Perimeter estimation based on grey level object representation

Nataša Sladoje[1] and Joakim Lindblad[2]

[1] Faculty of Engineering, University of Novi Sad,
Novi Sad, Serbia
sladoje@uns.ns.ac.yu

[2] Centre for Image Analysis,
Swedish University of Agricultural Sciences,
Uppsala, Sweden
joakim@cb.uu.se

Centre for Image Analysis, Uppsala, Sweden, Report 33, 2008.
ISSN 1100-6641

Abstract

We present a novel method that provides an accurate and precise estimate of the length of the boundary (perimeter) of an object, by taking into account gray-levels on the boundary of the digitization of the same object. Assuming a model where pixel intensity is proportional to the coverage of a pixel, we show that the presented method provides error-free measurements of the length of straight boundary segments in the case of non-quantized pixel values. For a more realistic situation, where pixel values are quantized, we derive optimal estimates that minimize the maximal estimation error. We show that the estimate converges toward a correct value as the number of gray-levels tends toward infinity. The method is easy to implement; we provide complete pseudo-code. Since the method utilizes only a small neighborhood, it is very easy to parallelize. We evaluate the estimator on a set of concave and convex shapes with known perimeters, digitized at increasing resolution.

Contents

1 Introduction **4**

2 Background **4**

3 Pixel coverage digitization **7**

4 Edge length estimation based on difference of column sums **8**
4.1 Non-quantized case 8
4.2 Gray-scale case 8
4.3 Binary case 9

5 Optimization to minimize maximal error **9**

6 Local computation of length **14**
6.1 Local computation of length based on 3×2 configurations 15
6.2 Local computation of length based on 3×3 configurations 16
6.3 Rotations of the plane 18
6.4 An algorithm for edge length computation 19

7 Evaluation of the estimator performance on synthetic test images **20**

8 Summary **22**

Appendix A Derivation of optimal scale factor $\gamma_n^{(i,j)}$ **24**

Appendix B Asymptotic behavior of the estimation error $|\varepsilon_n|$ **25**

Appendix C q for small values of n **25**

Appendix D Condition for the detection of $D_{(c,r)}$, so that $\tilde{d}_{(c,r)} = \tilde{d}_c$ **26**

Appendix E Lemmata for Subsection 6.3 **28**

Appendix F List of most often appearing symbols **30**

1 Introduction

The length of the boundary (perimeter) of an object in a two dimensional (2D) image is one of the most 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 by a digital image. Accordingly, a very 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., [1] and [2]. The binary model corresponds well with the output of the Gauss center point digitization scheme. It, however, discards a large amount of useful information, especially along the object boundary. Digital images are commonly acquired, not by taking point samples, but rather by the integration of intensity over finite sized detector regions. Such digitization provides gray-level images, where pixel intensities are related to the degree of overlap between the imaged object and the detector area. In this paper we show that a significant improvement in the accuracy and precision of perimeter estimates can be achieved by utilizing the gray-level information contained in digital images obtained by such pixel coverage digitization.

We base the theory for the length estimation on segments of the straight edges of half-planes, and perform an optimization to minimize the maximal estimation error for such objects when the length of the edge tends to infinity. We also show empirically that the developed estimator can successfully be applied for boundary length estimation of more general shapes.

The paper is organized as follows: The next section gives a brief overview of related work. Section 3 introduces the pixel coverage digitization model. In Section 4 we show how the length of a segment of a half-plane edge can be estimated from column-wise sums of pixel values. In Section 5 we perform optimization to minimize the maximal error of such estimates. In Section 6 we show that the column based estimate is equivalent to an estimate that uses only the pixel values of a 3×3 neighborhood. Section 7 illustrates the performance of the method when applied to a set of convex and non-convex shapes. Section 8 summarizes the paper. To improve the flow of reading, most of the theoretical derivations are given in a number of appendices; the last of them (Appendix F) contains a list of most often appearing symbols.

2 Background

Digital image analysis aims at measuring features of continuous (real world) objects on the basis of their digital images. In many applications, the length of the boundary of a digitized object is an important feature to measure. It should be noticed that measurements of this type can only be estimates, since the interest is seldom in the length of the boundary of the digital object, but rather in the length of the boundary of the original object that has been digitized.

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 centers as determined from the Freeman chain code [3]. This, 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.

Starting from an assumption that the boundary of an object is locally planar, optimal weights for the local steps along the border line have been derived, [4, 5], leading to improved perimeter estimates. Weights for the 2D case have been optimized to provide an unbiased estimator with minimal mean square error for straight lines with length tending to infinity, and have been proven to perform even better for curved contours [1]. This last property is important to notice since it provides much more general applicability for the estimator. Even though we perform optimization w.r.t. a different criterion, i.e., we minimize the maximum error [6, 7], we find a strong motivation in [1] to believe that similar holds for the approach taken in this paper, and that its applicability is not restricted to straight segments only. This is also verified in Section 7 of this paper.

In addition to the local type of estimators mentioned above, different non-local perimeter estimators have been developed; see e.g [2] for an overview. By basing the estimate on information from larger regions of the image, non-local estimators can be made to ensure convergence toward the true value, as the grid resolution increases [8]. These are often referred to as multigrid convergent estimators. Most such estimators are based on recognizing straight line segments and performing a polygonalization of the object. However, to find straight line segments, we can no longer use local algorithms.

Multigrid convergence, albeit being a theoretically appealing property, has only limited practical applicability. Resolution is rarely a parameter that can be increased indefinitely, or even chosen deliberately; it is imposed by the technical properties of the imaging device. Moreover, image analysis, and science in general, will most certainly continue to try to capture those phenomena that just barely can be resolved – the more powerful imaging devices available, the smaller objects become a challenge to image. The performance of feature estimation methods at low resolutions is therefore of constant practical significance. In addition, multigrid convergence of global estimators relies on the assumption of theoretical smoothness of the boundary of imaged objects, so that the edges become more and more similar to straight lines as resolution is increased. Physical objects do, in general, not have such a smoothness property and multigrid convergence based on approximation of the boundary segments with long straight lines is therefore a theoretical idealization that does not fit too well to reality.

On the other hand, local methods are not multigrid convergent. An analysis of this issue can be found in [9]. In spite of that, local methods are still often preferred to global ones, due to their several advantages. Local methods are in general easier to implement in an efficient manner than global ones. Local algorithms are very suited for parallel implementations, as opposed to global ones. This provides possibility for additional increase of speed by exploiting the extremely parallel architecture of modern graphics processing units. Further, methods based on summing a number of local contributions of bounded magnitude to form a global estimate are by definition rather insensitive to small changes of the input image (i.e., such changes lead to a small change in the output). That makes local methods stable, as opposed to

many global ones, where small changes can lead to changed polygonalization, and therefore more significant changes of the output. Moreover, if the input image is changed locally, only that part of the image needs to be traversed to update the estimate. Locality is also an appealing property when measuring features related to parts of the object boundary, such as, e.g., the common boundary of two adjacent image objects.

To improve the accuracy and precision of local estimators, methods that utilize gray-level information in images have been suggested. In [10], an arc length estimation method that uses normal vectors computed from gray 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. This enables estimates of higher precision. However, a thorough analysis of the performance of the method is not presented and no optimization of the local lengths is performed. The method presented in [11] relies on sampling theory. However, it includes some "practical choices" without full theoretical justification. It is based on transformation of object boundaries in gray-level images into corresponding volumes, where the length estimation problem is converted into a (simpler) problem of volume estimation. The method relies on 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 relation between the spatial resolution, the gray-level quantization (gray-level resolution), and the achievable reconstruction accuracy for certain types of images/objects is studied in [12]. It is shown that objects with straight edges can be reconstructed without error from an image with finite spatial resolution, if gray-levels are not quantized. In other words, it is concluded that low-resolution gray-scale images of polygonal silhouettes induce less ambiguity than high-resolution bi-level images.

These results encouraged us to further explore the potential of using gray-level information in length estimation. In this paper we present a novel length estimator based on local computations. It utilizes gray-levels assigned to pixels in the image in accordance to the pixel coverage digitization. Effects of quantization of gray-levels are considered. The optimal scale factor for the (Freeman-style) cumulative sum of local steps is derived, as a function of the number of gray-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; this definition of optimality is taken from [6]. The method is applied and evaluated on objects with non-straight boundaries as well. Tests include perimeter estimation for the set of objects suggested in [13]. The issue of a trade-off between spatial and gray-level 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 gray-level resolutions. The results show that the accuracy and precision of estimates rapidly increase with the increase of gray-level resolution, once a "reasonable" spatial resolution is provided.

In the following sections we present the suggested method and its application.

The research performed clearly shows that gray-level information can be successfully used for precise and accurate perimeter estimators.

3 Pixel coverage digitization

Most 2D imaging devices provide an output where photons are integrated over the finite area of individual detector elements, where each detector region is associated with a corresponding pixel in the image. Assume that we are imaging a white object on a black background, and that the imaging device is adjusted to provide maximal contrast without saturation, i.e., the pixels completely covered by the white object are assigned the maximal gray-level available, whereas the pixels of the background are assigned gray-level zero. If the imaging device has a linear response, then the partly covered pixels along the boundary of the object will exhibit gray values between the two extrema; the gray value assigned to a pixel is proportional to its coverage by the object. This model provides the basis for the *pixel coverage digitization*.

Let a square grid in 2D be given. The Voronoi region associated to a grid point $(i,j) \in \mathbb{Z}^2$ is called pixel $p_{(i,j)}$.

Definition 1 *For a given continuous object $S \subset \mathbb{R}^2$, inscribed into an integer grid with pixels $p_{(i,j)}$, the pixel coverage digitization of S is*

$$\mathcal{D}(S) = \left\{ \left((i,j), \frac{A(p_{(i,j)} \cap S)}{A(p_{(i,j)})} \right) \middle| (i,j) \in \mathbb{Z}^2 \right\} ,$$

where $A(X)$ denotes the area of a set X.

