Measuring perimeter of a discrete object

Nataša Sladoje

Summer School of Image Processing, Vienna

July 11, 2008

Outline

Introduction

Measuring perimeter of discrete binary objects

To introduce myself...

Nataša Sladoje

Assistant Professor at the Department of Mathematics
Faculty of Engineering
University of Novi Sad, Serbia

B.Sc. in Mathematics, University of Novi Sad
M.SC. in Discrete Mathematics, University of Novi Sad
Ph.D. in Image Analysis, Centre for Image Analysis, Uppsala, Sweden

Once a student at SSIP (Budapest, 2003)
Twice a teacher at SSIP (Szeged, 2006 and 2007)

My department is in the CEEPUS network “Medical Imaging & Medical Information Processing” since 2005, when Serbia joined CEEPUS programme.

To introduce myself...

Nataša Sladoje

Assistant Professor at the Department of Mathematics
Faculty of Engineering
University of Novi Sad, Serbia

B.Sc. in Mathematics, University of Novi Sad
M.SC. in Discrete Mathematics, University of Novi Sad
Ph.D. in Image Analysis, Centre for Image Analysis, Uppsala, Sweden

Once a student at SSIP (Budapest, 2003)
Twice a teacher at SSIP (Szeged, 2006 and 2007)

My department is in the CEEPUS network "Medical Imaging & Medical Information Processing" since 2005, when Serbia joined CEEPUS programme.

To introduce myself...

Nataša Sladoje

Assistant Professor at the Department of Mathematics
Faculty of Engineering
University of Novi Sad, Serbia

B.Sc. in Mathematics, University of Novi Sad
M.SC. in Discrete Mathematics, University of Novi Sad
Ph.D. in Image Analysis, Centre for Image Analysis, Uppsala, Sweden

Once a student at SSIP (Budapest, 2003)
Twice a teacher at SSIP (Szeged, 2006 and 2007)

My department is in the CEEPUS network “Medical Imaging & Medical Information Processing” since 2005, when Serbia joined CEEPUS programme.

To introduce myself...

Nataša Sladoje

Assistant Professor at the Department of Mathematics
Faculty of Engineering
University of Novi Sad, Serbia

B.Sc. in Mathematics, University of Novi Sad
M.SC. in Discrete Mathematics, University of Novi Sad
Ph.D. in Image Analysis, Centre for Image Analysis, Uppsala, Sweden

Once a student at SSIP (Budapest, 2003)
Twice a teacher at SSIP (Szeged, 2006 and 2007)

My department is in the CEEPUS network “Medical Imaging & Medical Information Processing” since 2005, when Serbia joined CEEPUS programme.

...and to introduce the topic

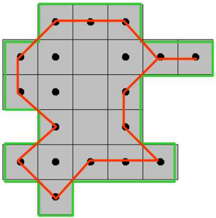
- The task of image analysis is to extract relevant information from images.
- Images contain discrete representations of real continuous objects.
- Our aim is usually to obtain information about continuous real objects, having available their discrete representations.
- Different numerical descriptors, such as area, perimeter, moments, of the objects are often of interest, for the tasks of shape analysis, classification, etc.
- Accurate and precise perimeter estimates of the real objects, based on their discrete representations, have been of interest for more than forty years, and many papers are published on that topic; the problem still attracts attention.

Formulation of the problem

Having a **discrete representation** of a real object, inscribed and digitized in an integer grid, **estimate** its perimeter (length of its border) with as small error as possible.

Small error provides not only accurate, but also precise estimates.
We obtain correct feature values - **accuracy**.
Repeated measurements provide very similar results - **precision**.

... and a way to solve it ...



...walk along the object boundary and accumulate **local step lengths**.

A digital boundary of an object (green) consists of **horizontal and vertical links** corresponding to pixel edges.

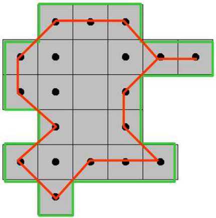
Its length is obtained by simply counting pixel edges traversed. Such a perimeter estimate leads to a large overestimate.

Freeman chain coding can be used instead. Eight directions of steps connecting consecutive pixel centres lead to **two step lengths**.

Accumulating the lengths of all the steps along the (inner) boundary of the object leads to perimeter estimation.

Boundary detection is required.

... and a way to solve it ...

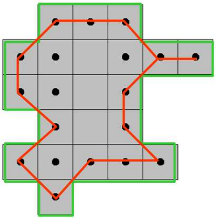


...walk along the object boundary and accumulate **local step lengths**.

A digital boundary of an object (green) consists of **horizontal and vertical links** corresponding to pixel edges.
Its length is obtained by simply counting pixel edges traversed. Such a perimeter estimate leads to a large overestimate.
Freeman chain coding can be used instead. Eight directions of steps connecting consecutive pixel centres lead to **two step lengths**.
Accumulating the lengths of all the steps along the (inner) boundary of the object leads to perimeter estimation.

Boundary detection is required.

... and a way to solve it ...

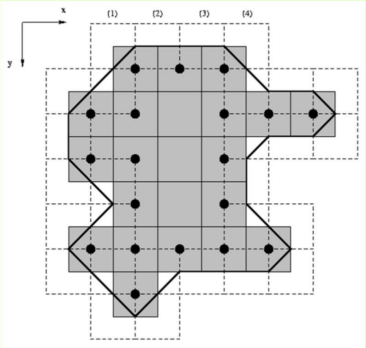


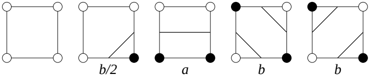
...walk along the object boundary and accumulate **local step lengths**.

A digital boundary of an object (green) consists of **horizontal and vertical links** corresponding to pixel edges.
Its length is obtained by simply counting pixel edges traversed. Such a perimeter estimate leads to a large overestimate.
Freeman chain coding can be used instead. Eight directions of steps connecting consecutive pixel centres lead to **two step lengths**.
Accumulating the lengths of all the steps along the (inner) boundary of the object leads to perimeter estimation.

