PFG 2013 / 4, 0297 – 0308
Stuttgart, August 2013

Article

Finding Poly-Curves of Straight Line and Ellipse Segments in Images

SUSANNE WENZEL & WOLFGANG FÖRSTNER, Bonn

Keywords: polygon simplification, circle and ellipse detection, grouping, model selection

Summary: Simplification of given polygons has attracted many researchers. Especially, finding circular and elliptical structures in images is relevant in many applications. Given pixel chains from edge detection, this paper proposes a method to segment them into straight line and ellipse segments. We propose an adaption of Douglas-Peucker's polygon simplification algorithm using circle segments instead of straight line segments and partition the sequence of points instead the sequence of edges. It is robust and decreases the complexity of given polygons better than the original algorithm. In a second step, we further simplify the poly-curve by merging neighbouring segments to straight line and ellipse segments. Merging is based on the evaluation of variation of entropy for proposed geometric models, which turns out as a combination of hypothesis testing and model selection. We demonstrate the results of `circlePeucker` as well as merging on several images of scenes with significant circular structures and compare them with the method of PATRAUCEAN et al. (2012).

Zusammenfassung: *Segmentierung von Pixelketten in Geraden- und Ellipsenelemente*. Die Detektion runder und elliptischer Strukturen ist relevant für viele Anwendungen. Die Reduktion der Komplexität gegebener Polygone ist für sich ein interessantes Forschungsthema. Diese Arbeit stellt ein Verfahren zur Segmentierung von Pixelketten einer Kantendetektion in Geraden- und Ellipsensegmente vor. Der erste Schritt besteht in einer Adaption des Douglas-Peucker Algorithmus, in der Kreise anstelle von Geraden zur Partitionierung verwendet werden und die Punkt- statt der Kantensequenz partitioniert wird. Das Verfahren ist robust und reduziert die Komplexität der gegebenen Polygone stärker als der originale Algorithmus. In einem zweiten Schritt vereinfachen wir diese Vorsegmentierung durch das Verschmelzen benachbarter Segmente zu Geraden- und Ellipsensegmenten und stützen uns dabei auf die Entropieänderung. Wir zeigen die Ergebnisse der Vorsegmentierung als auch der folgenden Vereinfachung anhand verschiedener Bilder von Szenen, die signifikante kreisförmige Strukturen aufweisen und vergleichen sie mit dem Algorithmus von PATRAUCEAN et al. (2012).

1 Introduction

Polygon simplification is interesting from several points of view. First, in terms of compact description of spatial data, e.g. in the context of image description. Second, in terms of generalisation, e.g. in the context of cartography or resolution dependent visualization of polygons.

On the other hand finding circular and elliptical structures in images is relevant in terms of compact image description and further image interpretation. Most image interpretation systems which use bottom up image features, thus not just pure pixel information, are based on key point or edge detection. Directly identifying circular and elliptical structure gives rise to much more informative image features from bottom up (CHIA et al. 2012, JURIE & SCHMID 2004).

In this paper we propose a two-step polygon simplification algorithm that approximates a given set of ordered points in 2D by a sequence of straight line and ellipse segments. The poly-curves are intended to be at least C^0, thus positional continuous. Although the algorithm is applicable to any kind of ordered 2D points we assume pixel chains within images, see Fig. 1. The first intuition behind our ap-

 www.schweizerbart.de
DOI: 10.1127/1432-8364/2013/0178 1432-8364/13/0178 $ 3.00

Fig. 1: Finding line and ellipse sections. Left: pixel chains. Middle: segmentation into circle chains using `circlePeucker`. Right: aggregation and classification into straight line and ellipse segments. Blue: lines. Orange: circular and elliptical arcs.

proach is that arbitrary smooth curves can be locally characterised by an osculating circle. We will use this in the first step of the algorithm where we simplify the given set of points by a sequence of circle segments.

But due to perspective distortions, in general there will be almost no circles in images. All circles in object space are projected to ellipses in image space, ellipses in object space are projected to ellipses, anyway. Only in rare cases, the image of 3D circles or 3D ellipses are mapped to hyperbola, namely in case they partially are behind the camera. The situation is different if the circles are sitting in a set of parallel planes and the viewing direction intentionally has been taken orthogonal to these planes or the image has been rectified to mimic this situation. Then almost no ellipses will occur in the images, and the proposed method can directly be transferred by replacing ellipses by circles.

Therefore, eventually the pixel chain is represented by a sequence of straight line and ellipse segments. This way we are more flexible representing curved lines. Please note, that circles are part of the representation, as they are just special ellipses.

The proposed method consists of two steps, see Fig. 1. Given the pixel chains within the image, we first iteratively segment the region boundary into circular segments. This yields an over-segmentation due to the non existence of real circle segments. Second, we merge neighbouring segments to straight line and ellipse segments based on statistical reasoning, namely hypothesis testing and model selection. This step optimally estimates lines and ellipses in a least squares sequence.

One might argue, why not directly segment a pixel chain into ellipses, but first look for circle segments, and then group them to ellipses. There are two main reasons for the two-step procedure:

1.) The slope, curvature or curvature change functions of the ellipse are no simple functions, which allow to identify elliptical segments, as this is the case for straight lines (constant slope) and circles (constant curvature), 2.) there is no simple local measure telling whether a local segment belongs to an ellipse or not: Analysing the curvature, distinguishes circles and straight lines. One would need the second derivatives of the curvature to capture the properties of a local ellipse element, as an ellipse has two more degrees of freedom, than a circle. But determining fourth derivatives is very unstable.

