

# CHAIN-WISE GENERALIZATION OF ROAD NETWORKS USING MODEL SELECTION

Dimitri Bulatov<sup>a</sup>, Susanne Wenzel<sup>b</sup>, Gisela Häufel<sup>a</sup>, Jochen Meidow<sup>a</sup>

<sup>a</sup> Fraunhofer IOSB, Ettlingen, Germany, dimitri.bulatov@iosb.fraunhofer.de

<sup>b</sup> Institute of Geodesy and Geoinformation, Photogrammetry Group, University of Bonn, Germany, wenzel@igg.uni-bonn.de

## Commission II, WG II/III

**KEY WORDS:** Road Extraction, Generalization, Polyline Simplification, Model Selection, Traffic Roundabout

### ABSTRACT:

Streets are essential entities of urban terrain and their automatized extraction from airborne sensor data is cumbersome because of a complex interplay of geometric, topological and semantic aspects. Given a binary image, representing the road class, centerlines of road segments are extracted by means of skeletonization. The focus of this paper lies in a well-reasoned representation of these segments by means of geometric primitives, such as straight line segments as well as circle and ellipse arcs. We propose the fusion of raw segments based on similarity criteria; the output of this process are the so-called chains which better match to the intuitive perception of what a street is. Further, we propose a two-step approach for chain-wise generalization. First, the chain is pre-segmented using `circlePeucker` and finally, model selection is used to decide whether two neighboring segments should be fused to a new geometric entity. Thereby, we consider both variance-covariance analysis of residuals and model complexity. The results on a complex data-set with many traffic roundabouts indicate the benefits of the proposed procedure.

## 1. INTRODUCTION AND PREVIOUS WORK

Roads are very important entities of any geographic database. Since they are man-made objects, road networks exhibit, often and particularly in urban terrain, regular structures, e. g., straight lines, circles, clothoids, ellipses, as well as orthogonality and symmetry. However, automatic detection of these regularities from airborne captured image or laser data is difficult. Occlusions and shadows are probably the most intuitive challenges when one thinks about road extraction. In fact, roads mostly are occluded by tree crowns, balconies of buildings, or (groups of densely) parking vehicles and their shadows. This is where semantics becomes relevant: Which obstacles can be ignored and which cannot? A separation line between two highway directions may look similar to a queue of trucks, hence, differentiation between both groups of objects should ideally be made. The third aspect, valid in case of aerial images, is that roads have often very homogeneous textures as well as moving objects; hence noise and outliers often make acquisition of 3D information difficult. However, elevation data, extractable, for example by means of depth maps (Rothermel et al., 2012), has turned out to be essential for reconstruction of roads in urban terrain (Hinz and Baumgartner, 2003; Wegner et al., 2015).

All these challenges let the automatically extracted road networks – which are mostly stored in geographic bases in form of vector data for street centerlines – appear extremely wriggled and should be corrected or generalized within a post-processing step. There are several contributions related to generalization of road networks but for most of them, Chaudhry and Mackaness (2006) for example, data noise is not a significant problem. Our work is more similar to (Bulatov et al., 2016b; Mena, 2006), where segments are extracted from the actual sensor data and finally generalized either by the well-known algorithm of Douglas and Peucker (1973) or by higher order, e. g., Bézier curves. Both modules (Douglas-Peucker and Bézier curves) were modified in the way that the polygonal chains do not cross obstacles, such

as buildings and trees. There are two major drawbacks of this approach: Neither variance-covariance error analysis was carried out nor any kind of hypothesis testing which of two models – a straight line or a smooth Bézier curve – is actually relevant for the current segment. Besides, the approach was applied to very short segments that are defined between two branch points resulting from a skeletonization algorithm. The effect of generalization was thus barely visible, in particular, because segment endpoints are fixed.