Definition 1 assumes assignment of non-quantized real coverage values to the pixels of the grid. However, when using digital approaches (computers) to represent, store, and analyze images, we are limited to a finite number of gray-levels to represent coverage of individual pixels. This leads to the following version of pixel coverage digitization:

Definition 2 *For a given continuous object $S \subset \mathbb{R}^2$, inscribed into an integer grid with pixels $p_{(i,j)}$, the n-level quantized pixel coverage digitization of S is*

$$\mathcal{D}^n(S) = \left\{ \left((i,j), \frac{1}{n} \left\lfloor n \frac{A(p_{(i,j)} \cap S)}{A(p_{(i,j)})} + \frac{1}{2} \right\rfloor \right) \middle| (i,j) \in \mathbb{Z}^2 \right\} ,$$

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

We denote the set of numbers representing pixel coverage values in n-level quantized pixel coverage digitization by $\mathcal{Q}_n = \left\{ 0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n} = 1 \right\}$. This set corresponds to the set of gray-levels available; e.g., $n = 1$ for a binary image, while $n = 255$ provides the set of gray-levels for an 8-bit pixel representation.

4 Edge length estimation based on difference of column sums

4.1 Non-quantized case

A well known formula for computing the arc length of a function $y = f(x)$ over an interval $[a, b]$ is given by $l = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx$. 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

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

Given a halfplane H defined by $y \leq kx + m$, $k \in [0, 1]$, $m \geq 0$, we can use Equation (1) to compute the length of the edge segment $y = kx+m$, $x \in [0, N], N > 0$. 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, $s_c = \int_{c-\frac{1}{2}}^{c+\frac{1}{2}} y(x)\, dx + \frac{1}{2}$, 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, $I = \mathcal{D}(H)$, the length l in Equation (1) can be computed as

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

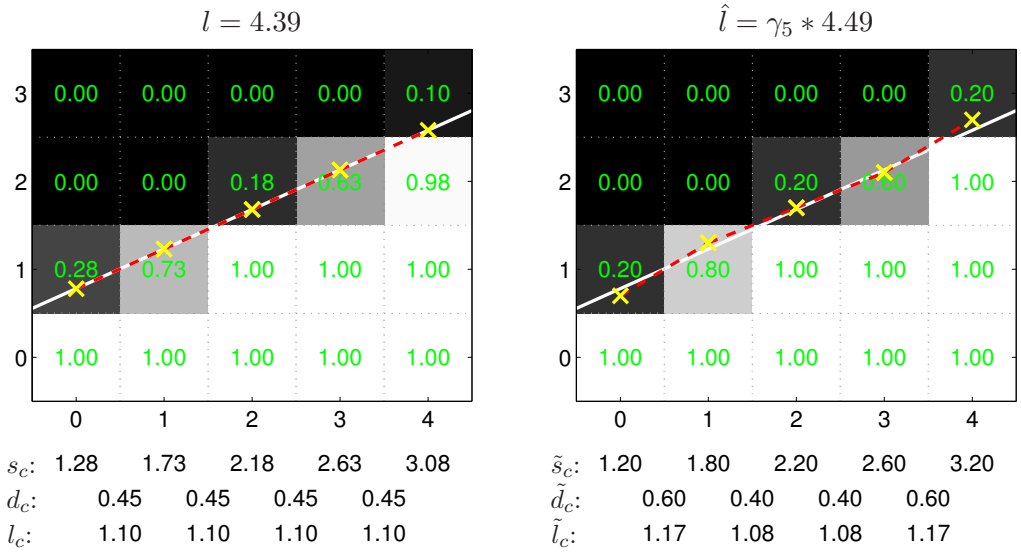
where the value $d_c = s_{c+1} - s_c$ is the difference of two consecutive column sums of the pixel values of the image. This is illustrated in Figure 1(a). The corresponding results for $k \notin [0, 1]$ follow by symmetries of the square grid.

4.2 Gray-scale case

If the observed image I is, instead, an n-level quantized pixel coverage digitization $I = \mathcal{D}^n(H)$, 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 $\mathcal{Q}_n$ and an edge with real valued slope $k \in [0, 1]$ is thereby approximated using local steps with slopes from $\mathcal{Q}_n$. An illustration is given in Figure 1(b). Due to the quantization, and the edge line only being approximated, an error is unavoidable. We therefore introduce a scale factor γ_n, providing the following formula for the estimation of the length of the edge present in the image:

$$\hat{l}(I) = \sum_{c=0}^{N-1} \hat{l}_c\,, \quad \text{where} \quad \hat{l}_c = \gamma_n\sqrt{1 + \tilde{d}_c^2}\,. \tag{3}$$

In Section 5 we derive a formula for the optimal value of the scale factor γ_n as a function of the number of gray-levels n so as to minimize the estimation error of (3) and we show that $\lim_{n\to\infty} \gamma_n = 1$; this is when formula (3) reduces to formula (2).



(a) Non-quantized (b) Quantized $n = 5$

Figure 1: 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 $y \leq 0.45x + 0.78$. 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 and $\times$ marking the ends of each step.

4.3 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. [4, 5]. 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 [7] a value $\gamma_1 \approx 0.9604$ is shown to minimize the maximal error for the binary case.

5 Optimization to minimize maximal error

Let a set of linearly independent vectors $\mathbf{S} = \left\{\mathbf{S}_i = (1, \frac{i}{n}),\ i \in \{0, 1, \ldots, n\}\right\}$ (in Cartesian coordinate system) be given. Their slopes are $\frac{i}{n} \in \mathcal{Q}_n$ 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 $\mathbf{S}_i$ is $S_i = \sqrt{1 + \left(\frac{i}{n}\right)^2}$.

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

$$\mathbf{l} = \frac{(j - nk)N}{j - i}\mathbf{S}_i + \frac{(nk - i)N}{j - i}\mathbf{S}_j \,. \tag{4}$$

The length of $\mathbf{l}$ can be estimated by using Equation (4) as

$$\hat{l} = \gamma_n^{(i,j)} \left(\frac{(j - nk)N}{j - i} S_i + \frac{(nk - i)N}{j - i} S_j \right). \tag{5}$$

This corresponds to an edge segment such that $\tilde{d}_c \in \left\{ \frac{i}{n}, \frac{j}{n} \right\}$ for all c, for which Equation (3) is equivalent to Equation (5).

As mentioned in the previous section, using $\gamma_n^{(i,j)} = 1$ in Equation (5) leads to an overestimate for all edges which are not in the direction of either $\frac{i}{n}$ or $\frac{j}{n}$. We derive $\gamma_n^{(i,j)}$ as a function of n to minimize the maximal error of estimation formula (5).

In the given context, the coefficients $\frac{(j-nk)N}{j-i}$ and $\frac{(nk-i)N}{j-i}$ of Equation (5) represent the non-negative number of repetitions of each of the local steps $\mathbf{S}_i$ and $\mathbf{S}_j$ in approximation of $\mathbf{l}$ and are therefore required to be integers. This condition is, however, rather difficult to impose in the general case. We, therefore, derive the theory for segments of infinite length ($N \to \infty$), and by that avoid restrictions to integer valued coefficients in the decompositions of the observed segments (see also [7]).

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

$$\varepsilon_n^{(i,j)}(k) = \frac{\hat{l} - l}{l} = \gamma_n \frac{(j - nk)S_i + (nk - i)S_j}{(j - i)\sqrt{1 + k^2}} - 1 \,. \tag{6}$$

To get a visual impression of the error function given by Equation (6), we plot $\varepsilon_n^{(i,j)}(k)$ as a function of k in Figure 2(a), for the case $\gamma_n = 1$, $n = 5$ and for a number of combinations of i and j.

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

$$\max_{k \in [\frac{i}{n}, \frac{j}{n}]} \varepsilon_n^{(i,j)}(k) = - \min_{k \in [\frac{i}{n}, \frac{j}{n}]} \varepsilon_n^{(i,j)}(k) \tag{7}$$

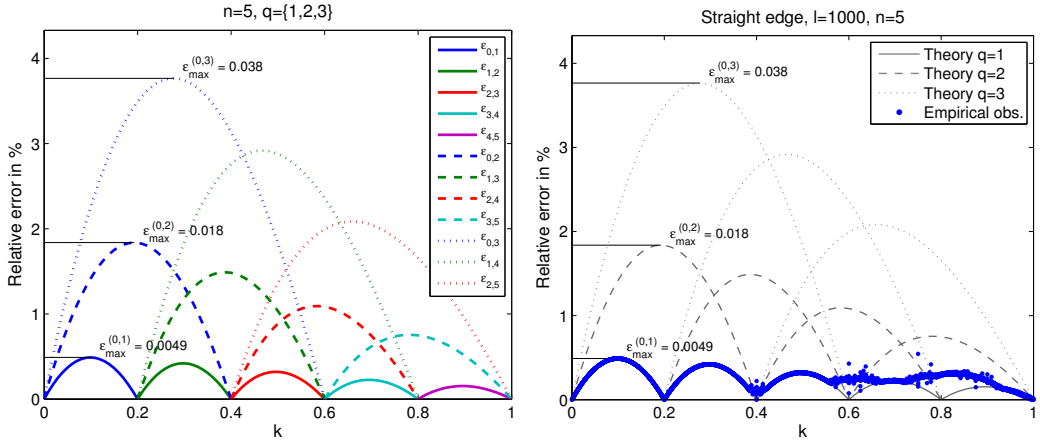
which gives the following optimal value for $\gamma_n^{(i,j)}$

$$\gamma_n^{(i,j)} = \frac{2(j - i)n}{(j - i)n + \sqrt{\left(\sqrt{n^2 + i^2}\sqrt{n^2 + j^2} - (n^2 + ij)\right)^2 + n^2(j - i)^2}} \,, \tag{8}$$

whereas, for two given vectors S_i and S_j, the maximal error is

$$|\varepsilon_n^{(i,j)}| = 1 - \gamma_n^{(i,j)}. \tag{9}$$

The derivation of this result is given Appendix A.



(a) Theoretical (b) Empirical

Figure 2: Relative error $\varepsilon_n^{(i,j)}(k)$, for $k \in [0,1]$, $\gamma_n = 1$, $n = 5$. The values $\varepsilon_{\max}^{(i,j)}$ are indicated. (a) Theoretically derived behavior of $\varepsilon_n^{(i,j)}(k)$, for $i \in \{0,1,2,3,4\}$, and $q = j - i \in \{1,2,3\}$. (b) Empirically observed values of $\varepsilon_n^{(i,j)}(k)$ 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).