There are two main contributions of this paper. First, for region boundary segmentation we propose an adaption to Douglas-Peucker's algorithm (DOUGLAS & PEUCKER 1973) which is based on circles as basic geometric elements and partitions the sequence of points instead of the sequence of edges. Second, we adapt the idea of variation of entropy by BEDER (2005) and statistically optimal merge neighboured segments while optimally fitting lines and ellipses. The whole process depends on two parameters, namely the precision of the edge extraction and the expected accuracy of the straight line and ellipse segments. The first one is an internal precision which guides the edge extraction, the second one is what the user defines to be and might guide the degree of generalization. Both can be estimated from training data.

Therefore, setting these parameters once is sufficient: The process works stable for all of our experiments using the same parameter set.

Related work

To our knowledge there is no work about an adaption of Douglas-Peucker's algorithm to the use of circles instead of lines as basic elements. However, proposals exists to simplify polygons by sets of circular arcs for the efficient storage of polylines. GÜNTHER & WONG (1990) proposed the so called Arc Tree which represents arbitrary curved shapes in a hierarchical data structure with small curved segments at the leaves of a balanced binary tree. MOORE et al. (2003) proposed a method for polygon simplification using circles. They aim on closed poly-

gons given by a set of 2D points. Based on medial axis from Voronoi polygons they propose a population of circles which they afterwards filter to get a set of circles which best approximate the given polygon. The final representation of the polygon consists of circles represented by centre and radius and tangents which link neighbouring circles. No work on using ellipses for improving storage requirements are known to us.

Finding ellipses in images has attracted many researchers. Some of them use Hough-transform methods which tend to be slow. Most techniques start from pixel chains. Early works focussed on ellipse fitting, e.g. PAVLIDIS (1983) and PORRILL (1990), later focussed on unbiased estimates, e.g. WU (2008) and LIBUDA et al. (2006). We are interested in the more general problem of describing the pixel chains by sequences of line and ellipse segments, a problem already addressed in ALBANO (1974), however, neither enforcing ellipses, nor looking for a best estimate for ellipses. WEST & ROSIN (1992) and ROSIN & WEST (1995) performed a segmentation of sequences into lines and ellipses in a multistage process. They first segment a 2D-curve into straight lines. Afterwards sequences of line segments are segmented into arcs restricted to their endpoints. One might interpret this step as merging sequences of lines to elliptical arcs. Model selection is done implicitly by evaluating a significance measure to each proposed segment, which is based on its geometry, purely. However, their criteria are non-statistical, thus cannot easily be adapted to varying noise situations. JI & HARALICK (1999) criticised this and proposed a statistically valid criterium. Starting from Rosin's output of arc segmentation they merge pairs of arcs belonging to the same ellipse. Moreover they also group non-adjacent arcs and exploit the sign of the arcs for grouping. Proposals for merging are validated via hypothesis testing. They showed only few results on comparably easy images. NGUYEN & KERAUTRET (2011) also addressed the segmentation of pixel chains into lines and ellipses. It is based on a discrete representation of tangents, circles, and an algebraic fitting through neighbouring arcs only using some key points (boundary and midpoint, instead of the complete pixel chain. Recently PATRAUCEAN et al. (2012) proposed a parameterless line segment and elliptical arc detector. They use an ellipse fitting algorithm which uses both, the algebraic distance of the conic equation and deviation from the gradient direction. Their model selection aims at avoiding false negatives, by controlling the number of false positives. Realizing the principle of "non-accidentalness" their method adapts to noise, which explains their visually appealing results. Their validation and model selection criteria, however, are based on fixed tolerance bands. Also they do not enforce any continuity between neighbouring segments.

Our scope is to segment pixel chains into straight line and ellipse segments, exploiting the knowledge about their statistical properties both w.r.t. detectability as well as w.r.t. accuracy.

Notation

Geometric elements are named with calligraphic letters, e.g. $\mathcal{X}$ is the name of a point, whereas $\boldsymbol{x}$ is its Cartesian representation. Homogeneous vectors and matrices are denoted with upright bold letters, e.g. $\mathbf{x}$ and C.

The rest of the paper is organized as follows. First, we describe the segmentation of region boundaries into circle elements based on the idea of Douglas-Peucker's algorithm. The merging procedure to obtain line and ellipse segments is explained in section 3. This section also gives details about model selection by variation of entropy and by the principle of minimum description length. Finally, section 4 presents results on synthetic and real data and compares them with the method of PATRAUCEAN et al. (2012).

2 Region Boundary Segmentation

Given a set of ordered points in 2D we aim at a partitioning into groups joining a common geometric element, specifically circular segments. We use the feature extraction procedure as described in FÖRSTNER (1994) and FUCHS & FÖRSTNER (1995). It includes an automatic noise estimation and an edge preserving filter as described in FÖRSTNER (2000). In contrast to many

other procedures it delivers region boundaries as well as thin lines in the form of chains of points with sub-pixel coordinates. For finding fine details, we use 0.7 pixel for the differentiation and 1.0 pixel for the integration scale. No blow up of the images is performed, as proposed by KÖTHE (2003) for fully exploiting the resolution.

