COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **40**, 311–333 (1987)

Length Estimators for Digitized Contours

LEO DORST

*Department of Applied Physics, Delft University of Technology, Delft, The Netherlands**

AND

ARNOLD W. M. SMEULDERS

Department of Medical Informatics, Erasmus University, Dr., Molenwaterplein, Rotterdam, The Netherlands

Received July 15, 1986; accepted March 5, 1987

The estimation of the length of a continuous curve from discrete data is considered. For ideal straight chaincode strings, optimal estimators are given. Comparisons are performed with known methods and recommendations given. A sampling density vs. accuracy trade-off theorem is presented. The applicability to nonstraight strings is discussed. For curves that may be considered to be composed of circular arcs good length estimators are found.

1. INTRODUCTION

The determination of the length of a discrete contour is frequently required in image analysis. Many methods have been proposed [1–6]. It is the purpose of this paper to review these methods within a proper framework, to evaluate them and to give recommendations for use on computational, practical, and theoretical grounds.

Discrete measurement of properties of continuous curves is a nontrivial task. Some preliminary observations can be made concerning the basics of measurement methods.

First and foremost, it should be realized that the length measurement of a discrete contour is essentially an estimation problem. The interest is **not** in the length of the digital contour but in the length of the original, predigitized line. We therefore conceive of the set of discrete data points as a *digitized* line rather than a *digital* line. A digitized line is the digitization of a particular continuous original, whereas a digital line is a subset of the pixels in the two-dimensional grid. The difference between digital and digitized may seem very subtle and irrelevant but in fact it causes a considerable difference in the accuracy of the length measurement, as will appear in the paper. In the digitization, the exact original, continuous line is lost. The only thing left is a discrete approximation of it. As the exact copy of the continuous line is lost, the length of the original line can only be *estimated* rather than exactly known. In this paper, we will review to which degree known length measurement methods succeed in providing an accurate estimate for the length of the original line.

To illustrate the difference between digital and digitized lines somewhat more, consider a discrete line the way it is usually drawn, as in Fig. 1b. The discrete line—which is in fact a set of grid *points*!—is indicated by a broken line interconnecting the pixels of the discrete line. In digital image analysis, this broken line is

*Now with Philips Labs, 345 Scarborough Rd., Briarcliff Manor, NY 10510.

0734-189X/87 $3.00

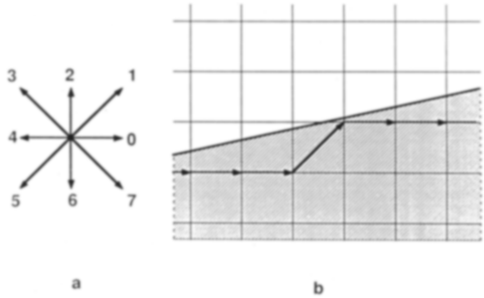

FIG. 1. (a) The 8-connected chaincode scheme; (b) a line in the standard situation.

not an actually existing entity but only a way to indicate which pixels belong to the discrete line. Still, the length of this broken line is an old and widely spread method to find the length of a discrete line in image processing [1]. But, as the broken line is not the original one, it is not truly relevant in finding a length measure for the predigitized original. In this paper other, more accurate, length estimates will be evaluated, some of which are computationally equally efficient as the above method by Freeman [1]. Also, resampling the broken line taking 5 samples in each straight connection and 7 in each diagonal connection, as was proposed in [17] to reduce the anisotropy of the grid, is not a proper way to conceive of a digitized figure.

Another important point to realize is that it is impossible to reconstruct the continuous contour from the discrete data. This means that many possible contours correspond to a specific discrete realization, all with different lengths. Some of these contours are very curvy. To develop practical length estimators, some reasonable assumption about the original contour has to be made. The simplest assumption is that it consists of piecewise straight parts, and to gauge the estimators accordingly. Therefore, in this paper we will develop and evaluate length estimators for *ideal straight strings*, the discrete images of continuous straight line segments. Deviations from the ideal case, such as noisy straight lines, will be only marginally be considered here. Dealing with noise is a problem-specific task, which should proceed from the estimation treated in this paper. We will, however, investigate ideal nonstraight curves, particularly chains of circular arcs. Third, the estimated length should by some criterion correspond to the original length. The criterion by which this match is to be judged should be chosen sensibly: on the one hand, it should be mathematically tractable; on the other hand, it should correspond to something that is meaningful in practice. We will use the mean square error (MSE) as a criterion to evaluate the estimators.

The recommendations of this paper are based on theoretical and practical considerations. Experiments to determine the best estimators for use in practice are described in Sections 6 (straight line segments) and 7.3 (circular arcs). These sections can be read fairly independently of the rest of the paper. They will appeal to a general reader who is just interested in a good estimator suited to his or her application. Readers with more conceptual interest in length estimators should read the whole paper, which is organized as follows. Section 2 contains some necessary preliminaries, reducing the various ways in which ideal straight strings may occur to a standard situation for the subsequent treatment. Section 3 describes the basic framework for straight line length estimation and gives a procedure to find optimal estimators. This provides the setting for the rest of the paper. Section 4 is on "characterizations," describing the string information essentially used in an estimator. The optimal estimators for various often-used characterizations are given. Section 5 describes the estimators of greatest practical use: the simple estimators. These estimators are not optimal, but often used because of their computational simplicity. Section 5.6 compares the estimators on straight line length estimation and gives recommendations for use on practical and theoretical grounds. Section 6 discusses an alternative approach, of fitting in least squares sense a line through the data points and taking the length of the best fit as the result. Section 7 discusses the validity and applicability for nonstraight curves, especially circular arcs. Section 8 concludes the paper with recommendations.

2. PRELIMINARIES

Digitization of a continuous object contour leads to a set of discrete contour points. These points are standardly denoted by a chaincode string (Fig. 1). In the case of digitization of an ideal straight continuous contour, this string has special properties: it satisfies the linearity conditions as given in [7] and [8]. Let us call such a string a *straight string*.

In the following, we will restrict ourselves to a standard situation with the commonly used 8-connected Freeman chaincode strings on the usual square grid of discrete points. Such an 8-connected chaincode string consists of two types of elements; we will call the elements aligned along the grid the *even* chaincode elements, and the diagonal elements the *odd* chaincode elements.

Other connectivities on regular grids (such as the 6-connected hexagonal grid, or the 4-connected square grid) can be transformed into this standard situation by an affine transformation [5, 9], so the restriction to the standard situation implies no loss of generality. A different connectivity is not conceptually different from the 8-connected square grid: the only effect is that coefficients occurring in estimators are different in value. For many estimators, this has already been described elsewhere [3, 5, 9].

The discrete line segments will be supposed to be derived from continuous straight line segments by object boundary quantization (OBQ) [1]. Again, this is a convenience rather than a restriction: for straight lines, the other form of digitization, grid intersection quantization (GIQ) [4] can be reduced to it (see e.g., [10]).

3. FRAMEWORK FOR LENGTH ESTIMATION

The usual procedure in digital length measurement is to characterize a string by a number of integer parameters. An example of this characterization is the commonly used "number of even chaincode elements" (n_e) and "number of odd chaincode elements" (n_o). This "tuple" of parameters (n_e, n_o) can then be used in the familiar Freeman length formula: $L_F = n_e + \sqrt{2}\, n_o$ [1]. Note that of all information present in the chaincode string, only the values of n_e and n_o are used in the length formula. We will call such a tuple of extracted parameters a *characterization*. It will appear that the characterization is a key notion in the assessment of the quality of estimators. In the next sections we will be considering different characterizations, their computational complexity, and the quality of estimator which can be based on each particular characterization.

In general, many continuous straight line segments lead to strings with the same tuple—the exact set depends on the type of characterization used. Since the tuple is the only information used in the length estimator, the estimated length will be the same for all of these continuous line segments. Originally, before digitization, the lengths of all these segments were certainly not identical, however. This means that a length value $L(t)$ attributed to a particular tuple t is only exact for a small fraction of the set $R(t)$ of all line segments that lead to this tuple t; for other line segments of this set $R(t)$, an error is made in the length measurement. One should clearly realize that these errors are fundamentally *unavoidable*! They cannot be reduced to zero, because information is lost in the digitization: slightly different continuous lines have become indistinguishable. Good estimators can minimize the errors—according to some criterion—but they can never eliminate them.