If we assume that $j - i = q$ is constant, i.e., that vectors $\mathbf{S}_i$ and $\mathbf{S}_{i+q}$ are used for the approximation of a given line, then

$$\gamma_n^{(i,i+q)} = \frac{2qn}{qn + \sqrt{\left(\sqrt{n^2+i^2}\sqrt{n^2+(i+q)^2} - (n^2+i(i+q))\right)^2 + n^2q^2}} \,. \tag{10}$$

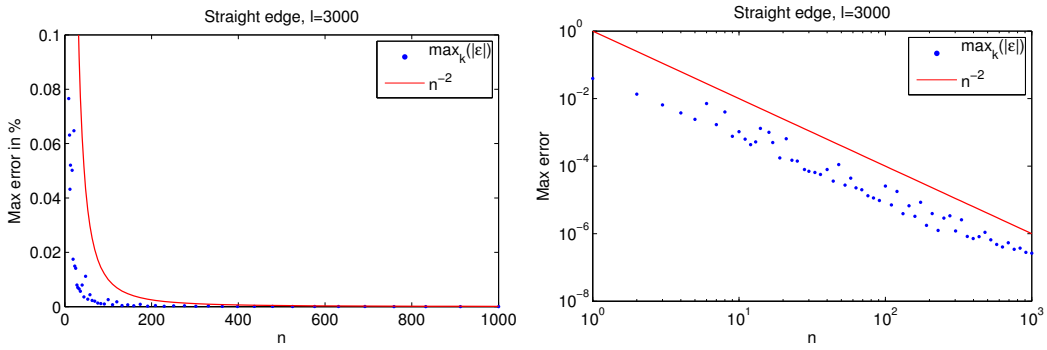
The value $\sqrt{n^2+i^2}\sqrt{n^2+(i+q)^2} - (n^2+i(i+q))$ decreases with i increasing. Consequently, $\gamma_n^{(i,i+q)}$ increases with i increasing and is the smallest for $i = 0$. This implies that the error $|\varepsilon_n^{(i,i+q)}|$ is the largest for $i = 0$. To minimize the maximal error for $i \in \{0,1,\ldots,n\}$, it is sufficient to observe $\gamma_n^{(0,q)}$:

$$\gamma_n^{(0,q)} = \frac{2q}{q + \sqrt{(\sqrt{n^2+q^2} - n)^2 + q^2}} = \frac{2}{1 + \sqrt{(\sqrt{(\frac{n}{q})^2+1} - \frac{n}{q})^2 + 1}} \,. \tag{11}$$

The value $\gamma_n^{(0,q)}$ is denoted γ_n and is used as optimal scale factor in the length estimation defined by Equation (5). The corresponding maximal estimation error, $|\varepsilon_n|$, is given by

$$|\varepsilon_n| = |\varepsilon_n^{(0,q)}| = 1 - \gamma_n^{(0,q)} \,. \tag{12}$$

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



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

Figure 3: Asymptotic behavior of the maximal error for straight edge length estimation using $\gamma_n = \gamma_n^{(0,1)}$; theoretical (line) and empirical (points) results.

It can be noticed that for $n = 1$, corresponding to a binary image, and with $q = 1$, being the only option for $q \leq n$, Equation (11) provides that the scale factor γ_1 that minimizes the maximal length estimation error is

$$\gamma_1 = \gamma_1^{(0,1)} = \frac{2}{1 + \sqrt{(\sqrt{2}-1)^2 + 1}} \approx 0.9604 \tag{13}$$

which is a well known optimal result, [7]. The corresponding estimation error is $|\varepsilon_1| = 1 - \gamma_1 \leq 4\%$.

More generally, observing the estimation error corresponding to $\gamma_n = \gamma_n^{(0,q)}$ as a function of n, as given by Equation (11), we conclude that for any constant $q \ll n$

$$|\varepsilon_n| = \mathcal{O}(n^{-2}) \,, \tag{14}$$

which gives an asymptotic upper bound for the estimation error as $n \to \infty$. See Appendix B for derivation of this result.

This behavior is also illustrated in Figure 3, where a result of an empirical study, performed on digital straight line segments, is presented. For a range of values of n, $n \in [1, 1000]$, maximal error of the length estimate $\hat{l}$ is presented. Observed edges are of length 3000 and for each observed value n, 1000 lines with randomly chosen $k, m \in [0, 1]$ are generated, and the maximal length estimation error is plotted as one point in the graph. Asymptotic bound, $\frac{1}{n^2}$, is also presented, for visual comparison with the empirical results.

The value of $q = j - i$, appearing in the factor γ_n, deserves some more attention. It reflects the difference in slope of the vectors $\mathbf{S}_i$ and $\mathbf{S}_j$ used in the linear combination, Equation (4). Their slopes $\frac{i}{n}$ and $\frac{j}{n}$ correspond to the column differences $\tilde{d}_c$ of the image $I = \mathcal{D}^n(H)$. A larger value of q leads to a larger error (according to Eq. (11)). For the purpose of minimization w.r.t. the maximal error, i.e., we observe the worst case situation, the value q should therefore reflect the range of possible column differences $\tilde{d}_c$ for any n-level digitization of a halfplane edge for any given slope $k \in [0, 1]$

To determine q for $n \geq 2$, we study how much $\tilde{d}_c$ can differ from the true slope of the edge, $d_c = k$. The column sum of a quantized image, denoted $\tilde{s}_c$ and

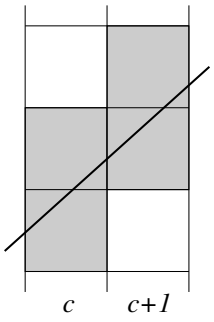


Figure 4: Not more than 4 pixels of two adjacent columns are intersected by an edge with $k \in [0,1]$ and can get quantized pixel values different from their corresponding non-quantized ones.

obtained as the sum of quantized pixel values, may differ from s_c, corresponding to the sum of non-quantized pixel values. Up to two pixels per column may be affected by quantization, since not more than two pixels of one column c are intersected by an edge with $k \in [0,1]$, and can get values different from 0 or 1 (see example in Figure 4). Therefore it holds that $\tilde{s}_c \in (s_c - \frac{1}{n}, s_c + \frac{1}{n}]$ and correspondingly $\tilde{d}_c \in (d_c - \frac{2}{n}, d_c + \frac{2}{n})$. That implies that an edge of slope $k = d_c$ may in the general case be estimated by local steps of *up to four* consecutive slope values $\tilde{d}_c$ from $\mathcal{Q}_n$, such that $\frac{i}{n} < \frac{i+1}{n} < k < \frac{i+2}{n} < \frac{i+3}{n}$.

Assume the worst case situation, i.e., only $\frac{i}{n}$ and $\frac{i+3}{n}$ are used in all steps along the observed edge. Then $q = 3$ and, according to Equation (11), the optimal scale factor is:

$$\gamma_n^{(0,3)} = \frac{6}{3 + \sqrt{(\sqrt{n^2+9} - n)^2 + 9}} \; . \tag{15}$$

This leads to the maximal estimation error for the general case:

$$|\varepsilon_n| = 1 - \gamma_n^{(0,3)} \; . \tag{16}$$

Combining this result with the asymptotic convergence as $n \to \infty$, Equation (14), the "multi-gray-level" convergence of the length estimate follows straightforwardly:

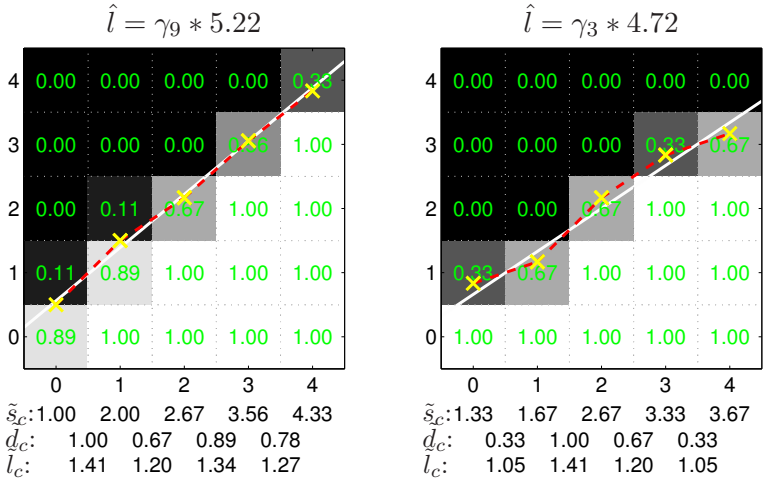
Theorem 1. *Let $H : y \leq kx + m, k \in [0,1]$ be a halfplane and let the length of a straight line segment $y = kx + m$ for $x \in [0, N]$ be estimated by using Formula* (5)*. Then for the maximal estimation error, Equation* (6)*, it holds that*

$$\lim_{n \to \infty} |\varepsilon_n| = 0 \; . \tag{17}$$

However, for small n, the worst case situation with $q = 3$ does not appear. As already noticed, for $n = 1$ only two different slopes are available, and thus q cannot be greater than 1.

We prove in Appendix C that q relates to n as follows:

$$q \leq \begin{cases} 1 & , \quad n \leq 2, \\ 2 & , \quad 3 \leq n \leq 8, \\ 3 & , \quad n \geq 9 \end{cases} \tag{18}$$



(a) $y = 0.83x + 0.56$, $n = 9$ (b) $y = 0.67x + 0.66$, $n = 3$

Figure 5: Two examples where the upper bounds of q, in relation (18), are reached. a) The slope of an edge $y \leq 0.83x + 0.56$, digitized using $n = 9$ quantization levels, is estimated by 4 different values ($q = 3$). b) The slope of an edge $y \leq 0.67x + 0.66$, digitized using $n = 3$ quantization levels, is estimated by 3 different values ($q = 2$).

Examples where the upper bounds of q, in relation (18), are reached do exist, two such are given in Figure 5.