2.1 *Algorithm*

Our concept of region boundary segmentation is based on the well known Douglas-Peucker algorithm (DOUGLAS & PEUCKER 1973). This algorithm is designed to simplify polygons. Therefore, it recursively splits the sequence of polygon edges into larger edges, until the distance of an eliminated point to the corresponding edge is below a threshold t. Thus neighbouring edges share a common point. In contrast, we want to recursively split the sequence of points $\mathcal{X} = \{\boldsymbol{x}_i\}$ until each subsequence can be approximated by a circular arc well enough. Thus neighbouring sequences are meant not to share a common point.

We realize this by first determining the mid points $\boldsymbol{x}'_i = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_{i+1}), i = 1, ..., I-1$, leading to a sequence $\mathcal{X}' = \{\boldsymbol{x}'_i\}$, which is a factor 2 smoother in variance than the original. Each edge in the sequence $\mathcal{X}'$ corresponds to a point $\boldsymbol{x}_i$ in the original sequence, except for the start and the end point. We now recursively partition the sequence of edges of $\mathcal{X}'$ into segments, which approximate the points $\boldsymbol{x}'_i$ by a circular arc up to a pre-specified tolerance t. A segment is split at that point $\boldsymbol{x}'_i$ where the distance to the circular arc is maximum. In order to enforce continuity, we fix the start and end point of the segments and determine the best fitting arc, see below.

The algorithm for approximating a polyline by a sequence of circles, called `circlePeucker`, is given in Alg. 1. It uses (1) function `fitArc`($\mathcal{X}$) for fitting a circular arc segment $\mathcal{S}$ to a given set of points $\mathcal{X}$ constraining it to the start and end point, and (2) a function `distXS`($\mathcal{X}, \mathcal{S}$) for determining the index i_b and the distance $d_{\max}$ of the point with the largest distance of the points $\mathcal{X}$ to an arc segment $\mathcal{S}$. The algorithms recursively splits the chain until the largest distance of a point to the corresponding arc is below a

In: Ordered set of points $\mathcal{X} = \{x_1 \dots x_I\}$, tolerance t
Out: List of segments O
1 **if** $I = 2$ **then** $O = \{1, I\}$, **return**;
2 $\mathcal{S}$ = `fitArc`($\mathcal{X}$);
3 $(d_{\max}, i_b)$ = `distXS`($\mathcal{X}, \mathcal{S}$);
4 **if** $d_{\max} > t$ **then**
5 partition at i_b:
$\mathcal{X}_1 = \{x_1 \dots x_{ib}\}$,
$\mathcal{X}_2 = \{x_{ib} \dots x_{\text{end}}\}$;
6 O_1 = `circlePeucker`($\mathcal{X}_1, t$);
7 O_2 = `circlePeucker`($\mathcal{X}_2, t$);
8 $O = O_1 \cup O_2$;
9 **end**
10 **return**

Algorithm 1: function `circlePeucker`

pre-set threshold t. As result we get a list O of N circle segments, each segment represented as a list of indices $\{i'_n\}, n = 1, ..., N$. Thus, we call O'=`circlePeucker`($\mathcal{X}', t$). The edges $(i', i'+1)$ of the segments in O' correspond to the sought points x_i, except for the start and the end point, which are added to the first and the last segment. This yields the required partitioning O of the original point sequence.

2.2 *Fitting Circle Segments*

The algorithm `fitArc`($\mathcal{X}$), needed in Alg. 1 line 2, constrains the circle to the starting and the endpoint of the current polygon segment. Additionally, we determine the distances $\boldsymbol{d} = [d_i]$ of the involved points x_i to the arc segment between x_1 and x_I of $\mathcal{S}$, not to the whole circle. Thus, the distance of a point to a segment is the minimum of the distance to the footpoint on the segment or the distance to the start or endpoint.

A circle usually has three degrees of freedom, but by restricting the arc to two points there is just one degree of freedom left. We parametrize the arc segment by its height h and solve the following optimization problem

$$\hat{h} = \text{argmin}_h \left(\|\boldsymbol{d}(\mathcal{X}, \mathcal{S}(\boldsymbol{x}_s, \boldsymbol{x}_e, h))\|_L \right). \quad (1)$$

For a robust estimate we choose the L_1-norm ($L = 1$), thus we optimize h such that the sum

of absolute distances of all points to the arc segment is minimized.

3 Merging

Given the pre-segmentation O of section 2 which is assumed to be over-segmented, we aim at a simplification by merging neighbouring segments which share the same model.

The pre-segmentation is based on circles, but in general there are almost no circles in natural images as they suffer from perspective distortions. Thus, our final segmentation is meant to consist of segments of straight lines and ellipses. From the pre-segmentation we just take the information about which points belong to one segment and ignore the parameters of the fitted circles.

The final representation is achieved by fitting straight lines and ellipses through neighboured segments and single segments using all points belonging to them. This is different from ROSIN & WEST (1995) who only use the endpoints from the pre-segmentation.

3.1 *Fitting Ellipses*

We perform maximum-likelihood estimations for fitting lines and ellipses, respectively, to the data. For line fitting we refer to standard literature, e.g. MCGLONE (2004).

Fitting ellipses is not trivial. We have to make sure an arc segment to be an ellipse and not a parabola or hyperbola.