Boundary detection is required.

Marching Squares technique

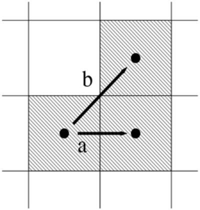




$4^2 = 16$ configurations reduce to 4 different perimeter contributions.

By checking local (2×2) pixel configurations, local perimeter contributions can be assigned.
Boundary detection simultaneously with perimeter estimation!

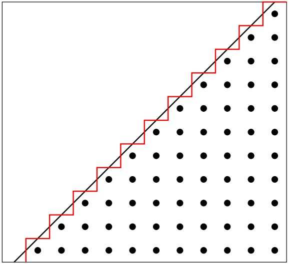
Important question - how to assign local step lengths



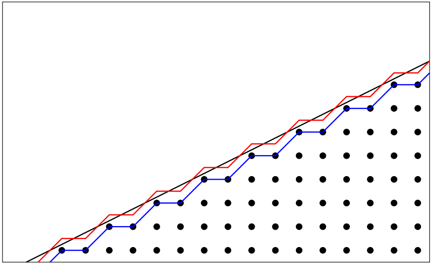
Freeman 1970: $a = 1, \quad b = \sqrt{2}$

Important question - how to assign local step lengths

If we use 4 directions.



If we use 8 directions.



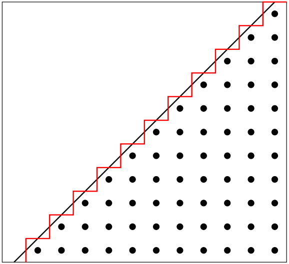
Digital edge $\sqrt{2}$ times longer than true edge.

Edge 1.08 times longer than true edge.

$a = 1$, $b = \sqrt{2}$ lead to an overestimate.

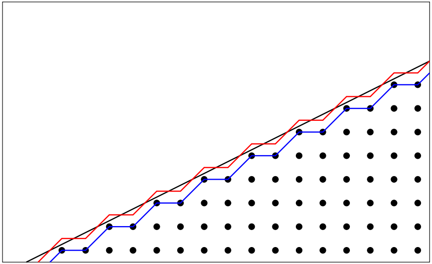
Important question - how to assign local step lengths

If we use 4 directions.



Digital edge $\sqrt{2}$ times longer than true edge.

If we use 8 directions.

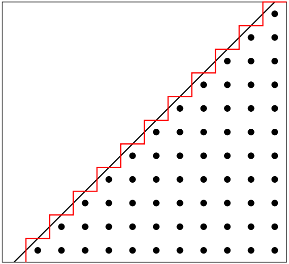


Edge 1.08 times longer than true edge.

$a = 1$, $b = \sqrt{2}$ lead to an overestimate.

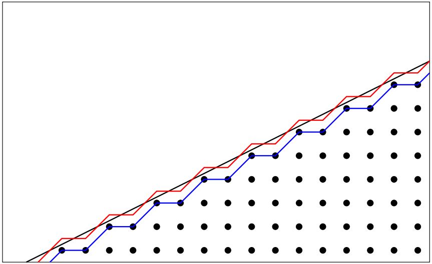
Important question - how to assign local step lengths

If we use 4 directions.



Digital edge $\sqrt{2}$ times longer than true edge.

If we use 8 directions.

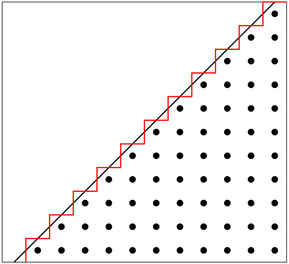


Edge 1.08 times longer than true edge.

$a = 1$, $b = \sqrt{2}$ lead to an **overestimate**.

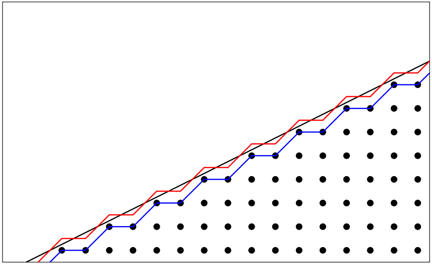
Important question - how to assign local step lengths

If we use 4 directions.



Digital edge $\sqrt{2}$ times longer than true edge.

If we use 8 directions.



Edge 1.08 times longer than true edge.
$a = 1$, $b = \sqrt{2}$ lead to an **overestimate**.

Optimization

- Observe straight lines of different directions.
- Restrict to lines with slopes $k \in [0, 1]$. Other cases follow by symmetries.
- Decide what error to minimize
  - The mean square error (MSE) minimization leads to estimators that, in average, perform well for lines of all (uniformly distributed) directions.
  - The maximal error minimization leads to estimator with a better "controllable" error. It is better suited for certain polygonal-shaped objects.
- Optimize step weights so that the chosen estimation error for length estimation of a straight segment is minimized. That leads to a scaling factor $\gamma \in (0, 1)$.

Optimization

- Observe straight lines of different directions.
- Restrict to lines with slopes $k \in [0, 1]$. Other cases follow by symmetries.
- Decide what error to minimize
  - The mean square error (MSE) minimization leads to estimators that, in average, perform well for lines of all (uniformly distributed) directions.
  - The maximal error minimization leads to estimator with a better "controllable" error. It is better suited for certain polygonal-shaped objects.
- Optimize step weights so that the chosen estimation error for length estimation of a straight segment is minimized. That leads to a scaling factor $\gamma \in (0, 1)$.

Optimization

- Observe straight lines of different directions.
- Restrict to lines with slopes $k \in [0, 1]$. Other cases follow by symmetries.
- Decide what error to minimize
  - The mean square error (MSE) minimization leads to estimators that, in average, perform well for lines of all (uniformly distributed) directions.
  - The maximal error minimization leads to estimator with a better “controllable” error. It is better suited for certain polygonal-shaped objects.
- Optimize step weights so that the chosen estimation error for length estimation of a straight segment is minimized. That leads to a scaling factor $\gamma \in (0, 1)$.

