}, \;\; \alpha_l^{(j)} \in \{0,1\},\; l = 1, 2, \dots, L,\; j = 1, 2, \dots, m,$$

assuming that $\{x_i^{(j)}\}_i$, $j = 1, 2, \dots, m$, are binary sequences.

This section yields a brief overview of a decoding based approach for security evaluation of the combination keystream generators.

Fast Correlation Attack. A central weakness of a nonlinear combination keystream generator has been demonstrated in [106]. Assuming certain nonlinear functions it is shown in [106] that it is possible to reconstruct independently initial states of the LFSRs, i.e. parts of the secret key (and accordingly the whole secret key as well) based on the correlation between the keystream generator output and the output of each of the LFSRs. The reported approach is based on exhaustive search through all possible nonzero initial states of each LFSR. A substantial improvement of the previous approach which yields nonexponential complexity with the LFSR length has been proposed in [97]. This approach is called fast correlation attack (FCA), and its extensions and refinements, as well as its analysis are presented in a number of papers including [29], [26], [73], [28] and [6].

The basic ideas of all reported FCAs include the following two main steps: (i) Transform the cryptographic problem into a suitable decoding one; (ii) Apply (devise) an appropriate decoding algorithm.

In the following, correlation means that the mod-2 sum of the LFSR output and the generator output can be considered as a realization of a binary random variable taking values 0 and 1 with probabilities $1-p$ and p, respectively, with $p < 0.5$ (or $p \neq 0.5$). Consequently, the problem of the LFSR initial state reconstruction based on the keystream generator output sequence can be considered as the decoding problem of a punctured simplex code (defined by the feedback connections of the LFSR) after transmission over a binary symmetric channel (BSC) with crossover probability p uniquely determined by the correlation. More precisely, the fast correlation attack on a particular LFSR in a nonlinear combining generator given the segment of the generator output can be considered as follows: (i) The N-bit segment of the output sequence from the length-L LSFR is a codeword of an (N, L) punctured simplex code; (ii) The corresponding N-bit segment of the nonlinear combination generator output is the corresponding noisy codeword obtained through a BSC with crossover probability p; (iii) The problem of the LFSR initial state reconstruction, assuming its characteristic polynomial is known, is the problem of decoding the (n, k) punctured simplex code transmitted over a BSC with crossover probability p.

Two main classes of the reported FCAs are one-step decoding and iterative decoding based fast correlation attacks.

FCAs based in One-Step Decoding. Powerful approaches for FCAs realization based on one-step decoding have been reported in [29], [26], and further developed in a number of references including [73]. These techniques are based on a threshold decoding for reconstruction of all information bits under a hypotheses of certain B bits in conjunction with exhaustive search over all 2^B possibilities. The analysis of these algorithms include the results reported in [63] implying the high efficiency assuming an enough long sample for cryptanalysis.

FCAs based on Iterative Decoding. Certain approaches for FCAs based on iterative decoding which have performance invariant on the weight of the LFSR feedback polynomial have been reported in [29] (IDA) and [28]. These methods employ a number of moderate-weight parity checks available under assumption of certain exhaustive search in conjunction with a iterative decoding techniques. Four different iterative decoding techniques have been considered in [28] and their performance have been experimentally justified showing efficiency when only the short samples are available for cryptanalysis. The origins for the iterative based decoding FCAs reported in [30] include [32], [33] and [31].

Implications on Security Evaluation and Design of Stream Ciphers. The considered one-step decoding based FCAs appear as a powerful tool for security evaluation of certain stream ciphers assuming that enough long sample for cryptanalysis is available. The performance of these FCAs can be heavily degraded if only a short sample is available for cryptanalysis. In these scenarios, when only short samples are available, the iterative decoding based FCA considered in this section appears as a suitable alternative.

Accordingly, the security evaluation and the design guidelines should take into account considering both the one-step and iterative decoding based FCAs.

3. A Low-Complexity and High-Performance Algorithm for the Fast Correlation Attack

As an in-details illustration of the topic, this section provides a self-contained presentation of an algorithm for FCA which has the following two advantages over the related previously reported ones: (i) it is more powerful and (ii) it provides a high-speed software implementation, as well as a simple hardware one, suitable for (highly) parallel architectures. This chapter is mainly based on the results reported in [49], [46], [29], [26], [30], [28] and [6].

The discussed algorithm is a method for the fast correlation attack with significantly better performance in comparison with the previously reported methods, assuming a lower complexity and the same inputs. The algorithm is based on decoding procedures of the corresponding binary block code with novel constructions of the parity-checks, and employment of the following two decoding approaches: the a posterior probability based threshold decoding and the belief propagation based bit-flipping iterative decoding. These decoding procedures offer good tradeoffs between the required sample length, overall complexity and performance. The discussed algorithm is compared with previously reported fast correlation attacks based on convolutional codes and turbo decoding: The underlying principles, performance and complexity are compared, and the gain obtained is pointed out.

3.1. Preliminaries. An important method for attack or security examination of certain stream ciphers based on nonlinear combination keystream generators composed of several linear feedback shift registers (LFSR's) (see [98], for example) are the basic correlation attack [106], and particularly the fast correlation attacks considered in a number of papers, including [97], [49], [46], [74], [87], [88] and [33]. Developing or improving techniques for realization of the fast correlation attack is still an important topic of cryptology.

The basic ideas of all reported fast correlation attacks include the following two main steps:

–Transform the cryptographic problem into a suitable decoding one;

–Apply (devise) an appropriate decoding algorithm.

There are two main approaches for realization of the fast correlation attack. The first one is based on decoding techniques for block codes (introduced in [97] and [109]), and the second one is based on decoding techniques for convolutional codes (proposed in [87] and [88]).

The main underlying ideas for a number of the fast correlation attacks based on linear binary block codes decoding is the iterative decoding principle introduced in [79]. For example, the fast correlation attacks reported in [97], [109], [49], [46] and [74], could be considered as variants of iterative decoding based on simple bit-flipping (BF) [79] or iterative extensions of *a posterior probability* (APP) decoding. Most of these methods (practically all except the method from [49]) are restricted on the LFSR feedback polynomials of low weight. Due to the established advantages of belief propagation (BP) based iterative decoding over iterative APP (see [32], for example), the application of BP based iterative decoding for realization of the fast correlation attack has been reported in [33]. The main goal of [33] was to report the potential gain and its origins when BP based iterative decoding is employed instead of APP based decoding, assuming the same construction method of the parity-checks and the same overall structure of the algorithm for fast correlation attack. A comparison of the iterative decoding approaches based on simple, APP and BF based decodings for the fast correlation attack is reported in [46].

Alternative approaches for fast correlation attack based on the theory of convolutional codes are given in [87]-[88]. They can be applied to arbitrary LFSR feedback polynomials, in opposite to the previous methods, which mainly focus on feedback polynomials of low weight. The proposed algorithm transforms a part of the code **C** steaming from the LFSR sequence into a convolutional code, based on finding suitable parity check equations for **C**. The approach considers a decoding algorithm that includes memory, but still has a low decoding complexity. With respect to the previous methods, this allows looser restrictions on the parity check equations that can be used, leading to many more equations. As the final decoding method, the Viterbi algorithm with memory orders of 10-15 was used. The results reported in [87] improve significantly the few previous results for high weight feedback polynomials, and are in many cases comparable with that corresponding to low weight feedback polynomials. Further developments of the idea for fast correlation attack based on decoding of certain convolutional codes are presented in [88] where new methods employing the techniques used for constructing and decoding turbo codes are proposed. The most powerful technique presented in [88] is based on the turbo decoding approach with M component convolutional codes and iterative APP decoding employing the BCJR algorithm [67].

Interests and the advances in developing algorithms for the fast correlation attack have raised a natural question of further improvements of the fast correlation attack, especially in the light of fast implementations.

The main goal of this section is to discuss the algorithm reported in [29] for the fast correlation attack suitable for a high-speed software implementation, as well as for a simple hardware one. Most previously reported algorithms can be considered as inappropriate ones for this goal assuming an LFSR feedback polynomial of arbitrary weight. Accordingly, the intention is to point out to an algorithm which employs *mod*2 additions and simple logical operations for processing, so that it is suitable for highly parallel architectures and high speed software or hardware implementations. Also, our goal is to point out to an algorithm which yields possibility for trade-offs between length of the required sample, overall complexity and performance.

In this section, a powerful algorithm for the fast correlation attack [29] is presented. The discussed algorithm is based on a novel method for constructing the parity-checks, motivated by the approach of [87] and [88], and two decoding approaches of the corresponding binary block code, APP threshold decoding and iterative decoding employing BP-like BF (see [79]). The construction of the parity-checks is based on searching for certain parity-check equations and theirs linear combinations employing the finite-state machine model of an LFSR with primitive characteristic polynomial. The expected numbers of parity-checks per parity bit are derived, showing that a large number of appropriate parity-checks can be constructed. An analysis of the algorithm performance and complexity is presented. The novel algorithm is compared with recently proposed improved fast correlation attacks based on convolutional codes and turbo decoding. The underlying principles, performances and complexities are compared, and the gains obtained with the novel approach are pointed out. It is shown that assuming the same input, the novel algorithm yields better performance and lower complexity than the best algorithm reported before it.

This section is organized as follows. Subsection 2 presents preliminaries. Subsection 3 points out the main underlying results for the construction of a novel algorithm for the fast correlation attack. Complete specification of the proposed algorithm is given in subsection 4. Experimental analysis of the performance is presented in subsection 5, as well as a discussion of the complexity issue. Comparisons between the previously reported fast correlation attacks, and in [29] proposed algorithm are given in subsection 6. Finally, the main issues are summarized in subsection 7.

3.2. Decoding Concept for the Fast Correlation Attack.

Recall that, the correlation means that the mod 2 sum of corresponding outputs of the LFSR and the generator can be considered as a realization of a binary random variable which takes value 0 and 1 with the probabilities $1 - p$ and p, respectively, $p \neq 0.5$.

The fast correlation attack on a particular LFSR, with primitive feedback polynomial, in a nonlinear combining generator given the segment of the generator output can be considered as follows:

- The n-bit segment of the output sequence from the length-k LSFR is a codeword of an (n, k) punctured simplex code;
- The corresponding n-bit segment of the nonlinear combination generator output is the corresponding noisy codeword obtained through a BSC with crossover probability p;
- The problem of the LFSR initial state reconstruction, assuming known characteristic polynomial, is equivalent to the problem of decoding after transmission over a BSC with crossover probability p.

The decoding approach employed in this section is based on combination of a restricted exhaustive search over a set of hypotheses and a one-step or an iterative decoding technique. The exhaustive search is employed in order to provide a possibility for construction of suitable parity-check equations relevant for high performance of complete decoding. This approach could be considered as a particular combination of the minimum distance decoding and another decoding technique.

Recall that a parity-check equation which involves a smaller number of bits is more powerful than a higher weight one. Also note that performance associated with a set of the parity-checks depends on its cardinality as well as on the parity-check weight distribution. Finally, the overall complexity of a decoding procedure depends on the number and weights of the employed parity-checks. Accordingly, from performance and complexity point of views, a favorable situation corresponds to the availability of a large number of low-weight parity-checks.

In the following, x_n, $n = 1, 2, \ldots, N$, denotes an LFSR output sequence which is a codeword $\mathbf{x}$ of a binary (N, L) punctured simplex code $\mathbf{C}$ where N is codeword length and L is number of information bits. $\mathbf{x}_0 = [x_1, x_2, \ldots, x_L]$ is the vector of information bits identical to the LFSR initial state; $\{z_n\}$ denotes the degraded sequence $\{x_n\}$ after transmission over a BSC with crossover probability p. Accordingly, $z_n = x_n \oplus e_n$, $n = 1, 2, \ldots, N$, where the effect of the BSC with error probability p is modeled by an N-dimensional binary random variable $\mathbf{E}$ defined over $\{0, 1\}^N$ with independent coordinates E_n such that $\Pr(E_n = 1) = p$, $n = 1, 2, \ldots, N$, and e_n is a realization of E_n. Applying a codeword $\mathbf{x} = [x_n]_{n=1}^N \in \mathbf{C}$, to the input of the BSC, we obtain the random variable $\mathbf{Z} = \mathbf{E} \oplus \mathbf{x}$ as a received codeword at its output. Let $\mathbf{z} = [z_n]_{n=1}^N$ and $\mathbf{e} = [e_n]_{n=1}^N$ denote particular values of the random vector variables $\mathbf{Z}$ and $\mathbf{E}$, respectively.

3.3. Parity-Check Sets.

This section points out novel sets of the parity-check equations relevant for construction of an algorithm for the fast correlation attack which will be proposed in the next section. Also, this section points out the expected cardinalities of these sets.

3.3.1. Preliminaries.

An LFSR can be considered as a linear finite state machine. Recall that a linear finite state machine is a realization or an implementation of certain linear operator. Accordingly, a state of a length-L LFSR after t clocks is given by the following matrix-vector product over GF(2):

$$\mathbf{x}_t = \mathbf{A}^t \mathbf{x}_0, \quad t = 1, 2, \ldots,$$

where $\mathbf{x}_t$ is an L dimensional binary vector representing the LFSR state after t clocks, $\mathbf{x}_0$ is an L dimensional binary vector representing the initial LFSR state (in notation that it has index L at the top and index 1 at the bottom), and $\mathbf{A}^t$ is the t-th power over GF(2) of the state transition $L \times L$ binary matrix $\mathbf{A}$. Assuming the LFSR characteristic polynomial $f(u) = 1 + \sum_{i=1}^{L} b_i u^i$, the matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} b_1 & b_2 & b_3 & \ldots & & b_L \\ 1 & 0 & 0 & \ldots & & 0 \\ 0 & 1 & 0 & \ldots & & . \\ . & . & . & \ldots & & . \\ 0 & & & \ldots & 1 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ . \\ \mathbf{A}_L \end{bmatrix}, \tag{3.1}$$

where each $\mathbf{A}_i$, $i = 1, 2, \ldots, L$, represents a $1 \times L$ binary matrix (a row-vector).

Powers of the matrix $\mathbf{A}$ determine algebraic replica of the LFSR initial state bits, i.e. linear equations satisfied by the bits of the codewords from the dual code. Accordingly, they directly specify the parity-checks.

Since our approach assumes an exhaustive search, over the first B information bits, the parity checks are obtained:

–directly from the powers of the matrix $\mathbf{A}$ corresponding to an arbitrary subset of the first B bits of the LFSR initial state and no more than three bits from the remaining $L - B$ bits of the initial state and the bit of the LFSR output sequence;

–as the mod2 sum of any two parity checks determined from the powers of the matrix **A** when this sum includes an arbitrary number of the first B bits of the LFSR initial state, at most one bit from the remaining $L-B$ bits of the initial state, and the two bits of the LFSR output sequence.

–as the mod2 sum of any three parity checks determined by the powers of matrix **A** when this sum includes an arbitrary number of the first B bits of the LFSR initial state, no bit from the remaining $L-B$ bits of the initial state, and the three corresponding bits of the LFSR output sequence.

As previously in this section pointed out, a desirable situation is that corresponding to as many low-weight parity-checks as possible. Following this fact and due to the comparison purposes with recently reported improved fast correlation attacks [87]–[88], we focus our intention mainly to parity-checks of effective weight three (i.e. without considering the first B bits), but also employ some of parity-checks of effective weight four as well (also note that parity-checks of an arbitrary weight could be considered).

3.3.2. Methods for Construction and Specification of the Parity-Check Sets. This subsection presents two methods for obtaining appropriate sets of parity-checks. The developed methods are related to the *information bits* (Method A) and to the *parity bits* (Method B) of the underlying punctured simplex code.

Method A: Parity-check sets related to the *information bits* of the underlying punctured simplex codeword. Note that $x_{L+n} = \mathbf{A}_1^n \mathbf{x}_0$, $n = 1, 2, \ldots, N-L$, where $\mathbf{A}_1^n$ is the first row of the n-th power of the state transition matrix **A**. Accordingly, the basic parity-check equations (defined on the noisy sequence) are given by:

$$c_{L+n} = z_{L+n} \oplus \mathbf{A}_1^n \mathbf{z}_0, \quad n = 1, 2, \ldots, N-L,$$

where $\mathbf{z}_0 = [z_1, z_2, \ldots, z_L]$.

Assuming that the first B information bits are known, appropriate parity-check equations for the i-th information bit, $i = B+1, B+2, \ldots, L$ can constructed according to the following definition.

Definition 3.1. The set Ω_i of parity-check equations associated with information bit-i is composed of:

- All parity-check equations corresponding to the vectors $\mathbf{A}_1^n$ such that each $\mathbf{A}_1^n$ has arbitrary values in the first B coordinates, has value one at the i-th coordinate, and has two ones in all other information bit coordinates;
- All parity-check equations obtained as the mod2 sum of two other basic parity-check equations, $(z_m \oplus \mathbf{A}_1^m \mathbf{z}_0) \oplus (z_n \oplus \mathbf{A}_1^n \mathbf{z}_0)$, where m and n have arbitrary values providing that the vector sum $\mathbf{A}_1^m \oplus \mathbf{A}_1^n$ has arbitrary values in the first B coordinates, value one at the i-th coordinate, and value zero in the all other coordinates.

Note that for given parameters N, L, and B, the sets Ω_i, $i = B+1, B+2, \ldots, L$, can be constructed in advance through a search procedure in a preprocessing phase, and later used for any particular application with these given parameters.

Method B: Parity-check sets related to the *parity bits* of the underlying punctured simplex codeword. First, an appropriate form of the parity check matrix of a punctured simplex code is pointed out. Then a method for constructing the parity checks is given and the parity checks to be employed by the algorithm are specified by Definition 3.2.

Recall, that the fast correlation attack has been modelled by the decoding of an (N, L) punctured simplex code used over a BSC. Accordingly, the following statement points out an appropriate form of the code parity-check matrix. This particular form has a one-to-one correspondence with the finite-state machine model of an LFSR with primitive characteristic polynomial.

Proposition 3.1. *The parity-check matrix* $\mathbf{H} = [\,\mathbf{P}^T, \mathbf{I}_{N-L}\,]$ *of a punctured simplex code* (N, L) *with corresponding polynomial* $f(u) = 1 + \sum_{i=1}^{L} b_i u^i$, *where the binary matrix* $\mathbf{P}$ *is the* $L \times (N - L)$ *matrix of parity checks,* $\mathbf{P}^T$ *is its transpose, and* $\mathbf{I}_{N-L}$ *is the identity matrix of dimension* $(N - L) \times (N - L)$, *is specified by the following:*

$$\mathbf{P}^T = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \cdot \\ \cdot \\ \mathbf{P}_{N-L} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^{(1)} \\ \mathbf{A}_1^{(2)} \\ \cdot \\ \cdot \\ \mathbf{A}_1^{(N-L)} \end{bmatrix},$$

where the m*-th row of the matrix* $\mathbf{P}^T$, *is an* L*-dimensional row vector* $\mathbf{A}_1^{(m)}$ *equal to the first row of the* m*-th power,* $\mathbf{A}^m$, *of the matrix* $\mathbf{A}$ *given in* (3.1).

The construction of the parity-checks is based on searching for certain linear combinations of rows in an appropriate form of the parity-check matrix given by Proposition 3.1. Accordingly, the preprocessing phase of the algorithm includes the construction of the parity-checks according to the following algorithm which generates a set of parity checks for each parity bit. Each parity check includes certain B information bits, and no more than $W + 1$ other arbitrary check bits.

Note that $W + 1$ is used here instead three to illustrate that a straightforward generalization is possible where not only the parity-checks of effective weight equal to three are considered.

Algorithm for the construction of the parity checks

- *Input:* The parity check matrix $\mathbf{H} = [\,\mathbf{P}^T, \mathbf{I}_{N-L}\,]$.
- *Processing Steps:* For each parity bit, generate a set of parity check equations employing the following procedure.
  - For $n = L + 1, L + 2, \ldots, N$ and each w, $1 \leqslant w \leqslant W$, proceed as follows:
    - Calculate the *mod*2-sum of the n-th row of the parity-check matrix $\mathbf{H} = [\mathrm{P}^T, \mathbf{I}_{N-L}]$ and any possible w other rows.
    - If the values at positions $i = B + 1, B + 2, \ldots, L$, are all zeros, where $B < L$, is a predetermined parameter, record the considered combination into the set Ω_n^*.
- *Output:* The sets of parity check equations Ω_n^*, $n = L + 1, L + 2, \ldots, N$.

Definition 3.2. The set Ω_n^* generated by the above algorithm is the set of all considered parity-check equations related to the n-th parity bit of codewords in the punctured (N, L) simplex code.

Note that each parity-check in Ω_n^* consists of α of the first B information bits with $0 < \alpha \leqslant B$, none of the remaining last $L - B$ information bits and at most $W + 1$ of the $N - L$ parity check bits, including bit-n.

3.3.3. Expected Cardinalities of the Parity-Check Sets.

Lemma 3.1. *In any set Ω_i, specified by Definition 3.1, $i = B+1, B+2, \dots, L$, a tight approximation about the expected number $|\bar{\Omega}|$ of the parity-checks is given by the following:*

$$|\bar{\Omega}| = 2^{B-L}\left[(N-L)\binom{L-B-1}{2} + \binom{N-L}{2}\right].$$

Note that Lemma 3.1 motivates the construction of Ω_i given in Definition 3.1. For each type of check sums in Ω_i, that corresponding to minimum weight with non negligible contribution to $|\bar{\Omega}|$ is chosen.

As an illustration, note that for $N = 40000$, $L = 40$ and $B = 18, 19, 20, 21, 22$, Lemma 3.1 yields that the expected cardinality, $|\bar{\Omega}|$ is equal to 192, 384, 768, 1534, 3066, respectively.

Lemma 3.2. *In any set Ω_n^*, specified by Definition 3.2, $n = L+1, L+2, \dots, N$, a tight approximation about the expected number $|\bar{\Omega}^*|$ of the parity-checks is given by the following:*

$$|\bar{\Omega}^*| = 2^{-L+B}\sum_{w=1}^{2}\binom{N-L-1}{w}.$$

As an illustration, note that for $L = 40$, and (N, B)=(1024,26), (4096,22), (8192,20), and (16384,18), Lemma 3.2 yields that the expected cardinalities, $|\bar{\Omega}^*|$ are equal to 29.5, 31.4, 31.7, and 31.9, respectively.

Note that Lemmas 3.1 and 3.2 show that Definitions 3.1 and 3.2 yield large numbers of the parity-checks relevant for an error-correction procedure.

Also, note that Lemmas 3.1 and 3.2 imply that the expected cardinalities of the parity-check sets specified by Definitions 3.1 and 3.2 do not depend on the LFSR characteristic polynomial, and particularly on its weight.

3.4. Algorithm for Fast Correlation Attack. The main underlying principles for construction of the novel fast correlation attack include the following:

- General concepts of linear block codes decoding, and particularly:
  - decoding of information bits only, employing an APP based threshold decoding;
  - iterative decoding of the parity bits employing a reduced complexity BP based iterative decoding.
- A novel method for constructing parity checks of a punctured simplex code based on linear finite state machine model of an LFSR (see [49]);

- The idea (implicitly given in [87]) of employing a partial (restricted) exhaustive search in order to enhance performance of the fast correlation attack. The developed algorithm assumes exhaustive search over the first B information bits in conjunction with appropriate decoding approaches.