Let us call the set of all line segments leading to the same tuple t the *region* of t, denoted $R(t)$. This region is most easily visualized in the parameter space of straight line segments, which will now be introduced.

An infinite continuous line is given by the familiar equation

$$y = \alpha x + e. \tag{1}$$

The restriction—for mathematical convenience—to strings which consist of codes 0 and/or 1 implies a restriction of α to the range $0 \leq \alpha \leq 1$. Digitization leads to a series of discrete points, together forming the discrete line. Axes of the Cartesian (x, y)-coordinate system are chosen such that the discrete points forming the discrete line segment under study lie in the columns $x = 0$ to $x = n$. The origin is chosen in a grid point such that $0 \leq e < 1$.

A part of the continuous straight contour is thus given by three parameters: e, α, and n. Considering all segments with fixed n, the segment is represented as a point (e, α) in (intercept, slope)-space, called (e, α)-space for short. Conversely, a point in (x, y)-space is represented, through Eq. (1), by a line in (e, α)-space (Fig. 2).

The region $R(t)$ of tuple t is the set of continuous line segments that have t as their characterizing tuple. This region is represented in (e, α)-space by a set of

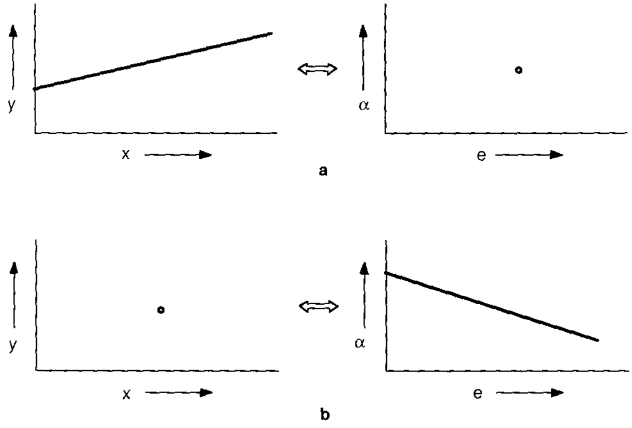

FIG. 2. The duality of points and lines in the (x, y)-space and (e, α)-space.

points, which will also be called $R(t)$ (this causes no confusion). Now, as we have reasoned above, the length of the continuous line segment in such a region is not constant. For a string of n elements, the total length of the part of the continuous contour enclosed between the columns $x = 0$ and $x = n$ is

$$f(\alpha; n) = n\sqrt{(1 + \alpha^2)}\,. \tag{2}$$

This is the "ground truth" for a continuous segment with parametrization (e, α) for each n. Note that the correct lengths indeed vary over the region $R(t)$, since they are dependent on α. Note that the length does not depend on e.

By the definition of a characterization, the estimator $L(t)$ serves as the length attributed to all strings with characterization t. This length estimate can be compared, in MSE sense, to the original lengths of all continuous line segments leading to t. This comparison allows relative assessment of different estimators, but also formulation of *optimal estimators*, as follows.

The continuous segments leading to a particular tuple t are all found in the region $R(t)$. A good length estimate corresponding to the tuple t represents this range of values over $R(t)$ in some way. For the minimal MSE criterion, the optimal estimator is the average value of $f(\alpha; n)$ over $R(t)$, the so-called BLUEstimator [6]:

$$L_{\text{BLUE}}(t) = \iint_{R(t)} f(\alpha; n)\, p(e, \alpha)\, de\, d\alpha. \tag{3}$$

Here $p(e, \alpha)$ is the probability density function of the lines in (e, α)-space (see [6]):

$$p(e, \alpha) = \sqrt{2}\,(1 + \alpha^2)^{-3/2}. \tag{4}$$

Note that eq. (3) defines a different estimate for each tuple. The BLUEstimator can be proved to be *optimal* for a given type of characterization in the sense that it is the only linear estimator that minimizes the MSE. Hence also its name: BLUE means *best linear unbiased estimator* (see, e.g., [11]). In the next section formulas for the BLUEstimator corresponding to various characterizations are considered.

Another estimator that will be seen to be of value is the MPO-estimator, defined as the length for the *most probable original* value of α in the region $R(t)$. If this value is $\alpha_M(t)$, then the formula for the MPO-estimator is

$$L_{\text{MPO}}(t) = n\sqrt{(1 + \alpha_M^2(t))}\,. \tag{5}$$

This estimator is the lowest order term in the Taylor expansion of $L_{\text{BLUE}}(t)$, under quite weak restrictions on the shape of $R(t)$ [9]. It is usually simpler to compute than $L_{\text{BLUE}}(t)$. Therefore it is also considered.

As a measure of the error for the length estimator L we will use a quantity dubbed RDEV(L, n). It is the root mean square difference between original length and estimated length, averaged over all strings of n elements, and divided by n:

$$\text{RDEV}(L(t), n) = \frac{1}{n}\left(\sum_t \iint_{R(t)} \{L(t) - f(\alpha; n)\}^2 p(e, \alpha)\, de\, d\alpha\right)^{1/2}. \tag{6}$$

The division by n makes the measure scale-invariant, and allows interpretation as the relative deviation (hence RDEV) in the length estimation of all line segments with a projected length of unity and slopes varying between 0 and 1, when the sampling density is n^2 points per square unit.

4. CHARACTERIZATIONS, REGIONS, AND OPTIMAL ESTIMATORS

A characterization of a string is a tuple of parameters extracted from it, to be used in the computation of estimators. In this section various characterizations often used in length measurement are defined, their regions $R(t)$ given, and the corresponding BLUE- and MPO-estimators computed.

4.1. (n)-Characterization

In the most primitive of characterizations, a chaincode string is simply characterized by counting the number of chaincode elements n. The region corresponding to the "tuple" (n) consists of all line segments passing through n columns of the grid and a slope α roughly between 0 and 1. See Fig. 3a.

The MPO-estimator of the length is the value of $f(\alpha; n) = n\sqrt{(1 + \alpha^2)}$ at the most probable value of α, which is $\alpha = 0$ [9]:

$$L_{\text{MPO}}(n) = n. \tag{7}$$

The BLUEstimator is, by Eq. (3), the average length of lines in the regions for this characterization. For large n, this can be computed to be

$$L_{\text{BLUE}}(n) = \int_0^1 \int_0^1 n\sqrt{(1 + \alpha^2)}\, de\, d\alpha = \frac{\pi}{4}\sqrt{2}\, n = 1.1107n. \tag{8}$$

The relative error RDEV for a chaincode string of n elements can be computed by Eq. (6) and is found to be asymptotically constant. For the MPO-estimator, the constant value is 0.1581, for the BLUEstimator 0.1129. Thus even with infinite sampling density the (n)-characterization based length estimators are never more accurate than 11%.

Note that Eq. (7) is also the length estimate one obtains when one simply counts the number of pixels on an 8-connected contour. This estimate is consistently too low. The argument above shows that multiplication by 1.1107 (for straight line segments!) reduces the error, since it makes the estimator asymptotically unbiased. The error then remaining is the variance, which is 11%. This is a gain of 4% over the biased estimate.

4.2. (n_e, n_o)-Characterization

A commonly used string characterization in image analysis is the number of even and odd chaincode elements, denoted by n_e and n_o, respectively. This is thus an (n_e, n_o)-characterization. An alternative but equivalent characterization is by n and n_o.

The regions of this characterization in (e, α)-space are sketched in Fig. 3b. The one region we had for the (n)-characterization is now subdivided into $n + 1$ regions, since each discrete end point of a line in column n leads to a different value of n_o (where $0 \leq n_o \leq n$).