It is well-known, however, that for road nets, not only geometric but also topological correctness, that is, connectivity between roads, becomes extremely relevant. Several authors (Türetken et al., 2013; Wegner et al., 2015) exploited this fact for road extraction and in this work, we exploit topology for post-processing. We establish neighborhoods between segments and generalize chain-wise. The advantages are on the one hand *semantic* – since the chains satisfy better the intuitive notion what a street is – and on the other hand *geometric*, since more points are considered for the upcoming generalization, increasing thus the redundancy. The process of generalization itself consists of fitting geometric primitives, namely, straight line segments, circle and ellipse arcs into chains of points. We first create an over-segmentation of circular segments using a modified version of Douglas and Peucker (1973) and finally merge neighboring segments using iterative model selection.

We ought to mention that the problem of fitting geometric primitives in pixel chains and 2D meshes has been extensively treated in the past. For example, Günther and Wong (1990) propose the so-called Arc Tree, which represents arbitrary shapes in a hierarchical data structure with small curved segments at the leaves of a balanced binary tree. Moore et al. (2003) propose a method for polygon simplification using circles. They aim at closed polygons given by a set of 2D points. Finding ellipses in images has attracted many researchers (Porrill, 1990; Patraucean et al., 2012). But these works start from pixel-chains, which is not the case in our application. We are interested in the more general

problem of describing polygonal chains by sequences of straight line, circle, and ellipse segments, a problem which was similarly addressed in Albano (1974), however, neither enforcing ellipses, nor looking for a best estimate for ellipses.

Most related to our approach is the work by Rosin and West (1995) where segmentation of point sequences into straight lines and ellipses is performed within a multistage process. 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 and Haralick (1999) criticize this and modified their idea by a hypothesis testing framework. Again, these approaches are applied to images and pixel chains, respectively.

The main contribution of this work is to combine the *semantical* approach of fusing road segments and assuming that they – as a typical man-made object – can be approximated by some geometric primitives with the *statistical* approach of model selection which allows to decide whether neighboring segments can be represented by a single primitive. The approach of model selection is based on information theory since not only coordinates’ residuals but also model complexity is taken into consideration. Note that after the generalization, the data is not necessarily consistent anymore; for example, lines and circles are not guaranteed to intersect. However, the adjacency information is not lost and can be used to create junctions of appropriate size so that the geometric inconsistencies are not visible. This is exactly the way the road networks are managed in most urban terrain simulation systems and many other applications.

For reasons of completeness, we provide in Sec. 2 a brief summary of methods we applied in order to fit geometric primitives, such as straight lines, circles, and ellipses. By ellipses, we strive to approximate clothoids, which are more often employed to provide a smooth transition of curvature for curvy road courses; however, clothoids turn out to be less handy for the chain forming module. The process of chain forming, applied once a raw road network had been extracted from the classification result, is explained in Sec. 3. In Sec. 4, we present our algorithm on chain-wise generalization. Our results in Sec. 5 verify that road networks generalized chain-wise with multiple primitives are visually more appealing than the results of segment-wise generalization with multiple attributes. In Sec. 6, main conclusions and ideas for future work are provided.

## 2. BASICS

Given the set of  $N$  observed points  $\mathcal{X} = \{\mathbf{x}_n\}, n = 1 \dots N$ , we aim at the best fitting straight line, circle or ellipse, which we represent as homogeneous elements. In each case, we look for the statistically best fitting parameter vector as well as its covariance. We need this when merging neighboring lines based on their statistical properties. A detailed discussion of uncertain homogeneous points and lines can be found in Förstner and Wrobel (2016) and Meidow et al. (2009). We assume i.i.d. coordinates of each point, sharing the same isotropic covariance  $\Sigma_{\mathbf{x}_n \mathbf{x}_n} = \sigma_n^2 I_2$ .

**Straight Line** In (Förstner and Wrobel, 2016, Sec. 9.4.2), it is shown that the statistically best fitting line passes through the centroid of given points and that its direction is given by the principal axis of their moment matrix. We obtain the estimated homogeneous coordinates of line  $\hat{\mathbf{l}}$  and the covariance matrix  $\Sigma_{\hat{\mathbf{l}}}$ .