According to these principles an algorithm for the fast correlation attack (based on a linear block code decoding approach) has been developed. The algorithm is based on the methods for constructing the appropriate parity-checks presented in the previous section, and its processing phase includes the following three techniques: (i) hypothesis testing, (ii) decoding of a punctured simplex code and (iii) correlation check. The algorithm employs two different decoding procedures in order to provide desired trade-offs between necessary length of the sample, i.e. the rate of underlying code, performance and overall complexity.

Algorithm for the Fast Correlation Attack

INPUT:

- values of the parameters N, L, B, and the threshold T;
- the noisy received bits $z_1, z_2, \ldots, z_N$;
- for each information bit i, $i = B+1, B+2, \ldots, L$, the set Ω_i of corresponding parity-check equations (constructed in the preprocessing phase based on Definition 3.1), and for each parity bit n, $n = L+1, L+2, \ldots, N^*$, $N^* \leqslant N$, the set Ω_n of corresponding parity-check equations (constructed in the preprocessing phase based on Definition 3.2).

PROCESSING STEPS:

(1) *setting the hypothesis*
From the set of all possible 2^B binary patterns, select a not previously considered pattern $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_B$, for the first B information bits. If no new pattern is available, go to the Output (b).

(2) *decoding*
Employ one of the following two decoding algorithms for estimating a candidate for the information bits (i.e. LFSR initial state):

- One-Step Decoding Algorithm (OSDA) using parity-checks specified by Definition 3.1;
- Iterative Decoding Algorithm (IDA) using parity-checks specified by Definition 3.2.

(3) *correlation check*
Check if the current estimation of the information bits (obtained from the decoding step) $\hat{\mathrm{x}}_0 = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_L]$, is the true one, according to the following:
For $\hat{\mathrm{x}}_0$, generate the corresponding sequence $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N$, and calculate $S = \sum_{n=1}^{N} \hat{x}_n \oplus z_n$.
If $S \leqslant T$ go to Output (a), otherwise go to Step 1.

OUTPUT:

(a) the considered vector $\hat{\mathrm{x}}_0$ of information bits is the true one;

(b) the true vector of information bits is not found.

The threshold scalar T is used for checking a hypothesis over all the information bits. For given N, L, B, p, the threshold T is calculated based on the method presented in [106].

The specifications of the employed decoding algorithms OSDA and IDA are given in the following.

3.4.1. One-Step Decoding Algorithm-OSDA. OSDA decodes the noisy received sequence $[z_1, z_2, \ldots, z_N]$ for the (N, L) truncated simplex code employing an APP threshold decoding and the sets Ω_i of parity-check equations, specified by Definition 3.1, $i = B+1, B+2, \ldots, L$ according to the following.

- *parity-checks calculation*
  For each information bit position i, $i = B+1, B+2, \ldots, L$, calculate the parity-check values employing the parity check equations from the set Ω_i.
- *error-correction*
  For each i, $i = B+1, B+2, \ldots, L$ do the following:
  – if the number of satisfied parity-check equations for the considered information bit is smaller than the threshold $T_1(i)$ set $\hat{x}_i = z_i \oplus 1$, otherwise set $\hat{x}_i = z_i$.

The algorithm employs a vector threshold $\mathbf{T}_1 = [T_1(i)]_{i=B+1}^{L}$ which contains values for the APP threshold decoding of certain information bits.

Elements of the threshold vector $\mathbf{T}_1$ are determined based on the posterior error probabilities computed by using the parity-checks specified by Definition 3.1. We assume that for each codeword bit, the parity-checks used are orthogonal on that bit, meaning that except for that bit, every other involved unknown bit appears in exactly one of the parity-checks. Finally, assuming as an appropriate approximation, that all the parity-check equations involve exactly two unknown bits beside the considered one, for any $i = B+1, B+2, \ldots, L$, the threshold $T_1(i)$ is equal to the smallest integer such that the following inequality holds:

$$\frac{p}{1-p}\left(\frac{1+(1-2p)^2}{1-(1-2p)^2}\right)^{|\Omega_i|-2T_1(i)} \leqslant 1,$$

where $|\Omega_i|$ denotes the number of parity-check equations related to the i-th information bit.

3.4.2. Iterative Decoding Algorithm-IDA. For a given $N^* \leqslant N$, IDA decodes the received sequence $[z_1, z_2, \ldots, z_{N^*}]$ for the (N^*, L) punctured simplex code employing a BP based bit-flipping (BP-BF) iterative decoding and the sets Ω_n^* of parity-check equations, specified by Definition 3.2, $n = L+1, L+2, \ldots, N^*$.

BP-BF based iterative decoding (see [79], for example) includes the following main difference in comparison with simple BF.

- For each bit n, and each combination of $|\Omega_n^*| - 1$ parity-checks out of the $|\Omega_n^*|$ parity checks associated with bit-n, make $|\Omega_n^*|$ estimate of the nth bit value associated with these combinations.

Accordingly, we employ the following iterative BP-BF based decoding algorithm.

- *Initialization*: $\hat{x}_n = z_n$ and $\hat{x}_{nm} = z_n$.

- *Iterative Processing*
  (1) *Step* 1:
  (a) For each n and for each $m \in \Omega_n^*$, evaluate:
  $\sigma_n(m) = \sum_{n' \in \omega(m)} \hat{x}_{n'm}$ $[mod2]$.
  (b) If all $\sigma_n(m) = 0$ go to Step 3 (a). If some maximum number of iterations (e.g. 30) is exceeded go to Step 3 (b).
  (2) *Step* 2: For each n, do the following:
  (a) If $\sum_{m}^{|\Omega_n^*|} \sigma_n(m) \geqslant |\Omega_n^*|/2$, then $\hat{x}_n = \hat{x}_n \oplus 1$.
  (b) If $\sum_{m'}^{|\Omega_n^* \setminus m|} \sigma_n(m') \geqslant |\Omega_n^* \setminus m|/2$, then $\hat{x}_{nm} = \hat{x}_{nm} \oplus 1$.
  If no complementation was performed go to Step 3 (b); otherwise go to Step 1.
  (3) *Step* 3:
  (a) $\hat{\mathbf{x}} = [\hat{x}_n]$ is the decoding result.
  (b) Algorithm halts and a warning is declared that a valid decoding is not reached.

3.5. Performance and Complexity.

3.5.1. Performance. The performance of the novel algorithm is experimentally considered when the LFSR characteristic polynomial is chosen as $1 + u + u^3 + u^5 + u^9 + u^{11} + u^{12} + u^{17} + u^{19} + u^{21} + u^{25} + u^{27} + u^{29} + u^{32} + u^{33} + u^{38} + u^{40}$ and $N = 40000$ (i.e. assuming the same example as was considered in [87]–[88]). Note that the proposed algorithm can be applied for values of L significantly longer than $L = 40$, but this value was employed in all numerical and experimental illustrations for comparison with previously reported results.

Results of the performance analysis are presented in Table 1. This table displays the error-rate of the LFSR initial state reconstruction as a function of the correlation noise p when the algorithm employs:

(i) OSDA with B=18, 19, 20, 21, 22,
(ii) IDA with $N^* = 4096$, $B = 22$, and at most 20 iterations.

Each error-rate given in the table is obtained by calculation over a corresponding, randomly selected, set of 1000 samples. Recall that "error-rate" indicates the fraction of trials for which we obtain incorrect decoding (and accordingly incorrect reconstruction of the secret key).

3.5.2. Complexity. Recall that the overall complexity assumes time and space complexity requirements. The complexity analysis yields that according to the structure of the proposed algorithm:

–The algorithm requires a space for the input. Space requirements for the decoding process are as follows: when OSDA is employed, decoding processing does not require memory; IDA requires a memory proportional to the parameter N^*;

–Time complexity is specified by the following corollaries.

Corollary 3.1. *Assuming that OSDA is employed, that $|\Omega|$ denotes the average cardinality of the parity-check sets $|\Omega_i|$, that ω denotes the average number of bits*

TABLE 1. Performance of the novel algorithm-experimental analysis: Error-rate of the LFSR initial state reconstruction, as a function of the correlation noise p when the LFSR length is $L = 40$, the characteristic polynomial weight is 17, and the length of the sequence available for processing is $N = 40000$ bits.

p	Error rate of LFSR initial state reconstruction					
	OSDA $B = 18$	OSDA $B = 19$	OSDA $B = 20$	OSDA $B = 21$	OSDA $B = 22$	IDA $B = 22$, $N^* = 4096$
0.25	0.009	0.000	0.000	0.000	0.000	0.000
0.26	0.015	0.000	0.000	0.000	0.000	0.000
0.27	0.024	0.000	0.000	0.000	0.000	0.000
0.28	0.081	0.000	0.000	0.000	0.000	0.000
0.29	0.159	0.006	0.000	0.000	0.000	0.000
0.30	0.254	0.023	0.000	0.000	0.000	0.000
0.31	0.384	0.041	0.002	0.000	0.000	0.000
0.32	0.569	0.098	0.002	0.000	0.000	0.000
0.33	0.696	0.226	0.020	0.000	0.000	0.000
0.34	0.838	0.356	0.053	0.001	0.000	0.000
0.35	0.915	0.542	0.114	0.002	0.001	0.000
0.36	0.955	0.743	0.225	0.019	0.022	0.000
0.37	0.983	0.865	0.450	0.080	0.062	0.001
0.38	0.990	0.932	0.652	0.210	0.208	0.023
0.39	0.997	0.980	0.850	0.445	0.399	0.052
0.40	1.000	0.988	0.935	0.663	0.651	0.267

in a parity-check, and that w denotes the weight of the LFSR characteristic polynomial, the implementation complexity of the proposed algorithm is proportional to $2^B[(L-B)|\Omega|\omega + (N-L)w]$ mod2 additions.

Corollary 3.2. *Assuming that IDA is employed, that $|\Omega^*|$ denotes the average cardinality of the parity-check sets $|\Omega_n|$, that ω^* denotes the average number of bits in a parity-check, that I denotes the number of iterations, and that w denotes the weight of the LFSR characteristic polynomial, the implementation complexity of the proposed algorithm is proportional to $2^B[I(N^* - L)|\Omega^*|(|\Omega^*| - 1)\omega^* + (N-L)w]$ mod2 additions.*

Note also that from the structure of the proposed algorithms, it is readily seen that the proposed algorithms are suitable for fast software implementation, as well as for simple hardware implementation: the algorithms employ only simple arithmetic operations (*mod*2 addition) and simple logical operations.

Also, since the decoding process is mainly memoryless, note that a reduction of the time complexity specified by the previous corollaries can be obtained by an appropriate time-memory complexity trade-off.

Finally note that in the presented experiments, the decoding step has employed the underlying codeword lengths $N = 40000$ and $N^* = 4096$ for OSDA and IDA, respectively. This is an illustration that OSDA and IDA yield a trade-off between the length of the required sample (i.e. the code rate) and the decoding complexity.

3.6. Comparison of the Discussed Algorithm with Previously Reported Fast Correlation Attacks. This section presents a comparative analysis of the underlying principles, performance and complexity of recently proposed improved fast correlation attacks [88] and the novel algorithm, assuming the same input.

3.6.1. Comparison of the Underlying Principles. Comparison of the underlying principles employed in [87]–[88] and in the novel algorithm for the fast correlation attack can be summarized as follows.

- The approaches of [87]-[88] are based on decoding of convolutional codes and turbo codes with convolutional codes as the component codes constructed over the LFSR sequence. The novel approach is based on decoding punctured simplex block codes corresponding to the LFSR sequence.
- The algorithms [87]-[88] and the novel algorithm employ different parity-checks.

  The parity-checks employed in [88]-[87] are constructed by searching for these parity checks which include the following bits: currently considered bit, bits from a subset of B previous bits, and no more than two other bits.

  The parity-checks employed in the novel algorithm are constructed by searching for these parity checks which include the following bits:

  (i) currently considered information bit, bits from a subset of B first information bits, and two other information bits with the corresponding parity-bit, or two arbitrary parity bits only, or

  (ii) currently considered parity bit, bits from a subset of B first information bits, and no more than two other parity bits.

  Note that these different approaches in the parity-check constructions imply different number of parity-checks per bit, as well.
- The decoding techniques employed in [87]-[88] are Viterbi decoding, BCJR decodings, and MAP turbo decoding (see [67]). On the other hand the novel algorithm employs the following two low-complexity decoding techniques: (i) APP threshold decoding, and (ii) BP based BF iterative decoding.
- The fast correlation attacks from [87]-[88] implicitly include an exhaustive search over a set of dimension 2^B through employment of the Viterbi or BCJR decodings due to the trellis search. The novel algorithm employs an explicit search over all 2^B possible patterns corresponding to the first B information bits.
- A decoding process based on the Viterbi or BCJR algorithm requires a memory of dimension proportional to 2^B. On the other hand, OSDA does not require memory, and IDA requires a memory proportional to the parameter N^*.

3.6.2. Comparison of the Performance and Complexity. For the performance comparison of the novel and turbo based fast correlation attacks [88] the same inputs are employed and relevant parameters are selected so that the novel

TABLE 2. Comparison of the algorithms performance, assuming the same inputs, and lower complexity of the novel algorithm in comparison to the turbo algorithm [9]: Limit noise for which the algorithms yield, with probability close to 1, correct reconstruction of the initial LFSR state, when the LFSR characteristic polynomial is $1 + u + u^3 + u^5 + u^9 + u^{11} + u^{12} + u^{17} + u^{19} + u^{21} + u^{25} + u^{27} + u^{29} + u^{32} + u^{33} + u^{38} + u^{40}$, and the available sample is 40000 bits.

ALGORITHM	Limit Noise
turbo algorithm [88]: $B = 15$, $M = 2$	0.27
novel algorithm with OSDA: $B = 19$	0.28
turbo algorithm [88]: $B = 15$, $M = 4$	0.29
novel algorithm with OSDA: $B = 21$	0.33
turbo algorithm [88]: $B = 15$, $M = 16$	0.30
novel algorithm with OSDA: $B = 22$	0.34
novel algorithm with IDA: $N^* = 4096$, $B = 22$	0.36

algorithm always has significantly lower overall implementation complexity than the algorithm [88].

According to [88], the time complexity of the turbo decoding is proportional to $2^B IMJm$ real multiplications where I denotes the number of the iterations, M the number of the component codes, J the number of processed bits, and m the number of employed parity-checks per bit. The time complexity of the novel algorithm is given in Corollaries 3.1 and 3.2.

Also note that the space complexity of the approach from [88] is proportional to 2^B due to employment of the BCJR algorithm. If OSDA is employed no space complexity is required, and if IDA is employed it is usually significantly smaller than 2^B due to its linear rather than exponential nature.

An illustrative performance comparison is presented in the Table 2. Note that, in each case, the complexity of the proposed algorithm could be considered as significantly lower than complexity of the turbo decoding [88] although the proposed algorithm assumes search over a much larger set of hypotheses, since: (i) [88] employs iterative processing with M component codes and (ii) the dominant arithmetic operation in the proposed algorithm is *mod*2 addition against real multiplication for the turbo based decoding of [88].

Finally, note that the actual time for performing the attack by the novel algorithm strongly depends on the implementation constraints so that a straightforward comparison is not appropriate. Also, the approaches of [87]-[88] can be modified to involve *mod*2 additions, but at the expense of performance degradation.

3.7. Concluding Notes.

The considered algorithm for the fast correlation attack is based on decoding procedures of the corresponding binary block code with novel constructions of the parity-checks, independent of the LFSR feedback polynomial weight, and the following two decoding approaches are employed: an APP based threshold decoding and a BP based BF iterative decoding. The constructions of the parity-checks are based on searching for certain parity-check equations and their linear combinations employing the finite-state machine model of an LFSR with primitive characteristic polynomial. The expected numbers of the parity-checks per parity bit have been derived, showing that a large number of appropriate parity-checks can be constructed.

The experimental consideration of the algorithm performance shows that the algorithm is a powerful one.

The overall implementation complexity has been specified. As dominant operations the algorithm employs *mod*2 additions and simple logical operations, so that it is very suitable for high-speed software implementation as well as for simple hardware implementation.

The algorithm offers good trade-offs between required sample length (i.e. rate of the underlying code), overall complexity and performance. The one-step threshold decoding approach yields high performance assuming long enough sample, and the iterative decoding approach can reach the same performance using a significantly shorter sample but at the expense of increased complexity.

The algorithm has been compared with recently reported improved fast correlation attacks based on convolutional codes and turbo decoding. The underlying principles, performance and complexity have been compared, and the essential gain obtained with the novel approach is pointed out. The developed algorithm has the following two main advantages over other previously reported ones:

(a) Assuming a lower overall complexity, and the same inputs, the algorithm yields significantly better performance.

(b) It is suitable for high-speed software implementation as well as for simple hardware implementation and highly parallel architectures.

4. Certain Approaches for Randomized Stream Ciphers

This section discusses design and security evaluation issues regarding a class of stream ciphers known as randomized stream ciphers. The security evaluation involves computational complexity as well as information theoretic ones. After certain introductory notes an approach for design of stream ciphers based on joint employment of pseudorandomness, randomness and dedicated coding, is in-details considered. As a generalization of the discussed approach, a generic framework for developing randomized stream ciphers is pointed out and elements of its security evaluation from information-theoretic and computational-complexity points of view are given. This section is mainly based on the results reported in [4], [51], [61] with origins in [7] and [9].

4.1. Introduction. Randomized symmetric key encryption as an alternative encryption paradigm has been reported in [103]. According to [103], the randomized encryption is a procedure which enciphers a message by randomly choosing a ciphertext from a set of ciphertexts corresponding to the message under the current encryption key, and the following is claimed, [103]: "At the cost of increasing the required bandwidth, randomized encryption procedures may achieve greater cryptographic security than their deterministic counterparts ...".

Stream ciphers are an important class of encryption techniques for providing data secrecy. Traditional stream ciphers do not include any randomness in generation of the outputting ciphertext: They are based on the deterministic operations which expand a short secret seed into a long pseudorandom sequence. This paper points out to a novel approach for design of stream ciphers based on a combination of the pseudo-randomness and randomness.

Usefulness of involvement pure randomness into a cryptographic primitive has been recognized in a number of reported designs and particularly in the following ones. McEliece public-key system [96], based on decoding complexity after a noisy channel, is the classical and a very illustrative example of the randomness involvement. In [103], a number of approaches for including randomness in the encryption techniques have been discussed mainly regarding block and stream ciphers.

In [70], a pseudorandom number generator based on the Learning from Parity with Noise (LPN) problem, derived from an older proposal of one-way function based on the hardness of decoding a random linear code, has been reported. (Informally note that the LPN problem can be considered as the problem of solving a system of linear equations corrupted by noise. or a problem of decoding a linear code). Recently a number of randomized symmetric key encryption techniques has been reported [81], [4], [51] [65] and [61].

In [81], a probabilistic private-key encryption scheme named LPN-C whose security can be reduced to the hardness of the LPN problem has been proposed and considered. Recently, in [65] a symmetric encryption scheme similar to the one reported in [81] is reported and its security and implementation complexity are analyzed. The symmetric encryption schemes reported in [81] and [65] appears as interesting and stimulating for further considerations (having in mind improvements as well) particularly because the security is related to the recognized hard (LPN) problem.

A different approach for achieving secrecy of communication has been reported in [108] assuming that the channel between the legitimate parties is with a lower noise in comparison with the channel via which a wire-tapper has access to the ciphertext. The method proposed in [108] does not require any secret: It is based on a specific coding scheme which provides a reliably communications within the legitimate parties and prevents, at the same time, the wire-tapper from learning the communication's contents. Wire-tap channel coding is based on assigning multiple codewords to the same information vector and from that point of view, it shares the same underlying idea employed in the homophonic coding, or homophonic substitution (see [86], for example). A basic cryptographic application of homophonic coding is to convert the plaintext into a sequence of completely random (equiprobable and independent) code letters. An approach to provide secrecy employing an error-correcting code, in a scenario similar to the wire-tap channel, has been reported in [105]. Under the assumption that an attacker is forced to wire-tap the communications via a channel with a noise, the following scheme for providing secrecy is proposed in [105]: To encrypt a bit, the sender randomly selects a bit sequence whose parity is equal to the message bit, choosing this sequence to be long enough so that, due to the noise in the wire-tap channel, the attacker is unable to determine the parity of the codeword.

Also, the following results have been reported regarding security usefulness of pure randomness in cryptographic primitives. Effects of random noise and wire-tap channel coding regarding certain quantum stream ciphers have been considered in [7]. The trapdoor cipher TCHo has been proposed in [66] where the additive noise has been employed to mask a pseudorandom sequence generated by an LFSR with feedback polynomial, which has a low-weight multiple, used as the trapdoor. An approach for design of stream ciphers employing error-correction coding and certain additive noise degradation of the keystream has been reported in [89]. A message is encoded before the encryption so that the decoding, after mod 2 addition of the noiseless keystream sequence and the ciphertext, provides its correct recovery. Resistance of this approach against a number of general techniques for cryptanalysis, has been also considered in [89].

4.2. A Stream Cipher Based on Embedding Pseudorandomness and Randomness. This section yields and analyzes an approach for design of stream ciphers based on joint computing over random and secret data. Feasibility of encryption/decryption computation when the ciphertext involve pure random data is shown. The core element of the proposed approach for stream ciphering is a pseudorandom embedding of the random bits into the ciphertext and this embedding plays role of a homophonic encoding. The initial ciphertext with the embedded random bits is further on intentionally degraded by its exposure to a moderate noise which can be modelled as the binary symmetric channel effect. A security evaluation of the proposed approach implies that its security appears as a consequence of hardness of the LPN problem, as well. The developed design has potential of providing that complexity of recovering the secret key in the known plaintext attack scenario is close to the complexity of recovering the secret key via the exhaustive search, i.e. close to the maximal possible one for the given size of the secret key. The proposed approach can be considered as a trade-off between the increased security and decreased communications efficiency which in a number of scenarios appears as a suitable one.

4.2.1. Introduction. The discussed construction originates from a consideration of the possibilities for some novel approaches for inclusion of pure randomness into a stream cipher framework. The main goal of employment the pure randomness is to provide a supporting element for achieving the maximum possible security of a stream cipher, i.e. to make it as high as it can be for the given secret key dimension. Also, the involvement of the randomness is considered in a manner that provides a low-complexity implementation as well as a low communications overhead. As the result, this paper yields the following: (i) a proposal of stream ciphers class which involve pure randomness; (ii) a discussion of the impact of randomness on the security of the proposed class of stream ciphers and for a particular family of the class the security statement based on the LPN problem hardness; (iii) a discussion on the implementation complexity and the communications overhead of the proposed class of stream ciphers.

This subsection is organized as follows. Part 4.2.2 contains certain preliminaries. Part 4.2.3 yields the underlying ideas for the design and the framework of the proposed stream ciphers. Part 4.2 4 specifies the related encryption and decryption algorithms as well as a particular instantiation of the proposed stream ciphers. A preliminary security evaluation of the proposed stream ciphers framework is given in the part 4.2.5, and a formal security evaluation of a particular instantiation is given in the part 4.2.6. Part 4.2.7 yields a consideration of the implementations complexity and the communications overhead. Finally, some concluding notes are pointed out in the part 4.2.8.

4.2.2. Preliminaries. This section introduces certain notations and, as a background, yields a brief overview of the LPN problem.

Notations. This paper employs the following particular notations.