We represent conics with the symmetric 3×3-matrix $\mathsf{C} = \begin{bmatrix} \boldsymbol{C}_{hh} & c_{h0} \\ c_{0h}^T & c_{00} \end{bmatrix}$ using homogeneous coordinates $\mathbf{x}$ for the points on the conic $\mathbf{x}^\mathsf{T}\mathsf{C}\mathbf{x} = 0$. To ensure the conic to be an ellipse the homogeneous part of the conic must fulfil $|\mathbf{C}_{hh}| > 0$. Therefore, we use Fitzgibbon's constraint (FITZGIBBON et al. 1999) which is equivalent to

$$|\mathbf{C}_{hh}| = 1\,. \tag{2}$$

This is a valid choice, as the conic representation is homogeneous. We end up with a maximum likelihood estimation following a Gauss-Helmert model with the constraint (2). Parameters are initialized using the direct method of Fitzgibbon (FITZGIBBON et al. 1999).

As a result we not only obtain the ellipse parameters but also the estimated variance σ^2 of the data and covariance matrix $\mathbf{\Sigma}$ of the parameters, which we use for the subsequent tests.

3.2 *Merging Segments Based on Variation of Entropy*

Deciding whether two neighbouring segments belong to the same model may be based on a statistical hypothesis test. As hypothesis tests aim at rejecting the null hypothesis, they can be used as sieve for keeping false hypothesis: Therefore, we use hypothesis testing for reliably identifying breakpoints between segments *not* belonging to the same model, by testing the null-hypothesis that they belong to the same segment.

Deciding which model fits the data best, i.e. whether a curved line is best approximated by a line or an ellipse, is a typical model selection problem and may be solved by the principle of minimal description length (MDL). This may be directly applied to isolated segments.

Merging segments based on hypothesis testing lacks on the risk of accepting large changes in geometry, in case the parameters of the proposed model are very uncertain. Therefore, we follow the idea of variation of entropy by BEDER (2005). He derives an information theoretical measure for the increase of uncertainty of a model due to adding new observations. This is equivalent to the change of entropy of the probability density function of the model's parameters. Following BEDER (2005), the change of entropy can be split into two parts. One depends on the increase of randomness due to new observations and is related to hypothesis testing. The other depends on the change of geometric uncertainty due to new data, respectively.

The differential entropy of a probability density function $p(x)$ is given by $h(p) = -\int p(x) \log p(x) \mathrm{d}x$. It reflects the randomness of a stochastic variable x. In case of a D-dimensional normally distributed random variable $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ the entropy is given by (COVER & THOMAS1991)

$$h(p) = 0.5 \log \left[(2\pi e)^D |\mathbf{\Sigma}| \right]. \tag{3}$$

Now, assume a segmentation O of points

$\mathcal{X} = \{\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_N\}$ into N segments. Further assume, we already found a model $\mathcal{M}_n$ fitting the points $\mathcal{X}_n$ of segment n, e.g. a line $\mathcal{M}_n = \mathcal{l}_n$. Now, we propose the points $\mathcal{X}_{n+1}$ of the neighbouring segment to belong to model $\mathcal{M}_n$, too. Without loss of generality we might argue on the neighbouring segments $n = 1$ and $n + 1 = 2$. Now, let the parameters of proposed model $\mathcal{M}_1$ be $\hat{\boldsymbol{\theta}}_1 \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1)$, with the empirical covariance matrix $\hat{\boldsymbol{\Sigma}}_1 = \hat{\sigma}_1^2 \boldsymbol{\Sigma}_1$ of the parameters, depending on the theoretical covariance matrix $\boldsymbol{\Sigma}_1$ and the estimated variance factor $\hat{\sigma}_1^2 = \Omega_1 / R_1$, derived from the weighted sum Ω_1 of the squared residuals and the redundancy R_1 of the estimation process.

When adding new observations $\mathcal{X}_{n+1}$ we estimate $\hat{\boldsymbol{\theta}}_2 \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_2)$ from $\mathcal{X}_{n,n+1} = \{\mathcal{X}_n \cup \mathcal{X}_{n+1}\}$ and obtain $\hat{\sigma}_2^2$ and the theoretical covariance matrix $\boldsymbol{\Sigma}_2$.

To validate the agreement of such two groups of observations concerning one of the two models $\mathcal{M}_1$ and $\mathcal{M}_2$ we analyse the change of entropy caused by adding new observations:

$$\Delta h_{\mathcal{M}} = h(\mathcal{N}(\hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}_2)) - h(\mathcal{N}(\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1)) \quad (4)$$

The parameters of a model $\mathcal{M}$ typically are given by adjustment theory. Thus, we know the variance factor $\hat{\sigma}^2$ and the empirical covariance matrix $\hat{\boldsymbol{\Sigma}} = \hat{\sigma}^2 \boldsymbol{\Sigma}$. Using (3) we get

$$\Delta h_{\mathcal{M}} = \underbrace{0.5 \log \left(\hat{\sigma}_2^2 / \hat{\sigma}_1^2 \right)}_{\Delta h_0} + \underbrace{0.5 \log(|\boldsymbol{\Sigma}_2| / |\boldsymbol{\Sigma}_1|)}_{-\Delta h_g} \quad (5)$$

The first term Δh_0 is closely related to the Fisher test statistic

$$\hat{\sigma}_2^2 / \hat{\sigma}_1^2 \sim \mathcal{F}(\Delta R, R_1) \quad (6)$$

with redundancy R_1 and $\Delta R = R_2 - R_1$, which is used to test whether the second set of observations fits the model estimated by the first set. It reflects the increase of randomness due to including new observations. The term Δh_g reflects the increase in randomness due to the geometric change of the model.