**Ellipse** We use the homogeneous representation of conics to express the parameters of the ellipse. Thereby, we represent conics with the symmetric  $3 \times 3$ -matrix

$$\mathbf{C} = \left[ \begin{array}{cc|c} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ \hline c_{13} & c_{23} & c_{33} \end{array} \right] = \left[ \begin{array}{cc} C_{hh} & c_{h0} \\ c_{0h}^T & c_{00} \end{array} \right]. \quad (1)$$

Any point  $\mathbf{x} = [x, y, 1]^T$  on the conic fulfills  $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$ . For estimating the parameters we use the implicit polynomial representation of the conic  $\mathbf{y}^T \mathbf{c} = 0$ , with the vector of unknowns  $\mathbf{c} = [c_{11}, c_{12}, c_{22}, c_{13}, c_{23}, c_{33}]^T$  and the observations  $\mathbf{y} = [x^2, 2xy, y^2, 2x, 2y, 1]^T$ . To ensure the conic to be an ellipse,  $|C_{hh}| > 0$  must be fulfilled. Thus, we impose the quadratic constraint  $c_{11}c_{22} - c_{12}^2 = 1$ , which is a valid choice, as the conic representation is homogeneous and all parameters can be divided by any non-zero scale factor. This leads to a non-linear Gauss-Helmert model. Using initial parameters estimated by means of the direct method of Fitzgibbon et al. (1999), we follow Wenzel (2016, Sec. 2.1.3, p. 47ff) to obtain the estimated parameters  $\hat{\mathbf{c}}$  of the conic and their covariance matrix  $\Sigma_{\hat{\mathbf{c}}}$ .

**Circle** A circle is a special regular conic for which the matrix  $C_{hh} \propto I_2$  in (1). Instead of using the over-parametrized conic representation, we represent circles by their implicit homogeneous equation  $\mathbf{z}^T \mathbf{p} = 0$ , where we collect the coordinates of a point  $\mathbf{x}$  in a vector  $\mathbf{z} = [x^2 + y^2, x, y, 1]^T$  and the parameters within vector  $\mathbf{p} = [A, B, C, D]^T$ , from which we easily obtain the circles parameters,  $x_0, y_0, r$ . Note that setting  $A = 0$  allows us to represent circles with infinite radius, thus, straight lines. Given at least three observations, the resulting linear equation system can be solved using a SVD-based method, which we refer on as direct method. Instead, we follow Förstner and Wrobel (2016, Sec. 3.6.2.5) and derive the covariance matrix of the circles’ parameters  $[x_0, y_0, r]$  directly from observed points. Finally, using variance propagation, we yield estimated parameters  $\hat{\mathbf{p}}$  and the according covariance matrix  $\Sigma_{\hat{\mathbf{p}}}$ .

## 3. ROAD-NET EXTRACTION AND CHAIN FORMING

Usually, classification results are represented by binary maps. The first step of our pipeline thus consists of vectorizing these maps. We obtain a set of polygonal chains, to which we will refer as polylines. The result may appear noisy and thus, we must filter out those polylines which do not correspond to our understanding of what a (part-of-a-)road is. These steps are explained in Sec. 3.1. Our next task is fusion of the remaining polylines into chains, which is done for two main reasons. Firstly, since the polylines connect just neighboring junctions, the chains conform better with our perception of street than the raw polyline. Think about the Oxford Street in London. Its name remains the same throughout its course, even though multiple side roads decompose it into several polylines. Secondly, chains are more suitable for generalization, since the whole geometry of the entity may be captured. More details are explained in Sec. 3.2. The contents of this section are visualized on a running example in Fig. 1.