Drawing from a distribution. Given a finite set G and a probability distribution Δ on G, $g \leftarrow \Delta$ denotes the drawing of an element of G according to Δ. $g \leftarrow G$ denotes the random drawing of an element of G according to the uniform probability distribution.

Bernoulli distributions. Ber_η denotes the Bernoulli distribution with the parameter $\eta \in [0, 1/2]$, i.e. a bit $\nu \leftarrow \mathrm{Ber}_\eta$ is such that $\Pr[\nu{=}1] = \eta$ and $\Pr[\nu{=}0] = 1 - \eta$. Vectorial distribution $\mathrm{Ber}_{n,\eta}$ is defined as follows: An n-bit vector $\mathbf{v} \leftarrow \mathrm{Ber}_{n,\eta}$ is such that each bit ν of $\mathbf{v}$ is independently drawn according to Ber_η.

Oracles. $\mathcal{U}_n$ denote the oracle returning independent uniformly random n-bit strings. LPN oracle: For a fixed k-bit string s, $\Pi_{s,\eta}$ will be the oracle returning independent $(k+1)$-bit strings according to the distribution (to which we will informally refer to as an LPN distribution):

$$\left\{\mathrm{a} \leftarrow \{0,1\}^k \,;\, \nu \leftarrow \mathrm{Ber}_\eta : (\mathrm{a}, \mathrm{a} \cdot \mathrm{s} \oplus \nu)\right\}$$

The LPN Problem. Informally, Learning from Parity with Noise (LPN) problem can be described as learning an unknown k-bit vector s given noisy versions of its scalar product $\mathbf{a} \cdot \mathrm{s}$ with randomly selected vectors $\mathbf{a}$.

In a formal manner, the LPN problem is the problem of retrieving s given access to the oracle $\Pi_{s,\eta}$. For a fixed value of k, we will say that an algorithm $\mathcal{A}(T, q, \delta)$-solves the LPN problem with noise parameter η if $\mathcal{A}$ runs in time at most T, makes at most q oracle queries, and

$$\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k : \mathcal{A}^{\Pi_{s,\eta}}(1^k) = s\right] \geqslant \delta$$

By saying that the LPN problem is hard, we mean that any efficient adversary solves it with only negligible probability. There is a significant amount of literature dealing with the hardness of the LPN problem. It is closely related to the problem of decoding a random linear code and it is NP-hard.

It is NP-hard to even find a vector $\mathbf{x}$ satisfying more than half of the equations outputted by $\Pi_{s,\eta}$. The LPN average-case hardness has also been extensively investigated and one of the currently best algorithms for this case has been reported in [9].

TABLE 3. The framework of the main operations at the sender's and receiver's sides: "Embedding" assumes interleaving of the effective and random (dummy) bits and "splitting" assumes separation of the effective and dummy bits.

Sender :	Encode → Encrypt → Embedding & Additive Nose Degradation
Receiver :	Splitting → Decrypt → Decode (Decimation)

4.2.3. A Framework for the Stream Ciphers Design. This section yields underlying ideas for design of stream ciphers which involve pure randomness and the architecture of the proposed stream ciphers.

Underlying Ideas. The novel design assumes the following: (i) a source of pure randomness is available (for example, as an efficient hardware module); and (ii) a suitable error-correcting coding (ECC) techniques is available. The availability means that the implementation complexities of the source of randomness and ECC do not imply a heavy implementation overhead in suitable implementation scenarios.

The main design goal is the following one: Any method for cryptanalysis of a novel stream cipher scheme should have complexity close to the complexity of the exhaustive search. Particular origins for achieving the design goals include the results reported in [7], [9], [6] and [5], where certain issues regarding coding and randomness, complexity of the LPN problem, and generic time-memory-data trade-off method for recovering the secret key are considered.

The novel approach for design of stream ciphers is based on the following:

–employment of the pure randomness for the intentional data degradation;

–employment of a dedicated homophonic-like coding which involves pure randomness.

Note that in the considered scenario, the homophonic coding does not have the same role as in its traditional applications where the role is to provide randomness of the plaintext. Here, a homophonic coding is employed to provide additional confusion at the attacker's side.

So, the main underlying ideas of a framework for stream ciphers which involves pure randomness and provide low-complexity implementation include the following:

- Encoding/Decoding of the plaintext;
- Encryption/Decryption of the encoded plaintext/ciphertext;
- Homophonic encoding via embedding random bits and an intentional degradation of the codewords before transmission.

Accordingly, the framework of the main operations at the sender's and receiver's sides is given in Table 3.

Regarding the underlying design ideas given in this section and some of the previously reported ones, note the following. Certain randomized stream cipher approaches based on the insertions of random bits are considered in [103] including the following relevant ones: (i) pseudo-random interspersing random bits after encryption, (ii) random interspersing random bits before encryption and pseudo-random encryption of the random control sequence. Note that these approaches are based on the random bits embedding but do not include neither employment of error-correction codes neither the additive degradation by random bits. On the other hand, independently of this paper, recently in [89], it has been proposed an approach for stream ciphers which includes error-correction coding and deliberate additive random degradation. The approach [89] is based on error-correction coding of the plaintext so that it can be correctly recovered when a randomized keystream is employed for encryption. Randomization of the keystream is performed via its degradation by randomly selected error patterns which are such that provide the decodability. Note that the approach from [89] does not include any embedding of the random bits in the employed processing. Finally, regarding a comparison with the approaches reported in [103] and [89], note that the underlying ideas of the design given in this section include joint employment of randomness via the embedding and the additive degradation, as well as employment of the dedicated error-correction coding, implying a noticeable conceptual difference between the proposed approach and the reported ones.

Components, Roles and Architecture. In comparison with a traditional stream cipher which performs "encoding & encryption", the structure of the proposing one has the following three additional components:

(1) a source of pure randomness called RAND-box;
(2) a component, which at the encryption side performs homophonic encoding of the ciphertext via embedding the random bits and at the decryption side performs "decoding" via the (corresponding) decimation which provides splitting of the embedded bits;
(3) a component at the encryption side which simulates a binary symmetric channel with controllable crossover probability.

Let's call ECC-box a box which encodes the plaintext in order to provide correction of the random errors. Note that, in the proposing stream cipher, ECC-box encodes the plaintext so that it can be recovered correctly after corruption due to the errors introduced intentionally in the ciphertext (in a general setting, certain noise in the public channel can be involved as well).

Block scheme of the considered stream cipher family is depicted in Fig. 1. The "white" boxes in Fig. 1 correspond to the boxes in a traditional stream cipher which performs "encoding+encryption" in order to perform reliable operation over a noisy communication channel, and the "gray" boxes are the additional ones.

The role of the employed homophonic encoding, implemented via the embedding of the random bits, is to provide a heavy masking of the keystream generator sequences so that they appear as very uncertain for a given ciphertext even when the plaintext is known.

Accordingly, the main features of the proposed stream ciphers framework are as follows.

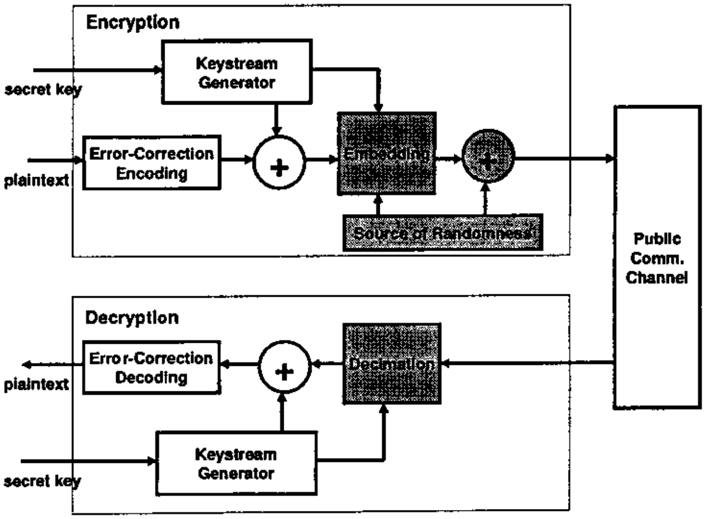


FIGURE 1. A framework of randomized stream ciphers.

- The resulting ciphertext consists of effective bits and dummy ones embedded in a manner controlled by the secret key.
- At the receiving part, the dummy bits are simply discarded and the effective bits are those which are employed for the deciphering. The decimation assumes splitting of the effective and dummy bits.
- The security is based on the impossibility of correct separation of effective bits from the dummy ones via the decimation of the available (embedded) sequence without the secret key.

4.2.4. Encryption&Decryption Algorithms and a Particular Instantiation. This section specifies the encryption and decryption algorithms in the proposed class of stream ciphers and, as an instantiation of the general framework, a particular family of the ciphers is defined.

Encryption and Decryption Algorithms

Encryption Algorithm

- *Input*: The message organized as a string of l-dimensional binary vectors $\{\mathrm{x}_t\}_t$, the secret key & non-secret initial vector which control the keystream generator, and the algorithm parameters m, n and η.
- *Encryption Steps.* For each t do the following.
  (1) Encode $\mathrm{x}_t \in \{0,1\}^l$ into the codeword $C(\mathrm{x}_t) \in \{0,1\}^m$ employing the selected ECC suitable for a binary symmetric channel with the crossover probability η.
  (2) Employing the output vector $\mathrm{y}_t \in \{0,1\}^m$ from the keystream generator compute $C(\mathrm{x}_t) \oplus \mathrm{y}_t$, where $\oplus$ denotes bit-by-bit *mod*2 addition.

(3) Generate by the RAND-box a random vector $\vec{\rho_t} \leftarrow \{0,1\}^{n-m}$ and perform psudorandom embedding (controlled by the keystream generator) of the bits from the vectors $C(\mathbf{x}_t) \oplus \mathbf{y}_t$ and $\vec{\rho}$ as follows: $(C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t})\mathbf{P}_t$, where $\mathbf{P}_t$ is an $n \times n$ permutation matrix which corresponds to the considered embedding and $||$ denotes the concatenation.

(4) Generate by the RAND-box a random $\vec{\nu_t} \leftarrow \mathrm{Ber}_{n,\eta}$ and generate the ciphertext vector as follows:

$$\mathbf{z}_t = (C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t}\ . \tag{4.1}$$

- *Output*: The ciphertext $\{\mathbf{z}_t\}_t$.

Decryption Algorithm

- *Input*: The ciphertext organized as a string of n-dimensional binary vectors $\{\mathbf{z}_t\}_t$, the secret key & non-secret initial vector which control the keystream generator, and the algorithm parameters m, n and η.
- *Decryption Steps*
  For each t do the following.

  (1) Perform decimation of $\mathbf{z}_t$ corresponding to the embedding performed in the encryption step 3 as follows:

$$\mathbf{z}_t \mathbf{P}_t^{-1} = (C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t}) \oplus (\vec{\nu_t}\mathbf{P}_t^{-1}),$$

$$tcat_m(\mathbf{z}_t \mathbf{P}_t^{-1}) = C(\mathbf{x}_t) \oplus \mathbf{y}_t \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1}),$$

  where $\mathbf{P}_t^{-1}$ denotes the inverse permutation of $\mathbf{P}_t$ (which is the transpose of $\mathbf{P}_t$), and $tcat_m(\cdot)$ denotes the truncating of the argument to the first m bits.

  (2) Employing the output vector $\mathbf{y}_t \in \{0,1\}^m$ from the keystream generator compute

$$tcat_m(\mathbf{z}_t \mathbf{P}_t^{-1}) \oplus \mathbf{y}_t = C(\mathbf{x}_t) \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1})$$

  (3) Perform decoding $C^{-1}(\cdot)$ according to the employed ECC and recover $\mathbf{x}_t$ as follows:

$$\mathbf{x}_t = C^{-1}(C(\mathbf{x}_t) \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1}))$$

- *Output*: The message in the form of the string $\{\mathbf{x}_t\}_t$.

Regarding the employed ECC we assume the following. It should be such that provides reliable decoding for the given parameter η and characteristics of the public communication channel. In the scenarios when the public communication channel is noiseless and the employed ECC is an $[m, l]$ binary block code with the decoding capability of correcting up to d errors, the lower bound on the probability of correct decoding $P(m, \eta)$ is determined by the following:

$$P(m,\eta) \geqslant \sum_{i=0}^{d} \binom{m}{i} \eta^i (1-\eta)^{m-i}.$$

Assuming that the probability of the acceptable decoding error is ϵ, the employed ECC $[m, l]$ should be such that $\epsilon \leqslant 1 - P(m, \eta)$. Finally note that in details discussion of suitable ECC selection is out of the scope of this paper.

An Equivalent Representation and a Particular Instantiation of the Proposed Stream Ciphers Family The following statement points out an equivalent analytical representation of the encryption algorithm given in the previous section which is suitable for specification of a particular and illustrative instantiation of the proposed family of stream ciphers.

Equivalent Representation

Proposition 4.1. *An equivalent analytical expression of the encryption specified by* (4.1) *is given by the following:*

$$\mathbf{z}_t = (C(\mathbf{x}_t)||\vec{\rho_t})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus \vec{\nu_t}, \tag{4.2}$$

where $\mathbf{0}^{n-m}$ *denotes the all zeros* $(n-m)$*-dimensional vector.*

Proof. We have

$$\begin{aligned}(C(\mathbf{x}_t) \oplus \mathbf{y}_t||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} &= (C(\mathbf{x}_t) \oplus \mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{0}^m||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} \\ &= (C(\mathbf{x}_t)||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{0}^m||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} \cdot \\ &= (C(\mathbf{x}_t) \oplus \mathbf{0}^m||\mathbf{0}^{n-m} \oplus \vec{\rho_t})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus \vec{\nu_t},\end{aligned}$$

which implies the proposition statement. □

Particular Instantiation. According to the encryption and decryption algorithms and Proposition 4.1, an instantiation of the proposed stream cipher framework is specified by the following definition.

Definition 4.1. Let $\mathbf{S}$ be a secret $k \times n$ binary matrix, and $\mathbf{P}_0$ be a secret $n \times n$ secret permutation matrix. Let $\mathbf{a}_t$ be a k-dimensional random vector which is publicly available, $t = 1, 2, \ldots$. Finally, let $\mathbf{P}_t = f(\mathbf{a}_t, \mathbf{P}_{t-1})$, where $f(\cdot)$ is a suitably selected function. For $t = 1, 2, \ldots$, encryption of $\mathbf{x}_t$ into $\mathbf{z}_t$ is

$$\mathbf{z}_t = (C(\mathbf{x}_t)||\vec{\rho_t})\mathbf{P}_t \oplus \mathbf{a}_t \cdot \mathbf{S} \oplus \vec{\nu_t},$$

and accordingly, decryption of $\mathbf{z}_t$ into $\mathbf{x}_t$ is as follows:

$$\mathbf{x}_t = C^{-1}(tcat_m((\mathbf{z}_t \oplus \mathbf{a}_t \cdot \mathbf{S})\mathbf{P}_t^{-1}) \, . \tag{4.3}$$

4.2.5. A Preliminary Security Evaluation of the Proposed Framework. This section yields a preliminary and informal discussion on the security of the proposed stream ciphers framework which points out to the security origins.

The role of the employed homophonic encoding, implemented via the random bits embedding, is to provide a heavy masking of the keystream generator sequences so that they appear as very uncertain for a given ciphertext even when the plaintext is known.

The proposed paradigm for providing the security is based on the following: (i) impossibility of correct decimation i.e. splitting of the effective from the dummy bits of the ciphertext without the secret key; and (ii) availability of the noisy sample only, due to the employed additive noise degradation of the ciphertext before its transmission via a public communications channel.

The main role of the additive random degradation of the ciphertext is to introduce uncertainty into a sample available for cryptanalysis preventing a possibility of mounting the generic time-memory trade-off approaches for cryptanalysis (see [84] and [69]) in order to employ a generic approach more efficient than the exhaustive search. When an error-free sample is available the time-memory (and time-memory-data) trade-off based attacks can be directly mounted in order to recover the secret key **K**. On the other hand, when the sample for cryptanalysis is not error-free, the time-memory trade-off approach, in a general case, does not work.

The above arguments are a background for the security evaluation of the proposed framework and for a conjecture that the complexity of cryptanalysis is determined by the complexity of exhaustive secret keys search.

Note that the following are basic approaches for cryptanalysis of any stream cipher: (i) the generic key recovery attacks based on different search techniques (including the trade-off ones); (ii) the dedicated key recovery attacks based on particular weaknesses of the underlying structure; (iii) a number of different not key recovery oriented attacks (distinguishing attacks, ...).

In a known plaintext attack scenario, the goal of cryptanalysis is to recover the key **K**. There are the following two basic approaches for achieving this goal:

–recovering **K** based on the given ciphertext $\{\mathbf{z}_t\}_t$,

–recovering certain pesudorandom sequences specified by **K** based on $\{\mathbf{z}_t\}_t$ and then recovering **K** based on these sequences.

For achieving any of these goals, an attacker faces the following two main problems:

- the inverse mapping without knowledge of the secret key in order to recover the considered pseudorandom sequences based on $\{\mathbf{z}_t\}_t$;
- impact of the noise sequence $\{\vec{\nu}_t\}_t$ to complexity of any generic technique for recovering the secret key beside the exhaustive search over the space of all possible keys which has complexity $O(2^K)$.

Hardness of the above problems is elaborated by the following.

Note that even in the case of a noiseless public communication channel, there are the following two problems at the attacker's side:

–removing the dummy bits from the ciphertext without knowledge of the secret key;

–uncertainty due to effect of the binary symmetric channel with crossover probability $p^* < 1/2$ which corrupts the data before theirs availability to the attacker.

The uncertainty at the attacker's side can be considered as a consequence of a noise corresponding to a channel with the bits insertion and complementing which corrupts the sample for cryptanalysis. Because, the legitimate parties share the secret key, they face a lower noise (corresponding only to the bits complementing) in comparison with the noise which an attacker faces.

Accordingly, security of the scheme appears as consequence of the employed wiretap channel like encoding which provides confusion of an attacker which faces much more heavy equivalent noise in comparison with the legitimate receiver because the attacker does not posses the employed secret key. This heavy noise implies that the attacker can not learn about the keystream generator output sequences.

Without knowledge about the employed secret, it is not possible to efficiently remove dummy bits and to learn about (noisy) sequences from the keystream generator. On the other hand, without reliable knowledge on the keystream generator output sequences, it is not possible to construct a more efficient approach for cryptanalysis than a hypothesis testing. Accordingly, the corruption of the output sequences by the noise $\{\vec{\nu}_t\}_t$ implies (as discussed above) that the time-memory trade-off hypotheses testing based attacks are not feasible because the entire system appears as a stochastic one which makes the algebraic approaches not feasible. So, the exhaustive search over the space of all possible keys appears as the only one option.

The above discussion implies that the security appears as a consequence of the uncertainty at the attacker's side which is jointly implied by: (i) pseudorandom homophonic encoding; (ii) effect of the intentional corruption of the data which are available only via a binary symmetric channel.

4.2.6. A Formal Security Evaluation of the Particular Instantiation. This section yields a formal security evaluation of the stream ciphers specified by Definition 4.1.

Security Evaluation Background. One of the security goals is the indistinguishability (IND): IND deals with the secrecy provided by the scheme in the following sense: An adversary must be unable to distinguish the encryption of two (chosen) plaintexts. This definition was introduced in the context of public-key encryption as a more practical equivalent to semantic security and recently employed for security evaluation of the schemes reported in [66] and [81]. Accordingly, and particularly following [81], this paper adopts IND as the security criterion for a formal security consideration. For the IND considerations we assume the following traditional approach. An adversary is considered as a pair of algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and they operate through two phases as follows.

- $\mathcal{A}_1$ is employed during the first phase and at the end of this phase, $\mathcal{A}_1$ outputs a pair of plaintexts $(\mathrm{x}_1, \mathrm{x}_2)$.
- One of the given plaintexts is selected with probability equal 1/2, then encrypted, and the obtained ciphertext is delivered to $\mathcal{A}_2$-this represents $\mathcal{A}$'s challenge. The success of $\mathcal{A}$ is determined according to correctness of decision whether x_1 or x_2 was encrypted.

The adversary $\mathcal{A}$ is classified according to the oracles (encryption and/or decryption) available in each of the phases. $\mathcal{A}$ is labelled as PX-CY, where P stands for the encryption oracle and C for the decryption oracle, and where $X, Y \in 0, 1, 2$ indicates when $\mathcal{A}$ is allowed to access the oracle:

–0: $\mathcal{A}$ never accesses the oracle;

–1: $\mathcal{A}$ can only access the oracle during phase 1, i.e. before seeing the challenge (also termed non-adaptive);

–2: $\mathcal{A}$ can access the oracle during phases 1 and 2 (also termed adaptive).

The following lemma states that the hardness of the LPN problem implies that the two oracles $\mathcal{U}_{k+1}$ and $\Pi_{s,\eta}$ are indistinguishable.

Lemma 4.1. [90, Lemma 1]. *Assume there exists an algorithm $\mathcal{M}$ making q oracle queries, running in time T, and such that*

$$\left|\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k : \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \geqslant \delta.$$

Then there is an algorithm $\mathcal{A}$ making $q' = O(q \cdot \delta^{-2} \log k)$ oracle queries, running in time $T' = O(T \cdot k\delta^{-2} \log k)$, and such that

$$\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k : \mathcal{A}^{\Pi_{s,\eta}}(1^k) = \mathrm{s}\right] \geqslant \frac{\delta}{4}.$$

An Analysis of the Security. This section yields a method for reducing security evaluation of the stream ciphers specified by Definition 4.1 to the problem of distinguishing $\mathcal{U}_{k+1}$ and $\Pi_{s,\eta}$ and according to Lemma 4.1 further on to the LPN problem.

Theorem 4.1. *Assume there is an adversary $\mathcal{A}$, running in time T, and attacking the stream cipher specified by Definition* 1 *with parameters (l, m, k, n, η) in the sense of IND-P1-C0 with advantage δ by making at most q queries to the encryption oracle. Then there is an algorithm $\mathcal{M}$ making $O(q)$ oracle queries, running in time $O(T)$, and such that*

$$\left|\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k : \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \geqslant \frac{\delta}{n}.$$

Proof. The proof is an adaptation of the proof technique reported in [81, Theorem 1]. Particularly note that non-adaptive CPA-security (P1) implies adaptive CPA-security (P2), and so it is possible to restrict the consideration to adversaries accessing the encryption oracle only during the first phase of the attack i.e. before seeing the challenge ciphertext (see [81], for example). □

In the following, the same notation as in the previous sections is used, but the index t has been omitted for the simplicity.

The proof proceeds by a hybrid argument based on the following hybrid distributions on $\{0,1\}^{k+n}$. For $j \in [0, \ldots, n]$, let $\mathbf{S}'$ denotes a $k \times (n-j)$ binary matrix. We define the probability distribution $\mathcal{P}_{j,S',\eta}$ as

$$\left\{\mathbf{a} \leftarrow \{0,1\}^k\,;\ \mathbf{r} \leftarrow \{0,1\}^j\,;\ \vec{\nu} \leftarrow \mathrm{Ber}_{(n-j),\eta} : \mathbf{a}||\mathbf{r}||(\mathbf{a} \cdot \mathbf{S}' \oplus \vec{\nu})\right\}.$$

Accordingly, we obtain the vector $\mathbf{a}||\mathbf{b}$ such that the first j bits of $\mathbf{b} = \mathbf{r}||(\mathbf{a}{\cdot}\mathbf{S}' \oplus \vec{\nu})$ are uniformly random, whereas the last $(n-j)$ bits are distributed according to the $(n-j)$ independent LPN distributions related to the respective columns of $\mathbf{S}'$.