Optimization

- Observe straight lines of different directions.
- Restrict to lines with slopes $k \in [0,1]$. Other cases follow by symmetries.
- Decide what error to minimize
  - The mean square error (MSE) minimization leads to estimators that, in average, perform well for lines of all (uniformly distributed) directions.
  - The maximal error minimization leads to estimator with a better “controllable” error. It is better suited for certain polygonal-shaped objects.
- Optimize step weights so that the chosen estimation error for length estimation of a straight segment is minimized. That leads to a scaling factor $\gamma \in (0,1)$.

Optimal steps

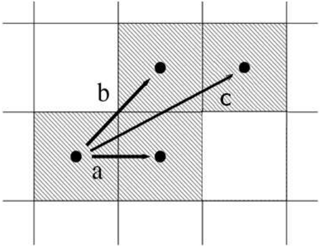
According to Verwer 1991,

- To minimize MSE, use:
  $a = 0.9481$ and $b = 1.3408$.
  Root Mean Square (RMS) Error is 2.33%.
- To minimize MaxErr, use:
  $a = 0.9604$ and $b = 1.3583$.
  Maximal Error is 3.95%.

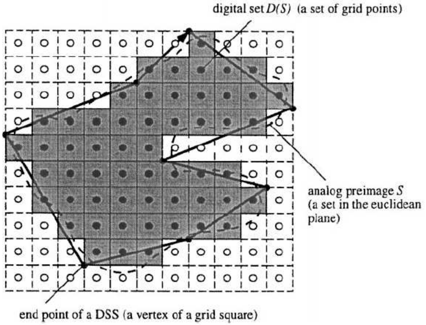
Observe: $b = a\sqrt{2}$ holds.

Further improvements

- Count corners in a crack code (Proffit and Rosen 1979) and subtract that number from the number of links.
- Extend neighbourhood (5×5 and further) and introduce more (longer) steps, such as "knight's move". Weights suggested: $a = 0.9865$, $b = 1.3951$, $c = 2.2058$. $MaxErr = 1.36\%$



Alternative approach



One possibility is to determine a maximum-length digital straight segment (DSS) approximation, and to compute its length.
Alternative polygonal approximations of the discrete object may be used.

Local and non-local estimators

Non-local estimators

Use information from larger (unbounded) regions of the image.

- Difficult to parallelize, if at all possible
- Often of higher complexity
- May suffer from stability problems
- Small change of the image requires global recomputation
- Multigrid convergent

Local estimators

Use information from a small region of the image to compute a local feature estimate. The global feature is computed by a summation of the local feature estimates over the whole image.

- Easy to implement
- Trivial to parallelize
- If a local change in the image, only that part has to be traversed to update the estimate
- Stable, if the local estimate is bounded
- Not multigrid convergent

Another view

- Work related to fuzzy shape analysis showed that information contained in grey levels can significantly improve performance of estimators.
- Results obtained for geometrical moments of fuzzy segmented shapes rely on a good theoretical background.
- Perimeter estimator based on fuzzy shape representation, my first PhD project task, performs really well, but only statistical studies have been performed.
- It felt tempting to try to improve perimeter estimation using theoretically supported method which relies on the knowledge how much the pixel is covered by an object.

Another view

- Work related to fuzzy shape analysis showed that information contained in grey levels can significantly improve performance of estimators.
- Results obtained for geometrical moments of fuzzy segmented shapes rely on a good theoretical background.
- Perimeter estimator based on fuzzy shape representation, my first PhD project task, performs really well, but only statistical studies have been performed.
- It felt tempting to try to improve perimeter estimation using theoretically supported method which relies on the knowledge how much the pixel is covered by an object.

Another view

- Work related to fuzzy shape analysis showed that information contained in grey levels can significantly improve performance of estimators.
- Results obtained for geometrical moments of fuzzy segmented shapes rely on a good theoretical background.
- Perimeter estimator based on fuzzy shape representation, my first PhD project task, performs really well, but only statistical studies have been performed.
- It felt tempting to try to improve perimeter estimation using theoretically supported method which relies on the knowledge how much the pixel is covered by an object.

Another view

- Work related to fuzzy shape analysis showed that information contained in grey levels can significantly improve performance of estimators.
- Results obtained for geometrical moments of fuzzy segmented shapes rely on a good theoretical background.
- Perimeter estimator based on fuzzy shape representation, my first PhD project task, performs really well, but only statistical studies have been performed.
- It felt tempting to try to improve perimeter estimation using theoretically supported method which relies on the knowledge how much the pixel is covered by an object.

What did we learn from others?

- Local estimators have sufficiently many advantages compared to global ones, to deserve to be studied further.
- Local estimators are, however, not multigrid convergent.
- “Grey levels can improve the performance of binary image digitizers” - N. Kiryati and A. Bruckstein, 1991.
- Work of Eberly and Lancaster, 1991, and Verbeek and van Vliet, 1993, showed attempts to use grey level information for length estimation, but appeared to be surprisingly “non-inspirative” for the scientific ancestors.

What did we learn from others?

- Local estimators have sufficiently many advantages compared to global ones, to deserve to be studied further.
- Local estimators are, however, not multigrid convergent.
- "Grey levels can improve the performance of binary image digitizers" - N. Kiryati and A. Bruckstein, 1991.
- Work of Eberly and Lancaster, 1991, and Verbeek and van Vliet, 1993, showed attempts to use grey level information for length estimation, but appeared to be surprisingly "non-inspirative" for the scientific ancestors.

What did we learn from others?

- Local estimators have sufficiently many advantages compared to global ones, to deserve to be studied further.
- Local estimators are, however, not multigrid convergent.
- “Grey levels can improve the performance of binary image digitizers” - N. Kiryati and A. Bruckstein, 1991.
- Work of Eberly and Lancaster, 1991, and Verbeek and van Vliet, 1993, showed attempts to use grey level information for length estimation, but appeared to be surprisingly “non-inspirative” for the scientific ancestors.