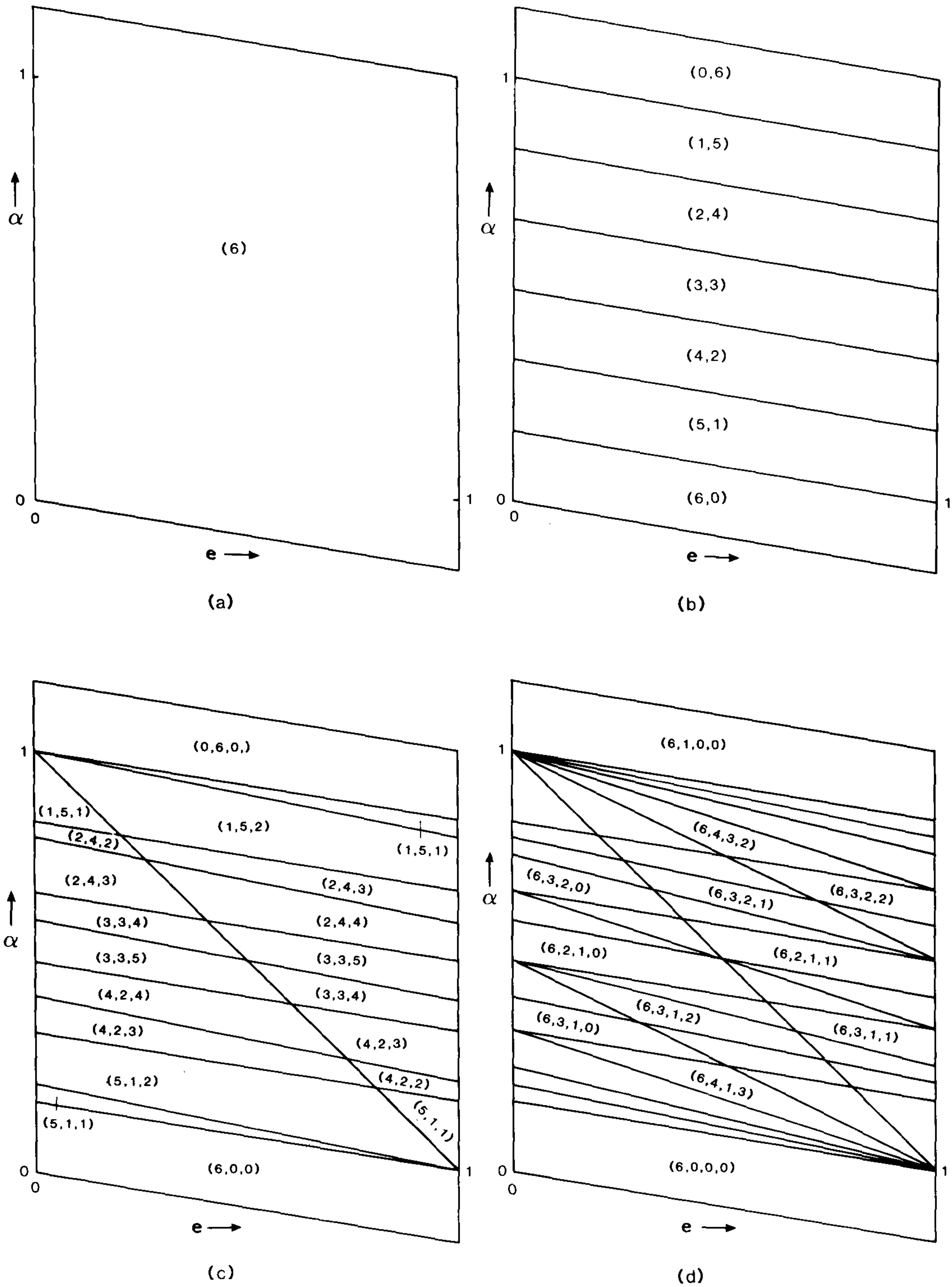

FIG. 3. The regions corresponding to various characterizations in (e, α)-space for $n = 6$, with the tuples indicated (in d not all are indicated for the sake of clarity). From [6].

The MPO-estimator for the length is the value of $f(\alpha) = n\sqrt{(1 + \alpha^2)}$ at the most probable value of α. This is the value where the e-dimension of a region is largest, which is $\alpha = n_o/n$,

$$L_{\mathrm{MPO}}(n, n_o) = \sqrt{(n^2 + n_o^2)}\,. \tag{9}$$

The BLUEstimator is the average over a region. This can be computed to be [9]

$$\begin{aligned} &L_{\mathrm{BLUE}}(n, m) \\ &\quad = \frac{1}{n}\left\{\frac{m+1}{n}\operatorname{atan}\left(\frac{m+1}{n}\right) - 2\frac{m}{n}\operatorname{atan}\left(\frac{m}{n}\right) + \frac{m-1}{n}\operatorname{atan}\left(\frac{m-1}{n}\right)\right. \\ &\qquad \left. - \frac{1}{2}\log\left(1 + \left(\frac{m+1}{n}\right)^2\right) + \log\left(1 + \left(\frac{m}{n}\right)^2\right) - \frac{1}{2}\log\left(1 + \left(\frac{m-1}{n}\right)^2\right)\right\} \Big/ \\ &\qquad \left\{\sqrt{1 + \left(\frac{m+1}{n}\right)^2} - 2\sqrt{1 + \left(\frac{m}{n}\right)^2} + \sqrt{1 + \left(\frac{m-1}{n}\right)^2}\right\}, \end{aligned} \tag{10}$$

where we denoted n_o by m, for convenience.

In the (n_e, n_o)-characterization, the MPO-estimator is the first term in the Taylor expansion of the BLUEstimator. Therefore the asymptotic behavior is identical to that of L_{BLUE}. Since Eq. (10) is more complicated than Eq. (9) and since we are looking for estimators that are simple to compute, we will only treat $L_{\mathrm{MPO}}(n, n_o)$.

The asymptotic order of the MPO-estimator can be computed to be [9]

$$\mathrm{RDEV}(L_{\mathrm{MPO}}(n, n_o), n) = 1/6n \qquad \text{for large } n. \tag{11}$$

This is directly related to the fact that the α-extent of the regions in (e, α)-space is of the order $O(n^{-1})$.

Note that the MPO-estimator in Eq. (9) is just the Euclidean distance between the end points of the discrete line segment. One would intuitively be inclined to think that this is the most accurate length estimator possible. However, the tuple (n, n_o) does not characterize a straight string completely and therefore not all information present in the string is used for the computation of the length. And indeed, more accurate length estimators can be found.

4.3. (n_e, n_o, n_c)-Characterization

In [5] an extra parameter for the characterization of 8-connected strings was introduced, called the "corner count." It is an extension of the corner count introduced by [3] for 4-connected strings. The corner count n_c of a string is defined as the number of occurrences of consecutive unequal chaincode elements in the string. In [5] it is shown that the improvement over the use of only (n_e, n_o) as characterizing tuple is that apart from the discrete points in the columns $x = 0$ and $x = n$ also the points in the columns $x = 1$ and $x = (n - 1)$ contribute to the characterization.

The regions of this characterization are sketched in Fig. 3c. There are now $5n - 4$ regions, of various shapes. The average extent in the α-dimension appears to be still of order $O(n^{-1})$.

Since the shape of the regions is fairly irregular, calculation of the MPO- and BLUEstimators is more complicated. The BLUEstimators can be found in [5]. The expressions for the MPO-estimators are fairly complicated and not given here. The asymptotic order of RDEV experimentally appears to be $O(n^{-1})$. The importance of this characterization lies not so much in the MPO- and BLUEstimators (the order of convergence is the same as for the simpler (n_e, n_o)-characterization) but rather in the simple estimators: $L(n_e, n_o, n_c) = an_e + bn_o + cn_c$, which will be treated in Section 5.3.