Therefore, we argue in the sense of hypothesis testing. Given a threshold $T_S = \mathcal{F}^{-1}(S, \Delta R, R_1)$ with significance level S by the inverse of Fisher distribution, there is no statistical reason to reject the hypothesis that both sets of observations fit the model if

$$\Delta h_0 < 0.5 \log T_S \quad (7)$$

which means that both sets of observations fit the model due to uncertainty in estimated parameters.

To bound the risk of large changes in geometry we further bound the increase of entropy by Δh_g. BEDER (2005) found this bound to be at the same order of magnitude as the increase of Δh_0. We use $T_g = T_0 + \frac{1}{2} \log T_S$ with a model dependent additional constant T_0 which we empirically found to be equal to the number of parameters of the current model, e.g. $T_0 = 2$ in case of lines or $T_0 = 5$ in case of ellipses. This compensates for a decrease in condition number of the covariance matrix caused by merging, therefore increasing with the number of parameters.

In case $R_1 = 0$ we cannot use an estimated variance factor $\hat{\sigma}_1^2$, but use the theoretical value σ_1^2 instead. Thus, (5) degenerates to

$$\Delta h_{\mathcal{M}} = 0.5 \log \left(\hat{\sigma}_2^2 / \sigma_1^2 \right) + 0.5 \log(|\boldsymbol{\Sigma}_2| / |\boldsymbol{\Sigma}_1|). \quad (8)$$

Now, the ratio $\hat{\sigma}_2^2 / \sigma_1^2 \sim \chi^2_{R_2}$ and we derive the threshold T_S from the inverse of χ^2distribution.

Please note, that the proposed approach is asymmetric in evaluation of $\mathcal{X}_{n,n+1}$ and $\mathcal{X}_{n+1,n}$. The asymmetry is compensated by always checking whether the smaller of two neighbouring segments can be merged with the larger one, thus the larger segment is taken to be $\mathcal{M}_1^*$.

3.3 *Model Selection*

We have seen how to use the variation of entropy to merge neighbouring segments to lines or ellipses, respectively. But the entropy criterion may not favour one of these two models. Then we select the one with smallest description length. This happens in case of long segments having very small curvature. Here the segments may be approximated either by a long line or by an ellipse segment having small curvature.

We evaluate the description length for merged models from their residuals. We use the modified Akaike criterion (AKAIKE 1974) $\mathrm{MDL}_{\mathrm{AIC}} = -2 \log p(\boldsymbol{l} | \hat{\boldsymbol{\theta}}) + 2U$ using

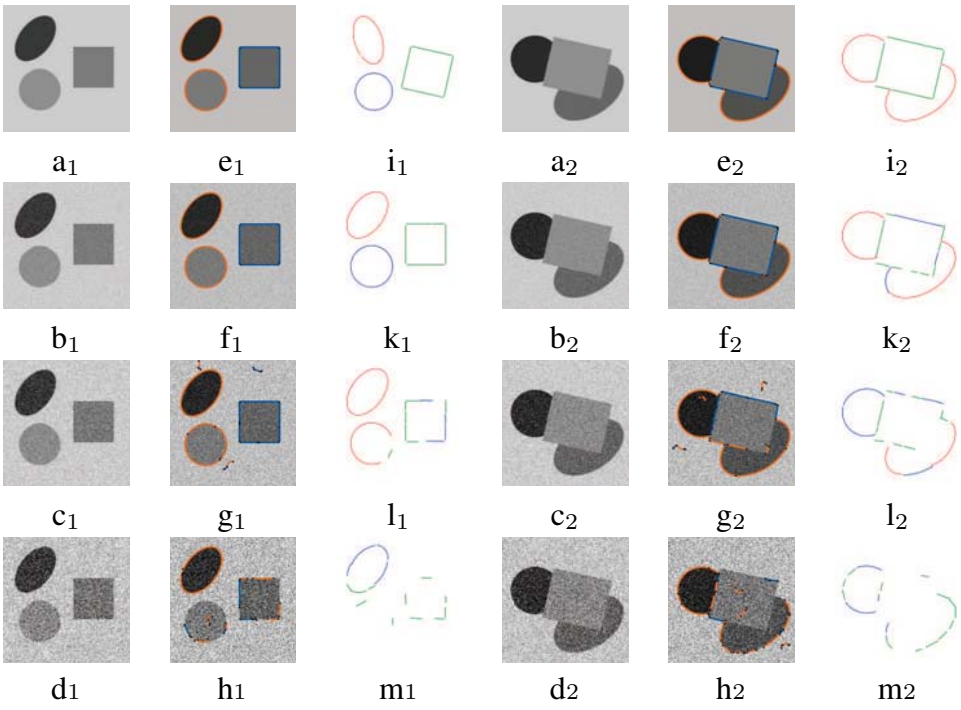

Fig. 2: Comparison to ELSD under different noise conditions (Best viewed in colour). a$_*$: given image. b$_*$ - d$_*$: $\sigma_n = 10, 20, 40$ grayvalues. e$_*$ - h$_*$: our final results. Colours, see Fig. 1. i$_*$ - m$_*$: results of ELSD. Red - elliptical arcs. Blue - circular arcs. Green: lines.

the log-likelihood function of data $\boldsymbol{l}$ and estimated parameters $\widehat{\boldsymbol{\theta}}$ and the number of parameters U. In case of normally distributed observations the log-likelihood function is equal to the sum of weighted squared residuals and we get

$$\text{MDL}_{\text{AIC}} = \Omega^2 + 2U. \tag{9}$$

Now, after having collected all criteria, we start the simplification of the given pre-segmentation. This is done in a greedy manner where we try to simplify the polygon while considering given pixel chains and while keeping the change of geometry slow.