Despite explicit examples that $q = 3$ is required for $n > 8$, empirical studies show that the situations where more than two different slopes appear in the estimation are very rare in practice. Observing the plot in Figure 2(b), we see that very few edges have an unscaled error greater than $\varepsilon_{\max}^{(0,1)}$, which supports the idea that computing γ_n using $q = 1$ is a better choice in practice, than using the theoretically derived worst case value $q = 3$. To use $\gamma_n^{(0,1)}$ rather than $\gamma_n^{(0,3)}$ is also observed as a better choice in our empirical test performed on other shapes (see Section 7).

In Figure 6 we show the effect of different choices of γ_n to the estimation error. It is apparent that choosing γ_n by using $q = 1$ in Equation (11) provides the best empirical result, even though $q = 2$ is the theoretically required choice ("dealing" with the worst case situation error for $n = 5$). This is in accordance with the existence of (very) few "outliers", exceeding the $\varepsilon_{\max}^{(0,1)}$ for an unscaled estimate $\gamma_5 = 1$.

6 Local computation of length

Local computation of the edge length relies on local computation of the $\tilde{d}_c$ values of Equations 3. How to estimate the slope of the observed straight edge from a small neighborhood is the main topic of this section, particularly considering the effects of quantization on the estimation algorithm. We also analyze the situations when $k \notin [0, 1]$ and conclude the section with pseudo-code for an explicit algorithm to compute the suggested boundary length estimate.

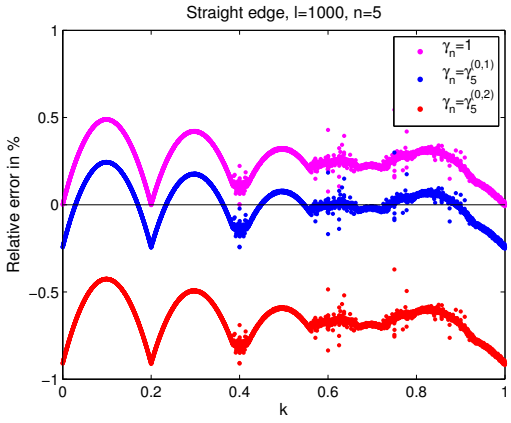


Figure 6: Relative error $\varepsilon_{i,j}(k)$, for different values of γ_n, $k \in [0,1]$, $n = 5$.

6.1 Local computation of length based on 3×2 configurations

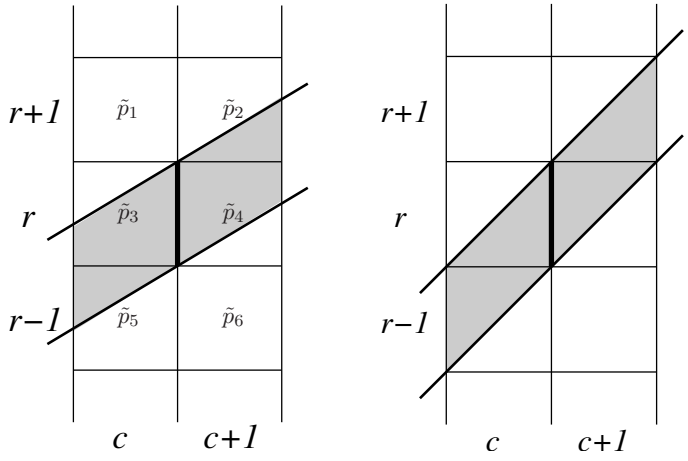
For lines of a slope $k \in [0,1]$, each value $\tilde{d}_c$ depends on at most six pixels, located in a 3×2 rectangle; the remaining pixels in the column pair do not contribute to the difference, neighboring pixels in each row being the same. The difference $\tilde{d}_c$ of two columns can thus be computed 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) (see Figure 7). We need to formulate a criterion to detect which $D_{(c,r)}$, for each observed pair of neighboring columns, is intersected by the straight edge.

Let $\tilde{d}_{(c,r)}$ denote the difference between the sum of (quantized) pixel values in the right and the left column of $D_{(c,r)}$. Let the (non-quantized) intersection of the halfplane edge, having slope $k \in [0,1]$, with the line $x = c + \frac{1}{2}$ be denoted by u; $u = k(c + \frac{1}{2}) + m$. If $r - \frac{1}{2} \leq u \leq r + \frac{1}{2}$, i.e., if the edge intersects the thick vertical line in Figure 7, then the halfplane edge does not intersect any pixels of the two observed neighboring columns that are outside of $D_{(c,r)}$. That implies that all pixels in column c and $c+1$ above and below $D_{(c,r)}$ are row-wise equal and do not contribute to $\tilde{d}_c$; the fulfillment of the condition $u \in [r - \frac{1}{2}, r + \frac{1}{2}]$ provides that the difference $\tilde{d}_c$ is equal to $\tilde{d}_{(c,r)}$.

This condition, however, cannot be directly checked, since knowledge of the real line, and therefore its intersection u with $x = c + \frac{1}{2}$ is not available. We instead estimate the intersection of the halfplane edge with the line $x = c + \frac{1}{2}$ from the pixel values of the configuration. We refer to the quantized pixel values of a configuration $D_{(c,r)}$ as $\tilde{p}_i$, $i = 1, 2, \ldots, 6$, indexed row-wise, top to bottom from left to right (see Figure 7(a)). The quantized value $\tilde{p}_i$ may differ from p_i, the corresponding true coverage of a pixel i, of a non-quantized coverage digitization, by up to $\frac{1}{2n}$ (n being the number of equally spaced quantization levels).

Observing that

$$u = k(c + \frac{1}{2}) + m = \frac{1}{2} \int_{c-\frac{1}{2}}^{c+\frac{3}{2}} (kx + m)\, dx = \frac{1}{2} \sum_{i=1}^{6} p_i + (r - \frac{3}{2}) \,,$$



(a) $k = 0.6$ (b) $k = 1$

Figure 7: Regions where lines $y = kx + m$ with k, m such that $u = k(c + \frac{1}{2}) + m \in [r - \frac{1}{2}, r + \frac{1}{2}]$, intersect a 3×2 configuration $D_{(c,r)}$.

we approximate the value u with

$$\tilde{u} = \frac{1}{2} \sum_{i=1}^{6} \tilde{p}_i + (r - \frac{3}{2}) \, .$$

Due to quantization, $\tilde{u} \in (u - \frac{1}{n}, u + \frac{1}{n}]$. Consequently, $\tilde{u} \in [r - \frac{1}{2}, r + \frac{1}{2}]$ does not imply $u \in [r - \frac{1}{2}, r + \frac{1}{2}]$. However, it can be proved that the fulfillment of the condition $\tilde{u} \in [r - \frac{1}{2}, r + \frac{1}{2}]$ is sufficient to assure that the difference $\tilde{d}_{(c,r)}$ computed from the corresponding $D_{(c,r)}$ provides the correct column difference $\tilde{d}_c$; see Lemma 1 in Appendix D. Therefore, the computation of the length $\hat{l}$ can be performed by using local length contributions derived from 3×2 pixel configurations, instead of from differences of full column sums.

The following formula provides the same edge length estimate $\hat{l}$ as Equation (3). The total length is computed as a sum of local length contributions $\hat{l}^D_{(c,r)}$ of all 3×2 regions of the image:

$$\hat{l}(I) = \sum_{D_{(c,r)} \subset I} \hat{l}^D_{(c,r)} \, , \quad \text{where} \quad \hat{l}^D_{(c,r)} = \begin{cases} \gamma_n \sqrt{1 + \tilde{d}^2_{(c,r)}} \, , & \tilde{u} \in \left(r - \frac{1}{2}, r + \frac{1}{2}\right) \\ \frac{\gamma_n}{2} \sqrt{1 + \tilde{d}^2_{(c,r)}} \, , & \tilde{u} = r \pm \frac{1}{2} \\ 0 \, , & \text{otherwise} \, . \end{cases} \tag{19}$$

Remark: The boundary cases, $\tilde{u} = r \pm \frac{1}{2}$, is treated so that a symmetric split of the edge contribution between the two (vertically) adjacent 3×2 configurations is considered.

6.2 Local computation of length based on 3×3 configurations

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

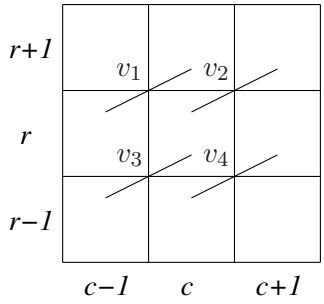


Figure 8: Edge localizations at the boundary points of the $\tilde{u}$-intervals in a 3×3 configuration $T_{(c,r)}$

the method, we suggest to not use two different region sizes for the two situations ($|k| \leq 1$ 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 to the 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, top to bottom and from left to right. That means that 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 $D_{(c,r)}$.

An extension of Equation (19) to estimations from 3×3 configurations can be made so that each 3×3 configuration is assigned a local length value that is the average of the length values of the two contained 3×2 sub-configurations. However, it is not necessary to treat the left and the right sub-configurations equally. To provide a more accurate localization of the edge length contribution with respect to the observed central pixel of the 3×3 configuration we propose the following:

Observe the border cases for the interval containing $\tilde{u}$, $\tilde{u} = r \pm \frac{1}{2}$, indicated with v_1, v_2, v_3, and v_4 in Figure 8. For edges of slope $k \in [0, 1]$, the vertexes v_2 and v_3 are intersected when the line passes through the central pixel of the configuration, whereas when v_1 and v_4 are intersected, the line does not pass through the central pixel. We therefore include, for a configuration $T_{(c,r)}$, the upper point for the 3×2 sub-configurations to the right (v_2), and the lower point for the 3×2 sub-configurations to the left (v_3), in the local edge length contribution. We assign the upper end point for the "left side" (v_1), and the lower end point for the "right side" (v_4), to neighboring configurations. The above can be summarized to the following local edge length $\hat{l}^T_{(c,r)}$, assigned to one 3×3 configuration $T_{(c,r)}$:

$$\hat{l}^T_{(c,r)} = \hat{l}_l + \hat{l}_r \text{ , where} \tag{20}$$

$$\hat{l}_l = \begin{cases} \frac{\gamma_n}{2}\sqrt{1 + d^2_{(c-1,r)}}\,, & \tilde{u}_{c-1} \in \left[r - \frac{1}{2}, r + \frac{1}{2}\right) \\ 0\,, & \text{otherwise}\,, \end{cases}$$

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

Finally,

$$\hat{l}^T(I) = \sum_{T_{(c,r)} \subset I} \hat{l}^T_{(c,r)} \tag{21}$$

is the length estimate over the image I. This formula gives the same edge length estimate as Equations (3) and (19), except for the leftmost and the rightmost columns of the image which are only included to 50% here.