What did we learn from others?

- Local estimators have sufficiently many advantages compared to global ones, to deserve to be studied further.
- Local estimators are, however, not multigrid convergent.
- “Grey levels can improve the performance of binary image digitizers” - N. Kiryati and A. Bruckstein, 1991.
- Work of Eberly and Lancaster, 1991, and Verbeek and van Vliet, 1993, showed attempts to use grey level information for length estimation, but appeared to be surprisingly “non-inspirative” for the scientific ancestors.

...so, a natural decision was to...

work on development of a **new perimeter estimator**, that

- utilizes grey levels available in an image;
- has a strong theoretical foundation;
- is local;
- is fast;
- is simple to implement.

What follows now is a presentation of the result of this work, jointly performed with Dr. Joakim Lindblad.

High precision boundary length estimation by utilizing gray-level information

Area coverage digitization

Definition, non-quantized case

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

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, quantized case

Definition

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

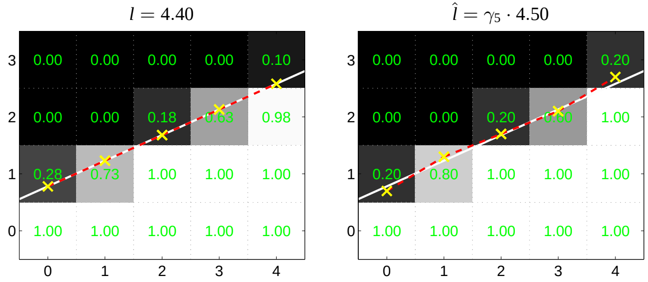
$$\mathcal{D}_n(S) = \left\{ \left((i,j), \frac{1}{n} \left\lfloor n \frac{A(p_{(i,j)} \cap S)}{A(p_{(i,j)})} + \frac{1}{2} \right\rfloor \right) \,\middle|\, (i,j) \in \mathbb{Z}^2 \right\},$$

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

$\mathcal{Q}_n = \{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n} = 1\}$ is the set of numbers representing area coverage values in n-level quantized area coverage digitization.

Slopes and difference of column sums

From continuous to quantized case



	0		1		2		3		4
s_c:	1.28		1.73		2.18		2.63		3.08
d_c:		0.45		0.45		0.45		0.45	
l_c:		1.10		1.10		1.10		1.10	

(a) Non-quantized (exact)

	0		1		2		3		4
$\tilde{s}_c$:	1.20		1.80		2.20		2.60		3.20
$\tilde{d}_c$:		0.60		0.40		0.40		0.60	
$\tilde{l}_c$:		1.17		1.08		1.08		1.17	

(b) Quantized $n = 5$

Figure: Edge length estimation based on the differences d_c of column sums s_c for a segment ($N = 4$) of a halfplane edge given by $y \leq 0.45x + 0.09$. The true halfplane edge is shown as a solid white line. The approximating local steps of slope d_c and $\tilde{d}_c$, respectively, are shown as dashed lines with $\times$ marking the ends of each step.

Estimation formula

The approximation formula for the length of the line is

$$\hat{l} = \gamma_n \sum_{c=0}^{N-1} \sqrt{1 + d_c^2}\ .$$

where the appropriate choice of the scale factor γ_n leads to a minimal estimation error.

Estimation formula

The approximation formula for the length of the line is

$$\hat{l} = \gamma_n \sum_{c=0}^{N-1} \sqrt{1 + d_c^2}\ .$$

where the appropriate choice of the scale factor γ_n leads to a minimal estimation error.

Optimization

Length estimate, in other words

...

The line segment l, represented as the vector $\mathbf{l} = (N, kN)$ with slope $k \in [0, 1]$ can be expressed as a linear combination of two of the vectors, $\mathbf{S}_i = (1, \frac{i}{n})$, $\mathbf{S}_j = (1, \frac{j}{n})$, $i, j \in \{0, 1, \ldots, n\}$, having slopes $k_i = \frac{i}{n}, k_j = \frac{j}{n} \in \mathcal{Q}_n$ such that $k_i \leq k \leq k_j$.
Its length on the interval $[0, N]$ can be estimated by

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

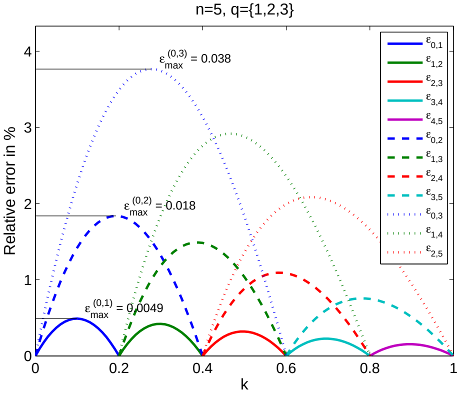
where $S_i = \sqrt{1 + \left(\frac{i}{n}\right)^2}$.

...and its error

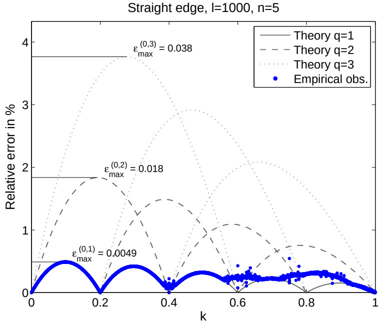
The relative error of the length estimation of the line segment with slope k, such that $k \in [\frac{i}{n}, \frac{j}{n})$:

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

This is how it looks, for $n = 5$



Empirical study of the estimation error



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.