3.4 *Algorithm*

For each segment $o_n \in O$ we initialize lines $\mathcal{l}_n$ and ellipses $\mathcal{C}_n$, if possible, i.e. we estimate model parameters $\{\hat{\boldsymbol{\theta}}_1\}_n$, covariances $\{\boldsymbol{\Sigma}_1\}_n$ and residuals $\{\boldsymbol{v}\}_n$. Let us call them models $\mathcal{M}_n^{\text{l}}$ and $\mathcal{M}_n^{\text{C}}$, respectively. For all neighbouring elements we propose merging, i.e. estimate parameters $\{\hat{\boldsymbol{\theta}}_2\}_{n,n+1}$, covariances $\{\boldsymbol{\Sigma}_2\}_{n,n+1}$ and residuals $\{\boldsymbol{v}\}_{n,n+1}$ of all potentially merged models. Let us call them models $\mathcal{M}_{n,n+1}^{\text{l}}$ and $\mathcal{M}_{n,n+1}^{\text{C}}$, respectively. For these models $\mathcal{M}_n^*$ and $\mathcal{M}_{n,n+1}^*$ we evaluate $T_S = \mathcal{F}^{-1}(S, \Delta R, R_1)$, Δh_0 and Δh_g using (5) or (8). To simplify notation, we avoid the index $(n, n+1)$ in the following. If $\Delta h_0 < \frac{1}{2} \log T_S$ and $\Delta h_g < T_0 + \frac{1}{2} \log T_S$ we add the proposed model to the set of merging proposals $\mathcal{P}$.

We require the geometrical change to be as small as possible when merging two segments. Therefore, we may choose the model $\mathcal{M}$ from $\mathcal{P}$ with smallest Δh_g. But note that we can not compare changes in entropy between line and ellipses. These are different models of different complexity, thus we are not allowed to pick the model with smallest Δh_g from the whole set $\mathcal{P}$. In a greedy process we start with lines, i.e. first merging all lines, which fulfil the requirements and afterwards merging all ellipses. More precisely, we pick these proposed merged line $\mathcal{l}^*$ from segments o_n and o_{n+1} with smallest Δh_g. If merging these two segments to an ellipse is a valid choice, too, we choose the line model if $\text{MDL}(\mathcal{l}^*) < \text{MDL}(\mathcal{C}^*)$.

After merging two segments, we update Δh for all affected segments and again pick the best proposal concerning Δh_g. If there are no line proposals left we continue with ellipses the same way, except from evaluating MDL.

All segments left, e.g. those segments that could not merge, are tested whether their curvature is significantly different from 0. If so, they become an ellipse, if not they become a line. To be precise, we perform variance propagation on the curvature and perform hypothesis tests on the 95% significance level.

4 Results

This section presents some results. We give details about parameters, show the resulting segments and discuss the success of the merging step by means of some statistics. We compare our results to those from ELSD (elliptical line segment detector) (PATRAUCEAN et al. 2012) as this is state of the art and there exists code as well as an online demo to process own images using fixed parameters.

Parameter setting

There are just a few parameters to choose and these are well understandable and stable for all tested images.

From experiments we found the standard deviation of edge pixels $\sigma_e = 0.1$ [pixel]. For this we set the tolerance t for the pre-segmentation using `circlePeucker` to $t = 3 \cdot \sigma_e$. Due to compression artefacts and image distortions, lines in images often are not that smooth and we set the variance for grouping a factor three larger than σ_e. Thus, for fitting and merging lines and ellipses, the uncertainty of each pixel is assumed to be isotropic $\boldsymbol{\Sigma}_{pp} = (3 \cdot \sigma_e)^2 \boldsymbol{I}_2$.

The significance level for the Fischer-test-statistic in (7) is set to $S = 0.95$. The additive constant for evaluating the bound of Δh_g in (5) is set to the number of parameters of the current model, $T_0 = U$.

As our purpose is the segmentation of given pixel chains and not the interpretation of the image the identification of spurious scatter is out of scope. Our algorithm works stable even for very small chains. Nevertheless, to simplify the visualization we do not show short pixel chains, say shorter than 10 or 20 [pixels], depending on the structure of the image.

Synthetic data

First we investigate the noise sensitivity of the procedure using synthetic images, see Fig. 2. When changing the noise σ_n of an image from $\sigma_n(0)$ to $\sigma_n(k)$ the standard deviation of edge pixels by $\sigma_e(\sigma_n(k)) = \sqrt{1 + \frac{\sigma_n^2(k)}{\sigma_n^2(0)}} \cdot \sigma_e(0)$ is adapted, where we assumed the noise of the image to be $\sigma_n(0) = 2$ [gr]. The parameters of edge detection are not changed.

Please note that the proposed algorithm works quite stable up to a certain degree of noise. As long as the contrast is high, geometric elements are reliable and accurate detected.

Douglas-Peucker vs. `circlePeucker`

Next we show the effect of the circle-version of the Peucker-Algorithm. We compare the results of `circlePeucker` to the original Douglas-Peucker algorithm when used as pre-segmentation for the final merging step. The results are given in Fig. 3 and Tab. 1.