6.3 Rotations of the plane

Let us now observe a general situation – a 3×3 configuration $T_{(c,r)} \subset I$, where the image $I = \mathcal{D}^n(H)$ is an n-level quantized coverage digitization of a halfplane $H : y \leq kx + m$ or $H : y \geq kx + m$, where the slope k is in $[-\infty, \infty]$. In the following we describe how to *isometrically* transform every such configuration $T_{(c,r)}$ so that the obtained configuration $T'_{(c,r)}$ corresponds to that of a halfplane $H' : y \leq k'x + m'$, such that $k' \in [0, 1]$. In that way we provide applicability of the estimation formula (21), derived in the previous subsection, to the general case.

If the slope k of the edge $y = kx + m$ of the halfplane H is available and if it is clear which one of the two halfplanes, $y \leq kx + m$, or $y \geq kx + m$, H denotes, the transformation to be applied to a configuration $T_{(c,r)}$ is easy to determine: If $y \geq kx + m$, symmetry w.r.t. the line $y = r$ is applied to $T_{(c,r)}$; If $k < 0$, then symmetry w.r.t. the line $x = c$ is applied; Finally, if $|k| > 1$, symmetry w.r.t. the line $x + y = r + c$ is applied. To obtain $T'_{(c,r)}$, with $k' \in [0, 1]$, the sequence of the described symmetries is applied to $T_{(c,r)}$, in the order as they are listed above, upon the fulfillment of the respective conditions.

Unfortunately, neither the value of k, nor the knowledge regarding the inequality ("side" of the edge line) defining the halfplane, are known. We have to extract that information from the local configuration $T_{(c,r)}$.

To define criteria for selection of the appropriate transformation to be applied to an observed configuration $T_{(c,r)}$, we rely on the auxiliary statements given in Appendix E. They lead to the following statement:

For any 3×3 configuration $T_{(c,r)}$, containing quantized pixel values $\tilde{p}_i$ of an n-level quantized pixel coverage digitization of $H : y \leq kx + m$, $k \in [0, 1]$, the following (systems of) inequalities hold:

$$\tilde{p}_1 \leq \tilde{p}_4 \leq \tilde{p}_7 \,, \qquad \tilde{p}_2 \leq \tilde{p}_5 \leq \tilde{p}_8 \,, \qquad \tilde{p}_3 \leq \tilde{p}_6 \leq \tilde{p}_9 \,, \tag{22a}$$

$$\tilde{p}_1 \leq \tilde{p}_2 \leq \tilde{p}_3 \,, \qquad \tilde{p}_4 \leq \tilde{p}_5 \leq \tilde{p}_6 \,, \qquad \tilde{p}_7 \leq \tilde{p}_8 \leq \tilde{p}_9 \,, \tag{22b}$$

$$\tilde{p}_2 \leq \tilde{p}_4 \,, \qquad \tilde{p}_3 \leq \tilde{p}_5 \leq \tilde{p}_7 \,, \qquad \tilde{p}_6 \leq \tilde{p}_8 \,. \tag{22c}$$

By using, in addition, the systems of inequalities derived from those above by the rule of contraposition, the following conclusion can be made: If any of the inequalities in (22a) does not hold then $y \not\leq kx + m$. If any of the inequalities in (22b) or (22c) does not hold, then $k \notin [0, 1]$. If any of the inequalities in (22b) is strict, then $k > 0$, and if any of the inequalities in (22c) is strict, then $k < 1$.

When the systems of equalities hold in either (22b) or (22c), it is, in the quantized case (as opposed to the non-quantized), not possible to detect if $k \in [0, 1]$, or not. However, such situations (systems of equalities) provide invariance of the observed

configuration $T_{(c,r)}$ w.r.t. the particular symmetry that should (or should not) be applied.

As a consequence, the following rules for the transformation of $T_{(c,r)}$ to $T'_{(c,r)}$ can be formulated:

1. Let $\tilde{\alpha} = \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3$.
   If $\tilde{\alpha} > 0$, then H corresponds to $y \leq kx + m$.
   If $\tilde{\alpha} < 0$, then H is given by $y \geq kx + m$ and the 1st and the 3rd row of $T_{(c,r)}$ are to be exchanged.
   If $\tilde{\alpha} = 0$, then all rows of the observed configuration are equal and there is no reason to exchange them (no mater if H corresponds to $y \leq kx + m$ or to $y \geq kx + m$).

2. Let $\tilde{\beta} = \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7$.
   If $\tilde{\beta} > 0$, then $k > 0$.
   If $\tilde{\beta} < 0$, then $k < 0$ and the 1st and the 3rd column of $T_{(c,r)}$ should be exchanged.
   If $\tilde{\beta} = 0$, then all columns of the observed configuration are equal and there is no reason to exchange them (no mater if $k > 0$ or $k < 0$).

3. Let $\tilde{\delta} = \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6$.
   If $\tilde{\delta} > 0$, then $k < 1$.
   If $\tilde{\delta} < 0$, then $k > 1$ and the symmetry w.r.t. $x + y = r + c$ is to be performed.
   If $\tilde{\delta} = 0$, then $T_{(c,r)}$ is symmetric w.r.t. the line $x + y = r + c$ and there is no reason to apply symmetry w.r.t the line $x + y = r + c$ (no mater if $k > 1$ or $k < 1$).

This result is utilized in the algorithm for edge length estimation, described in the next Subsection.

6.4 An algorithm for edge length computation

The following algorithm is suggested 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, to object edge adjacent only, 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 (21).

Algorithm 1.

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

---

if $\tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 < \tilde{p}_1 + \tilde{p}_2 + \tilde{p}_3$ /* $y \geq kx + m$ */
  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 $1 \leq \tilde{u}_l < 2$
  $\tilde{d}_l = \tilde{s}_2 - \tilde{s}_1$
  $\hat{l}_l = \frac{\gamma_n}{2}\sqrt{1 + \tilde{d}_l^2}$
else
  $\hat{l}_l = 0$
endif

if $1 < \tilde{u}_r \leq 2$
  $\tilde{d}_r = \tilde{s}_3 - \tilde{s}_2$
  $\hat{l}_r = \frac{\gamma_n}{2}\sqrt{1 + \tilde{d}_r^2}$
else
  $\hat{l}_r = 0$
endif

$\hat{l}^T_{(c,r)} = \hat{l}_l + \hat{l}_r$

Observe that, instead of computing the step lengths locally, but by counting the number of occurrences of each type of local step, all floating point operations (γ_n and $\sqrt{x}$) can be postponed to one final length computation. Keeping quantized pixel values as integers and performing the division by n also at the very end, all local computations can be made using integer arithmetics. We have not presented the algorithm in that way in order to keep the description as clear as possible.

7 Evaluation of the estimator performance on synthetic test images

We suggest to use the method presented in this paper (Algorithm 1) also for estimating the length of the boundary (perimeter) of more general shapes. Since the theoretical results of previous sections are derived for long straight edges, and not for curved boundaries, potentially with sharp corners, we complement the paper with an empirical study performed on a set of synthetic test images.

To study the accuracy and stability of the method applied to convex and non-convex curves, we evaluate the presented algorithm with respect to the accuracy of length estimation on a set of synthetic objects digitized using pixel coverage digitization. We use the set of test shapes proposed in [13] (also used in e.g. [2]), containing convex and non-convex objects with known perimeter, see Figure 9. The

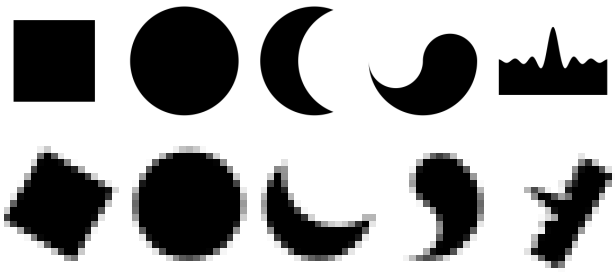
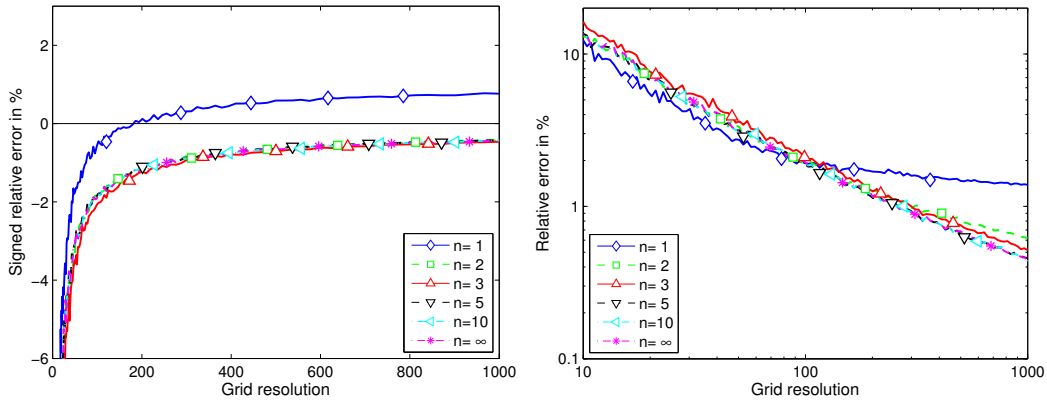


Figure 9: Pixel coverage digitizations of the test data set; one high resolution (grid resolution 512) (top) and one rotated low resolution (grid resolution 20) digitization (bottom).