### 3.1 Vectorization and Extraction of Road Polylines

A classification result is represented by the road-class binary image  $B$ . We denote by  $\partial B$  its boundary, which we usually smooth by morphological operations. Starting from  $B$ , we extract the medial axis by means of skeletonization and finally, we apply the vectorization tool of Steger (1998). The output of this method



Figure 1: Main steps of chain forming algorithm. Left: vectorization by means of medial axis; forbidden areas are indicated by the golden color. Middle: The implausible segments are filtered out. We also recommend fusing junctions (denoted by blue crosses) in order to increase the number of neighbors for chain forming (right image). Two segments are fused into a chain (cyan), because their partial dominant direction were similar. The red segment is a side road in this case. Chains and side roads are the main input for the chain-wise generalization, see Sec. 4.

is a set of open polygonal chains, which we call polylines. An endpoint of a polyline is always either a pixel on  $\partial B$  (usually, any concavity), or a branch point, for which at least three points, belonging to  $\partial B$ , have the same distance. In the first case, we refer to the polyline endpoint as to a *dead end* while in the second case, we denote it as a *junction*. Beside these two cases, a particular situation is observed if the polyline is closed, or, equally, if it is homeomorphic to the circular line. In this situation, both dead ends coincide with one of the vertices of the polyline.

To recognize whether a polyline endpoint is a dead end or a junction, the range-search procedure is applied. All endpoints are clustered by means of the generalized DBSCAN algorithm (Sander et al., 1998), which is the state-of-the-art tool for downsampling point clouds. A junction is a cluster with at least three vertices. The structure of junctions contains their 2D coordinates and the corresponding incident polylines. Since every concavity of  $B$  causes one polyline, discarding those road segments which exhibit a suspicious geometric appearance (too short, too broad, etc.) and at the same time do not contribute to the topological functionality of the road net has been proposed in the literature (Mena, 2006; Bulatov et al., 2016b). Thus, the iterative filtering procedure is based on polyline *attributes*, such as *width*, *length*, *type*, etc., which are calculated according to Bulatov et al. (2016a). More concretely, we delete within one iteration all polylines of which at least one endpoint is not a junction and whose length or width take on a suspicious value (e. g., the length below 2 m or width out of range [2 m; 50 m]). After every iteration, the attributes are updated. In order to remove redundant loops, for example, around isolated trees, an additional module was implemented, however, not employed since we wish to demonstrate the tools implemented in the next section to fit circle arcs.

### 3.2 Chain Forming

The previously discussed polylines serve, for the most part, as connection links between the junctions and do not correspond to the generally understood term of street. We wish to perform fusion of polylines into chains in order to generalize them chain-wise in the next step. The essential precondition of chain forming is establishing – geometric and topological – similarities between the polylines, which is done using the attributes mentioned in Sec. 3.1. That is, to find candidates for fusion, we have to search for similar attributes between pairs of polylines. The necessary condition for similarity is that two polylines are topologically neighboring; in other words, they must share a common

junction. This additionally simplifies the implementation since all the remaining steps of the algorithm run over junctions. Given two polylines gathered in a junction, the (dis)similarity of their geometric attributes is denoted as *cost*. The smaller the cost, the larger the likelihood of two polylines to be merged. After all  $n(n-1)/2$  costs are collected, where  $n$  is the number of polylines converging to a junction, pairs of candidates with minimum cost are collected for the upcoming fusion process. This may be done either by a greedy algorithm or using the Hungarian Method. We opted for the former one, our choice because of its simplicity and because only a few values of  $n$  exceed 4. The order of vertices of the merged polylines should be topologically correct. This means on the one hand, reordering the polyline segments to be fused and, on the other hand, flipping the order of points within one polyline, if necessary.

In the rest of this section, we outline different methods for comparing geometric attributes keeping in mind that we want to identify both straight and circular chains. First of all, we established a width gap: The necessary condition for two road segments to be neighbors is that the width of the narrower one, denoted by  $w_{\min}$ , and that of the broader one,  $w_{\max}$ , are similar, that is

$$(1 - \varepsilon)w_{\max} < w_{\min}, \text{ where } \varepsilon \approx 0.5. \quad (2)$$

This assumption is reasonable because a street usually has a constant width throughout its course. Note that even though this threshold may seem large, it is only a necessary condition. To make this condition also sufficient, we investigated two promising methods: First, the *partial dominant directions* of two neighboring polylines and second, the direct circle-fit method from Sec. 2 within a RANSAC framework (*circle-fit + RANSAC*).