4.4. (n, q, p, s)-Characterization

In [10] a quadruple of parameters was introduced, characterizing a straight string C faithfully: (n, q, p, s). In words, the meaning of the tuple of parameters is: n is the number of string elements, p/q is the simplest irreducible fraction that can be the slope of a line generating C (related to α in Eq. (1)), and s is a phase shift (related to e). As for the characterizations treated before, each string leads to one realization of the tuple of parameters. But only for the (n, q, p, s)-characterization, the converse is also true. In fact, from the tuple (n, q, p, s) the string C can be reconstructed; for the ith element c_i,

$$c_i = \left\lfloor \frac{p}{q}(i - s) \right\rfloor - \left\lfloor \frac{p}{q}(i - s - 1) \right\rfloor, \qquad i = 1, \ldots, n \tag{12}$$

Formulas to derive (n, q, p, s), given C, are given in the reference.

Since the string can be reconstructed from the tuple, all discrete points must contribute to the characterization. There is no extra loss of information after the digitization: the (n, q, p, s)-characterization can be viewed to be just a rewriting of the information present in the string C in a more convenient form. For this reason, the (n, q, p, s)-characterization is called a *faithful characterization*. There is no characterization that preserves more information than a faithful characterization, for the obvious reason that it preserves *all* information.

The regions of this characterization are the finest division in (e, α)-space that is still possible after digitization. These fundamental regions are called "domains" and are analyzed in [9–11]. They are depicted for $n = 6$ in Fig. 3d. The study of these domains and their mathematical properties is of great theoretical importance in the search for the ultimate accuracy that can be reached. It can be shown [9] that there are asymptotically n^3/π^2 of these regions (hence there are n^3/π^2 discrete straight line segments of n elements).

This is an increase in the number of regions of 2 orders compared to the (n_e, n_o)- or the (n_e, n_o, n_c)-characterization: considerably more strings have become distinguishable on the basis of this ultimate, faithful, characterization.

Evaluation of the BLUEstimator for this characterization leads to complicated expressions, to be found in [6] and not quoted here. The MPO-estimator is

$$L_{\mathrm{MPO}}(n, q, p, s) = n\sqrt{1 + \left(\frac{p}{q}\right)^2}, \tag{13}$$

p/q being the most probable value of α in the region $R(n, q, p, s)$ [10]. It can be shown that the asymptotic RDEV of this estimator is of the order $O(n^{-3/2})$ [9]. Surprisingly, this is better than the estimate "Euclidean distance between the end points," which is $L_{\mathrm{MPO}}(n_e, n_o)$ with an RDEV of $O(n^{-1})$!

Some reflection shows why this is so. According to Eq. (3), a good length estimate can be based on a estimate of the slope. Now, as slope estimate, p/q is more accurate than n_o/n. The value of p/q is based on the whole string, and mainly determined by the central section. Addition of extra points at the ends hardly, if at all, influences its value. The value n_o/n, on the contrary, is very sensitive to the position of the end points, which necessarily change by discrete grid steps. Therefore n_o/n is less accurate as a slope estimator than p/q, and consequently $L_{\mathrm{MPO}}(n_e, n_o)$ is less accurate than $L_{\mathrm{MPO}}(n, q, p, s)$.

5. SIMPLE LENGTH ESTIMATORS

Though for each characterization the optimal estimator is BLUE, and a good approximation to the optimal estimator is MPO, reasonable approximations are obtained by estimators that are linear in the characterizing parameters. These estimators are well known in literature [1–5] and popular because of their computational simplicity. We will analyze them for straight line length measurement in this section. Comparison with the more sophisticated estimators of Section 4 is detained until section 5.6.

5.1. (n)-Characterization

A simple estimator for the n-characterization has, by definition, the form

$$L(n) = an. \tag{14}$$

The natural (but suboptimal) choice $a = 1$ yields an "estimator" that simply counts the number of chaincodes as the length

$$L_0(n) = n. \tag{15}$$

This estimator is identical to $L_{\mathrm{MPO}}(n)$, treated in Section 4.1. It is biased over the ensemble of long straight line segments, with an asymptotic RDEV of 16%. Rescaling to make the estimator unbiased for the measurement of the length of all strings consisting of n elements yields the estimator

$$L_1(n) = 1.1107n. \tag{16}$$

This is $L_{\mathrm{BLUE}}(n)$ of Eq. (3). The asymptotic RDEV now equals 11%.

It is revealing to see what the "circles" corresponding to this length measure are, i.e., points at equal L_0- or L_1-distance of a fixed point. Asymptotically, they are squares, see Fig. 4a. For L_0, the square lies completely outside the Euclidean circle with the same "radius," for L_0 estimates lengths consistently too short. For L_1, the square is in a "best-fit" with the circle.

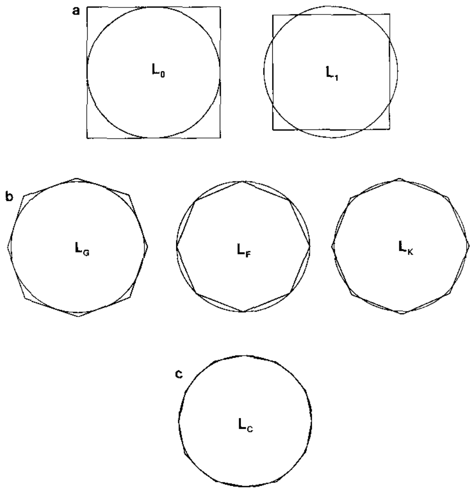

FIG. 4. The "circles" of the simple length estimators: (a) (n)-characterization: L_0 and L_1; (b) (n_e, n_o)-characterization: L_G, L_F and L_K; (c) (n_e, n_o, n_c)-characterization: L_C.

5.2. (n_e, n_o)-Characterization

For the (n_e, n_o)-characterization, the simple estimators are

$$L(n_e, n_o) = an_e + bn_o. \tag{17}$$

The first to use a formula of this kind was Freeman [1]. His is not an estimator in the proper sense, since it does not give an estimate of the length of the digital arc. The Freeman formula is obtained simply by counting even chaincode elements as having length 1 and odd chaincode elements as having length $\sqrt{2}$:

$$L_F(n_e, n_o) = n_e + \sqrt{2}\,n_o = 1.000n_e + 1.414n_o. \tag{18}$$

Considered as an estimator for the length of the original continuous curve before digitization, this formula gives biased results: the length it gives is consistently too long. For this reason, the asymptotic RDEV is high: 6.6%.

As a measure for the length of an 8-connected chaincode string, the L_F-estimator has been used for a long time and still is in common use. Nevertheless, the incorrectness of this approach was already pointed out in Kulpa [2] and, independently, by Groen and Verbeek [4]. They remarked that one should not compute the length of the digital arc, but of the original, continuous arc. This is an important point, essential to the development of optimal length measurement methods.

Starting from the same formula Eq. (17), [2] computes the coefficients a and b so as to minimize the expected error for measurement of the length of long line segments ($n \to \infty$). This yields

$$L_K(n_e, n_o) = 0.948n_e + 1.343n_o. \tag{19}$$

Now, the asymptotic RDEV decreases to 2.6%, a considerable improvement.

A look at the "circles" corresponding to these estimators shows what actually happens, see Fig. 4b. A "circle" of a length estimator L is the set of all continuous points with the same L-distance to the origin. Both the "circle" of the L_F- and of the L_K-measure is an octagon. Such a "circle" can be compared to the circle of the Euclidean distance measure. If this is done for "circles" of the same "radius," then the L_F-circle lies entirely inside the corresponding Euclidean circle, whereas the L_K-circle lies in a best-fit position. One can immediately see that this best-fit position of L_K-"circles" can be achieved by a rescaling of the L_F-"circles," i.e., multiplication of the L_F-measured "radius" by a properly chosen value. And indeed, comparing Eq. (18) and Eq. (19) we see that

$$L_K(n_e, n_o) = 0.948 L_F(n_e, n_o). \tag{20}$$

A calculation similar to that of Kulpa was independently performed by Groen and Verbeek [4]. They computed the expected lengths of an isolated even chaincode element and an isolated odd chaincode element, finding 1.059 and 1.183, respectively. A length estimator can be constructed from this:

$$L_G(n_e, n_o) = 1.059 n_e + 1.183 n_o. \tag{21}$$

Note that the coefficients differ considerably from those of $L_K(n_e, n_o)$! The estimator L_G is unbiased for strings with $n = 1$, for longer strings it is biased. The asymptotic RDEV for $n \to \infty$ (which is quite different from the $n = 1$ for which the estimator is optimized!) equals 8%, which is considerably higher than that of L_K. Nevertheless, L_G is the proper estimator to use for chaincode strings for which the elements may be considered completely uncorrelated. In practice, this is seldom encountered.