(a) lin-lin scale

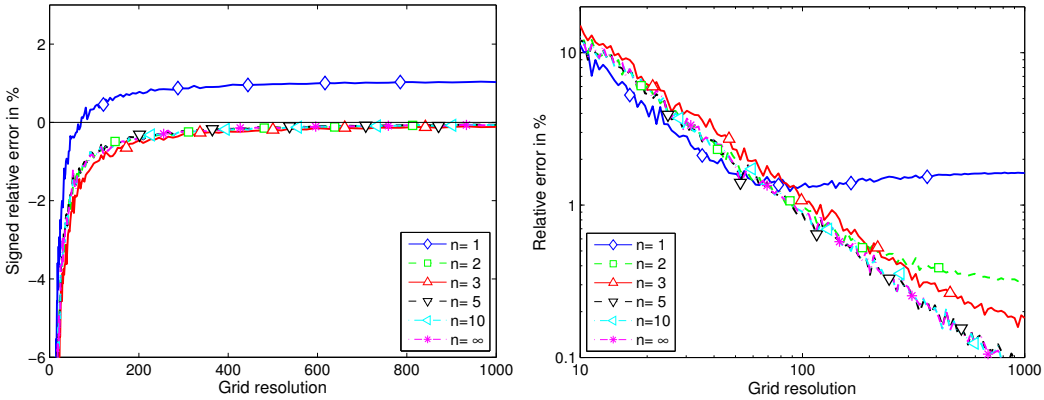
(b) log-log scale

Figure 10: Relative errors in percent for the test shapes (Figure 9) digitized at increasing resolution for 5 different quantization levels and a non-quantized ($n = \infty$).

test shapes are digitized at a range of resolutions, with random alignment in the digitization grid. Results of this evaluation are presented in Figure 10. We use $q = 1$ when computing the scale factor $\gamma_n = \gamma_n^{(0,q)}$, since that shows to provide empirically better results than using $q = 2$ or $q = 3$. For each test shape and for each of 120 resolutions in the range $[10, 1000]$, the pixel coverage digitizations for 10 different rotations and positions of the shape in the digital grid are computed. Quantization is performed with $n \in \{1, 2, 3, 5, 10\}$; the non-quantized digitization, indicated with $n = \infty$, is also included. The average performance of the method is plotted as a function of resolution. Since it is difficult to compute the true pixel coverage digitizations of the test shapes, they are instead approximated by 256×256 times subsampled Gauss digitizations.

As noticed in [13], the elongated peak of the Yin-yang curve slows down the convergence to the true value quite significantly. For complementary comparison we also show Figure 11, where the Yin-yang shape is removed from the test material.

As an additional test (also performed in [2]), we estimate the perimeter of a rotated square of size 128×128 and plot, in Figure 12, the estimate as a function of



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

Figure 11: Relative errors in percent when the Yin-yang shape is removed from the set of test shapes.

angle. As can be seen the rotational variation decreases rapidly with the increase of number of gray-levels, and is for $n \geq 3$ within $\pm 1\%$ 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 non-local multigrid convergent estimators evaluated in [2]. Note that this holds also for estimates based on relatively few gray-levels.

8 Summary

We present a novel and well performing local estimator of object boundary length. The method utilizes information existing in the gray-levels of an image, where pixel values are assumed to be proportional to the coverage of the respective pixels by the digitized shape. The method provides error-free estimates for non-quantized images. For quantized images, we derive the optimal scale factor to minimize the absolute estimation error.

The method is not multigrid convergent for any fixed number of gray-levels. However, the maximal error for the estimation of straight edges decreases as $\frac{1}{n^2}$ with the number of positive gray-levels n increasing. For an 8-bit image ($n = 255$) the maximal error for straight edges is less than 10^{-5} (Eq. (16), see also Figure 3).

The method is easy to implement and pseudo-code is given in Section 6.4. The algorithm is very fast, and the use of only a small local neighborhood makes the algorithm ideal for parallel implementations. All local computations and all comparisons can be done with integers only. The method, being based on a summation of local contributions of bounded magnitude, is by definition stable; a small change in the image renders a small change in the length estimate.

The method is derived for infinitely long straight edge segments. To show its applicability on more general shapes, we perform a statistical evaluation based on

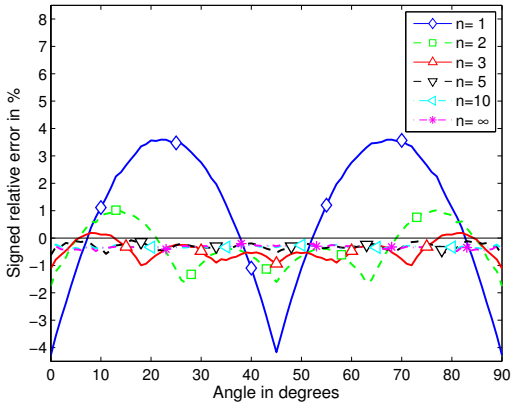


Figure 12: Relative errors in percent for a rotating square of size 128×128, from images with 5 different quantization levels and a non-quantized ($n = \infty$) one. Average of 20 random center locations per angle.

synthetic images. To facilitate easy comparison with other methods, we mimic the evaluation performed in [2]. The main conclusion we draw from such a comparison is that the suggested stable, fast, local, and discrete method provides length estimates well comparable with the best, often much more complex, non-local multigrid convergent methods.

Future work

An obvious path for future work is to extend the results of this paper to surface area measurements in 3D images. In that context, the use of a local method is even more beneficial in terms of complexity, both from an algorithmic and an implementation point of view. In addition, many multigrid convergent estimators for surface area have to solve a very difficult problem related to the task of decomposing a curved surface into plane segments. It has, for example, been shown that the task of finding a decomposition of a digital surface into a minimal number of plane segments is an NP-hard problem [14].

The work presented in this paper indicates the usefulness of utilizing gray-level information in images. Boundary length estimation is hardly the only problem of image analysis where the use of gray-level information is beneficial and the results of this work encourage our further research in this field.

Acknowledgments

DCI. Tom Barnaby and Prof. Samantha Ryan are greatly acknowledged for loads of inspiration. N. Sladoje is financially supported by the Ministry of Science of the Republic of Serbia through the Projects ON144018 and ON144029 of the Mathematical Institute of the Serbian Academy of Science and Arts.

Appendices

Appendix A Derivation of optimal scale factor $\gamma_n^{(i,j)}$

To minimize $|\varepsilon_n^{(i,j)}(k)|$ for $k \in [\frac{i}{n}, \frac{j}{n}]$, where $\varepsilon_n^{(i,j)}(k)$ is given by Equation (6), we observe the extrema of the function $\varepsilon_n^{(i,j)}(k)$ and chose $\gamma_n^{(i,j)}$ so that

$$\max_{k\in[\frac{i}{n},\frac{j}{n}]} \varepsilon_n^{(i,j)}(k) = -\min_{k\in[\frac{i}{n},\frac{j}{n}]} \varepsilon_n^{(i,j)}(k) \; .$$

The first derivative of $\varepsilon_n^{(i,j)}(k)$ is

$$\varepsilon_n'^{(i,j)}(k) = \gamma_n^{(i,j)} \frac{n(S_j - S_i) - k(jS_i - iS_j)}{(j-i)\sqrt{(1+k^2)^3}}$$

and

$$\varepsilon_n'^{(i,j)}(k) = 0 \quad \text{for} \quad k = k_{\max}^{(i,j)} = \frac{n(S_j - S_i)}{jS_i - iS_j} \; \in \left(\frac{i}{n}, \frac{j}{n}\right).$$

Noting that, for $\gamma_n^{(i,j)} > 0$, it holds that $\varepsilon_n'^{(i,j)}(k) > 0$ for $k \in \left(\frac{i}{n}, k_{\max}^{(i,j)}\right)$ and $\varepsilon_n'^{(i,j)}(k) < 0$ for $k \in \left(k_{\max}^{(i,j)}, \frac{j}{n}\right)$, we conclude that the function $\varepsilon_{i,j}(k)$ reaches a (local and global) maximum for $k = k_{\max}^{(i,j)}$, where

$$\varepsilon_n^{(i,j)}(k_{\max}^{(i,j)}) = \varepsilon_{\max}^{(i,j)} = \gamma_n^{(i,j)} \frac{\sqrt{n^2(S_j - S_i)^2 + (jS_i - iS_j)^2}}{j-i} - 1 \; . \tag{23}$$

Using $S_i = \sqrt{1 + (\frac{i}{n})^2}$, expression (23) becomes

$$\varepsilon_{\max}^{(i,j)} = \frac{\gamma_n^{(i,j)}}{(j-i)n} \sqrt{(\sqrt{n^2+i^2}\sqrt{n^2+j^2} - (n^2+ij))^2 + n^2(j-i)^2} - 1 \; . \tag{24}$$

The minima of $\varepsilon_n^{(i,j)}(k)$ are reached on the boundary of each interval, where for any $k = \frac{i}{n}$, Equation (6) gives

$$\varepsilon_{\min}^{(i,j)} = \gamma_n - 1 \; ,$$

which is the estimation error for any edge with a slope $k \in \mathcal{Q}_n$.

From $\varepsilon_{\max}^{(i,j)} = -\varepsilon_{\min}^{(i,j)}$, we derive (Equation (10))

$$\gamma_n^{(i,j)} = \frac{2(j-i)n}{(j-i)n + \sqrt{\left(\sqrt{n^2+i^2}\sqrt{n^2+j^2} - (n^2+ij)\right)^2 + n^2(j-i)^2}} \; .$$

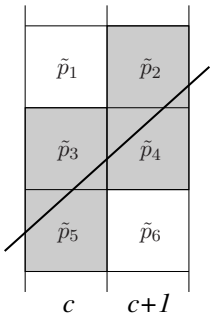


Figure 13: An edge with slope $k \in [0, 1]$ may intersect up to four pixels of a column pair $(c, c+1)$, leading to four quantized values that may differ from non-quantized pixel coverage values.

Appendix B Asymptotic behavior of the estimation error $|\varepsilon_n|$

Let us observe the error $|\varepsilon_n| = 1 - \gamma_n^{(0,q)}$, corresponding to $\gamma_n^{(0,q)}$ given by Eq. (11). The following holds:

$$\begin{aligned}
|\varepsilon_n| &= 1 - \gamma_n^{(0,q)} \\
&= \frac{\sqrt{(\sqrt{n^2+q^2}-n)^2+q^2}-q}{\sqrt{(\sqrt{n^2+q^2}-n)^2+q^2}+q} \\
&= \left(\frac{\sqrt{n^2+q^2}-n}{\sqrt{(\sqrt{n^2+q^2}-n)^2+q^2}+q}\right)^2 \\
&= \left(\frac{q^2}{\left(\sqrt{(\sqrt{n^2+q^2}-n)^2+q^2}+q\right)(\sqrt{n^2+q^2}+n)}\right)^2 \\
&\sim \frac{q^2}{16n^2} \text{ for } q \ll n \ .
\end{aligned}$$

Therefore, $\lim\limits_{n\to\infty} \dfrac{|\varepsilon_n|}{n^{-2}} < \infty$, i.e., $|\varepsilon_n| = \mathcal{O}(n^{-2})$.