Fig. 3 shows the results for two natural images when using the classical Douglas-Peucker algorithm and the new `circlePeucker`, respectively, as pre-segmentation for the final merging step as described in section 3. We see that both algorithms perfectly approximate the given data. This is due to the tight threshold for the maximum distance to a fitted geometric element which is the same in both cases. But we realize, just by visual inspection, that our new segmentation reduces the number of segments significantly. For a quantitative evaluation of this reduction, we count the total number of segments for each processed image when using the original Douglas-Peucker and our new segmentation, respectively. Tab. 1 gives these numbers for each processed image together with the total number of evaluated pixel chains and the final number of segments after the merging step. We see that the new circle-based pre-segmentation reduces the number of segments by almost 50 % compared to the line-based Douglas-Peucker algorithm. The merging step further reduces the number of segments by about 25 %.

We show some of the advantages of pre-segmentation using `circlePeucker` by some details. E.g. the capital O of the STOP-sign actually consists of four arcs instead of

Fig. 3: Pre-segmentation `circlePeucker` vs. Douglas-Peucker (Best viewed in colour). a_*: Given images as used in PATRAUCEAN et al. (2012). b_*: Pre-Segmentation using Douglas-Peucker. c_*: Pre-Segmentation using `circlePeucker`. e_* and f_*: Final segmentation using b_* and c_*, respectively. d_* results by ELSD. Colours, see Figs. 1 and 2.

Tab. 1: Statistics of simplification. The number of objects in the first column refers to the number of evaluated pixel chains per image. The second column gives the number of line segments using the classical Douglas-Peucker (DP) algorithm. The third column gives the number of circle segments using `circlePeucker` (new). Fourth and fifth column give the number of segments for the final segmentation results.

	No. objects	No. segments pre-segmentation DP	pre-segmentation new	final DP	final new
worm (Fig. 3)	187	1961	834	839	613
stop (Fig. 3)	159	1233	560	608	458
window (Fig. 5)	331	2226	868	941	726
icosahedron (Fig. 5)	1071	8991	6177	4517	3235
arcade (Fig. 4)	844	5908	2563	4675	2242

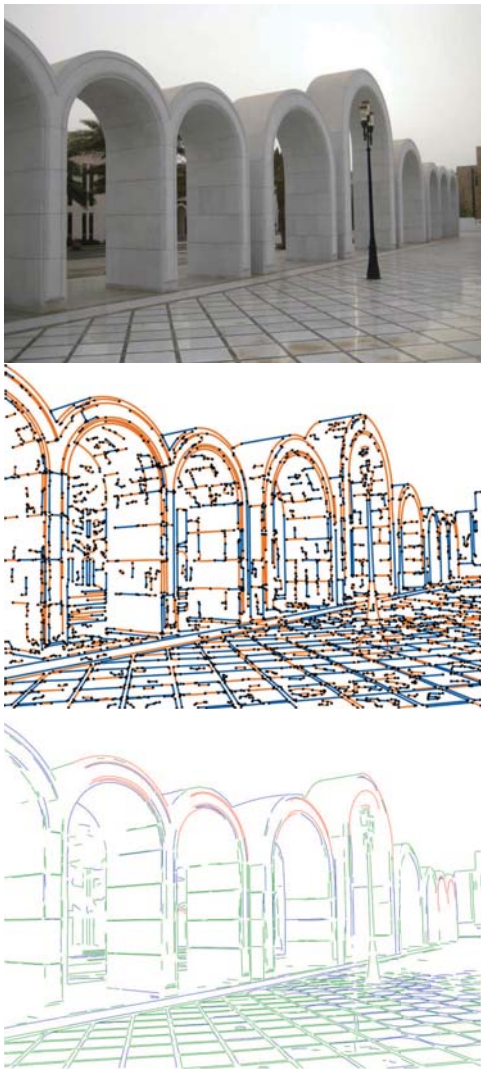

Fig. 4: More results on real images (Best viewed in colour). Arcade. From top to bottom: given image, our final result using `circlePeucker` for pre-segmentation, result of ELSD. Colours, see Figs. 1 and 2.

one ellipse. When using `circlePeucker` we get this result exactly. While using Douglas-Peucker tends to approximate arcs by lines, obviously. The main reason for this is the identification of break points candidates when evaluating the pre-segmentation. Obviously, `circlePeucker` identifies points of changing curvature more likely than Douglas-Peucker. The same effect can be observed for the boundary lines around the worm.

Comparison to ELSD

We give two more results on natural images in Figs. 4 and 5 and compare our results to those from from ELSD (PATRAUCEAN et al. 2012) in Figs. 2 to 5. Let us take the worm of the book cover shown in Fig. 3. ELSD resolves nearby edges, e.g. the black boundaries of the worm. The pre-processing of our method identifies these as (dark) lines, which are then simplified. The slightly curved boundaries of the letters W or B are straightened by ELSD, while better resolved by our method. ELSD simplifies too much, e.g. the ellipse of O in the STOP-sign. While ELSD detects the lines independently, our method segments the edge pixel chains, therefore at sharp corners occasionally an additional short segment is preserved, e.g. the rectangles within Fig. 2.

To summarize we see, the pre-segmentation using `circlePeucker` correctly identifies arc segments and especially their breakpoints. By itself these are promising results and improve the standard algorithm in terms of reducing the number of breakpoints of a given polygon while preserving the geometry.

The merging step identifies elliptical arcs correctly and further reduces the number of segments of most given pixel chains. Lines are identified in most cases, if not this might be due to distortions, especially for long lines.