Note that $\mathcal{P}_{n,S',\eta}$ corresponds to $\mathcal{U}_{k+n}$.

The next step is specification of the following hybrid encryption oracles $\mathcal{E}_{j,S',\eta}$ associated with the secret matrix $\mathbf{S}'$ and noise parameter η:

- On input the l-bit plaintext x, the encryption oracle performs a homophonic encoding and maps it to $[C(\mathrm{x})||\vec{\rho}]\mathbf{P}$, draws a random $(k+n)$-bit vector $\mathbf{a}||\mathbf{b}$ distributed according to $\mathcal{P}_{j,S',\eta}$, and returns $(\mathbf{a}, [C(\mathrm{x})||\vec{\rho}]\mathbf{P} \oplus \mathbf{b})$.

Recall that $\mathcal{M}$ has access to an oracle and wants to distinguish whether this is $\mathcal{U}_{k+1}$ or $\Pi_{s,\eta}$. In order to achieve its goal, the distinguisher $\mathcal{M}$ acts performing the following steps.

(1) On input the security parameter 1^k, $\mathcal{M}$ draws a random $j \in [1, \dots, n]$. If $j < n$, it also draws a random $k \times (n - j)$ binary matrix $\mathbf{S}'$. It then launches the first phase $\mathcal{A}_1$ of the adversary $\mathcal{A}$.

(2) Each time $\mathcal{A}_1$ asks for the encryption of some $\mathbf{x}$, $\mathcal{M}$ obtains a sample $(\mathbf{a}, z)$ from its oracle, and performs the following:
- draws a random $(j-1)$-bit vector $\mathbf{r} \leftarrow \{0,1\}^{j-1}$;
- draws a $(m-j)$-bit noise vector $\vec{\nu}$ distributed according $\mathrm{Ber}_{n-j,\eta}$;
- forms the masking vector $\mathbf{b} = \mathbf{r}||z||(\mathbf{a} \cdot \mathbf{S}' \oplus \vec{\nu}$, and returns $(\mathbf{a}, [C(\mathbf{x})||\tilde{\rho}]\mathbf{P} \oplus \mathbf{b})$.

(3) –The adversary $\mathcal{A}_1$ returns two plaintexts $\mathbf{x}_1$ and $\mathbf{x}_2$.
–The distinguisher $\mathcal{M}$ selects a uniformly random $\alpha \in 1, 2$ and returns to $\mathcal{A}_2$ the ciphertext corresponding to $\mathbf{x}_\alpha$ encrypted exactly as described just before.
–If the answer of $\mathcal{A}_2$ is correct, then $\mathcal{M}$ returns 1, otherwise it returns 0.

It is straightforward to verify the following

- when $\mathcal{M}$'s oracle is $\mathcal{U}_{k+1}$, $\mathcal{M}$ simulates the encryption oracle $\mathcal{E}'_{j,S',\eta}$, and
- when $\mathcal{M}$'s oracle is $\Pi_{s,\eta}$, then $\mathcal{M}$ simulates the encryption oracle $\mathcal{E}'_{j-1,S'',\eta}$ where $\mathbf{S}'' = \mathrm{s}||\mathbf{S}'$ is the matrix obtained as the concatenation of $\mathbf{s}$ and $\mathbf{S}'$.

So, the advantage of the distinguisher can be expressed as follows:

$$\begin{aligned}
\mathrm{Adv} &= \left|\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k : \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \\
&= \frac{1}{n}\left|\sum_{j=0}^{n-1} \Pr\left[\mathcal{A}^{\mathcal{E}'_{j,S',\eta}} \text{ succeeds}\right] - \sum_{j=1}^{n} \Pr\left[\mathcal{A}^{\mathcal{E}'_{j,S',\eta}} \text{ succeeds}\right]\right| \\
&= \frac{1}{n}\left|\Pr\left[\mathcal{A}^{\mathcal{E}'_{0,S',\eta}} \text{ succeeds}\right] - \Pr\left[\mathcal{A}^{\mathcal{E}'_{n,S',\eta}} \text{ succeeds}\right]\right|
\end{aligned}$$

Note that the encryption oracle $\mathcal{E}'_{0,S',\eta}$ is exactly the real encryption oracle.

On the other hand the encryption oracle $\mathcal{E}'_{n,M',\eta}$ encrypts all plaintexts by blinding them with uniformly random vectors $\mathbf{b}$ so that in this case the adversary $\mathcal{A}$ cannot do better (or worse) than guessing at random and has the success probability of 1/2. Accordingly, $\left|\Pr\left[\mathcal{A}^{\mathcal{E}'_{0,S',\eta}} \text{ succeeds}\right] - \Pr\left[\mathcal{A}^{\mathcal{E}'_{n,S',\eta}} \text{ succeeds}\right]\right|$ is exactly the advantage of the adversary which is greater than δ by the hypothesis, implying the theorem statement.

4.2.7. Implementation Complexity and Communications Overhead. This section yields a brief discussion on the implementation complexity and the communication overhead of the proposed framework for stream ciphers and it points out the main issues only. An in details consideration of the implementation complexity and the communications overhead requires focusing on particular instantiations of the proposed class of stream ciphers and it is out of the scope of this paper.

Complexity. The implementation complexity, in comparison with a traditional stream ciphering scheme which includes the error-correction coding is mainly due to requirement for the source of randomness (RAND-box) because the embedding and decimation operations could be considered as low complexity ones.

The implementation complexity of the additional components depends on the overall implementation scenario and particularly whether it is software only or a hybrid one. Assuming availability of a suitable RAND-box the dominant implementation complexity overhead appears as a very low one.

Overhead. In order to achieve the main security goal, the proposed stream ciphering approach includes the following processing with impacts on the communications overhead: (i) error-correction encoding of the messages; (ii) a homophonic encoding via random bits embedding which performs expansion of the "initial ciphertext". Both of these issues imply the communications overhead: Assuming that the error-correction encoding and the embedding introduce the expansion for the factors α_1 and α_2, respectively, the related communications overhead is determined by the factor $\alpha_1 \cdot \alpha_2$.

Accordingly, the proposed stream ciphers framework includes certain trade-off between the security and the communications overhead which in a number of scenarios can be considered as very appropriate.

4.2.8. Concluding Notes. This section proposes an alternative approach for design of stream ciphers which involve pure randomness and provide low-complexity of the implementation. The proposed framework employs a dedicated homophonic coding and a deliberate noise which, assuming the appropriate code and noise level provides at the attacker's side increased confusion close to the limit determined by the secret key length. The employed homophonic encoding/decoding is based on pseudorandom embedding/decimation of random bits, and it is specific in the following sense: (i) its only purpose is to introduce additional uncertainty at the attackers side, and (ii) decoding complexities with and without the secret key are extremely different. Generic encryption/decryption algorithms are proposed, an equivalent interpretation, and a particular family of stream ciphers.

Security evaluation implies that the proposed stream ciphering provides high security which can be very close to the maximum one indicated by the employed secret key length. Consequently, under certain conditions, a straightforward exhaustive search over all possible secret keys appears as very close to the most efficient method of cryptanalysis. For a particular family of the proposed stream ciphers it is formally shown that the security appears as a consequence of hardness of the LPN problem.

In order to achieve the main security goal, the proposed stream ciphering approach includes the following two encoding schemes with impacts on the communications overhead: (i) error-correction encoding of the messages; (ii) dedicated homophonic encoding via the random bits embedding which performs expansion of the initial ciphertext. Both of these issues imply the communications overhead: Accordingly, the proposed stream ciphers framework includes certain trade-off between the security and the communications overhead which, in a number of scenarios, can be considered as appropriate.

4.3. A Generic Framework of Randomized Stream Ciphers and Its Security Evaluation. Following the encryption approaches recently reported in [4] and [61], this section considers and analyzes from security point of view a generic model of randomized stream ciphers.

The section yields an analysis of security of a model of randomized stream cipher based on joint employment of pseudorandomness, randomness and dedicated coding. The considered scheme sequentially encrypts l-bit plaintext vectors into n-bit ciphertext vectors employing a keystream generator seeded by k-bit secret key, $m-l$, $l < m < n$, balanced random bits where ones and zeros appear with the same probability equal to $1/2$, n biased random bits where ones appear with the probability $p < 1/2$, and two linear encoding schemes for dedicated homophonic and error correction encoding. The security analysis has been performed assuming the chosen plaintext attack. The information-theoretic security evaluation was focussed towards the posterior uncertainty on the secret key. The equivocation of the secret key has been derived and analyzed. The equivocation expression shows that it can be kept to a nonzero value assuming appropriate selection of the encryption parameters $m-l$, n and p, when the sample available for cryptanalysis is limited. The previous imply that the scheme has potential of providing residual uncertainty on the secret key under certain conditions. Also, the considered encryption scheme is analyzed from computational complexity security point of view. The performed evaluation of the secret key recovery implies that it is as hard as decoding of a random linear block code after a binary symmetric channel with the additive noise (cross-over probability) parameter ϵ equal to $\frac{1}{2}(1-(1-2p)^{(m-l)/2})$. The analysis performed imply that the considered encryption paradigm provides a framework for design of provably secure stream ciphers which can provide low implementation complexity as well (noting that the implementation issues are out of the scope of this chapter).

4.3.1. Framework of Certain Randomized Stream Ciphers. We consider the randomized stream ciphers framework displayed in the following figure.

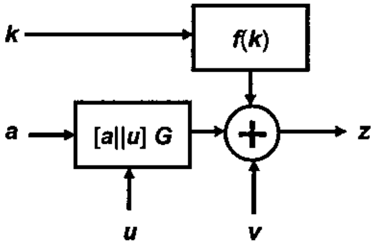


FIGURE 2. A generic randomized stream cipher encryption.

The analytical description of the considered encryption is as follows. Let:
–a$^{(l)}$ is a known l-dimensional binary vector;
–**G** is a known $(m \times n)$-dimensional binary matrix such that $\mathbf{G} = \mathbf{G}_H \mathbf{G}_{ECC}$ where $\mathbf{G}_H$ is the matrix of a linear homophonic encoding, and $\mathbf{G}_{ECC}$ is the generator matrix of a linear error-correcting code (ECC) designed to correct errors over a

binary symmetric channel (b.s.c.) with the crossover probability p;
–$\mathbf{u}^{(m-l)}$ is a realization of $(m-l)$-dimensional binary random variable $\mathbf{U}^{(m-l)}$ such that $\Pr(\mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) = \frac{1}{2^{m-l}}$;
–$\mathbf{v}^{(n)}$ is a realization of n-dimensional binary random variable $\mathbf{V}^{(n)}$ such that $\Pr(\mathbf{V}^{(n)} = \mathbf{v}^{(n)}) = p^{w}(1-p)^{n-w}$, $p < 0.5$, and $w = Hwt(\mathbf{v}^{(n)})$, and $Hwt(\cdot)$ denotes the Hamming weight.

Accordingly, we have the following algebraic representation of the ciphertext:

$$\mathbf{z}^{(n)} = [\mathbf{a}^{(l)} || \mathbf{u}^{(m-l)}]\mathbf{G} \oplus f^{(n)}(\mathbf{k}) \oplus \mathbf{v}^{(n)}.$$

The corresponding decryption process is as follows:

$$\mathbf{a}^{(l)} = tcat\{[ECC^{-1}(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}))]\mathbf{G}_H^{-1}\},$$

where $tcat\{\cdot\}$ is the operator of truncation to the first l bits, $ECC^{-1}(\cdot)$ denotes the decoding operator of the ECC with the generator matrix $\mathbf{G}_{ECC}$, and $\mathbf{G}_H^{-1}$ is the inverse matrix of $\mathbf{G}_H$.

4.3.2. Encryption of a Sequence of Vectors. We consider encryption of a sequence of vectors at the time instances $t = 1, 2, \ldots, \tau$, employing the following notation:
–$\mathbf{a}_t^{(l)}$ is a known l-dimensional binary vector at the time instance t;
–$f_t^{(t)}(\mathbf{k})$ is the keystream generator output segment of length n generated at the time instance t;
–$\mathbf{u}_t^{(m-l)}$ is a realization of $(m-l)$-dimensional binary random variable $\mathbf{U}_t^{(m-l)}$ such that $\Pr(\mathbf{U}_t^{(m-l)} = \mathbf{u}_t^{(m-l)}) = 2^{-m+l}$, at the time instance t;
–$\mathbf{v}_t^{(n)}$ is a realization of n-dimensional binary random variable $\mathbf{V}_t^{(n)}$ at the time instance t such that $\Pr(\mathbf{V}_t^{(n)} = \mathbf{v}_t^{(n)}) = p^{w_t}(1-p)^{n-w_t}$, $p < 0.5$, and $w_t = Hwt(\mathbf{v}_t^{(n)})$ denotes the Hamming weight of the vector $\mathbf{v}_t^{(n)}$.

Accordingly, the ciphertext vectors $\mathbf{z}_t^{(n)}$, $t = 1, 2, \ldots, \tau$, are specified by the following:

$$\mathbf{z}_t^{(n)} = [\mathbf{a}_t^{(l)} || \mathbf{u}_t^{(m-l)}]\mathbf{G} \oplus f_t^{(n)}(\mathbf{k}) \oplus \mathbf{v}_t^{(n)}, \quad t = 1, 2, \ldots, \tau.$$

4.3.3. Information-Theoretic Security Evaluation of a Single Encryption: On the Equivocation. This section consider the uncertainty on the secret key when a corresponding keystream generator output segment is known. As the first, the posterior probability that certain key $\mathbf{k}$ has been involved into generation of a keystream segment $\mathbf{z}^{(n)}$ is given and finally the equivocation[1] (which specifies the posterior uncertainty) is derived.

Lemma 4.2. *We have*

$$\Pr(\mathbf{K} = \mathbf{k} | \mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \frac{\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^{w}(1-p)^{n-w}}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^{w}(1-p)^{n-w}},$$

[1] see [104] or a textbook on information theory

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ is the number of different vectors $\mathbf{u}^{(m-l)}$ which, for given $\mathbf{k}$ and $\mathbf{z}^{(n)}$ imply the same $w = Hwt(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})$, and $\sum_{\mathbf{k}}(\cdot)$ denotes summation over all possible keys, assuming that $\mathbf{a}^{(l)}$ is known.

Proof. We have

$$\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \frac{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) \Pr(\mathbf{K} = \mathbf{k})}{\sum_{\mathbf{k}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) \Pr(\mathbf{K} = \mathbf{k})},$$

and when all the keys are equiprobable

$$\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \frac{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k})}{\sum_{\mathbf{k}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k})},$$

On the other hand we have the following

$$\begin{aligned}\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) &= \frac{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}, \mathbf{K} = \mathbf{k})}{\Pr(\mathbf{K} = \mathbf{k})} \\ &= \frac{1}{\Pr(\mathbf{K} = \mathbf{k})} \sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}, \mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \\ &= \frac{1}{\Pr(\mathbf{K} = \mathbf{k})} \sum_{\mathbf{u}^{(m-l)}} (\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \\ &\qquad \cdot \Pr(\mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}|\mathbf{K} = \mathbf{k}) \Pr(\mathbf{K} = \mathbf{k})) \\ &= \sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \Pr(\mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}).\end{aligned}$$

Further on:

$$\begin{aligned}&\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \\ &\qquad = \Pr(\mathbf{V}^{(n)} = \mathbf{v}^{(n)} = \mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})\end{aligned}$$

and accordingly

$$\begin{aligned}&(4.4) \quad \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) \\ &= \sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \Pr(\mathbf{V}^{(n)} = \mathbf{v}^{(n)} = \mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}) \\ &= \frac{1}{2^{m-l}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^{w} (1-p)^{n-w}, \quad w = Hwt(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}),\end{aligned}$$

and $\alpha_{\mathbf{k},\mathbf{z}}(w)$ is the number of different vectors $\mathbf{u}^{(m-l)}$ which, for given $\mathbf{k}$ and $\mathbf{z}$, yield the same w. The above imply the lemma statement. □

Corollary 4.1. *According to (4.4), when $\mathbf{a}^{(l)}$ is known, we have the following:*

- *when $p = 0$,*

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k})$$

$$= \begin{cases} \frac{1}{2^{m-l}}\alpha_{\mathbf{k},\mathbf{z}}(0) = 2^{-(m-l)} & \textit{if } \mathbf{z} = [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} \oplus f^{(n)}(\mathbf{k}) \\ 0 & \textit{otherwise} \end{cases}$$

- *when* $p = 1/2$,

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) = \frac{1}{2^{m-l}}\frac{1}{2^n}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w) = \frac{1}{2^{m-l}}\frac{1}{2^n}2^{m-l} = 2^{-n}.$$

Lemma 4.3. *The coefficients* $\{\alpha(w)\}_{w=0}^{n}$ *corresponds to the weight distribution of a modified linear block code with the generator matrix* $\mathbf{G}$.

Sketch of the Proof. Let a binary code C be specified by the generator matrix $\mathbf{G}$ of dimension $m \times n$. Each vector $[\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}$ is a codeword of C. When $\mathbf{a}^{(l)}$ is a vector constant, all the codewords are specified by the set of all possible 2^{m-l} vectors $\mathbf{u}^{(m-l)}$. Let a modified code C' be obtained from C via *mod*2 addition of the codewords of C with a given n-dimensional constant vector $c^{(n)}$. Accordingly, $\{\alpha(w)\}_{w=1}^{n}$ specifies the weight distribution of C' if each $\alpha(w)$ is equal to number of the codeword with Hamming weight equal to w, namely $Hwt(\mathrm{c}^{(n)} \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}) = w$. □

Theorem 4.2. *The equivocation of secret key in the known plaintext attack scenario (when* $\mathbf{a}^{(l)}$ *is known) is given by the following:*

$$(4.5)\quad H(\mathbf{K}|\mathbf{Z}^{(n)})$$

$$= 2^{-(|\mathbf{k}|+m-l)}\sum_{\mathbf{z}^{(n)}}\Big(\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}\Big)\cdot\log_2\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}$$

$$-2^{-(|\mathbf{k}|+m-l)}\sum_{\mathbf{z}^{(n)}}\sum_{\mathbf{k}}\Big(\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}\Big)\cdot\log_2(\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}),$$

where $|\mathbf{k}|$ *is length of the secret key* $\mathbf{k}$, $\alpha_{\mathbf{k},\mathbf{z}}(w) \geqslant 0$ *is the number of different vectors* $\mathbf{u}^{(m-l)}$ *which, for given* $\mathbf{k}$ *and* $\mathbf{z}^{(n)}$ *imply the same* $w = Hwt(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})$, *and* $\sum_{\mathbf{k}}(\cdot)$ *denotes summation over all possible keys.*

Proof. We have

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = E_{\mathbf{Z}^{(n)}}\{H(\mathbf{K}|\mathbf{z}^{(n)})\}$$

$$= \sum_{\mathbf{z}^{(n)}}\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\sum_{\mathbf{k}}\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\log_2\frac{1}{\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})}$$

and employment of Lemma 4.2 yields

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = \sum_{\mathbf{z}^{(n)}}\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})$$

$$\cdot\sum_{\mathbf{k}}\frac{\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}{\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}\log_2\frac{\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}{\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}\quad .$$

Via manipulations over the above expression we obtain the following:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) =$$

$$\sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) \frac{\log_2 \sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}} \sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}$$

$$+ \sum_{\mathbf{z}^{(n)}} \frac{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}}$$

$$\cdot \sum_{\mathbf{k}} \left(\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \right) \log_2 \frac{1}{\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}} .$$

On the other hand, when all the keys are equally-probable

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \sum_{\mathbf{k}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)} | \mathbf{K} = \mathbf{k}) \cdot \Pr(\mathbf{K} = \mathbf{k}),$$

and employing (4.4) we obtain

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{k}} (\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w})$$

where $|\mathbf{k}|$ denotes the secret key length. Combining the above directly implies the theorem statement. □

Corollary 4.2. *When $p = 0$,*

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = \begin{cases} |\mathbf{k}| + m - l - n, & \text{if } n < |\mathbf{k}| + m - l \\ 0, & \text{otherwise} \end{cases}$$

noting that $n > m - l$.

When $p = 1/2$, $H(\mathbf{K}|\mathbf{Z}^{(n)}) = |\mathbf{k}|$.

Proof. When $p = 0$, note that

$$\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} = \sum_{\mathbf{k}} \alpha_{\mathbf{k},\mathbf{z}}(0)$$

because only $p^0 = 1$ yields non-zero terms. On the other hand each $\alpha_{\mathbf{k},\mathbf{z}}(0) \in \{0, 1\}$, and

$$\sum_{\mathbf{k}} \alpha_{\mathbf{k},\mathbf{z}}(0) = \begin{cases} 2^{|\mathbf{k}|+m-l-n} & \text{if } n < |\mathbf{k}| + m - l \\ 1 & \text{if } n \geqslant |\mathbf{k}| + m - l \end{cases}$$

because a system of n equations with $|\mathbf{k}| + m - l > n$ unknowns has $2^{|\mathbf{k}|+m-l-n}$ equally likely solutions, noting as well that $m - l < n$. The above and Theorem 4.2 yields the corollary statement for $p = 0$.

When $p = 1/2$, note that $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{m-l}$ and so $\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{|\mathbf{k}|+m-l}$. Accordingly, statement of Theorem 4.2 implies the corollary for $p = 1/2$. □

Additionally, we point out to the following. Let $\pi = \pi(\mathbf{K}|\mathbf{Z}^{(n)})$ denotes the minimal probability of error which corresponds to employment of the maximum a-posteriori probability (MAP) decision rule for recovering the secret key. In the considered setting, $\pi(\mathbf{K}|\mathbf{Z}^{(n)})$ is specified by the next statement.

Corollary 4.3. *When $n > |\mathbf{k}| + m - l$,*

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \left\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \right\},$$

and

$$\pi = \begin{cases} 0 & \textit{if } p = 0 \\ 1 - 2^{|\mathbf{k}|} & \textit{if } p = 1/2 \end{cases}$$

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ denotes the number of vectors $\mathbf{u}^{(m-l)}$ which for given $\mathbf{a}^{(l)}$, $\mathbf{z}^{(n)}$ and $\mathbf{k}$ yield $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} = w$.

Proof. Taking into account the definition of π and the implication of the exhaustive search based minimum distance decoding paradigm, we have the following:

$$\pi = \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) \left[1 - \max_{\mathbf{k}} \Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\right]$$

$$= \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) \Bigg[1 -$$

$$\max_{\mathbf{k}} \frac{\sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \cdot \Pr(\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)})}{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})} \Bigg],$$

$$= \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) - \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \Bigg\{ \sum_{\mathbf{u}^{(m-l)}}$$

$$\Pr(\mathbf{Z}^{(n)} \oplus f^{(n)}(\mathbf{K}) \oplus [\mathbf{A}^{(l)}||\mathbf{U}^{(m-l)}]\mathbf{G} = \mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} \mid$$

$$\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \cdot \Pr(\mathbf{K} = \mathbf{k}) \cdot \Pr(\mathbf{U}^{(u-l)} = \mathbf{u}^{(m-l)}) \Bigg\},$$

where $\mathbf{a}^{(l)}$ is given.