In order to estimate the partial dominant directions, we build for each polyline a weighted histogram of directions modulo  $180^\circ$ , where the weights are proportional to the segments' lengths. A hill-and-dale analysis of the smoothed histograms yields, in essential, the partial dominant directions. Our cost function is thus given by the truncated absolute difference of the partial dominant directions corresponding to the relevant junction (Fig. 1, right). The more the dominant directions differ, the higher the cost.

For the second approach, the direct circle-fit method mentioned in Sec. 2 is the core function for the RANSAC algorithm over the union of vertices of both polylines. Here the dissimilarity is given by the percentage of outliers. The advantage of this method is that we can detect, consciously, circular structures around a junction. The problem, however, is extension of chains over pairs of polylines without storing the parameters of the fitted circles. Note that numerous further cost functions can be devised. Besides those mentioned in Bulatov et al. (2016a), we implemented the circle-fit function from the minimum set (the junction and two loose ends) of both polylines and measured once again the outlier percentage to build the cost function. Clearly, this was by far less accurate than the solution based on RANSAC. Alternatively, one could compare the curvatures of the adjacent polylines. However, the positions of vertices are often very noisy and, since building the second derivatives is hardly known to be a numerically stable process, we rejected this idea. Summarizing, partial dominant directions and circle-fit + RANSAC, both preceded by the width gap filter, are the best trade-offs between characterization of the local course of the polyline near the junction and the point of view of the numerical stability, for which possibly all vertices should be considered.

#### 4. GENERALIZATION

Given the chains from road-net extraction, we wish to represent them by sequences of straight line, circular, and ellipse segments. The proposed method consists of two steps, described in subsections 4.1 and 4.2. Given the chains, we first iteratively segment them into circular segments, which yields an over-segmentation. Second, merging neighboring segments is performed based on model selection. In this step, straight lines, circular arcs and ellipses are estimated optimally in a least squares sense. This way, we are more flexible representing curved courses than using straight lines, solely. It might be confusing that polylines just connected to chains are segmented into smaller parts again, which are then fused once more. Note that the chain-forming procedure is based on topology and the streets attributes, while pre-segmentation here is just based on the polylines geometry. Over-segmentation is a natural side-effect of the algorithm and is desired in order to generate proposals for later merging to larger and potentially more complex geometric objects, which are consistent to the given raw data on one hand but generalizations in terms of our intuition of streets on the other hand.

##### 4.1 Segmenting Point Sequences into Circle Segments

The concept of chain segmentation into circle segments is based on the `circlePeucker` algorithm (Wenzel and Förstner, 2013), which is an adaption of the well known Douglas-Peucker algorithm (Douglas and Peucker, 1973). The original algorithm is designed to simplify polylines by recursively splitting a sequence of polyline edges into larger edges until the distance of an eliminated point to the corresponding edge is below a threshold  $t$ .

Instead of straight lines, `circlePeucker` uses circle segments (Wenzel and Förstner, 2013). Given a sequence of points, it is recursively partitioned into segments which approximate the according points by a circular arc up to a pre-specified tolerance  $t$ . If applicable, a segment is split at that point  $\chi_n$ , where the distance to the circular arc is maximum. In order to enforce continuity, they fix the start and endpoint of the segments and determine the best fitting arc. As threshold  $t$  we use in our application half of the width of the smallest street part involved in the relevant group; the width gap mentioned in Eq. (2) widely guarantees uniformity of width values. As result we obtain a list of indices which represent the endpoints of sought segments. This yields the required partitioning of the original point sequence.