Minimization of the absolute maximal error

Maximal error is minimized for the optimal value of γ_n given by

$$\gamma_n^q = \frac{2q}{q + \sqrt{(\sqrt{n^2+q^2} - n)^2 + q^2}}$$

where $q = j - i$.
The maximal error is

$$|\varepsilon| = 1 - \gamma_n^q.$$

Rounding errors (quantization!), lead to:

$$\begin{aligned} n > 8 \quad &\Rightarrow \quad q = 3 \\ 3 \le n \le 8 \quad &\Rightarrow \quad q = 2 \\ n \le 2 \quad &\Rightarrow \quad q = 1. \end{aligned}$$

Illustration

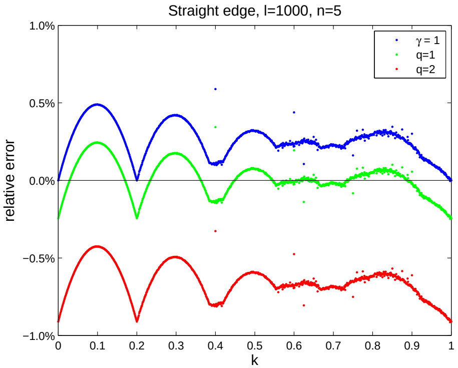
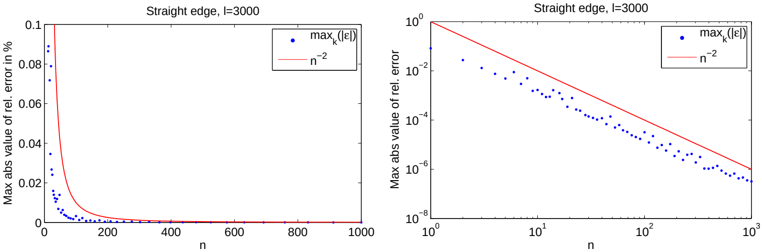


Figure: Relative error of length estimation. Non-optimized and optimized cases.

Asymptotic behaviour

Observing the estimation error as a function of n, we conclude that for any constant $q \ll n$

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



(a) lin-lin scale

(b) log-log scale

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

Local computations

To what to assign edge length

For lines of a slope $k \in [0, 1]$, each value d_c depends on at most six pixels, located in a 3×2 rectangle:

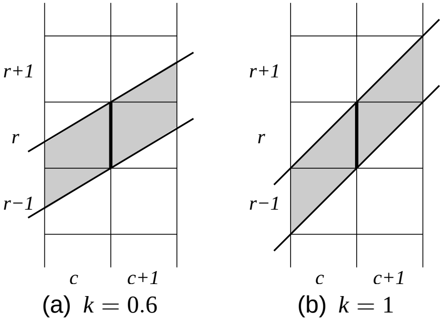


Figure: Regions where lines $y = kx + m$ with k, m such that $r - \frac{1}{2} \leq u = k(c + \frac{1}{2}) + m \leq r + \frac{1}{2}$, intersect a 3×2 configuration.

So, edge length is assigned to...

We need to check if the 3×2 neighbourhood of a pixel $p_{(c,r)}$ is "appropriate", by checking if $r - \frac{1}{2} \leq u = k(c + \frac{1}{2}) + m \leq r + \frac{1}{2}$, for a line $y = kx + m$

...whose equation we, unfortunately, do not know!

So, edge length is assigned to...

We need to check if the 3×2 neighbourhood of a pixel $p_{(c,r)}$ is “appropriate”, by checking if $r - \frac{1}{2} \le u = k(c + \frac{1}{2}) + m \le r + \frac{1}{2}$, for a line $y = kx + m$

...whose equation we, unfortunately, do not know!

To what to assign edge length, then? And what length?

Two pages are dedicated to a proof that we can **estimate** $u = k(c+\frac{1}{2}) + m$ by

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

and, in spite of rounding errors, successfully apply to detect “good” 3×2 configurations. Local contributions are calculated as

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

Rotations of the plane

What if $|k| \notin [0, 1]$

- If $|k| \notin [0, 1]$, we need 2×3 configuration to estimate the slope; when we exchange roles of the axes, we can apply the same algorithms as for the former case.
- Instead of changing size of configuration depending on k, we suggest to use 3×3 configurations in all cases.
- In any 3×3 configuration we (would like to) use the algorithms described for case $|k| \in [0, 1]$.
- To be able to do that, we need to apply isometric transformations to the 3×3 configurations correspondent to $|k| \notin [0, 1]$.

How to know what transformation to apply

Essentially,

- If $y \geq kx + m$, then symmetry w.r.t. the line $y = r$ should be applied;
- If $k < 0$, then symmetry w.r.t. the line $x = c$ should be applied;
- Finally, if $|k| > 1$, symmetry w.r.t. the line $x + y = r + c$ should be applied.

But, how to know...

- **IF** $y \geq kx + m$;
- **IF** $k < 0$;
- **IF** $|k| > 1$,

when we do not know analytical definition of the observed half-plane H, and when we are looking only at a small 3×3 neighbourhood??

Local conditions for isometries

Additional 3 pages of formulations and proofs that we can use **quantized** pixel values from a 3×3 neighbourhood in the following set of criteria:

Let

$$\begin{aligned}\tilde{\alpha} &= \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3 \,, \\ \tilde{\beta} &= \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7 \,, \\ \tilde{\delta} &= \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6 \,.\end{aligned}$$

If $\tilde{\alpha} > 0$ then $H : y \leq kx + m$. (all fine)
If $\tilde{\alpha} < 0$ then $H : y \geq kx + m$. Exchange the first and the third row.
If $\tilde{\alpha} = 0$ then all rows are equal. Leave as it is.
If $\tilde{\beta} > 0$ then $k > 0$. (all fine)
If $\tilde{\beta} < 0$ then $k < 0$. Exchange the first and the third column.
If $\tilde{\beta} = 0$ then all columns are equal. Leave as it is.
If $\tilde{\delta} > 0$ then $k < 1$. (all fine)
If $\tilde{\delta} < 0$ then $k > 1$. The symmetry w.r.t. $x + y = r + c$ is to be performed.
If $\tilde{\delta} = 0$ then “corners” are equal. Leave as it is.