5 Conclusion

We presented a line simplification approach which approximates given pixel chains by a sequence of lines and elliptical arcs. For this we proposed an adaption to Douglas-Peucker's algorithm for the use of circles instead of straight lines. Furthermore, we developed an approach for the simplification of such a segmentation by merging neighbouring segments due to their agreement to a joint geometric model in terms of bounded variation of entropy. The approach depends on just a few parameters which are clearly explained by a priori knowledge about edge detection accuracy. Depending on the assumed edge accuracy we showed very accurate results. We showed the effects of polyline segmentation and simplification on several images with comparable good results referring to an state of the art algorithm. We proved the success of merging in terms of the reduction rate of number of segments per object. We believe that the final segmentation gives rise to useful

Fig. 5: More results on real images (Best viewed in colour). Left: Gothic window. Right: cropped icosahedron. From top to bottom: given image, our final result using `circlePeucker` for pre-segmentation, result of ELSD. Colours, see Figs. 1 and 2.

high level image features as input for an image interpretation system.

Acknowledgement

The authors would like to thank the reviewers for their valuable suggestions which helped to improve the paper.

References

AKAIKE, H., 1974: A new look at the statistical model identification. – IEEE Transaction on Automatic Control **AC-19**: 716–723: System identification and time-series analysis.

ALBANO, A., 1974: Representation of digitized contours in terms of conic arcs and straight-line segments. – Computer Graphics and Image Processing **3** (1): 23–33.

BEDER, C., 2005: Agglomerative grouping of observations by bounding entropy variation. – DAGM, LNCS **3663**: 101–108.

CHIA, A., RAJAN, D., LEUNG, M. & RAHARDJA, S., 2012: Object recognition by discriminative combinations of line segments, ellipses and appearance features. – PAMI **34** (9): 1758–1772.

COVER, T.M. & THOMAS, J., 1991: Elements of Information Theory. – Wiley.

DOUGLAS, D.H. & PEUCKER, T.K., 1973: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. – Cartographica **10** (2): 112–122.

FITZGIBBON, A.W., PILU, M. & FISHER, R.B., 1999: Direct least-squares fitting of ellipses. – PAMI **21** (5): 476–480.

FÖRSTNER, W., 1994: A framework for low-level feature extraction. – ECCV **801**/1994: 383–394.

FÖRSTNER, W., 2000: Image preprocessing for feature extraction in digital intensity, color and range images. – Geomatic Methods for the Analysis of Data in Earth Sciences **95**/2000: 165–189.

FUCHS, C. & FÖRSTNER, W., 1995: Polymorphic grouping for image segmentation. – ICCV.

GÜNTHER, O. & WONG, E., 1990: The arc tree: An approximation scheme to represent arbitrary curved shapes. – CVGIP **51** (3): 313–337.

JI, Q. & HARALICK, R.M., 1999: A Statistically Efficient Method for Ellipse Detection. – ICIP: 730–734.

JURIE, F. & SCHMID, C., 2004: Scale-invariant shape features for recognition of object categories. – CVPR **2**: II–90 – II–96.

KÖTHE, U., 2003: Edge and Junction Detection with an Improved Structure Tensor. – DAGM, LNCS **2781**: 25–32.

LIBUDA, L., GROTHUES, I. & KRAISS, K.-F., 2006: Ellipse detection in digital image data using geometric features. – VISAPP: 175–180.

MCGLONE, J.C., 2004: Manual of Photogrammetry. – 5th edition edn. American Society for Photogrammetry and Remote Sensing.

MOORE, A., MASON, C., WHIGHAM, P.A. & THOMPSON-FAWCETT, M., 2003: The use of the circle tree for the efficient storage of polygons. – GeoComputation.

NGUYEN, T.P. & KERAUTRET, B., 2011: Ellipse detection through decomposition of circular arcs and line segments. – ICIAP: 554–564.

PATRAUCEAN, V., GURDJOS, P. & VON GIOI, R.G., 2012: A parameterless line segment and elliptical arc detector with enhanced ellipse fitting.. – ECCV **7573**: 572–585.

PAVLIDIS, T., 1983: Curve fitting with conic splines. – ACM Transaction Graphics **2** (1): 1–31.

PORRILL, J., 1990: Fitting ellipses and predicting confidence envelopes using a bias corrected kalman filter. – Image and Vision Computing **8** (1): 37–41.

ROSIN, P.L. & WEST, G.A.W., 1995: Nonparametric Segmentation of Curves into Various Representations. – PAMI **17** (12): 1140–1153.

WEST, G.A.W. & ROSIN, P.L., 1992: Multi-stage Combined Ellipse and Line Detection. – BMVC: 197–206.

WU, J., 2008: Robust Real-Time Ellipse Detection by Direct Least-Square-Fitting. – International Conference on Computer Science and Software Engineering **1**: 923–927.

Address of the Authors:

Dipl.-Ing. SUSANNE WENZEL, Rheinische Friedrich-Wilhelms-Universität zu Bonn, Institut für Geodäsie und Geoinformation, Photogrammetrie, Nußallee 15, D-53115 Bonn, Tel.: +49-228-73-2713, Fax: +49-228-73-2712 and Prof. Dr.-Ing. Dr. h.c. mult. WOLFGANG FÖRSTNER, Josef-Schell-Str. 34, 53121 Bonn, e-mail: {susanne}{wf}@ipb.uni-bonn.de

Manuskript eingereicht: März 2013
Angenommen: Mai 2013