The above implies

$$\pi = 1 - \sum_{\mathbf{z}^{(n)}} 2^{-(|\mathbf{k}|+m-l)} \max_{\mathbf{k}} \left\{ \sum_{\mathbf{u}^{(m-l)}} p^w (1-p)^{n-w} \right\},$$

where w is equal to the Hamming weight of the vector $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}$, and accordingly

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \left\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \right\},$$

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ denotes the number of vectors $\mathbf{u}^{(m-l)}$ which for given $\mathbf{a}^{(l)}$, $\mathbf{z}^{(n)}$ and $\mathbf{k}$ yield $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} = w$, noting that $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{m-l}$.

When $p = 0$, only the sum $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w)p^{w}(1-p)^{n-w}$ reduces to $\alpha_{\mathbf{k},\mathbf{z}}(0)$ taking into account that $p^0 = 1$ and $p^w = 0$, $w = 1, 2, \ldots, n$. On the other hand, for $n > |\mathbf{k}| + m - l$, we have $\alpha_{\mathbf{k},\mathbf{z}}(0) = 1$ because just one out of 2^{m-l} vectors provides $w = 0$. Accordingly, for $p = 0$, $\pi = 1 - \sum_{\mathbf{z}} 2^{|\mathbf{k}|+m-l} = 0$ because when $p = 0$ $\mathbf{z}^{(n)}$ runs over exactly $2^{|\mathbf{k}|+m-l}$ different patterns. When $p = 1/2$, for any $w = 0, 1, \ldots n$, $p^w(1-p)^{n-w} = 2^{-n}$ implying $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w)p^{w}(1-p)^{n-w} = 2^{m-l-n}$, and accordingly $\max_{\mathbf{k}}\{\cdot\} = 2^{m-l-n}$. Taking into account that when $p = 1/2$ the vector $\mathbf{z}^{(n)}$ can take 2^n different values we obtain $\pi = 1 - \sum_{\mathbf{z}^{(n)}} 2^{-(|\mathbf{k}|+m-l)}2^{m-l-n} = 1 - 2^{-|\mathbf{k}|}$.

Note that the distribution of $\alpha_{\mathbf{k},\mathbf{z}}(w)$ is implied by the following: Assuming $n > |\mathbf{k}| + m - l$, note the following:

- When $\mathbf{z}^{(n)}$ is generated employing the key $\mathbf{k}$ and the vector $\mathbf{u}^{(m-l)}$ we have:
$$\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} = \mathbf{v}^{(n)},$$
- When $\mathbf{z}^{(n)}$ is not generated employing the key $\mathbf{k}$ and the vector $\mathbf{u}^{(m-l)}$, the binary vector variable $\mathbf{Z}^{(n)} \oplus f^{(n)}(\mathbf{K}) \oplus [\mathbf{A}^{(l)}||\mathbf{U}^{(m-l)}]\mathbf{G}$ appears as a random one where each component of the vector takes values one and zero with the probability equal to 1/2. □

As a finalization of the above discussion we point out to the following statement.

Proposition 4.2. *Let i be an integer, $2 \leqslant i < 2^{|k|}$, such that $\frac{i-1}{i} \leqslant \pi \leqslant \frac{i}{i+1}$. Then the tight bounds on the equivocation are as follows:*

$$\log_2 i + i(i+1)\Big(\log_2 \frac{i+1}{i}\Big)\Big(\pi - \frac{i-1}{i}\Big) \leqslant H(\mathbf{K}|\mathbf{Z}^{(n)})$$
$$\leqslant \pi \log_2 \frac{1}{\pi} + (1-\pi)\log_2 \frac{1}{1-\pi} + \pi \log_2(2^{|k|} - 1)$$

where

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \left\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w)p^{w}(1-p)^{n-w} \right\}.$$

Proof. The Fano inequality [78] yields the following upper bound:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) \leqslant h(\pi) + \pi \log_2(2^{|\mathbf{k}|} - 1),$$
$$h(\pi) = \pi \log_2 \pi - (1-\pi)\log_2(1-\pi).$$

The following lower bound is implication of the general lower bound reported in [91]:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) \geqslant \log_2 i + i(i+1)(\log_2 \frac{i+1}{i})(\pi - \frac{i-1}{i})$$

where $2 \leqslant i < 2^{|k|}$, and $\frac{i-1}{i} \leqslant \pi \leqslant \frac{i}{i+1}$. Taking into account Corollary 4.3 which specifies π, we have the theorem statement. □

4.3.4. Computational Complexity Security Evaluation. This section analyzes the security of the proposed scheme from a computational complexity point of view in the chosen plaintext attacking (CPA) scenario. In this case, the security evaluation consists of establishing how hard it is to find the secret key based on the algebraic representation of the encryption. We will show in our complexity analysis that the hardness of recovering the key relies on the hardness of the LPN problem (see [71],[9], [92], for example). The analysis will pinpoint the features that the homophonic encoder should have so as to create an increased complexity of the underlying LPN problem in the average case.

Preliminaries. We consider the scenario where enough large samples $\{\mathbf{z}^{(t)}\}_{t=1}^{\tau}$ have been recorded by an attacker, who can now try to find the employed secret key $\mathbf{k}$ contained in $\mathbf{x}^t = \mathbf{x}^{(t)}(\mathbf{k})$ using

$$\mathbf{z}^{(t)} = C_{ECC}(C_H(\mathbf{a}^{(t)}||\mathbf{u}^{(t)})) \oplus \mathbf{x}^{(t)} \oplus \mathbf{v}^{(t)}, \quad t = 1, 2, \ldots, \tau,$$

since he has a probability of error in recovering the key which now tends to zero.

For the simplicity of exposition, we assume from now on that $|\mathbf{K}| = n$. We further perform the security evaluation under the following two assumptions:

- $\mathbf{x}^{(t)} = f^{(t)}(\mathbf{k}) = \mathbf{k}\mathbf{S}^t$, $t = 1, 2, \ldots, \tau$, where $\mathbf{S} = [s_{i,j}]_{i=1}^{n}\,{}_{j=1}^{n}$, is a binary matrix, and
$$\mathbf{S}^t = [\mathbf{S}_1^{(t)}, \mathbf{S}_2^{(t)}, \ldots, \mathbf{S}_n^{(t)}]$$
where each $\mathbf{S}_i^{(t)}$, $i = 1, 2, \ldots, n$, denotes a column of the tth power of the matrix $\mathbf{S}$; note that usually $f^{(t)}(\cdot)$ is a heavily nonlinear function, and its consideration as a linear one actually implies a scenario for evaluation of a lower bound of the complexity for the secret key recovery;
- we consider the chosen plaintext attack where the data is the whole zero vector, i.e. $\mathbf{a}^{(t)} = \mathbf{0}$, for each t.

Under the above assumptions, and recalling that both C_{ECC} and C_H are linear encoders, we can write $\mathbf{k}\mathbf{S}^t \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G} = \mathbf{z}^{(t)} \oplus \mathbf{v}^{(t)}$, from which we obtain an algebraic representation of the recovery problem in terms of a noisy system of linear equations, as seen by the adversary:

$$\begin{bmatrix} \mathbf{k}\mathbf{S}_1^{(t)} \\ \mathbf{k}\mathbf{S}_2^{(t)} \\ \vdots \\ \mathbf{k}\mathbf{S}_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_1 \\ [\mathbf{0}||\mathbf{u}^{(t)}]\,\mathbf{G}_2 \\ \vdots \\ [\mathbf{0}||\mathbf{u}^{(t)}]\,\mathbf{G}_n \end{bmatrix} = \begin{bmatrix} z_1^{(t)} \\ z_2^{(t)} \\ \vdots \\ z_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} v_1^{(t)} \\ v_2^{(t)} \\ \vdots \\ v_n^{(t)} \end{bmatrix}, \quad t = 1, 2, \ldots, \tau, \tag{4.6}$$

where $\mathbf{u}^{(t)} = [u_i^{(t)}]_{i=1}^{m-l}$ and $\mathbf{G}_i$ denotes the ith column of $\mathbf{G}$.

Remark 4.1. Note that in the set $\{[\mathbf{0}||\mathbf{u}^{(\mathbf{t})}]\,\mathbf{G}_i\}_{i=1}^{n}$ all the elements could be split into two non-overlapping subsets such that a subset contains k linearly independent elements, k at most $m - l$, and the other subset contains $n - k$ elements each of which is a linear combination of the elements from the first set, since $[0||\mathbf{u}^{(\mathbf{t})}]\,\mathbf{G}$ only involves the lower part of $\mathbf{G}$, which is an $(m - l) \times n$ matrix, which has thus at most $m - l$ linearly independent columns, and the other columns can be obtained as linear combinations.

On the LPN Problem. The problem of solving a system of linear equations in the presence of noise is directly related to LNP problem. What the LPN problem captures is that, given a security parameter k, a secret vector $\mathbf{x}$, and $\mathbf{g}_1, \ldots, \mathbf{g}_n$ randomly chosen binary vectors of length $n = O(k)$, it is possible knowing $y_i = \langle \mathbf{x} | \mathbf{g}_i \rangle$ and $\{\mathbf{g}_i\}_{i=1}^{n}$ to solve for $\mathbf{x}$ using standard linear-algebraic techniques as long as there is no noise. However, when each y_i is flipped (independently) with probability p, finding $\mathbf{x}$ becomes much more difficult. The problem of learning $\mathbf{x}$ in this latter case is refereed to as the learning parity in noise (LPN) problem.

Finally note that the LPN problem is equivalent to the problem of decoding of a general linear block code and it is known that this problem is NP-complete [68], and that relating security of an encryption technique to the LPN problem has been employed for security evaluation of certain stream ciphers (see [4]), for example).

Complexity Evaluation. A systematic way to solve a system of linear equations, with or without noise, is to perform a Gaussian elimination. If the system furthermore contains unknowns that we are not interested in finding, it is natural to start by removing them, so as to obtain a system with a smaller number of equations, where only the unknowns we would like to find are left. We will now show how such a strategy changes the noise present in the system of equations.

Lemma 4.4. *Consider the following system of N equations over the binary field $GF(2)$ to be solved for $x_1, \ldots, x_L$, $L \leqslant N$:*

$$\left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\right) \oplus y_i = z_i \oplus e_i, \quad i = 1, 2, \ldots, M,$$

$$\left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\right) \oplus \left(\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j\right) = z_i \oplus e_i, \quad i = M+1, M+2, \ldots, N,$$

where $\{z_i\}_{i=1}^{N}$, $\{\alpha_j^{(i)}\}_{j=1}^{L}\ {}_{i=1}^{N}$ and $\{\beta_j^{(i)}\}_{j=1}^{M}\ {}_{i=1}^{N}$ are known, $\{x_j\}_{j=1}^{L}$, $\{y_j\}_{j=1}^{M}$ and $\{e_i\}_{i=1}^{N}$ are unknown, and each e_i is a realization of a random variable E_i such that $\Pr(E_i = 1) = p < 1/2$, $i = 1, 2, \ldots, N$. *If*

(1) *the Hamming weight of each vector $[\beta_1^{(i)}, \ldots, \beta_M^{(i)}]$ is greater or equal to some parameter w, for $i = M+1, M+2, \ldots, N$,*

(2) *and no $\bigoplus_{j=1}^{M} \beta_j^{(k)} y_j$, $k \in \{M+1, M+2, \ldots, N\}$, is a linear combination of any other w or less $\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j$, $i \in \{M+1, M+2, \ldots, N\}$, i.e., there are at least w linearly independent sums $\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j$ among those $i \in \{M+1, \ldots, N\}$,*

then, the problem of recovering the unknown $x_1, x_2, \ldots, x_L$ is the problem of solving the following system of equations:

$$\left(\bigoplus_{k=1}^{M} \beta_k^{(i)} \left(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j\right)\right) \oplus \left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\right) = z_i \oplus \left(\bigoplus_{k=1}^{M} \beta_k^{(i)} z_k\right) \oplus e_i^{*},$$

for $i = M+1, M+2, \dots, N$, where e_j^* is a realization of a random variable E_j^* such that $\Pr(E_j^* = 1) > p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$.

Proof. For every $i \in \{M+1, M+2, \dots, N\}$, adding the following linear combination of the first M equations

$$\left(\bigoplus_{k=1}^{M}\beta_k^{(i)}\left(\bigoplus_{j=1}^{L}\alpha_j^{(k)}x_j\right)\right)\oplus\left(\bigoplus_{k=1}^{M}\beta_k^{(i)}y_k\right)=\bigoplus_{k=1}^{M}\beta_k^{(i)}(z_k\oplus e_k),$$

to the ith equations of the system yields:

$$\left(\bigoplus_{k=1}^{M}\beta_k^{(i)}\left(\bigoplus_{j=1}^{L}\alpha_j^{(k)}x_j)\right)\oplus\left(\bigoplus_{j=1}^{L}\alpha_j^{(i)}x_j\right)=z_i\oplus\left(\bigoplus_{k=1}^{M}\beta_k^{(i)}z_k\right)\oplus e_i\oplus\left(\bigoplus_{k=1}^{M}\beta_k^{(i)}e_k\right).$$

We are left to compute the probability $\Pr(E_i^* = 1)$, where

$$E_i^* = E_i \oplus \left(\bigoplus_{k=1}^{M}\beta_k^{(i)}E_k\right), \quad i = M+1, \dots, N.$$

Since $i \geqslant M+1$, E_i is independent of $\beta_k^{(i)}E_k$ for every $1 \leqslant k \leqslant M$. We are thus summing the components of the vector $[E_i, E_1\beta_1^{(i)}, \dots, E_M\beta_M^{(i)}]$ and

$$\Pr(E_i^* = 1) = 1 - \Pr(E_i^* = 0) = 1 - \Pr\left(E_i \oplus \left(\bigoplus_{k=1}^{M}\beta_k^{(i)}E_k = 0\right)\right).$$

Now the probability that an even number of digits are 1 in a sequence of $M+1$ independent binary digits is $\frac{1}{2}\big(1+(1-2p)^{M+1}\big)$ [79, Lemma1] if p is the probability that every digit is 1. Since $\frac{1}{2}\big(1+(1-2p)^{M}\big) > \frac{1}{2}\big(1+(1-2p)^{M+1}\big)$, $p < 1/2$, we have that $1-\frac{1}{2}\big(1+(1-2p)^{M}\big) < 1-\frac{1}{2}\big(1+(1-2p)^{M+1}\big)$, and

$$\begin{aligned}\Pr(E_i^* = 1) &= 1 - \Pr\left(E_i \oplus \left(\bigoplus_{k=1}^{M}\beta_k^{(i)}E_k = 0\right)\right)\\ &> 1 - \frac{1+(1-2p)^{w+1}}{2} = \frac{1-(1-2p)^{w+1}}{2}\end{aligned}$$

since by the assumption 1, the weight of each vector of $[\beta_1^{(i)}, \dots \beta_M^{(i)}]$ is at least w, and according to the assumption 2., there is no linear combination of the equations which can reduce the corruption noise value lower bounded by p_w (i.e., it cannot be reduced via any further linear processing of the system of equations). □

This leads to the following evaluation result.

Theorem 4.3. *The complexity of recovering the secret key* **k** *based on the algebraic representation of the scheme is lower bounded by the complexity of solving the* $LPN_{n,\epsilon}$ *problem where,* $\epsilon = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$ *and* n, w *and* p *are the parameters of the scheme, representing resp. the length of the key, a parameter of the homophonic encoder and the probability of the BSC.*

Proof. From (4.6), we have the following system of τn overdefined consistent but probabilistic equations over the binary field $GF(2)$:

$$\begin{array}{c} \mathbf{kS}_1^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_1 = z_1^{(t)} \oplus v_1^{(t)} \\ \mathbf{kS}_2^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_2 = z_2^{(t)} \oplus v_2^{(t)} \\ \vdots \\ \mathbf{kS}_n^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_n = z_n^{(t)} \oplus v_n^{(t)} \end{array} , \qquad t = 1, 2, \dots, \tau,$$

where each equation is correct with probability equal to p, $\mathbf{0}$ is a l-dimensional vector of all zeroes, and $\mathbf{u}^{(t)} = [u_i^{(t)}]_{i=1}^{m-l}$.

The above system of equations fits the setting of Lemma 4.3, since we have $N = \tau n$ equations, for $L = n$ unknown, where $\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j$, $k = 1, \dots, N$ correspond to $\mathbf{kS}_i^{(t)}$, $i = 1, \dots, n$, $t = 1, \dots, \tau$, and y_j, $j = 1, \dots, M$ together with $\bigoplus_{j=1}^{M} \beta_j^{(k)} y_j$ for $k = M+1, \dots, N$ correspond to $[\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_i$, $i = 1, \dots, M$, $t = 1, \dots, \tau$, since according to Remark4.1, we can indeed separate the $\{[\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_i\}_{i=1}^{n}$ for every t into one set of linear independent vectors, and another set which is obtained as linear combinations of the first set (M is then τk, where k is at most $m - l$).

Note that the above system of τn equations contains only $n + \tau(m - l)$ unknown variables, and that our goal is to recover $\mathbf{k}$ only, i.e., we do not have any interest in recovering $\{u_i^{(t)}\}_{i=1}^{m-l}$, $t = 1, 2, \dots, \tau$. Thus, via Gaussian elimination, we can remove the $\tau(m - l)$ unknown $\{u_i^{(t)}\}_{i=1}^{m-l}$, $t = 1, 2, \dots, \tau$, and obtain $\tau(n - m + l)$ equations where only $\mathbf{k}$ is unknown. This transforms the initial system of τn equations into the following one with $\tau(n - m - l)$ equations (in total) and n unknowns $\mathbf{k}$:

$$\begin{array}{c} \mathcal{L}_1^{(k)}(\mathbf{k}) = \mathcal{L}_1^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_1^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big) \\ \mathcal{L}_2^{(k)}(\mathbf{k}) = \mathcal{L}_2^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_2^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big) \\ \vdots \\ \mathcal{L}_{n-m+l}^{(k)}(\mathbf{k}) = \mathcal{L}_{n-m+l}^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_{n-m+l}^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big) \end{array} , \qquad t = 1, 2, \dots, \tau, \tag{4.7}$$

where $\mathcal{L}_j^{(k)}(\cdot)$, $\mathcal{L}_j^{(z)}(\cdot)$ and $\mathcal{L}_j^{(v)}(\cdot)$, $j = 1, 2, \dots, n - m + l$, are linear functions, all of them specified by the matrix $\mathbf{G}$ and the Gaussian elimination used to remove the random bits $\mathbf{u}^{(t)}$, while $\mathcal{L}_j^{(k)}(\cdot)$ further depends on the matrix $\mathbf{S}^t$. Note that the Gaussian elimination of the variables $\{u_i^{(t)}\}_{i=1}^{m-l}$, can be performed independently for each t, implying that the entire complexity (for $t = 1, 2, \dots, \tau$) is upper-bonded by $\tau O(n^{2.7})$ assuming employment of the most efficient algorithm for the Gaussian processing.

Lemma 4.3 and its underlying assumptions provide that each equation in (4.7) is correct with some probability lower than $1 - p_w$, where $p_w = \frac{1}{2}\big(1 - (1 - 2p)^{w+1}\big)$, since the noise $(\mathbf{v}_1^*)^{(t)} = \mathcal{L}_1^{(v)}([v_i^{(t)}]_{i=1}^{n}), \dots, (\mathbf{v}_{n-m+l}^*)^{(t)} = \mathcal{L}_{n-m+l}^{(v)}([v_i^{(t)}]_{i=1}^{n})$ has coefficients that are the realization of a random variable which takes value 1 with probability greater than $p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$. The above system of $\tau(n-m+l)$ equations can consequently be rewritten as:

$$\begin{aligned}
\mathcal{L}_1^*([k_i]_{i=1}^n) &= \mathcal{L}_1^{(z)}([z_i^{(1)}]_{i=1}^n)\\
\mathcal{L}_2^*([k_i]_{i=1}^n) &= \mathcal{L}_2^{(z)}([z_i^{(1)}]_{i=1}^n)\\
&\vdots\\
\mathcal{L}_{n-m+l}^*([k_i]_{i=1}^n) &= \mathcal{L}^{(z)}_{n-m+l}([z_i^{(1)}]_{i=1}^n)\\
\mathcal{L}_{n-m+l+1}^*([k_i]_{i=1}^n) &= \mathcal{L}_1^{(z)}([z_i^{(2)}]_{i=1}^n)\\
\mathcal{L}_{n-m+l+2}^*([k_i]_{i=1}^n) &= \mathcal{L}_2^{(z)}([z_i^{(2)}]_{i=1}^n)\\
&\vdots\\
\mathcal{L}_{\tau(n-m+l)}^*([k_i]_{i=1}^n) &= \mathcal{L}^{(z)}_{n-m+l}([z_i^{(\tau)}]_{i=1}^n)
\end{aligned}$$

where $\mathcal{L}_j^*$, $j = 1, 2, \ldots, \tau(n-m+l)$, are linear functions, and where each equation is incorrect with probability greater than $p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$.

We finally get:

$$\begin{aligned}
\langle \mathbf{k}|\mathbf{c}_1\rangle &= d_1\\
\langle \mathbf{k}|\mathbf{c}_2\rangle &= d_2\\
&\vdots\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l}\rangle &= d_{m-n+l}\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l+1}\rangle &= d_{m-n+l+1}\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l+2}\rangle &= d_{m-n+l+2}\\
&\vdots\\
\langle \mathbf{k}|\mathbf{c}_{\tau(n-m+l)}\rangle &= d_{\tau(m-n+l)}
\end{aligned}$$

where each equation is correct with a probability upper-bounded by $1-p_w = 1-\frac{1}{2}\big(1-(1-2p)^{w+1}\big)$, and where the n-dimensional binary vectors $\{\mathbf{c}_j\}_{j=1}^{\tau(n-m+l)}$ and $\{d_j\}_{j=1}^{\tau(n-m+l)}$ are known.

According to the definition of the LPN problem and the above representation, the problem of recovering the secret key is at least as hard as the $\mathrm{LPN}_{n,\epsilon}$ problem with $\epsilon = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$, which concludes the proof of the theorem. □

4.4. A Generalization of the LPN Problem and Its Hardness. The LPN problem has a number of equivalent formulations and under consideration of this section is a formulation which corresponds to the decoding problem. It has been shown in [68] that the decoding incarnation of the LPN problem is NP-complete which implies suitability of this problem as an underlying problem for certain cryptographic applications. The basic LPN problem can be considered as a problem of solving an overdefined and consistent system of linear but noisy equations corresponding to the following vector equation:

$$\mathbf{z} = \mathbf{k}\mathbf{S} \oplus \mathbf{v}, \tag{4.8}$$

where $\mathbf{k}$ is $|\mathbf{k}|$-dimensional binary vector of the variables which are target of recovering, $\mathbf{z}$ is given n-dimensional, $n \gg |\mathbf{k}|$, binary vector, $\mathbf{S}$ is known $|\mathbf{k}| \times n$ binary matrix, and $\mathbf{v}$ is unknown n-dimensional binary vector of independent identically distributed elements which take value 1 with the probability $\epsilon = p < 1/2$ and value 0 with the probability $1 - p$.