Local conditions for isometries

Additional 3 pages of formulations and proofs that we can use **quantized** pixel values from a 3×3 neighbourhood in the following set of criteria:

Let

$$\begin{aligned} \tilde{\alpha} &= \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3 , \\ \tilde{\beta} &= \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7 , \\ \tilde{\delta} &= \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6 . \end{aligned}$$

If $\tilde{\alpha} > 0$ then $H : y \leq kx + m$. (all fine)
If $\tilde{\alpha} < 0$ then $H : y \geq kx + m$. Exchange the first and the third row.
If $\tilde{\alpha} = 0$ then all rows are equal. Leave as it is.
If $\tilde{\beta} > 0$ then $k > 0$. (all fine)
If $\tilde{\beta} < 0$ then $k < 0$. Exchange the first and the third column.
If $\tilde{\beta} = 0$ then all columns are equal. Leave as it is.
If $\tilde{\delta} > 0$ then $k < 1$. (all fine)
If $\tilde{\delta} < 0$ then $k > 1$. The symmetry w.r.t. $x + y = r + c$ is to be performed.
If $\tilde{\delta} = 0$ then “corners” are equal. Leave as it is.

Local conditions for isometries

Additional 3 pages of formulations and proofs that we can use **quantized** pixel values from a 3×3 neighbourhood in the following set of criteria:

Let

$$\begin{aligned}\tilde{\alpha} &= \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3 \,, \\ \tilde{\beta} &= \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7 \,, \\ \tilde{\delta} &= \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6 \,.\end{aligned}$$

If $\tilde{\alpha} > 0$ then $H : y \leq kx + m$. (all fine)
If $\tilde{\alpha} < 0$ then $H : y \geq kx + m$. Exchange the first and the third row.
If $\tilde{\alpha} = 0$ then all rows are equal. Leave as it is.
If $\tilde{\beta} > 0$ then $k > 0$. (all fine)
If $\tilde{\beta} < 0$ then $k < 0$. Exchange the first and the third column.
If $\tilde{\beta} = 0$ then all columns are equal. Leave as it is.
If $\tilde{\delta} > 0$ then $k < 1$. (all fine)
If $\tilde{\delta} < 0$ then $k > 1$. The symmetry w.r.t. $x + y = r + c$ is to be performed.
If $\tilde{\delta} = 0$ then “corners” are equal. Leave as it is.

Local conditions for isometries

Additional 3 pages of formulations and proofs that we can use **quantized** pixel values from a 3×3 neighbourhood in the following set of criteria:

Let

$$\begin{aligned}\tilde{\alpha} &= \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3 \,, \\ \tilde{\beta} &= \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7 \,, \\ \tilde{\delta} &= \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6 \,.\end{aligned}$$

If $\tilde{\alpha} > 0$ then $H : y \leq kx + m$. (all fine)
If $\tilde{\alpha} < 0$ then $H : y \geq kx + m$. Exchange the first and the third row.
If $\tilde{\alpha} = 0$ then all rows are equal. Leave as it is.
If $\tilde{\beta} > 0$ then $k > 0$. (all fine)
If $\tilde{\beta} < 0$ then $k < 0$. Exchange the first and the third column.
If $\tilde{\beta} = 0$ then all columns are equal. Leave as it is.
If $\tilde{\delta} > 0$ then $k < 1$. (all fine)
If $\tilde{\delta} < 0$ then $k > 1$. The symmetry w.r.t. $x + y = r + c$ is to be performed.
If $\tilde{\delta} = 0$ then "corners" are equal. Leave as it is.

Local conditions for isometries

Additional 3 pages of formulations and proofs that we can use **quantized** pixel values from a 3×3 neighbourhood in the following set of criteria:

Let

$$\begin{aligned}\tilde{\alpha} &= \tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_2 - \tilde{p}_3 \,, \\ \tilde{\beta} &= \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9 - \tilde{p}_1 - \tilde{p}_4 - \tilde{p}_7 \,, \\ \tilde{\delta} &= \tilde{p}_4 + \tilde{p}_7 + \tilde{p}_8 - \tilde{p}_2 - \tilde{p}_3 - \tilde{p}_6 \,.\end{aligned}$$

If $\tilde{\alpha} > 0$ then $H : y \leq kx + m$. (all fine)
If $\tilde{\alpha} < 0$ then $H : y \geq kx + m$. Exchange the first and the third row.
If $\tilde{\alpha} = 0$ then all rows are equal. Leave as it is.
If $\tilde{\beta} > 0$ then $k > 0$. (all fine)
If $\tilde{\beta} < 0$ then $k < 0$. Exchange the first and the third column.
If $\tilde{\beta} = 0$ then all columns are equal. Leave as it is.
If $\tilde{\delta} > 0$ then $k < 1$. (all fine)
If $\tilde{\delta} < 0$ then $k > 1$. The symmetry w.r.t. $x + y = r + c$ is to be performed.
If $\tilde{\delta} = 0$ then “corners” are equal. Leave as it is.

Length estimation from 3×3 configurations

After all, what to assign to 3×3 configurations?

Now we have two 3×2 sub-configurations within the observed 3×3 one, and each contributes with equal proportion to the local estimate:

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

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

$$\tilde{l}^{N}_{(c,r)} = \tilde{l}^{N_l}_{(c,r)} + \tilde{l}^{N_r}_{(c,r)}$$

Complete algorithm

Algorithm

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

---

if $\tilde{p}_7 + \tilde{p}_8 + \tilde{p}_9 < \tilde{p}_1 + \tilde{p}_2 + \tilde{p}_3$ /* $y \geq kx + m$ */
  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

$\tilde{s}_1 = \tilde{p}_1 + \tilde{p}_4 + \tilde{p}_7$
$\tilde{s}_2 = \tilde{p}_2 + \tilde{p}_5 + \tilde{p}_8$
$\tilde{s}_3 = \tilde{p}_3 + \tilde{p}_6 + \tilde{p}_9$