What can be learned from the comparison of L_K with L_G is that an estimator is only optimally applicable for the range of n for which it was designed. In fact, for each n there is a specific optimal value for the coefficients a and b [5]. Typical accuracies for these estimators are errors of approximately 5%.

5.4. (n_e, n_o, n_c)-Characterization

For the (n_e, n_o, n_c)-characterization, the simple estimators have the form

$$L(n_e, n_o, n_c) = a n_e + b n_o + c n_c. \tag{22}$$

The question arises what a good choice for a, b, and c is. An inkling of their values can be obtained by a reasoning similar to the one leading to Eq. (18), considering the length of the digital arc. For 8-connected grids, the corner count n_c counts the number of "knight's moves," consecutive odd–even or even–odd sequences in the string. The length that is attributed to such sequences—in a reasoning congruent to that from Freeman for Eq. (19)—is $\sqrt{5}$. Every even and odd code element in such a knight's move has already been counted in n_e and n_o, respectively. So, counting properly, one has

$$\begin{aligned} L_{FC}(n_e, n_o, n_c) &= n_e + \sqrt{2} n_o + \tfrac{1}{2}(\sqrt{5} - \sqrt{2} - 1) n_c \\ &= 1.000 n_e + 1.414 n_o - 0.089 n_c. \end{aligned} \tag{23}$$

This estimator is biased and gives lengths that are consistently too long. In [5], the optimal coefficients for an estimator of the form of Eq. (23) were evaluated by computer experiments, minimizing the MSE between the estimate L_C and the Euclidean length, for all straight strings with $n = 1000$. The resulting estimator is

$$L_C(n_e, n_o, n_c) = 0.980n_e + 1.406n_o - 0.091n_c. \tag{24}$$

This will be called the *corner count estimator*. The coefficients are indeed close to those derived by the naive approach outlined above. The corner count estimator is unbiased over the ensemble of long straight strings. The asymptotic RDEV equals 0.8%, which is remarkably accurate!

Again, a look at the "circles" of this estimator reveals why this is so. For a line $y = \alpha x$, the corner count increases if α increases from 0 to $\frac{1}{2}$ and decreases in the interval $\frac{1}{2}$ to 1. The "circle" is now an irregular hexadecagon (16-gon), with vertices at $\alpha = 0$, $\alpha = \frac{1}{2}$, $\alpha = 1$, and so on, symmetrically (see Fig. 4c). For L_{FC}, the hexadecagon lies entirely inside the Euclidean circle with the same "radius." For L_C, the hexadecagon is in a "best-fit" and becomes very close to the Euclidean circle. This explains the small value for the error.

5.4. (n, q, p, s)-Characterization

Linearization of the formula $n\sqrt{(1 + (p/q)^2)}$ to $An + Bq + Cp - Ds$ is not considered useful. Therefore, there is no simple (n, q, p, s)-based estimator.

5.5 Asymptotic Analysis of the Simple Estimators; Borgefors Distances

It is possible to analyze all simple estimators simultaneously, by noting that all are in fact contained in Eq. (22). For L_0 and L_1, $a = b$ and $c = 0$; for L_F, L_K and L_G, $c = 0$; L_C is the unrestricted case. The complete analysis can be found in [9]. Let us here treat the case $c = 0$, the simple estimators for the (n_e, n_o)-characterization, in some detail.

For the two-parameter characterization, the estimator depends only on the parameters a and b. Each simple estimator is therefore represented as a point in (a, b)-space. Computation of the asymptotic ith moment $G^i(a, b; n)$ for $n \to \infty$ (integrated over all lines with strings of consisting of n elements 0 and/or 1) yields

$$G^i(a, b; n) = n\int_0^1 \left\{a + (b - a)\alpha - \sqrt{(1 - \alpha^2)}\right\}^i \sqrt{2}\,(1 + \alpha^2)^{-3/2}\,d\alpha. \tag{25}$$

Setting $G^1(a, b; n)$ equal to 0 gives the (a, b)-values of all asymptotically unbiased estimators. This is a line in (a, b)-space, with equation

$$a\sqrt{2} + b = \frac{\pi}{4}\sqrt{2}\,(1 + \sqrt{2}) \tag{26}$$

Computation of the MSE, which is $G^2(a, b; n)$, yields a biquadratic form in a and b. Curves of constant MSE are thus ellipses in (a, b)-space, aligned along the line of unbiased estimators. This is indicated in Fig. 5. The points (a, b) representing the estimators L_0, L_1, L_F, L_K, and L_G are all indicated.

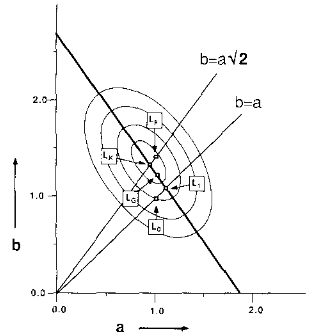

FIG. 5. (a, b)-space, the parameter space of the simple estimators for the (n) and (n_e, n_e)-characterization. The ellipses are curves of constant asymptotic MSE, the drawn line is the line of asymptotically unbiased estimators.

All rescalings of an estimator are represented in (a, b)-space by a line connecting the estimator to the origin. Making an estimator unbiased implies intersecting this line with the line of unbiased estimators. It is seen that L_K is L_F rescaled, in agreement with Eq. (20). The center of all ellipses represents the asymptotically most accurate simple (n_e, n_o)-based estimator, which is $L_K(n_e, n_o)$.[1]

If one performs the same kind of computation for the case $c \neq 0$, a triquadratic form in a, b, and c is found. Minimization yields exactly the coefficients of the corner count estimator (see [9]), found in [5] by a simulation.

In a recent paper on distance transformations [13], Borgefors describes distance measures similar in form to Eq. (22), but with rational coefficients a, b, and c. These estimators can be analysed in exactly the same way [9].

5.6. *Non-asymptotic Analysis; A Trade-off Theorem*

To obtain an insight into the behavior of the length estimators for the nonasymptotic case, a computer experiment was performed. The experiment was organized as follows. For each estimator L and for all straight strings C consisting of n elements, RDEV is computed as the weighted sum of the expected squared difference over the domain of C between the length estimate and the ground truth $n\sqrt{(1 + \alpha^2)}$, in agreement with Eq. (6). All quantities in Eq. (6) can be computed using the formulas developed in [9, 10]; thus the experiment is numerical rather than stochastic.

Figure 6 presents a plot of RDEV(L, n) as a function of n, and Table 1 the numerical results. It is observed that the ordering of estimators on the basis of their asymptotic behavior vindicates the calculations quoted in the previous sections.

Striking about Fig. 6 (but already explained in Section 5) is that the simple estimators reach a limit accuracy. The purport of this should be clearly realized. It means that if a line of fixed length is measured by taking increasing finer grids, then the length estimates do *not* become more accurate! In other words, there are

[1]Almost, but not quite, since in Eq. (25) optimization is done for all line segments leading to strings with the same number of elements, whereas L_K is computed by averaging over line segments with the same continuous length. See [9] for this point for gourmets. The ensuing difference in coefficients is small: 0.1%.