The goal is recovering of $\mathbf{k}$ with minimization of the probability of error and this goal corresponds to decoding of a linear block code. Accordingly, the goal can be achieved employing the minimum distance decoding paradigm based on an exhaustive search according to the following:

- For each possible candidate $\hat{\mathbf{k}}$ for $\mathbf{k}$ evaluate the Hamming weight, $Hwt(\cdot)$, of the vector $\mathbf{z} \oplus \hat{\mathbf{k}}\mathbf{S}$;
- Select as the true candidate $\hat{\mathbf{k}}$ the one which provides minimum $Hwt(\mathbf{z} \oplus \hat{\mathbf{k}}\mathbf{S})$.

A generalized LPN problem can be formulated as a problem of solving an overdefined and consistent system of linear but noisy equations corresponding to the following vector equation:

$$\mathbf{z} = [\mathrm{a}||\mathbf{u}]\mathbf{G} \oplus \mathbf{k}\mathbf{S} \oplus \mathbf{v}, \tag{4.9}$$

where a is known l-dimensional binary vector, $\mathbf{u}$ is an unknown random $(m - l)$-dimensional binary vector of independent identically distributed elements which take values 0 and 1 with the same probability equal to 1/2, $\mathbf{G}$ is known $m \times n$ binary matrix, $l < m < n$, and $\mathbf{z}$, $\mathbf{k}$, $\mathbf{S}$ and $\mathbf{v}$ are defined by the above basic LPN problem specification.

Regarding the generalized LPN problem specified by (4.9), a similar approach can be employed as regarding the basic one (4.8) but taking into account that not only $\mathbf{k}$ is unknown but $\mathbf{u}$ as well. Accordingly, we can employ the following approach for recovering unknown $\mathbf{k}$:

- For each possible candidate $\hat{\mathbf{k}}$ for $\mathbf{k}$ and all possible vectors $\mathbf{u}$ evaluate the Hamming weight, $Hwt(\cdot)$, of the corresponding vectors $\mathbf{z} \oplus [\mathrm{a}||\hat{\mathrm{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}$;
-
  - If a unique minimum $Hwt(\cdot)$ exists, select as the true candidate $\hat{\mathbf{k}}$ the one which yields this minimum value;
  - If the unique minimum $Hwt(\cdot)$ does not exist, make a list $\mathcal{L}$ of the final candidates $\hat{\mathbf{k}}$ for the true $\hat{\mathbf{k}}$ such that each final candidate yields:

$$Hwt(\mathbf{z} \oplus [\mathrm{a}||\hat{\mathrm{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}) \leqslant w_{thr},$$

where w_{thr} is certain threshold value.

Note that the outcome of the considered approach depends on the parameters $|\mathbf{k}|$, l, m and n. If $|\mathbf{k}| + m - l - n > 0$ after the exhaustive search we obtain $2^{|\mathbf{k}|+m-l-n}$ pairs $(\hat{\mathbf{k}}, \hat{\mathrm{u}})$ which yield that $Hwt(\mathbf{z} \oplus [\mathrm{a}||\hat{\mathrm{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}) = 0$ yielding that the approach outcome is $2^{|\mathbf{k}|+m-l}(1-p)^n$ equally likely candidates for the true $\mathbf{k}$. On the other hand, the all zeros noise pattern is not the most likely one, and in order not to miss inclusion into the list of candidates, the true one, instead of only minimum $Hwt(\cdot)$, all error patterns up to certain weight should be included as the eligible candidates yielding the list of candidates $\mathcal{L}$.

Note the following: For given $\mathbf{z}$ and $\mathbf{a}$ and assumed $\mathbf{k}$, 2^{m-l} different vectors $\mathbf{u}$ will yield 2^{m-l} different vectors $\mathbf{z} \oplus \mathbf{kS} \oplus [\mathbf{a}||\mathbf{u}]\mathbf{G}$ assuming appropriate matrix $\mathbf{G}$. Employing the random arguments, among these vectors $2^{m-l}\binom{n}{w}p^w(1-p)^{n-w}$ will have the Hamming weight w. Accordingly, the expected dimension of the list $\mathcal{L}$ can be estimated as follows:

$$|\bar{\mathcal{L}}| = \min\left\{2^{|\mathbf{k}|}, 2^{|\mathbf{k}|+m-l}\sum_{w=0}^{w_{thr}}\binom{n}{w}p^w(1-p)^{n-w}\right\}.$$

and lower-bounded as

$$|\bar{\mathcal{L}}| \geqslant \min\left\{2^{|\mathbf{k}|}, 2^{|\mathbf{k}|+m-l-n}\sum_{w=0}^{w_{thr}}\binom{n}{w}\right\}.$$

When $w_{thr} = pn$,

$$\sum_{w=0}^{w_{thr}}\binom{n}{w} \leqslant 2^{h(p)n}$$

where $h(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2(1-p)$.

The above consideration has the following implication: If the parameters of the generalized LPN problem are such that $m - l > n(1 - h(p))$, after the considered optimal search for the true hypothesis on $\mathbf{k}$, the expected number of equally-likely candidates can be close to the total number of candidates.

5. A Security Evaluation of Broadcast Encryption Key Management Schemes

5.1. Introduction. A conventional approach for access control to broadcasted (multicasted) data employs the following paradigm: the data are encrypted and only legitimate users are provided with the information on how to decrypt them. We consider schemes where the data encryption is performed based on a symmetric cipher and the updatable secret session encrypting key (SEK). To make SEK updating possible, another set of keys called the key-encrypting keys (KEKs) are involved. There are two basic approaches for establishing the required security based on the above paradigm. One approach uses static KEKs (see [99], [64] and [94], for example), and the other one employs updatable KEKs (see [101], [102], and [110], for example). BE schemes with static KEKs (stateless receivers) have the desirable feature that members do not need to be always connected online, which is especially preferable for applications over lossy channels. Since rekey messages in stateless schemes are independent of each other, members once being offline or inactive can always decrypt the latest group key instantly.

In this section the security evaluation of certain BE schemes with static KEKs is considered. In order to enhance the security of these schemes, before the encryption, the SEK is XOR-ed with the identifier (ID) of the KEK employed for its encryption, as proposed in [94] and [99].

This section points out to a weakness of certain broadcast encryption schemes in which the protected delivery of a session key (SEK) is based on XOR-ing this SEK with the IDs of the keys employed for its encryption is addressed. The weakness can be effectively explored assuming passive attacking which in the cases corresponding to a malicious legitimate user being the attacker, is a ciphertext only attack. A dedicated algorithm for cryptanalysis is discussed based on a generalized time-memory-data trade-off approach and its main characteristics are derived. The algorithm points out a security weakness of employing a block cipher with block length shorter than the key length in the considered BE schemes.

5.2. Models of Certain Broadcast Encryption and Problem Statement. Let KEK_i denote a KEK employed in the system, and let ID_i denote its name or ID, assuming that ID_i does not disclose any information on KEK_i itself. BE is based on the following approach. The system center generates all the employed KEKs. A user of the BE system is in advance provided with a subset of all KEKs employed in the system. Note that different users can have overlapping subsets of KEKs, but no pair of users have an identical subset.

In a basic BE setting, the procedures at the center and for each of the users are based on the following. When the current SEK should be updated, the center finds a subset $I = I(SEK)$ of KEKs $\{KEK_i\}_{i \in I}$ such that each of the legitimate users possesses at least one of these keys and none of the un-legitimate users possesses any of these keys. The center encrypts the data with SEK, generates all encrypted forms of SEK employing each KEK_i, $i \in I$, and broadcasts $\langle [header]; G_{SEK}(data) \rangle = \langle [\{(ID_i, E_{KEK_i}(SEK))\}_{i \in I}]; G_{SEK}(data) \rangle$, where $E(\cdot)$ and $G(\cdot)$ are certain encryption algorithms.

In order to address certain weaknesses of this basic BE model, in [94], an enhanced security approach for BE is proposed, which corresponds to the following BE header model:

$$[header] = [\{(ID_i, E_{KEK_i}(SEK \oplus ID_i))\}_{i \in I}], \tag{5.1}$$

where $\oplus$ denotes bit-by-bit XOR-ing of the vectors SEK and ID_i. This enhanced approach is employed in[64] as well.

The problem addressed in this section is the security evaluation of the BE schemes which follow the header model specified by (5.1). Recently, vulnerabilities of certain BE schemes have been reported in [10] and [8] and they provide the origins for the approach given in this letter. Particularly note that a security weakness of the approach proposed in [94] is reported in [10] employing an active attack scenario, while the scope of this consideration is restricted to passive attacking.

5.3. Scenario for the Security Evaluation. The considered settings for cryptanalysis originate from the following issues: (i) It is a realistic scenario that different schemes of the same class are deployed and are subject to malicious monitoring; (ii) In a typical BE scheme with stateless receivers KEKs are in a tamper-proof (resistant) hardware and accordingly the system should be considered as broken even if an attacker can recover only one of the KEKs employed in the system.

We assume a system where N BE schemes of the same structure but with different (non-overlapping) KEKs are employed and that the attacker can monitor J SEK updates in each of these schemes. In the considered system, in order to provide the legitimate users with the decryption key $SEK_j^{(n)}$, the set $S_j^{(n)}$ of the following pairs is publicly available:

$$S_j^{(n)} = \{(ID_i^{(n)}, C_{i,j}^{(n)})\}_{i\in I^{(n)}(SEK_j^{(n)})}, \tag{5.2}$$

where $C_{i,j}^{(n)} = E_{KEK_i^{(n)}}(SEK_j^{(n)} \oplus ID_i^{(n)})$, $j = 1,2,\dots,J$, $n = 1,2,\dots,N$, and where $E(\cdot)$ is a block cipher which employs length-K secret key and operates over L-dimensional binary blocks. We also assume that the following is valid:

- i_{max} KEKs are employed in each of the considered N BE schemes and for each KEK: (i) $KEK_i^{(n)}$ is a randomly generated binary vector of length-K and $2^K \gg Ni_{\max}$; (ii) $ID_i^{(n)}$ is a length-L binary vector, $2^L > i_{\max}$, which only indicates that the encrypted form of $SEK_j^{(n)}$ is obtained employing the key $KEK_i^{(n)}$ and does not provide any information on the binary vector $KEK_i^{(n)}$;
- For each SEK: (i) SEK is a binary vector of dimension L, $K/2 \leqslant L < K$; (ii) each $I(SEK)$ is a subset of $\{1,2,\dots,i_{max}\}$, and for simplicity, we assume that $|I(SEK)| = I$.
- The employed encryption algorithm $E(\cdot)$ is secure, so that the following holds: any $C_i = E_{KEK_i}(SEK \oplus ID_i)$ does not yield any information on KEK_i and SEK.

The attacker's prior knowledge is limited to the following: (i) The attacker knows the entire structure of the BE scheme under cryptanalysis including the employed encryption algorithm $E(\cdot)$; (ii) The attacker does not know any of the keys $KEK_i^{(n)}$, $i = 1,2,\dots,i_{max}$, $n = 1,2,\dots,N$, employed in the considered N BE schemes.

The goal of the attacker is to recover at least *one of the secret keys* $KEK_i^{(n)}$, $i = 1,2,\dots,i_{\max}$, $n = 1,2,\dots,N$, employed in the considered system of BE schemes.

The scenario for cryptanalysis assumes that the attacker has a suitable (large) collection of the following data: (i) the headers $S_j^{(n)}$ specified by (5.2), and (ii) the employed $SEK_j^{(n)}$.

5.4. A Method for Cryptanalysis of Certain Broadcast Encryption Schemes.

The developed technique for security evaluation of the considered class of BE schemes includes the following: (i) collecting a suitable sample via monitoring SEKs update in a number of different BE schemes of the considered class; (ii) employment of a sophisticated and dedicated "dictionary" with implicitly memorized words which provides a time-memory-data trade-off.

Assuming that $SEK_j^{(n)}$ is selected randomly and independently of $ID_i^{(n)}$, $i = 1,2,\dots,I$, the probability that the given $SEK_j^{(n)}$ is equal to one of $ID_i^{(n)}$, $i =$

$1, 2, \dots, I$, is equal to $I2^{-L}$ as all $ID_i^{(n)}$'s are distinct. Accordingly, the probability $P^*(k)$ that, for a given n, there are exactly k common elements in the sets $\{SEK_j^{(n)}\}_{j=1}^J$ and $\{ID_i^{(n)}\}_{i=1}^I$ is:

$$P^*(k) = \binom{J}{k} 2^{-k(L-\log_2 I)}(1 - 2^{-L+\log_2 I})^{J-k}. \tag{5.3}$$

Note that (5.3) implicitly assumes that all $SEK_j^{(n)}$'s are distinct, which is readily achieved for $2^L \gg J$. This implies that in the sets $\{SEK_j^{(n)} \oplus ID_i^{(n)}\}_{i=1}^I{}_{j=1}^J$, $n = 1, 2, \dots, N$, the expected number $\hat{D}$ of the the elements which sum to a certain pattern is given by:

$$\hat{D} = NJI2^{-L}. \tag{5.4}$$

In the following, for simplicity of the exposition, we consider the all zeros pattern $\mathbf{0}$, and in the same manner any other pattern can be employed.

A time-memory-data trade-off (TMD-TO) approach for cryptanalysis has been reported in [69] as a generalization of the time-memory trade-off based cryptanalysis [84]. In this section, a further generalized dedicated time-memory-data trade-off approach for cryptanalysis of the considered BE system is proposed, assuming that the parameters are such that $IJN > 2^L$. The main steps are:

- Perform a suitable pre-processing for a dedicated TMD-TO based cryptanalysis assuming that the encryption algorithm encrypts only all zeros L-dimensional binary vectors. The pre-processing output is a set of tables. This pre-processing should be done only once and is independent of the sample for cryptanalysis and the KEKs employed in the system.
- Collect the sample for processing consisting of the ciphertext corresponding to the vectors $SEK_j^{(n)} \oplus ID_i^{(n)} = \mathbf{0}$ (on average D values based on (5.4)).
- Using the tables prepared during pre-processing and the collected sample perform the cryptanalysis employing a dedicated TMD-TO based cryptanalysis and generate a list of candidates.
- From the list of candidates recover one or more KEKs via an additional check of each candidate.

5.4.1. Algorithm for Cryptanalysis.

Pre-Processing

- *Input Data:* The algorithm parameters K, L, M, T, and D such that $M^2D^2T = 2^{2L}$.
- *Pre-Processing Steps*
  For $u = 1, 2, \dots, 2^{K-L}$ and $i = 1, 2, \dots, M$, do the following:
  (1) Randomly select an L-dimensional binary vector X_0' and define $X_i(0) = X_0'||U$ where U is the length-$(K - L)$ binary representation of $u - 1$, and $||$ denotes the concatenation of two vectors.
  (2) For $t = 1, 2, \dots, T$, perform the following recursive calculation: $X = E_{X_i(t-1)}(\mathbf{0})$, $X_i(t) = X||U$.

(3) Memorize $X_i(0)$ in the first column and the first L elements of $X_i(T)$ in the second column of the i-th row of the $M \times 2$ matrix M_u.

- *Output*: Tables M_u, $u = 1, 2, \ldots, 2^{K-L}$, of the pairs memorized in step 3.

Processing

- *Input Data*: Set S_D of D different values $C^{(n)}_{i,j} = E_{KEK^{(n)}_i}(SEK^{(n)}_j \oplus ID^{(n)}_i = 0)$, $i \in I(j)$, $j \in \{1, 2, \ldots, J\}$, $n \in \{1, 2, \ldots, N\}$, $I = |I(j)|$.
- *Processing Steps*

  I. *Generation of the List of Candidates*

  For each triple (n, j, i) such that $C^{(n)}_{i,j} \in S_D$, and all tables M_u, $u = 1, 2, \ldots, 2^{K-L}$, do the following:

  (1) Set: $t = 0$, $X'_t = C^{(n)}_{i,j}$, and $X_t = X'_t||U$ where U is the length-$(K-L)$ binary representation of $u - 1$.

  (2) Check the identity of the considered X'_t to any of the second column elements of the matrix M_u; if an identity appears in the i-th row, go to step 4; otherwise go to step 3.

  (3) Set $t \to t + 1$. If $t \leqslant T$, calculate $X'_t = E_{X_{t-1}}(\mathbf{0})$, $X_t = X'_t||U$, and go to step 2; if $t > T$, go to step 5.

  (4) (a) Select the corresponding $X_i(0)$ and set $X_0 = X_i(0)$;

  (b) Perform the following iterative calculation: $X'_{v+1} = E_{X_v}(\mathbf{0})$, $X_{v+1} = X'_{v+1}||U$ until $X'_{v+1} = C^{n}_{i,j}$; Memorize X_v into the list of candidates *List*, and go to step 5.

  (5) Select a previously not considered $C^{(n)}_{i,j}$ and go to step 1; If all elements of S_D have been considered go to phase II.

  II. *Recovering KEKs from the List of Candidates*

  For each candidate Y from *List* do the following:

  (1) For $O(1)$ different randomly selected indices j, $j \in \{1, 2, \ldots, J\}$, check the equality of $E_{KEK_i = Y}(SEK^{(n)}_j \oplus ID^{(n)}_i) = C^{(n)}_{i,j}$;

  (2) If all the checks in the previous step are fulfilled, memorize the considered Y as the recovered $KEK^{(n)}_i$.
- *Output*: Set of the recovered KEKs obtained via the memorized pairs in step II.2.

5.4.2. Complexity of Cryptanalysis. Based on the structure of the considered algorithm, and the results on TMD-TO reported in [69] the following statements are readily proved.

Proposition 5.1. *The proposed algorithm has space complexity $2^{K-L}M$, preprocessing time complexity proportional to $2^{K+L}M^{-1}D^{-2}$ and expected processing time complexity $O(2^{K+L}M^{-2}D^{-2}) + O(2^{K-L})$, assuming $D > 1$ and the goal is recovering one KEK. It provides different possible trade-offs between the parameters T,M, D and L, assuming that the following trade-off condition holds $TM^2D^2 = 2^{2L}$.*

5.5. Security Evaluation of Certain BE Schemes.

A numerical illustration of Proposition 5.1 is given in Table 4 assuming employment of a block cipher which operates over 64-bit blocks and uses a length-K secret key with $K > 64$. The column regarding 80-bit KEKs can correspond to a block cipher with a variable-length key (see [77], for example). The column corresponding to 128-bit KEKs holds even if a highly secure block-cipher MISTY1 (accepted as a standard block-cipher in [85]) is employed, and the column corresponding to 112-bit KEKs holds when the Triple DEA (see also [85]) with two 56-bit keys, which is a standard encryption primitive in many commercial products, is employed.

TABLE 4. Complexity of recovering one KEK of K bits in a system with N BE schemes and $J = 2^{30}$ SEK updates in each one, when SEKs and IDs consists of $L = 64$ bits, $K > L$, and a secure block cipher which operates over $L = 64$-bit blocks is employed.

KEK dimension K	80 bits	112 bits	128 bits
number N of monitored BE schemes	2^{20}	2^{30}	2^{35}
number I of KEKs in a BE scheme	$\sim 2^{30}$	$\sim 2^{35}$	$\sim 2^{40}$
the algorithm parameter M	2^{40}	2^{21}	2^{21}
space complexity of the algorithm	$\sim 2^{56}$	$\sim 2^{69}$	$\sim 2^{85}$
time complexity of pre-processing	$\sim 2^{72}$	$\sim 2^{94}$	$\sim 2^{99}$
expected time complexity of recovering one KEK	$\sim 2^{32}$	$\sim 2^{70}$	$\sim 2^{78}$

5.6. Concluding Remarks. It has been shown that indirect encryption of SEKs (modified by XOR-ing with IDs of KEKs) employing KEKs longer than SEKs does not provide the desired protection of KEKs in a number of scenarios. The developed generalized TMD-TO algorithm for cryptanalysis shows that the employment of block ciphers which operate over blocks shorter than the key in certain BE schemes implies a security weakness of these schemes regardless of the security level of the considered block cipher. Particularly, note that the employment of even highly secure block cipher has no impact against the proposed technique for cryptanalysis because the performance of the proposed algorithm for cryptanalysis does not depend on the security of the employed cryptographic primitives but on the considered BE system parameters. In the process, we also generalized the algorithms [84, 69] to the case where the secret key length is larger than the length of the encrypted blocks.

6. Design of Certain Broadcast Encryption Schemes

This section addresses the following issues:

- An approach for the cryptographic keys management in the broadcasting scenario with a conditional access control is considered. It employs the reconfiguration concept, and it is based on a collection of the underlying structures-at each instant of time a structure from the collection is employed for updating the session key so that the communication overhead of updating is minimized.

A receiver has a fixed set of cryptographic keys and in a general case, each of these keys plays a different role determined by the employed underlying structure.

- The problem of minimizing the amount of secret information (secret bits) required for certain key management schemes related to data access control techniques is addressed. The importance of secret storage minimization originates from the fact that this storage should be both read-proof and tamper-proof. The proposed approach intends to minimize the protected (secret) storage by introducing public storage in conjunction with an efficient one-way mapping of the secret bits in the exchange of information from private to public storage. This method achieves reduction of the secret storage overhead at the user's side and provides an appropriate trade-off between the reduced private storage, and the required public storage and associated processing complexity. The method is applied to the minimization of the secret storage required by two recently proposed key management schemes, and the overheads of the modified schemes are compared with the related previously reported ones, pointing out the advantages of the novel approach.

6.1. Reconfigurable Broadcast Encryption. This section addresses a problem of conditional access control to the broadcasted data where at each instant of time only the legitimate receivers have the access assuming that the set of these receivers is a highly dynamical one. The same scenario as in [83] is considered. The broadcasting is towards receivers with the pre-configured and not updatable states and has the following main requirements: (i) Each user is initially given a collection of symmetric encryption keys; (ii) The keys do not change when other users join or leave the system.

This section considers an approach for the key management which employs the reconfigurability concept. A generic framework for the reconfigurable key management is shown and an illustrative particular scheme is discussed. In the considered particular case, the developed approach is compared with the best previously reported schemes, and the advantages of the novel one are pointed out.

6.1.1. Background: Conditional Access Control and Key Management. As pointed out in the previous section, when cryptography is employed for securing broadcasting communications, a session-encrypting key (SEK) is used to encrypt the data. Ensuring that only the valid members of the group have SEK at any given time instance is the key management problem in secure broadcasting/multicast communications. To make this updating possible, another set of keys called the key-encrypting keys (KEKs) should be involved so that it can be used to encrypt and transmit the updated SEK to the valid members of the group. Hence, the key management problem reduces to the problem of distributing KEKs to the members such that at any given time instant all the valid members can be securely updated with the new SEK.

In [99], a generic framework, is given by encapsulating several previously proposed revocation methods called Subset-Cover algorithms. These algorithms are based on the principle of covering all non-revoked users by disjoint subsets from a predefined collection, together with a method for assigning KEKs to subsets in the collection. For further discussion we assume that there are N users and R revocations.

Two types of revocation schemes in the Subset-Cover Framework, are proposed [99] with a different performance trade-off. Both schemes are tree-based, namely the subsets are derived from a virtual tree structure imposed on all receivers in the system. The first proposed scheme, Complete Sub-Tree (CST) scheme, requires a message length of $R\log_2(N/R)$ and storage of $1+\log_2 N$ keys at the receiver and constitutes a moderate improvement over previously proposed schemes. The second scheme, called the Subset Difference (SD) algorithm exhibits a more improvement: it requires a message length of $2R$. The improved performance of SD is primarily due to its more sophisticated choice of covering sets. Let i be any vertex in the tree and let j be any descendent of i. Then $S_{i,j}$ is the subset of leaves which are descendants of i but are not descendants of j. Note that $S_{i,j}$ is empty if $i = j$. Otherwise, $S_{i,j}$ looks like a tree with a smaller subtree cut out. The SD scheme covers any privileged set P defined as the complement of R revoked users by the union of $O(R)$ of these $S_{i,j}$ sets.