$\tilde{u}_l = (\tilde{s}_1 + \tilde{s}_2)/2$
$\tilde{u}_r = (\tilde{s}_2 + \tilde{s}_3)/2$

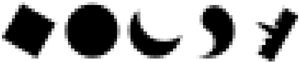
if $1 \leq \tilde{u}_l < 2$
  $\tilde{d}_l = \tilde{s}_2 - \tilde{s}_1$
  $\hat{l}_l = \frac{\gamma_n}{2}\sqrt{1 + \tilde{d}_l^2}$
else
  $\hat{l}_l = 0$
endif

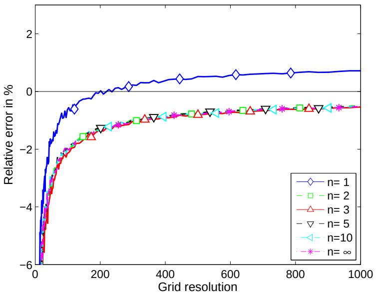
if $1 < \tilde{u}_r \leq 2$
  $\tilde{d}_r = \tilde{s}_3 - \tilde{s}_2$
  $\hat{l}_r = \frac{\gamma_n}{2}\sqrt{1 + \tilde{d}_r^2}$
else
  $\hat{l}_r = 0$
endif

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

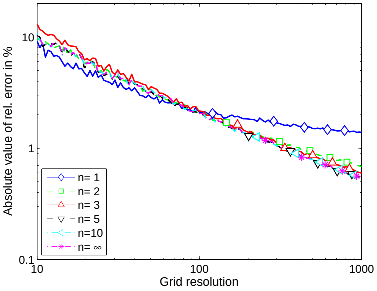
Evaluation and Examples

Estimation errors - synthetic images



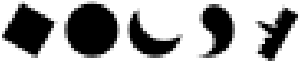


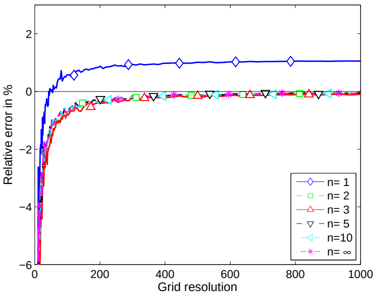
(a) lin-lin scale



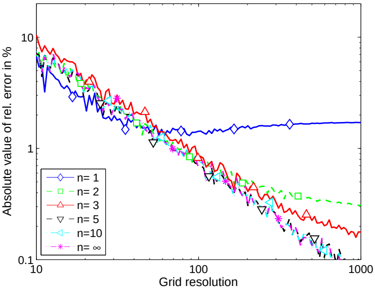
(b) log-log scale

Estimation errors - synthetic images





(c) lin-lin scale



(d) log-log scale

Rotating square

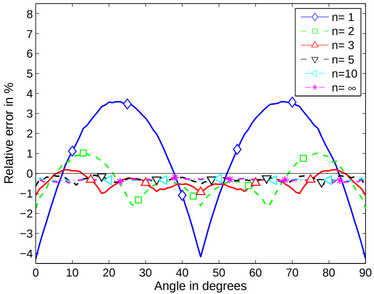


Figure: Relative errors in percent for a rotating square of size 128×128, from images with 5 different quantization levels and a non-quantized ($n = \infty$) one. Average of 20 random centre locations per angle.

A real object - straight edge

Acquisition of a test set

- We took 10 photos of the straight edge of a white paper on a black background at different angles using a digital camera in gray-scale mode.

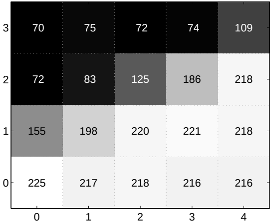


Figure: Close up of the straight edge of a white paper imaged with a digital camera. Estimated $k = 0.42$.

Length estimation - real images

A segmentation method required

- We need discrete representation of an object that corresponds well to the suggested area coverage digitization method.
- That assumes **one pixel thick** boundary, with grey levels in a range as wide as possible.
- That requires new segmentation methods.
- Segmentation methods are very application dependant.
- We suggest a simple one, based on double thresholding and morphological operations, that suits our purposes.

Length estimation - real images

A segmentation method required

- We need discrete representation of an object that corresponds well to the suggested area coverage digitization method.
- That assumes **one pixel thick** boundary, with grey levels in a range as wide as possible.
- That requires new segmentation methods.
- Segmentation methods are very application dependant.
- We suggest a simple one, based on double thresholding and morphological operations, that suits our purposes.

Length estimation - real images
A segmentation method required

- We need discrete representation of an object that corresponds well to the suggested area coverage digitization method.
- That assumes **one pixel thick** boundary, with grey levels in a range as wide as possible.
- That requires new segmentation methods.
- Segmentation methods are very application dependant.
- We suggest a simple one, based on double thresholding and morphological operations, that suits our purposes.

Length estimation - real images

A segmentation method required

- We need discrete representation of an object that corresponds well to the suggested area coverage digitization method.
- That assumes **one pixel thick** boundary, with grey levels in a range as wide as possible.
- That requires new segmentation methods.
- Segmentation methods are very application dependant.
- We suggest a simple one, based on double thresholding and morphological operations, that suits our purposes.

Length estimation - real images

A segmentation method required

- We need discrete representation of an object that corresponds well to the suggested area coverage digitization method.
- That assumes **one pixel thick** boundary, with grey levels in a range as wide as possible.
- That requires new segmentation methods.
- Segmentation methods are very application dependant.
- We suggest a simple one, based on double thresholding and morphological operations, that suits our purposes.

A simple segmentation method

1. Find a threshold couple, b and f, where pixels darker than b are to be assigned to the background, while pixels brighter than f are to be assigned to the foreground.
2. Provide that the pixels in between form a one pixel thick separating band.
3. Provide that the contrast between foreground and background, i.e., the difference $f - b$, is as large as possible.
4. The requirement of a thin gray border may be expressed using mathematical morphology; a 3×3 binary structuring element is used.
5.
   - Apply thresholding at a level f.
   - Apply dilation of a foreground.
   - Set threshold b at the max value of the region outside the dilated foreground.
   - Apply for all levels f.
   - Select the pair f, b providing the maximal difference $f - b$.