Appendix C q for small values of n

In the general case, a single edge with slope $k \in [0, 1]$ may be approximated with local steps having up to four different slopes $\tilde{d}_c \in \mathcal{Q}_n$ such that $\frac{i}{n} < \frac{i+1}{n} < k < \frac{i+2}{n} < \frac{i+3}{n}$. That corresponds to $q = 3$. For such situation to occur, it is required that for at least one column pair $(c, c+1)$, it holds that $\tilde{s}_c < s_c - \frac{1}{2n}$ and $\tilde{s}_{c+1} > s_{c+1} + \frac{1}{2n}$, providing that $\tilde{d}_c > d_c + \frac{1}{n}$, while for some other column pair $(c', c'+1)$ it holds that $\tilde{s}_{c'} > s_{c'} + \frac{1}{2n}$ and $\tilde{s}_{c'+1} < s_{c'+1} - \frac{1}{2n}$, giving $\tilde{d}_{c'} < d_{c'} - \frac{1}{n}$.

Using the notation of Figure 13, $\tilde{d}_c$ can be computed as $\tilde{d}_c = \tilde{p}_2 + \tilde{p}_4 + 1 - \tilde{p}_3 - \tilde{p}_5$. The two above situations can then be formulated as:

i) for at least one column pair $(c, c+1)$ it holds that
$\tilde{p}_2 > p_2$, $\tilde{p}_4 > p_4$, $\tilde{p}_3 < p_3$, and $\tilde{p}_5 < p_5$;

ii) while at the same time for at least one other column pair $(c', c'+1)$ it holds that
$\tilde{p}_2 < p_2$, $\tilde{p}_4 < p_4$, $\tilde{p}_3 > p_3$, and $\tilde{p}_5 > p_5$.

For case (i) to appear, it must hold that $p_2 \geq \frac{1}{2n}$ and $p_5 < 1 - \frac{1}{2n}$. This imposes a lower (monotonically decreasing with n) limit of the slope k:

$$k \geq \frac{n + 1 + \sqrt{2n+1}}{2n} \ . \tag{25}$$

For case (ii) to appear in addition to case (i), it is required that $p_3 \geq \frac{1}{2n}$ and $p_4 < 1 - \frac{5}{2n}$ (or that $p_3 \geq \frac{5}{2n}$ and $p_4 < 1 - \frac{1}{2n}$). This imposes an upper (monotonically increasing with n) limit of the slope k:

$$k < \frac{n}{(1+\sqrt{5})^2} \ . \tag{26}$$

The system of inequalities, $\frac{n+1+\sqrt{2n+1}}{2n} \leq k < \frac{n}{(1+\sqrt{5})^2}$ has no solutions for $n \leq 8$. That implies that for such values of n not more than three different values of $\tilde{d}_c$ can appear for the same edge. For $n = 2$, $q = 1$ is enough: the only possible values of three slopes for $n = 2$ are $\{0, \frac{1}{2}, 1\}$, where for $k < \frac{1}{2}$ only the two first appear, and for $k > \frac{1}{2}$, only the two last appear. For $k = \frac{1}{2}$ the estimate $\tilde{k}$ is constant, equal to $\frac{1}{2}$.

Appendix D Condition for the detection of $D_{(c,r)}$, so that $\tilde{d}_{(c,r)} = \tilde{d}_c$

Lemma 1. *Let $H : y \leq kx + m$, $k \in [0, 1]$, be a halfplane. Let I be the n-level quantized pixel coverage digitization of H in the square integer grid. Let $D_{(c,r)}$ denote a 3×2 configuration of pixels in I, such that the left pixel of the middle row is located at (c, r). Let $\tilde{d}_{(c,r)}$ be the difference of the right and the left column in $D_{(c,r)}$, and $\tilde{d}_c$ be the difference of sums of all the (quantized) pixels in the columns $c+1$ and c in I. Let $u = k(c + \frac{1}{2}) + m$ and $\tilde{u} \in (u - \frac{1}{n}, u + \frac{1}{n}]$. Then*

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

Proof. Assume that $\tilde{u} \in [r - \frac{1}{2}, r + \frac{1}{2}]$. If $u \in [r - \frac{1}{2}, r + \frac{1}{2}]$, then $\tilde{d}_{(c,r)} = \tilde{d}_c$ holds (as discussed in Section 6.1). If $u \notin [r - \frac{1}{2}, r + \frac{1}{2}]$, that means that one of the two following cases holds:

$$(i) \qquad \tilde{u} \in \left(r + \tfrac{1}{2} - \tfrac{1}{n}, r + \tfrac{1}{2}\right] \text{ and } u \in \left(r + \tfrac{1}{2}, r + \tfrac{1}{2} + \tfrac{1}{n}\right) \ ; \tag{27}$$

$$(ii) \qquad \tilde{u} \in \left[r - \tfrac{1}{2}, r - \tfrac{1}{2} + \tfrac{1}{n}\right) \text{ and } u \in \left(r - \tfrac{1}{2} - \tfrac{1}{n}, r - \tfrac{1}{2}\right) \ . \tag{28}$$

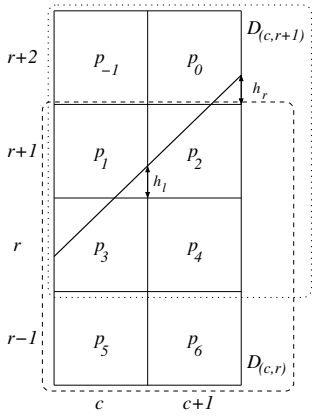


Figure 14: Configurations $D_{(c,r)}$ and $D_{(c,r+1)}$, and their four common pixels, contributing to $\tilde{d}_{(c,r)}$ and $\tilde{d}_{(c,r+1)}$.

Case (i) corresponds to the situation when the quantized value $\tilde{u}$ indicates that the appropriate configuration for calculating $\tilde{d}_c$ is $D_{(c,r)}$, while the corresponding non-quantized u indicates $D_{(c,r+1)}$ to be correct. Case (ii) corresponds to the situation when the quantized value $\tilde{u}$ indicates that $D_{(c,r)}$ should be used to calculate $\tilde{d}_c$, while its quantized value indicates that the appropriate configuration is $D_{(c,r-1)}$. We observe Case (i) and show that, under the assumption (27), $\tilde{d}_{(c,r)} = \tilde{d}_{(c,r+1)}$, by showing that the values below $D_{(c,r+1)}$ are all equal to 1, and that the values above $D_{(c,r)}$ are all equal to 0. In other words, only the four pixels common for both configurations $D_{(c,r)}$ and $D_{(c,r+1)}$ contribute to $\tilde{d}_c$. Since the vertical translation by ± 1 of the local edge does not influence the length estimate, the total edge length estimate is not affected by this possible "disagreement" of the values $\tilde{u}$ and u; any of $D_{(c,r)}$ and $D_{(c,r+1)}$ can be used. Analogous observations and conclusions can be made for Case (ii).

Assume that relations (27) hold. Since $u > r + \frac{1}{2}$, it holds that $(p_4 =) p_5 = p_6 = 1$ and therefore also $(\tilde{p}_4 =) \tilde{p}_5 = \tilde{p}_6 = 1$, see Figure 14. Since $u < r + \frac{3}{2}$, it holds that $p_{-1} = \tilde{p}_{-1} = 0$. Let us show that $\tilde{p}_0 = 0$ holds as well.

Using the notation of Figure 14, it holds that $p_0 = \frac{h_r^2}{2k}$. Since $u \in \left(r + \frac{1}{2}, r + \frac{1}{2} + \frac{1}{n}\right)$, and therefore $h_l = u - (r + \frac{1}{2}) < \frac{1}{n}$, for $k \in [0, 1]$ it holds that the value $h_r = k(c + \frac{3}{2}) + m - \left(r + \frac{3}{2}\right) = h_l + k - 1 \leq h_l < \frac{1}{n}$. Writing $k = h_r + 1 - h_l$, we get $p_0 = \frac{h_r^2}{2(h_r + 1 - h_l)}$, and $\frac{1}{2p_0} = \frac{1}{h_r} + \frac{1-h_l}{h_r^2} \geq \frac{1}{h_l} + \frac{1-h_l}{h_l^2} > n^2$. Thus $p_0 < \frac{1}{2n^2}$. Assuming $p_0 \geq \frac{1}{2n}$, we get $n < 1$, which is a contradiction. We conclude that $p_0 < \frac{1}{2n}$ and thus $\tilde{p}_0 = 0$, for all $n \geq 1$.

Equalities $\tilde{p}_5 = \tilde{p}_6 = 1$ and $\tilde{p}_{-1} = \tilde{p}_0 = 0$ ensure that only the common pixels of $D_{(c,r)}$ and $D_{(c,r+1)}$ contribute to the value $\tilde{d}_c = \tilde{d}_{c+1}$. Similar conclusion about $D_{(c,r)}$ and $D_{(c,r-1)}$ can be drawn if relations (28) hold. □

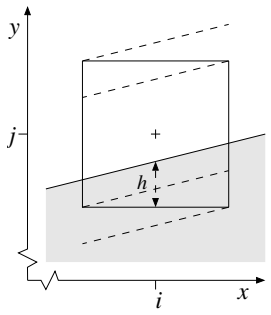


Figure 15: The coverage of a pixel at a position (i, j) by a halfplane $H : y \leq kx + m$ can be expressed as a function of $h = ki + m - \left(j - \frac{1}{2}\right)$.