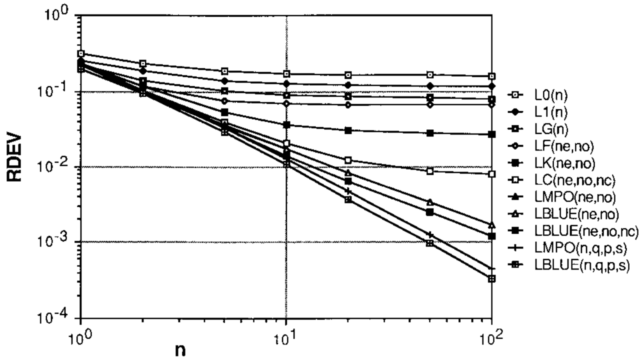

FIG. 6. Comparison of all length estimators for straight lines. See also Table 1. The "Euclidean length estimator" $L_{\text{MPO}}(n_e, n_o)$ virtually concides with $L_{\text{BLUE}}(n_e, n_o)$.

TABLE 1

A Comparison of Length Estimators for All Straight Strings Consisting of n Elements

	1	2	5	10	20	50	100	∞
$L_0(n)$	0.310	0.231	0.186	0.172	0.165	0.161	0.159	0.1581
$L_1(n)$	0.252	0.183	0.141	0.127	0.120	0.116	0.1143	0.1129
$L_G(n_e, n_o)$	0.217	0.141	0.103	0.091	0.085	0.082	0.0800	0.803
$L_F(n_e, n_o)$	0.223	0.117	0.0755	0.0682	0.0669	0.0664	0.0664	0.0664
$L_K(n_e, n_o)$	0.232	0.114	0.0534	0.0371	0.0307	0.0278	0.0270	0.0263
$L_C(n_e, n_o, n_c)$	0.228	0.103	0.0398	0.0208	0.0125	0.00879	0.00804	0.0077
$L_{\text{MPO}}(n_e, n_o)$	0.217	0.0937	0.0354	0.0172	0.00848	0.00337	0.00174	
$L_{\text{BLUE}}(n_e, n_o)$	0.223	0.0966	0.0356	0.0173	0.00849	0.00337	0.00170	
$L_{\text{BLUE}}(n_e, n_o, n_c)$	0.217	0.104	0.0329	0.0141	0.00644	0.00248	0.00124	
$L_{\text{MPO}}(n, q, p, s)$	0.217	0.103	0.0337	0.0127	0.00476	0.00127	0.00045	
$L_{\text{BLUE}}(n, q, p, s)$	0.196	0.0937	0.0291	0.0107	0.00379	0.00097	0.00034	

Note: The column marked ∞ contains predicted values. The values indicated are relative deviations (RDEV), compared to the continuous value. The experiment is described in Section 5.6.

accuracies that can never be achieved by certain estimators, no matter how high the sampling density is chosen. In those cases, improvement in accuracy cannot be reached by simply "oversampling"; rather, it should be achieved by using a more sophisticated estimator!

As was seen in Fig. 4 the explanation for the asymptotic tapering is that the "circles" of the simple estimators are squares, octagons, or hexadecagons. For each particular estimator, the polygon has a fixed shape, and this polygon can only approximate the Euclidean circle to a certain extent. Generally, the more elements the characterizing tuple has, the better the approximation is. It should be noted that the "corner count estimator" does a remarkably good job; its limit accuracy is 0.8%, and its error is within a factor of two of the best achievable accuracy for straight strings of up to about 10 elements. So, of all simple estimators, we recommend the corner count estimator explicitly for daily use.

For the more advanced MPO- and BLUEstimators, the asymptotic behavior also depends on the characterization employed. For the (n)-characterization the RDEV becomes constant, for the (n_e, n_o)- and (n_e, n_o, n_c)-characterization RDEV is of the order $O(n^{-1})$. The (n, q, p, s)-based estimators deserve special attention since they are the theoretically best possible. Their asymptotic behavior is of a different order, namely $O(n^{-3/2})$. The behavior of RDEV for the BLUEstimator is RDEV = $0.34n^{-3/2}$, where the factor 0.34 was determined experimentally. Since it is the theoretically optimal length estimator, this can be reformulated in the form of a "sampling vs. error trade-off theorem."

TRADE-OFF THEOREM. *For length measurement of straight strings, the sampling density d (per unit length) is related to the best achievable estimation error percentage p by*

$$d \geq 10.7p^{-2/3}. \tag{27}$$

Equality is reached with the length estimator $L_{\mathrm{BLUE}}(n, q, p, s)$ *only.*

A proof that the exponent is $-2/3$ can be found in [9]; the factor 10.7 is based on the experimental value 0.34 quoted above.

Equation (27) implies that for a desired average error of 1%, a sampling density of $d = 11$ pixels per unit length suffices when the optical length estimator is used. By way of comparison, the corner count estimator requires a sampling density of 40 to reach the same accuracy (Fig. 6). For the commonly used L_{F} and for L_{K} the accuracy of 1% is beyond reach, even with infinite sampling density!

6. NONIDEAL STRAIGHT LINES

Up to this point, we have limited the analysis to ideal straight lines. In dealing with noisy straight lines, the removal of the noise should precede the length estimation procedures as described above and can be treated separately. The removal of noise depends on the noise characteristics, and therefore is a too problem-specific task to be discussed here.

An exception is made for the least squares fitting of a line through the pixels of a digital line. An approximation of the requested line of the original line can then be found by taking the length of the best fit. Fitting a line through the pixels of the digital line is a completely different approach from the one in Sections 4 and 5. As is well known, fitting a line through a set of points assumes a linear relationship between the two independent variables and noise added to the data. In the case considered here, the noise is due to the quantization only. As the points of a digital straight line lie in a regular structure "around" the original line and knowledge about this structure is not exploited in the least square fitting of a line, the method cannot be as accurate as the BLUEstimation technique. In the BLUE technique, mentioned in Section 5.4, the structure of the set of all pixels was taken into account to determine the smallest possible set of original lines in the search for the length of this ensemble.

To make a rough estimate of the asymptotic accuracy which can be expected from the approach of the least squares fit, it is deducted from [9] that the spread in the contour points around the best fitting line is of the order $1/n$. In other words, the quantization noise has a spread of $O(n^{-1})$, and one may expect that the length approximation based on the fit has an accuracy of this order of magnitude. And indeed, an experiment, similar to the one described in Section 5.6, shows that the

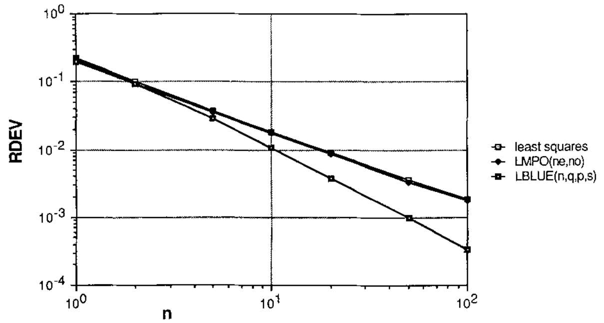

FIG. 7. Comparison of 3 estimators: The Euclidean distance between the ends of the digitized line, the length of the least squares fit to the data points, and the optimal BLUEstimator.

relative difference between the truth and the length obtained from the fit is of the order $O(n^{-1})$, see Fig. 7. The least squares fit shows almost the same asymptotic behavior as the Euclidean distance between beginning and endpoint of the digital curve, being less accurate than the estimates based on the (n, q, p, s)-characterization.

For the ideal, nonnoisy lines, the least squares fit cannot be recommended, as the Euclidean distance of the endpoints is as good, but computationally more efficient. For noisy straight lines, Fig. 7 shows the lower bound of the accuracy one can obtain with a least squares fit.

7. IDEAL NONSTRAIGHT LINES

In Section 4 increasingly accurate techniques were considered for the estimation of the length of straight lines only. The general trend that can be seen from the results in Section 6 is that the better estimation techniques are based on more extended characterizations of the string. In effect the better estimators make progressively more use of the fact that one knows that the string is derived from a continuous straight line. The ultimate in this is the faithful (n, q, p, s)-characterization, from which the straight string can even be reconstructed (Eq. (11)); this leads to the optimal estimators.