A simple segmentation method

1. Find a threshold couple, b and f, where pixels darker than b are to be assigned to the background, while pixels brighter than f are to be assigned to the foreground.
2. Provide that the pixels in between form a one pixel thick separating band.
3. Provide that the contrast between foreground and background, i.e., the difference $f - b$, is as large as possible.
4. The requirement of a thin gray border may be expressed using mathematical morphology; a 3×3 binary structuring element is used.
5.
   - Apply thresholding at a level f.
   - Apply dilation of a foreground.
   - Set threshold b at the max value of the region outside the dilated foreground.
   - Apply for all levels f.
   - Select the pair f, b providing the maximal difference $f - b$.

A simple segmentation method

1. Find a threshold couple, b and f, where pixels darker than b are to be assigned to the background, while pixels brighter than f are to be assigned to the foreground.
2. Provide that the pixels in between form a one pixel thick separating band.
3. Provide that the contrast between foreground and background, i.e., the difference $f - b$, is as large as possible.
4. The requirement of a thin gray border may be expressed using mathematical morphology; a 3×3 binary structuring element is used.
5.
   - Apply thresholding at a level f.
   - Apply dilation of a foreground.
   - Set threshold b at the max value of the region outside the dilated foreground.
   - Apply for all levels f.
   - Select the pair f, b providing the maximal difference $f - b$.

A simple segmentation method

1. Find a threshold couple, b and f, where pixels darker than b are to be assigned to the background, while pixels brighter than f are to be assigned to the foreground.
2. Provide that the pixels in between form a one pixel thick separating band.
3. Provide that the contrast between foreground and background, i.e., the difference $f - b$, is as large as possible.
4. The requirement of a thin gray border may be expressed using mathematical morphology; a 3×3 binary structuring element is used.
5.
   - Apply thresholding at a level f.
   - Apply dilation of a foreground.
   - Set threshold b at the max value of the region outside the dilated foreground.
   - Apply for all levels f.
   - Select the pair f, b providing the maximal difference $f - b$.

A simple segmentation method

1. Find a threshold couple, b and f, where pixels darker than b are to be assigned to the background, while pixels brighter than f are to be assigned to the foreground.
2. Provide that the pixels in between form a one pixel thick separating band.
3. Provide that the contrast between foreground and background, i.e., the difference $f - b$, is as large as possible.
4. The requirement of a thin gray border may be expressed using mathematical morphology; a 3×3 binary structuring element is used.
5.
   - Apply thresholding at a level f.
   - Apply dilation of a foreground.
   - Set threshold b at the max value of the region outside the dilated foreground.
   - Apply for all levels f.
   - Select the pair f, b providing the maximal difference $f - b$.

Segmentation of a straight edge

An example - segmentation of an image of an edge with slope $k = 0.42$.

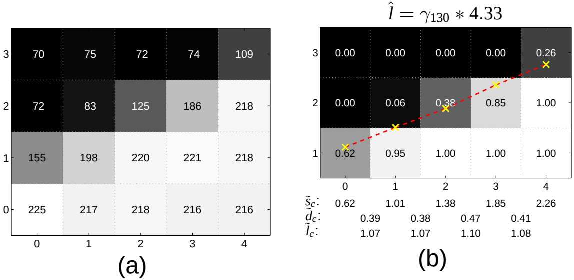


Figure: (a) Close up of the straight edge of a white paper imaged with a digital camera. (b) Segmentation result using 130 positive gray-levels. Approximating edge segments are superimposed (dashed lines) on the image .

Real images - estimation results

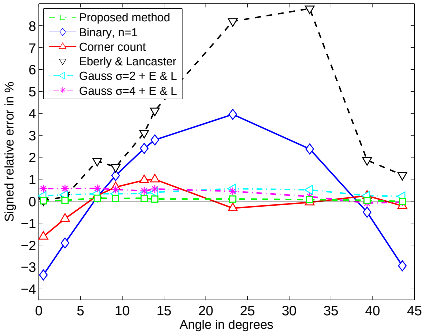


Figure: Relative errors for different methods when used to estimate the length of the edge of a white paper photographed with a digital camera at different angles. One image per angle, over 10 angle values, are observed.

Real images - comparative evaluation

To perform comparative evaluation, we apply a binary segmentation of the edge images (Otsu's thresholding method). We compared the following estimation methods:

- Two step directions, with weights as proposed by Verwer, 1991;
- The "corner count method" by Vossepoel and Smeulders 1982, where in addition to two weighted step directions a corner count correction is used. This may be seen as equivalent with using larger neighbourhood and additional step direction.
- Grey-scale based normal estimation method by Eberly and Lancaster;

The observed maximal errors for several estimation methods are as follows:

Proposed method 0.14%;

Verwer 3.95%;

Vossepoel and Smeulders 1.61%;

Eberly & Lancaster 8.78%;

Conclusions

- We described a novel and well performing local estimator of object boundary length.
- The method utilizes information from gray-level discrete representation of an object.
- Error-free estimates are obtained for non-quantized images.
- For quantized images, optimal scale factors to minimize the absolute estimation error is derived.
- The method is optimized for straight edge segments, but performs well as a perimeter estimator of more general shapes.
- The method is not multigrid convergent for a fixed number of grey levels.
- The method is "multi-grey level" convergent;
  max error for straight edges behaves as $\frac{1}{n^2}$ and is less than 10^{-5} for straight edges if $n = 255$ grey levels are used;

Further challenges

- Extension to 3D and surface area measurements.
- Extend the approach of utilizing grey-levels to other tasks of image analysis, especially feature estimation.
- Develop required “infrastructure” - segmentation methods that correspond well to the area coverage digitization.