Appendix E Lemmata for Subsection 6.3

Lemma 2. *Let $H : y \leq kx + m$, $k \in [0, 1]$, be a halfplane. Let I be the non-quantized pixel coverage digitization of H in the square integer grid. Then, for two pixels $p = p_{(c,r)}, q = q_{(i,j)} \in I$, located at (c, r) and (i, j), respectively, the following holds:*

$$c = i, r < j \quad \Rightarrow \quad p \geq q\,, \tag{29a}$$

$$c < i, r = j \quad \Rightarrow \quad p \leq q\,, \tag{29b}$$

$$c - r = i - j, c + r < i + j \quad \Rightarrow \quad p \geq q\,. \tag{29c}$$

Proof. Observe the function $I(i, j) : \mathbb{Z}^2 \to \mathbb{R}$ given by the non-quantized pixel coverage digitization, $I = \mathcal{D}(H)$. Let $h = ki + m - \left(j - \frac{1}{2}\right)$ for a pixel at a position (i, j), see Figure 15.

The the pixel coverage for the observed pixel can be computed as follows:

$$I(i,j) = I(h) \begin{cases} 0 & , & h & \leq -\frac{k}{2}\,, \\ \frac{(h+k/2)^2}{2k} = \frac{h^2}{2k} + \frac{h}{2} + \frac{k}{8} & , & -\frac{k}{2} < & h & < \frac{k}{2}\,, \\ h & , & \frac{k}{2} \leq & h & \leq 1 - \frac{k}{2}\,, \\ 1 - \frac{(1-h+k/2)^2}{2k} & , & 1 - \frac{k}{2} < & h & < 1 + \frac{k}{2}\,, \\ 1 & , & 1 + \frac{k}{2} \leq & h & . \end{cases} \tag{30}$$

We extend the function $I(h)$ to the real line $\mathbb{R}$, according to (30) and differentiate w.r.t. h.

$$I'(h) = \begin{cases} 0 & , & h & \leq -\frac{k}{2}\,, \\ \frac{h}{k} + \frac{1}{2} & , & -\frac{k}{2} < & h & < \frac{k}{2}\,, \\ 1 & , & \frac{k}{2} \leq & h & \leq 1 - \frac{k}{2}\,, \\ \frac{1-h}{k} + \frac{1}{2} & , & 1 - \frac{k}{2} < & h & < 1 + \frac{k}{2}\,, \\ 0 & , & 1 + \frac{k}{2} \leq & h & . \end{cases} \tag{31}$$

We conclude that $I'(h)$ is non-negative for any $k \in [0, 1]$ and, hence, that $I(h)$ is non-decreasing with h increasing. Computing $h_p = h(c, r)$ and $h_q = h(i, j)$ for the

preliminaries of (29), we obtain

$$\begin{aligned} c = i, r < j \quad &\Rightarrow \quad h_p > h_q \quad \Rightarrow \quad p \geq q \ , && \text{(32a)} \\ c < i, r = j \quad &\Rightarrow \quad h_p < h_q \quad \Rightarrow \quad p \leq q \ , && \text{(32b)} \\ c - r = i - j, c + r < i + j \quad &\Rightarrow \quad h_p > h_q \quad \Rightarrow \quad p \geq q \ . && \text{(32c)} \end{aligned}$$

□

Lemma 3. *Let $H : y \leq kx + m$, $k \in [0, 1]$, be a halfplane. Let I be the non-quantized pixel coverage digitization of H in the square integer grid. Then, for any 3×3 configuration $T \subset I$ with pixel values p_i, $i = 1, \ldots, 9$, indexed row-wise from left to right, the following (systems of) inequalities hold:*

$$\begin{aligned} p_1 \leq p_4 \leq p_7 \ , \qquad & p_2 \leq p_5 \leq p_8 \ , \qquad p_3 \leq p_6 \leq p_9 \ , && \text{(33a)} \\ p_1 \leq p_2 \leq p_3 \ , \qquad & p_4 \leq p_5 \leq p_6 \ , \qquad p_7 \leq p_8 \leq p_9 \ , && \text{(33b)} \\ p_2 \leq p_4 \ , \qquad & p_3 \leq p_5 \leq p_7 \ , \qquad p_6 \leq p_8 \ . && \text{(33c)} \end{aligned}$$

Proof. Follows immediately from Lemma 2. □

Remark: Similarly, the set of statements corresponding to the cases $k \leq 0$ and $k \geq 1$ can be proved. These statements consist of opposite inequalities to those given for p_i $i = 1, \ldots, 9$ in relations (33b) and (33c), respectively. It is easy to show that $k = 0$ is equivalent with the system of equalities in (33b), while $k = 1$ is equivalent with the system of equalities in (33c).

Lemma 4. *For any two real positive numbers p, q, and their corresponding quantized values $\tilde{p}$ and $\tilde{q}$, obtained by n-level quantization according to Definition 2, the following holds*

$$p < q \quad \Rightarrow \quad \tilde{p} \leq \tilde{q} \ .$$

In other words, ordering is preserved under quantization.

Proof. Assume $p < q$. Observe that $p \in [\tilde{p} - \frac{1}{2n}, \tilde{p} + \frac{1}{2n})$ and $q \in [\tilde{q} - \frac{1}{2n}, \tilde{q} + \frac{1}{2n})$. Assume that $\tilde{p} > \tilde{q}$. Then $\tilde{p} \geq \tilde{q} + \frac{1}{n}$, and

$$p \geq \tilde{p} - \frac{1}{2n} \geq \tilde{q} + \frac{1}{2n} > q$$

which is a contradiction. □

Appendix F List of most often appearing symbols

H A halfplane $y \leq kx + m$, observed for $x \in [0, N]$, $N \in \mathbb{Z}^+$.

$\mathbf{l}$ A vector representing the edge of H over $[0, N]$, $\mathbf{l} = (N, kN)$.

l The Euclidean length of the observed edge, $l = kN = \sum_{c=0}^{N-1} l_c$.

$\mathcal{D}(\mathcal{S})$ Non-quantized pixel coverage digitization of a shape S.

n The number of strictly positive gray-levels.

$\mathcal{D}^n(S)$ n-level quantized pixel coverage digitization of S.

$\mathcal{Q}_n$ The set of possible gray-levels of an n-level pixel coverage digitization.

I A (quantized or non-quantized) pixel coverage digitization of H over $[0, N]$.

s_c The sum of non-quantized pixel values in column c.

$\tilde{s}_c$ The sum of quantized pixel values in column c.

$d_c = s_{c+1} - s_c$

$\tilde{d}_c = \tilde{s}_{c+1} - \tilde{s}_c$

$l_c = \sqrt{1 + d_c^2}$

γ_n Scale factor for n-level quantized pixel coverage digitization.

$\hat{l}_c = \gamma_n \sqrt{1 + \tilde{d}_c^2}$

$\hat{l}$ The scaled length estimate, $\hat{l} = \sum_{c=0}^{N-1} \hat{l}_c$.

$\mathbf{S}_i$ A basis vector $\left(1, \frac{i}{n}\right)$.

S_i The Euclidean length of $\mathbf{S}_i$.

$q + 1$ The number of different $\tilde{d}_c$-values when estimating one given slope k.

$\varepsilon_n(k)$ signed relative error when approximating l with $\hat{l}$ using n positive gray-levels.

$D_{(c,r)}$ A 3×2 neighborhood where the middle left pixel is at (c, r).

$T_{(c,r)}$ A 3×3 neighborhood centered at (c, r).

p_i The non-quantized value of the i-th pixel in the given neighborhood.

$\tilde{p}_i$ The quantized value of the i-th pixel in the given neighborhood.

u The intersection of the halfplane edge with the line $x = c + \frac{1}{2}$; $u = k(c + \frac{1}{2}) + m$.

$\tilde{u}$ An approximation of u; $\tilde{u} = \frac{1}{2} \sum_{i=1}^{6} \tilde{p}_i + (r - \frac{3}{2})$.

References

[1] L. Dorst and A. W. M. Smeulders, "Length estimators for digitized contours," *Computer Vision, Graphics and Image Processing*, vol. 40, pp. 311–333, 1987.

[2] D. Coeurjolly and R. Klette, "A comparative evaluation of length estimators of digital curves," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 252–258, 2004.

[3] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, pp. 260–268, 1961.

[4] Z. Kulpa, "Area and perimeter measurement of blobs in discrete binary pictures," *Computer Graphics and Image Processing*, vol. 6, pp. 434–454, 1977.

[5] D. Proffit and D. Rosen, "Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges," *Computer Graphics and Image Processing*, vol. 10, pp. 318–332, 1979.

[6] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics and Image Processing*, vol. 34, pp. 344–371, 1986.

[7] B. Verwer, "Local distances for distance transformations in two and three dimensions," *Pattern Recognition Letters*, vol. 12, pp. 671–682, 1991.

[8] R. Klette and J. Žunić, "Multigrid convergence of calculated features in image analysis," *Journal of Mathematical Imaging and Vision*, vol. 13, pp. 173–191, 2000.

[9] M. Tajine and A. Daurat, "On local definitions of length of digital curves," in *Proceedings of the 11th International Conference on Discrete Geometry for Computer Imagery (DGCI)*, ser. LNCS, vol. 2886. Naples, Italy: Springer-Verlag, Nov. 2003, pp. 114–123.

[10] D. Eberly and J. Lancaster, "On gray scale image measurements: I. Arc length and area," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 6, pp. 538–549, 1991.

[11] P. Verbeek and L. van Vliet, "Estimators of 2D edge length and position, 3D surface area and position in sampled grey-valued images," *Bioimaging*, vol. 1, pp. 47–61, 1993.

[12] N. Kiryati and A. Bruckstein, "Grey levels can improve the performance of binary image digitizers," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 1, pp. 31–39, January 1991.

[13] R. Klette, V. Kovalevsky, and B. Yip, "Length estimation of digital curves," in *SPIE Proc. Vision Geometry VIII*, 1999, pp. 117–129, (expanded version available: Machine Graphics & Vision, vol. 9, pp. 673-703, 2000.).

[14] I. Sivignon and D. Coeurjolly, "Minimal decomposition of a digital surface into digital plane segments is NP-hard," in *Proceedings of the 13th International Conference on Discrete Geometry for computer Imagery (DGCI)*, ser. LNCS. Springer Verlag, Oct. 2006, pp. 674–685.