What is shown in [83] is that this collection of sets can be reduced: The basic idea of the Layered Subset Difference (LSD) scheme is to use only a small subcollection of $S_{i,j}$ sets employed by SD scheme which suffices to represent any P as the union of $O(R)$ of the remaining sets, with a slightly larger constant. Since there are fewer possible sets, it is possible to reduce the number of initial keys given to each user. In [83], it is shown that if we allow the number of sets in the cover to grow by a factor of two, we can reduce the keys storage from $O((\log_2 N)^2)$ to $O((\log_2 N)^{3/2})$.

6.1.2. Reconfigurable Key Management. Underlying Ideas. An approach for the key management is proposed in [19] and [18] which has the following basic characteristics: (i) It employs the reconfiguration concept, and it is based on a collection of the underlying structures-at each instant of time a structure from the collection is employed for updating the session key SEK so that the communication overhead of updating is minimized; (ii) A receiver has the single set of cryptographic keys which (and each of these keys) plays a role determined by the employed underlying structure.

A main component of the reconfigurable key management is a collection of the underlying structures, and regarding these structures note the following.

- The underlying structures could be very different but all of them should fulfil the following condition: They should be able to work with the same set of keys (KEKs) assuming that a key can be employed in different modes.
- Selection of the underlying structures for the collection depends on the functional requirements of the key management, and particularly on the identified most likely classes of the revocation patterns.

Generic Framework

- Center forms a collection of different underlying structures suitable for different revocation scenarios; usually, each underlying structure is a graph with the keys and users assigned to the graph nodes, and the employed graph is a tree where the users are assigned to the tree leaves, and the keys are assigned to all the tree nodes.

- Taking into account all the underlying structures, center selects a set of the keys (KEKs) to be assigned to each of the receivers, as an appropriate subset of all the keys related to the employed underlying structures. (Note that in a general case, the subsets corresponding to the different receivers are the overlapping ones.)
- For each of the session key, SEK, updating, center selects one of the underlying structures in such a way to minimize the communication overhead under given revocation requests.
- Center performs SEK updating by broadcasting the following: (i) encrypted forms of the new SEK obtained by employing different KEKs; (ii) labels of the employed KEKs, and (iii) the mod of KEKs use (which depend on currently used underlying structure).
- If not revoked, during the key management communication, a receiver will find the following information in the updating message: (i) which of its KEKs should be employed for the new SEK recovering, and (ii) in which mode the KEK should be used. Accordingly, a not revoked receiver is able to recover the new SEK.

Note that the proposed framework employs a reconfigurable logical key hierarchy (RLKH), and accordingly we call it RLKH. Also note that the proposed framework for design of particular reconfigurable, i.e. time varying, key management schemes could provide minimization of the cumulative main system overheads (storage and processing at receiver, and updating communications) over highly dynamical set of privileged users.

Implementation Issues. At the center side RLKH implementation includes establishing of the RLKH system, and during this phase the center selects the component key management schemes so that each of them is suitable for certain class of the revocation patterns. Accordingly, during the establishing phase the center forms a list of the following pairs: (*revocation pattern class; key management scheme*). Storage requirements for this list of pairs and related information on the component schemes is usually negligible in comparison with the number of keys which should be stored at the center. One-to-one correspondence between the revocation pattern class and the component scheme implies that RLKH employment does not require any additional processing for selecting a particular key management at any time instance.

At a receiver, in a general case, according to the extracted information from the broadcast, a mapping of a KEK should be performed. Note that this mapping is not a secret operation and usually it should be the cryptographic one-way hashing (see [98], for example).

6.1.3. Illustrative Example of Reconfigurable Key Management.

This section yields an illustrative toy example of the reconfigurable key management when only two underlying tree structures are employed. It is assumed that there are $N = 2^n$ receivers and that each receiver holds $1 + log_2 N$ keys. The following two underlying structures, Tree-A and Tree-B, are employed:

- Tree-A: a binary balanced tree graph of height $log_2 N$;
- Tree-B: a tree graph with M branches from the root, and a binary balanced sub-tree of height $log_2(N/M)$ rooted at each of M branches, where M is a parameter which value will be discussed bellow.

An usual assignment of the center, users and keys to the considered trees is assumed: (i) the center is assigned to the root; (ii) each receiver is a leaf of the tree; (iii) all the employed keys in the scheme are assigned to the tree nodes.

Illustrative examples of the considered trees are given in Fig. 3 and 4.

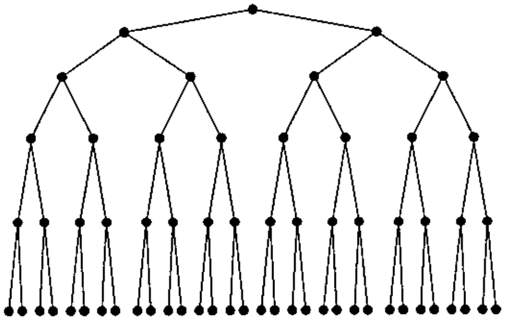


FIGURE 3. An illustration for the underlying structure Tree-A when $N = 32$.

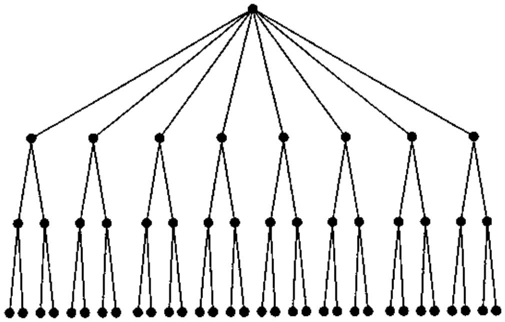


FIGURE 4. An illustration for the underlying structure Tree-B when $N = 32$ and $M = 8$.

The above trees are related to certain key management schemes according to the following:

- CST[99] is employed over Tree-A;
- basic LSD[83] is employed over each of M binary balanced sub-trees in Tree-B.

It is assumed that in the system developing phase the center has established one-to-one correspondence between the revocation pattern class and appropriate component scheme so that the appearance of the revocation pattern directly selects the scheme to be employed.

Having in mind that according to [83] employment of LSD based key management over a binary balanced tree of height $\log_2(N/M)$ requires that each receiver should store $(\log_2(N/M))^{1.5}+1$ keys (assuming appropriate values of N and M), and taking into account limit on total number of keys at a receiver, it appears that parameter M should be the smallest integer such that the following inequality holds,

$$\log_2 N \geqslant (\log_2(N/M))^{1.5}+1. \tag{6.1}$$

Assuming, for simplicity, the equality in (6.1), the following assigning and processing of the secret bits is employed.

- Each receiver stores the secret bits in form of the keys required for Tree-B with LSD.
- Required mapping of the keys for employment over Tree-A with CST is performed based on dedicated one-way (collisionful) hash functions.

According to the previous, and taking into account the results reported in [99] and [83], it can be shown that the considered reconfigurable key management has the following main characteristics.

Proposition 6.1. *The considered reconfigurable key management requires the following overhead for R revocations in total assuming that $R=\sum_{m=1}^{M} r_m$, where r_m is the number of revocations corresponding to the m-th subtree in Tree-B structure:*

- *dimension of the storage needed by a receiver:* $O(\log_2 N)$;
- *dimension of the communication overhead:* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\}$;
- *dimension of the processing at receiver overhead:*

$O(\log_2 log_2 N)$ *if* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\} = O(R\log_2(N/R))$, *or* $O(\log_2(N/M))$ *if* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\} = O(R)+\sum_{m=1}^{M}\delta_{0,r_m}$; *where* $\delta_{0,x}$ *takes value* 1 *for* $x=0$ *and 0 otherwise.*

Sketch of the Proof. According to [99], when Tree-A with CST is employed, the key management requires the following overhead:

(i) dimension of the communication overhead-$O(R\log_2(N/R))$;
(ii) dimension of the processing at receiver overhead-$O(\log_2\log_2 N)$;
(iii) dimension of the storage needed by a receiver-$O(\log_2 N)$.

When Tree-B with LSD is employed we have the following. Revocation of r_m receivers corresponding to the m-th subtree requires the communication overhead of $O(r_m)$. Accordingly, the communication overhead for revocation of all $R=\sum_{m=1}^{M} r_m$ receivers is $O(R)$. Also, the new session key should be sent, as well, to all clusters of users where no one revocation has been made, and this rekeying requires additional $\sum_{m=1}^{M}\delta_{0,r_m}$ messages. On the other hand, when LSD based key management is employed over a subtree of height $\log_2(N/M)$, according to[83], the processing at receiver overhead is no more than $O(\log_2(N/M))$.

Finally, note that at each instant of time, according to the current revocation request, the center selects the underlying structure which minimizes the communication overhead. The above, and taking into account (6.1), yields the proposition statement.

Accordingly, based on the characteristics of CST, SD and LSD reported in [99] and [83], a comparison of these schemes and the considered RLKH is summarized in the following Table 5 assuming N receivers and $R = \sum_{m=1}^{M} r_m$ revocations, $N > R$, $r_m \geqslant 0$, $m = 1, 2, \ldots, M$, that the parameters N, M are related by (6.1), and that $\delta_{0,x}$ takes value 1 for $x = 0$ and 0 otherwise.

TABLE 5. Comparison of the considered BE schemes.

	storage at receiver	processing at receiver	communication
CST [99]	$O(\log_2 N)$	$O(\log_2 \log_2 N)$	$O(R \log_2 \frac{N}{R})$
SD [99]	$O((\log_2 N)^2)$	$O(\log_2 N)$	$O(R)$
basic LSD [83]	$O((\log_2 N)^{1.5})$	$O(\log_2 N)$	$O(R)$
proposed RLKH	$O(\log_2 N)$	$\leqslant$ $O(\log_2(N/M))$	$\min\{O(R \log_2(N/R));$ $O(R) + \sum_{m=1}^{M} \delta_{0,r_m}\}$

6.1.4. Discussion. The main characteristics of the up to now reported key management schemes include employment of a static underlying structure for the key management, and addressing the subset covering problem over the entire underlying structure. Oppositely, the main underlying ideas for developing of the improved key management schemes based on RLKH include the following: (i) employment of a reconfigurable underlying structure; and (ii) in a general case employment of a divide-and-conquer approach over the underlying structure. RKLH appears as a very suitable approach for highly dynamic revocation scenarios.

Employment of RLKH for a SEK updating requires just a slight (almost negligible) increase of required processing at the both sides, at the center and at the receiver. On the other hand, RLKH requires a moderate processing at the center side in order to establish the system, but this operation should be done just once.

6.2. A Broadcast Encryption Approach Based on Coding.

6.2.1. Introduction. As already pointed out in Section 5.1, the KEKs are used to encrypt and deliver the updated SEK to the valid members of the group. In order to obtain the desired security, the KEKs must be kept in a protected storage called the secret storage. This storage should be as small as possible to allow an efficient implementation.

Generally, the employment of a key management scheme in a communications system introduces certain system overheads. The main ones are: (i) the required (protected) secret storage at the receiver; (ii) the required public storage at the receiver; (iii) the processing overhead at the receiver; (iv) the communications overhead. Different trade-offs between the overheads are possible. Some of these overheads are discussed in [72] and [102], for example, but the objective of this letter is towards different trade-offs.

A framework for minimization of the secret storage based on the secret-public storage trade-off has been reported in [17]. This section, according to [15] considers an alternative approach for minimizing the secret storage in certain key management schemes employing an appropriate trade-off between the required secret storage, the public storage and the processing overheads. The proposal employs a technique for one-way mapping of the secret bits whose security originates from the uncertainty associated with decoding a binary block code after transmission over a binary erasure channel.

A number of recently proposed key management schemes for broadcast encryption (SD [99], LSD [83], and reconfigurable key management schemes [19]-[18]) require a significant amount of secret data to be stored at a receiver. This constraint appears to be inappropriate in certain scenarios implying the need for small secret storage overhead. Consequently assigning multiple roles to the same secret bits via one-way mapping is required. A motivation for this work is to propose certain key management schemes with minimized secret storage employing a low complexity mapping technique. As a result, an implementation based only on very simple arithmetic and logical operations like mod2 additions and look-up table operations is possible. An additional motivation for this work is to yield appropriate techniques required for reconfigurable key management (RKM) [19]-[18], and to support the generic framework of assigning different roles to the secret key bits.

6.2.2. A Framework for Minimization of the Secret Storage. Following [17], this section provides a general framework for minimization of the required secret storage: This framework is based on the employment of a secret seed and its mapping into the required KEKs. Particularly note that this approach is very different from the one which employs encryption of KEKs by a master secret key and storing the encrypted forms of the KEKs in a public storage. The considered approach is not based on encryption of KEKs. Accordingly it does not require exposition of any transformation of KEKs in a public storage and its security does not depend on the security of any encryption technique, implying a number of related advantages.

Suppose that a key management scheme with non-updatable keys requires that the following I KEKs are stored in a secret (protected) storage at a receiver: $KEK_1, KEK_2, \ldots, KEK_I$.

Let: $f(\cdot)$ and $g(\cdot)$ be functions which fulfill certain requirements; S be arbitrary data; (R_i, Q_i), $i = 1, 2, \ldots, I$, be certain data such that the following holds:

$$KEK_i = g(f(S, R_i), Q_i), \quad i = 1, 2, \ldots, I. \tag{6.2}$$

Assuming that the composition of $f(\cdot)$ and $g(\cdot)$ yields the appropriate one-wayness, instead of keeping all $KEKs$ in a protected storage, the following strategy can be employed:
–keep S in a protected storage which provides the data secrecy;
–keep (R_i, Q_i), $i = 1, 2, \dots, I$, in a public storage;
–when required, calculate any KEK_i, employing (6.2).

The main architectural components for the implementation of the above method are the following: (i) temper-proof block T for the seed S and the mappings $f(\cdot)$ and $g(\cdot)$; and (ii) public storage for the non-secret data R_i, Q_i, $i = 1, 2, \dots, I$. The block T has two inputs and one output. The role of T is to yield KEK_i as its output for the given input pair (R_i, Q_i) preserving the secrecy of the secret seed S in a computationally secure manner.

6.2.3. Dedicated One-Way Mapping for Secret-to-Public Storage Exchange. Let S be a binary k-dimensional vector, and let K_i, $i = 1, 2, \dots, I$, be I different binary n-dimensional vectors, $k \geqslant n$. A goal is to propose certain methods for mapping the vector S into any of the vectors K_i, $i = 1, 2, \dots, I$, under the following conditions:

- it is computationally infeasible to recover S knowing all K_i, $i = 1, 2, \dots, I$, and all the related public information;
- the mapping of S into any K_i should *not* include public key cryptography;
- the mapping of S into any K_i should be a low complexity one and include only mod2 additions and simple logical operations.

Preliminaries For any input vector, a communication channel with erasures yields a vector of the same dimension in which a certain fraction of the symbols is no longer known due to erasures. Accordingly, the output of a binary erasure channel (BEC) yields exact information on the the input bits in a certain subset and no information on the erased e bits outside this subset.

The list decoding problem for a binary error-correcting code consists of outputting the list of all codewords that lie within a certain Hamming distance of the received word. The decoding is considered successful if and only if the correct codeword is included in the list.

Mapping Specification Let C be an (m, k) block code which maps k information bits into a codeword of length m.

For $i = 1, 2, \dots, I$, let the public information associated to each K_i be in the form of a pair of binary vectors (R_i, E_i) of dimensions n and m, respectively, where:

- R_i is selected randomly, and
- for a given S, the vector E_i is selected so that the non erased bits of the codeword generated by $\phi(S, R_i)$ yield K_i, where $\phi(\cdot)$ is a suitable nonlinear function.

Mapping E (where E corresponds to the erasure channel) is defined as follows:

(1) employing C, perform encoding of the vector $\phi(S, R_i)$ into the codeword C_{S,R_i} which is a binary m-dimensional vector;
(2) employing the vector E_i erase $e = m - n$ bits in the codeword C_{S,R_i};
(3) define K_i as the consecutive sequence of nonerased bits.

Proposition 6.2. *Assuming sufficiently large values for the parameters m and $e = m - n > m/2$, Mapping E can map any S into K_i, with probability close to 1.*

Proof. Given an m-dimensional binary vector, let $P(m,n)$ be the probability that one random n-tuple can be embedded into the given vector. Then we have the following (see [82], as well):

$$P(m,n) = \sum_{l=0}^{m-n} 2^{-(n+l)} \binom{n+l-1}{l} = 1 - 2^{-m} \sum_{l=0}^{n-1} \binom{m}{l}$$
$$\geqslant 1 - 2^{-m(1-h((n-1)/m))}$$

where $h(\cdot)$ is the binary entropy function. □

Proposition 6.3. *When $k > 2n + \delta$, $\delta > 0$, the complexity of recovering any K_i is proportional to 2^n and the complexity of recovering S is proportional to $2^{n+\delta}$, given all other vectors K_j, $j \neq i$, $j = 1, 2, \dots, I$, and all public information.*

Proof. Any unknown K_i can be recovered via simple guessing which has complexity proportional to 2^n, or via recovering of S and employment of Mapping E for obtaining K_i. The complexity of recovering S is determined by the following consideration. The capacity $C(\epsilon)$ of a BEC with erasure probability $\epsilon = e/m$ is given by $C(\epsilon) = 1 - e/m = n/m$. The code rate k/m is always greater than the capacity for $k > n$, implying that unique decoding is not feasible. Then, the best possible is list decoding which is equivalent to the following algorithm:

(i) fix certain $k^* \leqslant k$ bits so that the code rate of the modified code is below the capacity, i.e. $(k - k^*)/m \leqslant C(\epsilon)$;

(ii) generate a list for 2^{k^*} decodings.

When $k > 2n + \delta$, the list dimension becomes greater than $2^{n+\delta}$ implying that it is greater than the number of hypotheses required by the direct exhaustive search for any K_i. Consequently, the simple guessing approach provides the lower bound on the recovering complexity for any K_i. □

Proposition 6.4. *The implementation complexity is proportional to nk.*

Proof. In Mapping E, the implementation complexity mainly depends on step 1, i.e. encoding of S into the n codeword coordinates of C_{S,R_i} corresponding to the bits which are not erased. This requires a number of mod2 additions upperbounded by nk. □

6.2.4. SD and LSD Key Managements with Minimized Secret Storage. This section proposes and discusses modified SD [99] and LSD [83] key management schemes based on the proposed Mapping E. For further considerations we assume that the employment of the original SD and LSD schemes requires that each receiver stores in secret storage a sequence of n-dimensional binary vectors K_i, $i = 1, 2, \dots, I$. Assuming that $I^{(SD)}$ and $I^{(LSD)}$ denote the values of I related to SD and basic LSD, respectively, we have (see [99] and [83]):

$$I^{(SD)} = \frac{1}{2}[(\log_2 N)^2 + \log_2 N] + 1,$$

$$I^{(LSD)} = (\log_2 N)^{3/2} + 1,$$

where N denotes the number of users in the system.

Modified SD and LSD: We propose modified versions of SD and LSD as follows:

- In the modified SD/LSD, each receiver keeps in the secret storage the seed S in the form of a $3n$-dimensional binary vector, and employing Mapping E evaluates any of the required $I^{(SD)}/I^{(LSD)}$ vectors; all other issues are identical to that of the original SD/LSD.

Recall that the employment of Mapping E requires that certain information is stored in public storage and that certain processing is employed. Also note that the proposed modification does not affect the communication overhead (i.e., it is same as in the original schemes).

Security of Modified SD and LSD: The original SD and LSD schemes are only computationally secure ones, and the complexity of step by step straightforward recovering of all employed KEKs is upperbounded by $I^{(SD)}2^n$ and $I^{(LSD)}2^n$, respectively. On the other hand, the only security difference between the original and the modified schemes is that an attempt to recover the employed KEKs in the modified schemes could be done either directly employing the same complexity as in the original schemes, or via recovering S. Note that when $k = 3n$, Proposition 2 implies that the complexity of recovering S is proportional to 2^{2n}. Hence it is always greater than the upperbound of step-by-step recovering of all the KEKs because $2^n \gg I^{(SD)} > I^{(LSD)}$. Accordingly, Proposition 2, the nature of the modification, and the selected dimension of S directly imply that Mapping E preserves the security of the original schemes.

Comparison of Modified and Original SD and LSD: According to the results reported in [99] and [83], the nature of the modifications and the characteristics of Mapping E, Tables 6 and 7 provide a summary comparison of the main overheads regarding the modified and original SD and LSD, respectively, assuming a group of N users and that certain R users should be revoked.

TABLE 6. Comparison of original and modified SD schemes.

	proposed SD (Mapping E)	original SD [99]
secret storage overhead at receiver	$O(1)$	$O((\log_2 N)^2)$
public storage overhead at receiver	$O((\log_2 N)^2)$	–
processing overhead at receiver	$nk + O(\log_2 N)$	$O(\log_2 N)$
communications overhead	$O(R)$	$O(R)$

TABLE 7. Comparison of original and modified LSD schemes.

	proposed LSD (Mapping E)	original LSD [83]
secret storage overhead at receiver	$O(1)$	$O((\log_2 N)^{3/2})$
public storage overhead at receiver	$O((\log_2 N)^{3/2})$	–
processing overhead at receiver	$nk\ +\ O(\log_2 N)$	$O(\log_2 N)$
communications overhead	$O(R)$	$O(R)$

6.2.5. Concluding Remarks. A generic framework and a particular mapping technique have been pointed out for minimization of the required secret storage in certain key management schemes for broadcast encryption. The proposed modified SD and LSD based key management schemes require secret storage overhead independent of the parameter N of only $O(1)$, public storage overheads $O((\log_2 N)^2)$ and $O((\log_2 N)^{3/2})$, respectively, and a low additional processing overhead. For example, when the number of users is about one million, i.e. $N = 2^{20}$, and assuming that each KEK consists of 100 bits, the modified SD/LSD schemes require only 300 secret bits, while the original SD and LSD schemes require approximately 20000 and 9000 secret bits, respectively.

Finally note that in [19], with an additional refinement in [18], as well as discussed in Section 6.1, RKM has been proposed as an advanced technique for broadcast encryption (appropriate for high dynamic scenarios). RKM assumes that the secret bits can play different roles, and that the employed volume of secret information is as small as possible. Accordingly, the proposed method for minimizing the secret storage is also of direct interest for RKM.

7. Concluding Discussion

The considerations in this chapter could also be employed as particular guidelines for future research directions because they point out and provide background for further work regarding the following active and important topics of cryptology:

- security evaluation of stream ciphers and authentication protocols employing decoding techniques and algorithms for the LPN problem;
- design of advanced encryption techniques based on employment of coding theory and randomness moving towards provable secure encryption in information-theoretic sense;
- security evaluation and design of certain key management schemes based on the broadcast encryption paradigm.

Accordingly, note the following issues which support the above claims.

Significance of solving consistent and overdefined systems of algebraic equations which are true with certain probability (algebraic equations corrupted by noise) is a well recognized mathematical topic of the LPN and LPN-related problems, and of direct interest for decoding of linear block codes and for security evaluation of certain cryptographic techniques.