7.1. The Framework Revisited

An extension of length measurement to nonstraight strings is obviously the next problem to solve. As for straight strings, let us consider the ideal case (no noise) and investigate to what extent one may expect to find estimators, approximating reality on the basis of the discrete data. To do so, let us recapitulate the steps followed in the development of estimators for straight strings, as described in Section 3.

Step one was a *parametrization of the continuous curves* under consideration (for straight lines, this was (e, α)-space). Such a parametrization is always necessary, for it is the only way to describe each of the curves considered mathematically. For each point in this parameter space, the "ground truth"—the value of the continuous property—should be known. After all, it is the reconstruction of the ground truth that is the aim of the estimation. Since this ground truth can only be formulated if a parametrization is made, it is impossible to find estimators for properties of curves that cannot be parametrized.

Step two, was the *characterization of the discrete data* (for straight lines we used (n)-, (n_e, n_o)-, (n_e, n_o, n_c)-, and (n, q, p, s)-characterizations). Being the mathematical description of the discrete entities considered, this step is necessary before estimators can be formulated. If one desires optimal estimators, a faithful characterization should be made. This may be very difficult: we believe it is already impossible for circular arcs.

Step three was that the digitization process, described in terms of these parametrizations, led to a *description of the regions* in the continuous parameter space in a mathematically closed form. The regions imply the admissable values and the spread of the ground truth, given a specific realization of the discrete data. Therefore, for the development of estimators, this step too is essential.

Step four, *calculation*, depended on the estimator used. Some estimators, such as the MPO-estimators, took a reasonable value of the property in the region as the value of the property for the tuple considered. Other estimators, such as the BLUE- or simple estimators, required integration over the region. This presupposes the existence of a well-behaved probability density function, and hence the convergence of integration in the continuous parameter space.

An example may clarify these issues. Consider the estimation of the perimeter length of a circle of radius R with a center point at integer coordinates. The parameter space in this problem is the one-dimensional R-space. As a characterization of the corresponding digitization one could use the largest difference D between x-values of the discrete points. The regions of this (D)-characterization are the intervals $(D/2 - 1/2) \leq R < (D/2 + 1/2)$. The ground truth for the perimeter length of a circle of radius R is $L(R) = 2\pi R$. Estimators for this characterization and some specified error criterion can obviously be developed easily. It is an interesting, nontrivial puzzle to design a faithful characterization for this example and to develop corresponding optimal estimators—in fact, we believe that optimal estimators cannot be found in closed form.

As another example, consider an arbitrary circular arc. Such an arc can be parametrized by the position of the center and by the radius of the circle, together with two angles. Characterizations of the corresponding discrete arc can also be given (e.g., an (n_e, n_o)-characterization), but it is highly difficult (if at all possible) to specify the corresponding regions. This shows the difficulty involved in the development of length estimators for even slightly more general curves than straight line segments.

7.2. *The Polygonal Approach*

Instead of developing length estimators tailored to higher order curves (such as circular arcs) one might consider to use straight line length estimators on such curves. Note that this is in fact only allowed for continuous curves that are piecewise straight. For arbitrary curves, one could make a polygonal decomposition of the string into straight substrings and use the sum of the estimated lengths over these straight parts as an estimate for the length of the original contour. It will be clear that in the Euclidean case, the length measure thus obtained is consistently too small, for the model used in the estimation (piecewise straightness) does not correspond to the continuous reality (a curve).

There are algorithms known which perform a polygonal decomposition of a digitized curve into straight substrings of maximum length. Wu [7] presents an algorithm of order Σn_i^2 (where n_i is the number of elements in the ith substring). An algorithm of order Σn_i can also be given [14].

The length of the ith substring is estimated by the estimator L on the basis of the tuple t_i to be $L(t_i)$; the total length is then

$$L = \sum_i L(t_i). \tag{28}$$

Note that for the simple (n_e, n_o)-based estimators, explicit polygonal approximation is not necessary. Denoting the number of even and odd codes of the ith segment by $n_e(i)$ and $n_o(i)$, respectively, the total length estimated is

$$L = \sum_i (an_e(i) + bn_o(i)). \tag{29}$$

Using fixed values for a and b, and noting that $\Sigma_k n_e(k) = n_e$ and similarly for n_o, this becomes simply

$$L = an_e + bn_o. \tag{30}$$

Thus simple, (n_e, n_o)-based length estimators can be used for arbitrary strings without making a polygonal decomposition. In fact, these estimators consider the whole string as if it were a single straight string. Since the estimators were optimized to estimate the length of a straight string optimally, one expects an inferior behavior for arbitrary strings. An experiment on circular arcs, described in the next subsection, unexpectedly reveals that the opposite is the case!

7.3. *Circular Arcs*

The next step in complexity from the polygonal approximation is the approximation of general curves by second-order segments, such as circular arcs. In that case, one needs estimators for the length of a circular arc, rather than for the length of a chord. It was argued in Section 7.1 that it is difficult to find such estimators. One is therefore tempted to see how well simple estimators do the job.

An experiment was organized as follows. A circle with radius R was generated by Bresenham's circle generation algorithm [15]. In the string thus obtained, for a given n all digital arcs consisting of n consecutive elements were measured by the estimator L_K, with all points of a quarter circle serving in turn as the start point of an arc. Also, for each of these segments, the Euclidean distance L_E measured along a circular arc of radius R contained within the rays determined by the discrete end points of the segments was computed. The average normalized root MSE between these two measures

$$\left\{ \frac{4}{N} \sum_{i=1}^{N/4} \left(\frac{L_K(i)}{L_E(i)} - 1 \right)^2 \right\}^{1/2} \tag{31}$$

(N is the number of points of the complete discrete circular string) was computed. Figure 8 is a log–log plot of the result as a function of n/R and R.

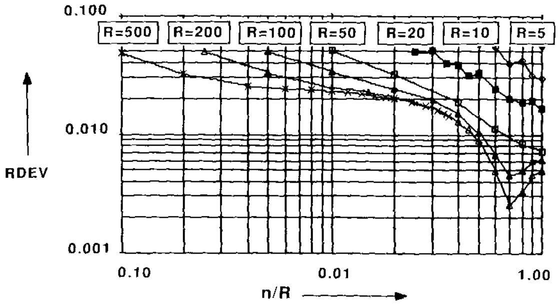

FIG. 8. The standard deviation of Eq. (31), as a function of n/R, for various R (see text).

It is seen that the error becomes zero for $n/R \simeq 0.7$. The reason why this happens is the key to the explanation of the whole figure. Consider Fig. 9 where an arc of $\pi/4$ is drawn. From the figure, it follows that for large R, when the circle may be considered as continuous, we have

$$n/R = (n_1 + n_2)/R = \sqrt{2} - \left((\sqrt{2} - 1)\sin\alpha + \cos\alpha\right)/\sqrt{2} \tag{32}$$

and

$$n_o/R = (m_1 + m_2)/R = \left(\sin\alpha + (\sqrt{2} - 1)\cos\alpha\right)/\sqrt{2} - \sqrt{2}. \tag{33}$$

It follows from Eq. (32) and Eq. (33) that a relation holds between $n_e = (n - n_o)$ and n_o, independent of α:

$$n_e + \sqrt{2}\, n_o = 2(\sqrt{2} - 1)R. \tag{34}$$

The L_K-measure for the length of the *chord* of $\pi/4$ is now, writing the proper value $\pi/8(1 + \sqrt{2})$ for the constant 0.9481 of Eq. (20):

$$L\left(\frac{\pi}{4}\right) = \frac{\pi}{8}(1 + \sqrt{2})\left(n_e + \sqrt{2}\, n_o\right) = \frac{\pi}{4}R \tag{35}$$

which is exactly the correct value for the length of the *arc* of $\pi/4$! Thus L_K is a

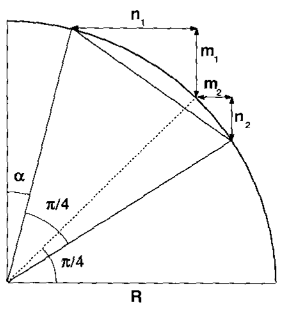

FIG. 9. Derivation of Eqs. (32) and (33).

consistent estimate for the length of an arc of $\pi/4$. This explains why the error at $n/R = 0.7$ is very small. Obviously, at multiples of $\pi/4$ the same happens.

For small values of n/R, one considers small arcs of a large circle. These arcs are virtually straight. Computing the error as in Eq. (31) means averaging over lines in all positions. One therefore expects an error curve as the curve for the L_K-estimator in Fig. 6, and this expectation is vindicated, see Fig. 8. Arcs that are not small, but still smaller than $\pi/4$, have restrained (but α-dependent) relations between n_e and n_o. This reduced variation with α reduces the variance and hence the MSE of Eq. (31): the MSE is smaller than that for straight lines. For arcs of $\pi/4$, the MSE becomes 0. With growing arclength, still an error of 0 is made for the part of the arc that spans $\pi/4$. For the remainder, the above sequence of events is repeated. The relative error is, through division by the complete arc length L_E, much smaller.

It is thus seen that the simple estimator L_K, already reasonably good for straight lines (RDEV > 2.6%) is even better for curved arcs. In an arbitrary curve that may be considered to be composed of continuous circular arcs, a relatively small error is made for each stretch spanning $\pi/4$. The main contribution to the error is due to the straight stretches and highly curved parts in the string.

8. CONCLUSIONS

The conclusions of this paper can be arranged with respect to the shape of the continuous figures of which the length is to be determined, and the estimators used.

8.1. *Straight Line Segments and Polygons*

If the continuous figure is piecewise straight, one can apply the length estimators developed for straight line segments. The properties of these estimators are found in the calculations of Sections 4 and 5, and the experiment of Section 5.6.

The main conclusions are:

—The optimal estimator is known, and described in [6], but highly mathematically complex and not very suited for common practice.

—The trade-off between sampling density d and optimal accuracy (expressed as percentage error p) is given by $d \geq 10.7p^{-2/3}$ (see Eq. (27).

—There exists a simple estimator that provides accurate results (accuracy up to 0.8%), namely the corner count estimator L_C of Eq. (24).

—There are better estimators for the length of a discrete straight line segment than the Euclidean distance between the end points. Computationally the simplest is $L_{MPO}(n, q, p, s)$ of Eq. (13).

Comparing the estimators with respect to the performance and complexity, there are a few that can be recommended, replacing some of those in use at present. Some of the others are no longer of use since there are other estimators of the same complexity that perform better. This is the case for $L_0(n)$, $L_F(n_e, n_o)$, and $L_{BLUE}(n_e, n_o, n_c)$. Others are of limited use under special circumstances: $L_G(n_e, n_o)$ for strings with little or no correlation between consecutive string elements and $L_{BLUE}(n, q, p, s)$ as the theoretically optimal estimator. The recommended selection is:

$L_1(n)$—	for use in extremely time-critical situations, or in simple image analysis. Accuracy up to 11%. Better than $L_0(n)$.
$L_K(n_e, n_o)$—	for use in time-critical situations, or when no high accuracy is needed. Accuracy up to 2.6%. Better than $L_G(n_e, n_o)$ and $L_F(n_e, n_o)$.
$L_C(n_e, n_o, n_c)$—	the "corner count" estimator. For normal use: simple to compute, reasonably high accuracy, up to 0.8%. Comparable to optimal result for straight strings with $n \lesssim 10$.
$L_{MPO}(n_e, n_o)$—	the "Euclidean distance" of beginning and end points. Simple to compute but more time-consuming that L_C. Accuracy $0.17n^{-1}$.
$L_{MPO}(n, q, p, s)$—	for use when high accuracy is required, or when high sampling density is expensive. Accuracy $0.46n^{-3/2}$.

It should be stressed that this section applies to the length estimation of straight strings only.

8.2. *Chains of Circular Arcs*

Some figures are not to be conceived of as polygons, but may be approximated as a concatenated series of circular arcs. For such figures, one might develop optimal estimators following the techniques outlined in Section 7.1, but this is left for future investigations.

In the experiment of Section 7.3 we used a length estimator developed for straight line segments to estimate the contour length. It was seen that the simple L_K estimator is remarkably accurate, in spite of the fact that it was not designed for this situation. Presumably, so is L_C. In fact, these estimators are better for the estimation of the length of an *arc* than for the estimation of a *chord*. For arcs of $\pi/4$, the L_K-length is unbiased. These simple estimators have the additional advantage that it is not necessary to actually perform the decomposition of the string into substrings, representing circular arcs, as was explained in Section 7.2.

The MPO line length estimators are less suited for figures consisting of circular arcs; since they estimate the chord length rather than the arc length, they give a biased result with a correspondingly high MSE.

8.3. *General Figures*

In Section 7.1, it is argued that for the general figures length estimators cannot be developed. This is interesting, since it means that a general equivalent in the discrete world of the continuous notion "length of a contour" cannot be given. For each situation, one should carefully define what one means by the length of the discrete contour; only then can one estimate it. This situation is reminiscent of the famous "coast of Brittany" problem in fractal geometry [16].

Thus, for general curves, at the moment of the best thing to do is to approximate such curves by straight line segments and circular arcs. For these, the results quoted above are applicable. It may be possible to derive a trade-off theorem similar to Eq. (27) for arbitrary curves using the two extreme approximations by straight line segments and circular arcs. Research on this is being performed.

ACKNOWLEDGMENT

The authors are indebted to Nellie Schipper for shuttling.

REFERENCES

1 H. Freeman, Boundary encoding and processing, In *Picture Processing and Psychopictorics* (B. S. Lipkin and A. Rosenfeld, Eds.), pp. 241–266, Academic Press, New York, 1970.
2 Z. Kulpa, Area and perimeter measurement of blobs in discrete binary pictures, *Comput. Graphics Image Process.* **6**, 1977, 434–454.
3 D. Proffitt and D. Rosen, Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges, *Comput. Graphics Image Proc.* **10**, 1979, 318–332.
4 F. C. A. Groen and P. W. Verbeek, Freeman code probabilities of object boundary quantized contours, *Comput. Graphics Image Process.* **7**, 1978, 391–402.
5 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.
6 L. Dorst and A. W. M. Smeulders, Best linear unbiased estimators for properties of digitized straight lines, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1986, 276–282.
7 L.-D. Wu, On the chain code of a line, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-4**, No. 3, 1982, 347–353.
8 S. H. Y. Hung, On the straightness of digital arcs, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, No. 2, 1985, 203–215.
9 L. Dorst, *Discrete Straight Line Segments: Parameters, Primitives and Properties*, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, June 10, 1986.
10 L. Dorst and A. W. M. Smeulders, Discrete representation of straight lines, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 4, 1984, 450–463.
11 T. O. Lewis and P. L. Odell, *Estimation in Linear Models*, Prentice–Hall, Englewood Cliffs, NJ, 1971.
12 M. D. McIlroy, A note on discrete representation of lines, *AT&T Tech. J.* **64**, No. 2, 1985.
13 G. Borgefors, Distance transformations in digital images, *Comput. Vision Graphics Image Process*, **34**, 1986, 344–371; *On Hierarchical Edge Matching in Digital Images Using Distance Transformations*, Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden, TRITA-NA-8602, April 1986.
14 A. W. M. Smeulders and L. Dorst, Straightness and characterization of tracked arcs: A linear-time algorithm, in preparation, 1987.
15 J. E. Bresenham, Algorithms for circular arc generation, in *Fundamental Algorithms in Computer Graphics, Proceedings, NATO Advanced Study Institute, Ilkley, Yorkshire, Great Britain, March 30–April 12, 1985*, R. A. Earnshaw, Ed., NATO ASI Ser. F, Vol. 17, pp. 197–217, Springer-Verlag, Berlin, 1985.
16 B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, New York, 1977.
17 B. Shahraray and D. J. Anderson, Uniform resampling of digitized contours, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, 674–681, 1985.