Coding theory has accumulated a huge amount of results of potential interest for design of advanced encryption schemes based on employment of pseudorandomness, randomness and dedicated coding. This approach also provides a framework for employment of certain results of information theory for achieving security which could be provable in information-theoretic sense.

Particular open research problems related to the topics discussed in this chapter include the following ones:

- Solving probabilistic systems of algebraic equations over GF(2);
- developing joint homophonic and error-correction coding schemes or the wire-tap channel coding schemes dedicated to particular encryption and authentication paradigms;
- developing graphs dedicated to certain subset-covering problems.

The results of the above addressed research problems could be employed for developing advanced cryptographic techniques regarding the following:

- Security evaluation of certain cryptographic primitives for encryption and authentication;
- developing of advanced encryption and authentication techniques which provide low implementation complexity and high level of provable security;
- developing of advanced key management schemes which provide low overhead to the system which employs these cryptographic keys.

Through an in details consideration of selected techniques for security evaluation and design of certain cryptographic primitives this chapter provides a background for further research activities as well as textbooks-like introduction of important topics of cryptology.

Particularly, this chapter points out to a number of mathematical techniques for addressing different problems of developing basic components for cyber-security issues.

Acknowledgement. This work is partially supported by the Serbian Ministry of Education and Science, the projects ON174008, "Advanced Techniques of Cryptology, Image Processing and Computational Topology for Information Security", and III44006, "Development of new information and communication technologies, based on advanced mathematical methods, with applications in medicine, telecommunications, power systems, protection of national heritage and education".

References

[1] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *State Recovery of Grain-v1 Employing Normality Order of the Filter Function*, IET Information Security 6, no. 2, pp. 55–64, June 2012.

[2] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *Internal state recovery of keystream generator LILI-128 based on a novel weakness of the employed Boolean function*, Information Processing Letters 112, pp. 805–810, Nov. 2012.

[3] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *Generic Cryptographic Weakness of k-normal Boolean Functions in Certain Stream Ciphers and Cryptanalysis of Grain-128*, Periodica Mathematica Hungarica (Springer), vol. 65, pp. 205–227, Dec. 2012.

[4] M. J. Mihaljević and H. Imai, *An approach for stream ciphers design based on joint computing over random and secret data*, Computing 85, no. 1–2, pp. 153–168, June 2009.

[5] M. J. Mihaljević, M. Fossorier and H. Imai, *Security Evaluation of Certain Broadcast Encryption Schemes Employing a Generalized Time-Memory-Data Trade-Off*, IEEE Communications Letters 11, no. 12, pp. 988–990, Dec. 2007.

[6] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Modeling Block Encoding Approaches for Fast Correlation Attack*, IEEE Transactions on Information Theory 53, no. 12, pp. 4728–4737, Dec. 2007.

[7] M. J. Mihaljević, *Generic framework for secure Yuen 2000 quantum-encryption employing the wire-tap channel approach*, Physical Review A 75, no. 5, pp. 052334-1-5, May 2007.

[8] M. J. Mihaljević, M. P. C. Fossorier and H.Imai, *Birthday Paradox Based Security Analysis of Certain Broadcast Encryption Schemes*, IEICE Trans. Fundamentals E90-A, pp. 1248–1251, June 2007.

[9] M. P. C. Fossorier, M. J. Mihaljević, H. Imai, Y. Cui and K. Matsuura, *An Algorithm for Solving the LPN Problem and its Application to Security Evaluation of the HB Protocols for RFID Authentication*, Lect. Notes Comput. Sci. 4329, pp. 48–62, Dec. 2006.

[10] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *Security Weaknesses of Certain Broadcast Encryption Schemes*, DRMtics2005, Lect. Notes Comput. Sci. 3919, pp. 228–245, 2006.

[11] M. J. Mihaljević, M. Fossorier and H. Imai, *A Novel Broadcast Encryption Based on Time-Bound Cryptographic Keys*, DRMtics2005, Lect. Notes Comput. Sci. 3919, pp. 258–276, July 2006.

[12] J. Wang, M. J. Mihaljević, L. Harn, and H. Imai, *A Hierarchical Key Management Approach for Secure Multicast*, ARCS2006, Lect. Notes Comput. Sci. 3894, pp. 422–434, March 2006.

[13] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *A General Formulation of Algebraic and Fast Correlation Attacks Based on Dedicated Sample Decimation*, AAECC2006, Lect. Notes Comput. Sci. 3857, pp. 203–214, 2006.

[14] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *Cryptanalysis of keystream generator by decimated sample based algebraic and fast correlation attacks*, INDOCRYPT2005, Lect. Notes Comput. Sci. 3797, pp. 155–168, Dec. 2005.

[15] M. J. Mihaljević, M. Fossorier and H. Imai, *Key management with minimized secret storage employing an erasure channel approach*, IEEE Communications Letters 9, no. 8, pp. 741–743, Aug. 2005.

[16] M. J. Mihaljević and H. Imai, *The decimated sample based improved algebraic attacks on nonlinear filters*, SCN 2004, Lect. Notes Comput. Sci. 3352, pp. 310–323, Jan. 2005.

[17] M. J. Mihaljević, M. Fossorier and H. Imai, *Secret-public storage trade-off for broadcast encryption key management*, ICICS 2004, Lect. Notes Comput. Sci. 3269, pp. 375–387, October 2004.

[18] M. J. Mihaljević, *Reconfigurable key management for broadcast encryption*, IEEE Communications Letters 8, pp. 440–442, July 2004.

[19] M. J. Mihaljević, *Key management schemes for stateless receivers based on time varying heterogeneous logical key hierarchy*, ASIACRYPT 2003, Lect. Notes Comput. Sci. 2894, pp. 137–154, Dec. 2003.

[20] M. J. Mihaljević, *On vulnerabilities and improvements of Fast Encryption Algorithm for Multimedia FEA-M*, IEEE Trans. Cons. Electr. 49, no. 4, pp. 1199–1207, Nov. 2003.

[21] M. Mihaljević, *Broadcast encryption schemes based on the sectioned key tree*, ICICS2003, Lect. Notes Comput. Sci. 2836, pp. 158–169, Oct. 2003.

[22] P. Camion, M. J. Mihaljević and H. Imai, *Two alerts for design of certain stream ciphers: Trapped LFSR and weak resilient function over GF(q)*, SAC2002, Lect. Notes Comput. Sci. 2595, pp. 196–213, Feb. 2003.

[23] L. Michael, M.J. Mihaljević, S. Haruyama and R. Kohno, *Security issues for software defined radio: Design of a secure download system*, IEICE Transactions on Communications E85-B, pp. 2588–2600, Dec. 2002.

[24] M. J. Mihaljević and R. Kohno, *Cryptanalysis of fast encryption algorithm for multimedia FEA-M*, IEEE Communications Letters 6, pp. 382–384, Sept. 2002.

[25] L. Michael, M. J. Mihaljević, S. Haruyama and R. Kohno, *A framework for secure download for software defined radio*, IEEE Communications Magazine 40, no. 7, pp. 88–96, July 2002.

[26] M. J. Mihaljević, M.P.C. Fossorier and H. Imai, *Fast Correlation Attack Algorithm with List Decoding and an Application*, FSE2001, Lect. Notes Comput. Sci. 2355, pp. 196–210, 2002.

[27] M. J. Mihaljević and H. Imai, *Cryptanalysis of TOYOCRYPT-HS1 stream cipher*, IEICE Transactions on Fundamentals E85-A, pp. 66–73, Jan. 2002.

[28] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *On decoding techniques for cryptanalysis of certain encryption algorithms*, IEICE Transactions on Fundamentals E84-A, pp. 919–930, Apr. 2001.

[29] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *A low-complexity and high-performance algorithm for the fast correlation attack*, FSE2000, Lect. Notes Comput. Sci. 1978, pp. 196–212, 2001.

[30] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *An algorithm for cryptanalysis of certain keystream generators suitable for high-speed software and hardware implementations*, IEICE Transactions on Fundamentals E84-A, pp. 311–318, Jan. 2001.

[31] M. J. Mihaljević and J. Golić, *A method for convergence analysis of iterative probabilistic decoding*, IEEE Transactions on Information Theory 46, pp. 2206–2211, Sept. 2000.

[32] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Reduced complexity iterative decoding of Low Density Parity Check codes based on Belief Propagation*, IEEE Transactions on Communications 47, pp. 673–680, May 1999.

[33] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Critical noise for convergence of iterative probabilistic decoding with belief propagation in cryptographic applications*, Applied Algebra, Algebraic Algorithms and Error Correcting Codes-AAECC 13, Lect. Notes Comput. Sci. 1719, pp. 282–293, 1999.

[34] M. J. Mihaljević and H. Imai, *A family of fast keystream generators based on programmable linear cellular automata over GF(q) and time variant table*, IEICE Transactions on Fundamentals E82-A, pp. 32–39, Jan. 1999.

[35] M. Mihaljević, Y. Zheng and H. Imai, *A family of fast dedicated one-way hash functions based on linear cellular automata over GF(q)*, IEICE Transactions on Fundamentals E82-A, pp. 40–47, Jan. 1999.

[36] M. Mihaljević, Y. Zheng and H. Imai, *A cellular automaton based fast one-way hash function suitable for hardware implementation*, PKC'98, Lect. Notes Comput. Sci. 1431, pp. 217–233, 1998.

[37] M. Mihaljević, *An improved key stream generator based on the programmable cellular automata*, ICICS'97, Lect. Notes Comput. Sci. 1334, pp. 181–191, 1997.

[38] M. J. Mihaljević, *Security examination of a cellular automata based pseudorandom bit generator using an algebraic replica approach*, AAECC12, Lect. Notes Comput. Sci. 1255, pp. 250–262, 1997.

[39] M. J. Mihaljević, *An iterative probabilistic decoding algorithm for binary linear block codes beyond the half minimum distance*, AAECC12, Lect. Notes Comput. Sci. 1255, pp. 237–249, 1997.

[40] M.J. Mihaljević, *A faster cryptanalysis of the self-shrinking generator*, ACISP'96, Lect. Notes Comput. Sci. 1172, pp. 182–188, 1996.

[41] M. J. Mihaljević, *A sequence comparison approach for decoding of general binary block codes after the binary symmetric channel with synchronization errors*, Zeitschrift fur Angewandte Mathematik und Mechanik-ZAMM 76, pp. 479–481, 1996.

[42] M. J. Mihaljević, *A correlation attack on the binary sequence generators with time-varying output function*, ASIACRYPT'94, Lect. Notes Comput. Sci. 917, pp. 67–79, 1995.

[43] M. J. Mihaljević, *On message protection in crypto systems modeled as the generalized wire-tap channel II*, Lect. Notes Comput. Sci. 829, pp. 13–24, 1994.

[44] M. J. Mihaljević, *An approach to the initial state reconstruction of a clock-controlled shift register based on a novel distance measure*, AUSCRYPT'92, Lect. Notes Comput. Sci. 718, pp. 349–356, 1993.

[45] M. J. Mihaljević and J. Golić, *Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence*, EUROCRYPT'92, Lect. Notes Comput. Sci. 658, pp. 124–137, 1993. (reprinted in Lect. Notes Comput. Sci. 1440, 1999)

[46] M. J. Mihaljević and J.Dj. Golić, *A comparison of cryptanalytic principles based on iterative error-correction*, Advances in Cryptology-EUROCRYPT '91, Lect. Notes Comput. Sci. 547, pp. 527–531, 1991.

[47] J. Golić and M. Mihaljević, *A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance*, J. of Cryptology 3, pp. 201–212, 1991.

[48] J. Golić and M. J. Mihaljević, *A noisy clock-controlled shift register cryptanalysis concept based on sequence comparison approach*, EUROCRYPT'90, Lect. Notes Comput. Sci. 473, pp. 487–491, 1991. (reprinted in Lect. Notes Comput. Sci. 1440, 1999)

[49] M. J. Mihaljević and J.Dj. Golić, *A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence*, Advances in Cryptology-AUSCRYPT '90, Lect. Notes Comput. Sci. 453, pp. 165–175, 1990.

[50] J. Golić and M.J. Mihaljević, *Minimal linear equivalent analysis of a variable memory binary sequence generator*, IEEE Transactions on Information Theory 36, pp. 190–192, Jan. 1990.

[51] M. J. Mihaljević, *A Framework for Stream Ciphers Based on Pseudorandomness, Randomness and Error-Correcting Coding*, in Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes, Editors B. Preneel, S. Dodunekov, V. Rijmen and S. Nikova, Vol. 23 in the NATO Science for Peace and Security Series-D: Information and Communication Security, pp. 117–139, IOS Press, Amsterdam, The Netherlands, June 2009.

[52] M.J. Mihaljević, *Decimation Based Correlation and Algebraic Attacks and Design of Boolean Functions*, in Boolean Functions in Cryptology and Information Security, Editors B. Preneel and O. A. Logachev, Vol. 18 in the Series Information and Communication Security, pp. 183–199, IOS Press, Amsterdam, The Netherlands, July 2008.

[53] M. Matsui and M. J. Mihaljević, *Security Evaluation Techniques for Symmetric Cryptography*, Chapter in Information Security Handbook, Editors H. Imai and E. Okamoto, IEICE, Ohmusha Ltd., Tokyo, Japan, pp. 145–160, 2004. (ISBN 4–274–07980–5)

[54] *Japan Patent JP 4863283*: M. J. Mihaljević and H. Watanabe, *Authentication System Using Light-Weight Authentication Protocol*, November 18, 2011.

[55] *United States Patent US 8023649*: M. J. Mihaljević and J. Abe, *Method and apparatus for cellular automata based generation of pseudorandom sequences with controllable period*, 21 pages, September 2011.

[56] *China Patent CN 1698306*: M. J. Mihaljevicć and J. Abe, *Data processing method*, 51 pages, October 2010.

[57] *Japan Patent JP 4432350*: M. J. Mihaljevic and J. Abe, *Data Processing Method, Program Thereof, Data Processor, and Receiver*, 35 pages, March 2010.

[58] *United States Patent US 7502941*: L. Michael and M. J. Mihaljević, *Wireless data communication method and apparatus for software download system*, 36 pages. March 2009.

[59] *Japan Patent JP 3918578*: R. Morelos-Zaragoza and M. J. Mihaljević, *Method and apparatus for loss correction and limited reception in streaming data, and data communication apparatus*, May 2007.

[60] M.J. Mihaljevic, *An Approach for Light-Weight Encryption Employing Dedicated Coding*, to appear in IEEE GLOBECOM 2012 Proceedings, 7 pages (Anaheim CA, USA, 03–07 Dec. 2012).

[61] M. J. Mihaljević and H. Imai, *A Security Evaluation of Certain Stream Ciphers which Involve Randomness and Coding, 2010 IEEE Int. Symp. on Inform. Theory and its Appl.-ISITA 2010*, Taichung, Taiwan, Oct. 17–20, 2010, Proceedings, pp. 789–794, IEEE, 2010.

[62] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Decimation Based Fast Correlation Attack, 2007 IEEE Int. Symp. Inform. Theory-ISIT'2007*, Nice, France, June 24–29, 2007, Proceedings, pp. 456–460 (ISBN 1–4244–1429–6).

[63] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *A Unified Analysis for the Fast Correlation Attack, 2005 IEEE Int. Symp. Inform. Theory-ISIT'2005*, Adelaide, Australia, Sept. 2005, Proceedings, pp. 2012–2015 (ISBN 0–7803–9151–9).

[64] Advanced Access Content System (AACS): Introduction and Common Cryptographic Elements, Feb. 2006. Available at http://www.aacsla.com.

[65] B. Applebaum, D. Cash, C. Peikert and A. Sahai, *Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems*, CRYPTO 2009, Lect. Notes Comput. Sci. 5677, pp. 595–618, Aug. 2009.

[66] J.-P. Aumasson, M. Finiasz, W. Meier, S. Vaudenay, *STCHo: A Hardware-Oriented Trapdoor Cipher*, ACISP 2007, Lect. Notes Comput. Sci. 4586, pp. 184–199, 2007.

[67] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inform. Theory IT-20, pp. 284–287, March 1974.

[68] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the Inherent Intractability of Certain Coding Problems*, IEEE Trans. Info. Theory 24, pp. 384–386, 1978.

[69] A. Biryukov and A. Shamir, *Cryptanalytic time/memory/data tradeoffs for stream ciphers*, ASIACRYPT 2000, Lect. Notes Comput. Sci. 1976, pp. 1–13, 2000.

[70] A. Blum, M. Furst, M. Kearns and R. Lipton, *Cryptographic Primitives Based on Hard Learning Problems*, CRYPTO 1993, Lect. Notes Comput. Sci. 773, pp. 278291, 1994.

[71] A. Blum, A. Kalai and H. Wasserman, *Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model*, J. of the ACM 50, no. 4, pp. 506–519, July 2003.

[72] R. Canetti, T. Malkin and K. Nissim, *Efficient communication-storage tradeoffs for multicast encryption*, EUROCRYPT'99, Lect. Notes Comput. Sci. 1592, pp. 459–474, 1999.

[73] P. Chose, A. Joux and M. Mitton, *Fast Correlation Attacks: An Algorithmic Point of View*, EUROCRYPT2002, Lect. Notes Comput. Sci. 2332, pp. 209–221, 2002.

[74] A. Clark, J. Dj. Golić, and E. Dawson, *A comparison of fast correlation attacks*, Fast Software Encryption-FSE'96, Lect. Notes Comput. Sci. 1039, pp. 145–157, 1996.

[75] N. T. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, EUROCRYPT'2003, Lect. Notes Comput. Sci. 2656, pp. 345–359, 2003.

[76] N. T. Courtois, *Fast algebraic attacks on stream ciphers with linear feedback*, CRYPTO'2003, Lect. Notes Comput. Sci. 2729, pp. 176–194, 2003.

[77] Ecrypt: Network of Excellence in Cryptology, EU FP6 Project, 2004–2008. http://www.ecrypt.eu.org.

[78] R. M. Fano, *Transmission of Information: A Statistical Theory of Communication.* New York: MIT, 1961.

[79] R. G. Gallager, *Low-density parity-check codes*, IRE Trans. Inform. Theory IT-8, pp.21–28, Jan. 1962.

[80] H. Gilbert, M. J. B. Robshaw and Y. Seurin, *$HB^{\#}$: Increasing the Security and Efficiency of HB^{+}*, EUROCRYPT2008, Lect. Notes Comput. Sci. 4965, pp. 361–378, 2008.

[81] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, *How to Encrypt with the LPN Problem*, ICALP 2008, Part II, Lect. Notes Comput. Sci. 5126, pp. 679–690, 2008.

[82] J.Dj. Golić and L. O'Connor, *Embedding and probabilistic correlation attacks on clock-controilled shift registers*, EUROCRYPT'94, Lect. Notes Comput. Sci. 950, pp. 230–243, 1995.

[83] D. Halevy and A. Shamir, *The LCD broadcast encryption scheme*, CRYPTO 2002, Lect. Notes Comput. Sci. 2442, pp. 47–60, 2002.

[84] M. E. Hellman, *A cryptanalytic time-memory trade-off*, IEEE Transactions on Information Theory 26, pp. 401–406, July 1980.

[85] ISO/IEC Standard 18033–3:2005.

[86] H. N. Jendal, Y. J. B. Kuhn, and J. L. Massey, *An information-theoretic treatment of homophonic substitution*, EUROCRYPT'89, Lect. Notes Comput. Sci. 434, pp. 382–394, 1990.

[87] T. Johansson and F. Jonsson, *Improved fast correlation attacks on stream ciphers via convolutional codes*, Advances in Cryptology-EUROCRYPT'99, Lect. Notes Comput. Sci. 1592, pp. 347–362, 1999.

[88] T. Johansson and F. Jonsson, *Fast correlation attacks based on turbo code techniques*, Advances in Cryptology-CRYPTO'99, Lect. Notes Comput. Sci. 1666, pp. 181–197, 1999.

[89] O. Kara and I. Erguler, I, *A New Approach to Keystream Based Cryptosystems*, SASC 2008, Workshop Record, pp. 205–221, 2008.

[90] J. Katz, *Efficient Cryptographic Protocols Based on the Hardness of Learning Parity with Noise*, Cryptography and Coding 2007, Lect. Notes Comput. Sci. 4887, pp. 1–15, 2007.

[91] V. A. Kovalevskij, *The problem of character recognition from the point of view of mathematical statistics*, in Character Readers and Pattern Recognition, pp. 3–30, New York: Spartan, 1968. Russian edition 1965.

[92] E. Levieil and P.-A. Fouque, *An Improved LPN Algorithm*, SCN 2006, Lect. Notes Comput. Sci. 4116, pp. 348–359, 2006.

[93] J. Lotspiech, S. Nusser and F. Prestoni, *Broadcast encryption's bright future*, IEEE Computer 35, pp. 57–63, Aug. 2002.

[94] J. Lotspiech, V. Mirles, D. Naor and I. Nin, *Coincidence-free media key block for content protection for recordable media*, United States Patent 6,883,097, filed May 2000.

[95] J. Massey, *Some Applications of Source Coding in Cryptography*, European Transactions on Telecommunications 5, pp. 421–429, July-August 1994.

[96] R. J. McEliece, *A public key cryptosystem based on algebraic coding theory*, DSN progress report 42–44, pp. 114–116, 1978.

[97] W. Meier and O. Staffelbach, *Fast Correlation Attacks on Certain Stream Ciphers*, J. of Cryptology 1, pp. 159–176, 1989.

[98] A. Menezes, P. C. van Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Roton, 1997.

[99] D. Naor, M. Naor and J. Lotspiech, *Revocation and tracing schemes for stateless receivers*, CRYPTO 2001, Lect. Notes Comput. Sci. 2139, pp. 41–62, Aug. 2001.

[100] D. Naor and M. Naor, *Protecting cryptographic keys: The trace-and-revoke approach*, IEEE Computer 36, pp. 47–53, July 2003.

[101] B. Pinkas, *Efficient state updates for key management*, Proc. IEEE 92, pp. 910–917, June 2004.

[102] R. Poovendran and C. Bernstein, *Design of secure multicast key management schemes with communication budget constraint*, IEEE Commun. Lett. 6, pp. 108–110, March 2002.

[103] R. Rivest and T. Sherman, *Randomized Encryption Techniques*, Advances in Cryptology: Proceedings of CRYPTO '82, Plemum, New Yourk, pp. 145–163, 1983.

[104] C. E. Shannon, *Communication theory of secrecy systems*, Bell Systems Technical J. 28, pp. 656–715, Oct. 1949.

[105] N. J. A. Sloane, *Error-correcting codes and cryptography–part I*, Cryptologia 6, pp. 128–153, 1982.

[106] T. Siegenthaler, *Decrypting a Class of Stream Ciphers Using Ciphertext Only*, IEEE Transactions on Compututers C-34, pp. 81–85, 1985.

[107] C.K. Wong, M. Gouda, and S.S. Lam, *Secure group communications using key graphs*, IEEE/ACM Trans. Networking 8, pp. 16–31, Feb. 2000.

[108] A. D. Wyner, *The wire-tap channel*, Bell Systems Technical J. 54, pp. 1355–1387, 1975.

[109] K. Zeng and M. Huang, *On the linear syndrome method in cryptanalysis*, Advances in Cryptology-CRYPTO '88, Lect. Notes Comput. Sci. 403, pp. 469–478, 1990.

[110] W. T. Zhu, *Optimizing the tree structure in secure multicast key management*, IEEE Commun. Lett. 9, (5), pp. 477–479, May 2005.