##### 4.2 Merging Line Primitives Based on Model Selection

Given the preliminary, over-segmented partition of the chain, we aim at a simplification by merging neighboring segments which share the same geometric model instance. Deciding whether two neighboring segments belong to the same model instance may be based on a statistical hypothesis test. As these tests aim at rejecting the null hypothesis, they can be used as sieve for keeping false hypotheses. Merging segments merely based on hypothesis testing, however, fails due to the risk of accepting large changes in geometry, in case the parameters of the proposed model are uncertain. On the other hand, deciding which model fits the data best, i. e., whether it should be approximated by a straight line, circle or an ellipse, is a typical model selection problem.

The domain of models we use is {straight line, circle, ellipse}, which differ in the number of parameters. Here the term accuracy is related to the residuals,  $\mathbf{v}$ , caused by deviations of the points to the selected model. Let us consider a number of  $N$

normally, i. d. observations  $\mathbf{l}$  with covariance  $\widehat{\Sigma}_{ll}$ . We are looking for an  $U$ -dimensional parameter vector  $\widehat{\boldsymbol{\theta}}$ , whereby observations and parameters are related by the Gauss-Markov functional model  $\mathbf{l} + \widehat{\mathbf{v}} = f(\widehat{\boldsymbol{\theta}})$ . Using the usual definition  $\Omega = \widehat{\mathbf{v}}^T \widehat{\Sigma}_{ll}^{-1} \widehat{\mathbf{v}}$ , Schwarz (1978) derived the Bayesian Information Criterion

$$\text{BIC} = \Omega + U \ln N, \quad (3)$$

as a criterion for model selection. The lower the complexity of the model, given by the number of parameters  $U$ , the lower BIC. A large number  $N$  of observations increases the relative precision of the parameters and thus the reliability of the model. It can be shown that the BIC is closely related to the *description length* from information theory. Thus, we use these terms synonymously and wish to minimize Eq. (3) to select the best model.

From the pre-segmentation, we only take the information which points belong to the same segment and ignore the parameters of the fitted circle segments. The final representation is achieved by fitting straight line, circle and ellipse segments through chains and side roads, respectively, using all points belonging to them. Again, given a set of  $N$  observations  $\mathcal{X} = \{\mathbf{x}_n\}, n = 1 \dots N$ , where we assume i. i. d. coordinates of each point, sharing the same isotropic covariance  $\widehat{\Sigma}_{\mathbf{x}_n \mathbf{x}_n} = \sigma_n^2 \mathbf{I}_2$ , we aim at the best fitting line  $\widehat{\mathbf{l}}$ , circle  $\widehat{\mathbf{p}}$ , or ellipse  $\widehat{\mathbf{c}}$  as described in Sec. 2. Thereby we assume  $\sigma_n = 1$  and take the threshold  $t$ , given above, as a priori variance factor  $\sigma_0^2$  in order to scale the variance of observations. For each model, we look for the statistically best fitting parameter vector as well as its covariance by estimating the weighted sum of squared residuals  $\Omega = \sum_n \widehat{\mathbf{v}}_n^2 / \sigma_n^2$  as measure of precision and the estimated variance factor  $\widehat{\sigma}_0^2 = \Omega / (N - U)$ .

Let us assume a segmentation of points  $\mathcal{X} = \{\mathcal{X}_m\}$  into  $M$  segments. We call the current parameter vector of the  $m$ -th segment  $\boldsymbol{\theta}_m$ . Thus,  $\boldsymbol{\theta}_m$  acts as placeholder for  $\mathbf{l}_m$ ,  $\mathbf{p}_m$  or  $\mathbf{c}_m$  and includes the number  $U_m$  of parameters (2, 3, and 5 respectively) needed to define the current model, which is our measure of complexity. Initially, we select the best model for each segment by minimizing its description length in terms of the BIC

$$\widehat{\boldsymbol{\theta}}_m = \underset{\boldsymbol{\theta}_m}{\operatorname{argmin}} \text{BIC}(\mathcal{X}_m, \boldsymbol{\theta}_m) = \underset{\boldsymbol{\theta}_m}{\operatorname{argmin}} (\Omega_m + U_m \ln N_m), \quad (4)$$

where  $\Omega_m = \Omega(\mathcal{X}_m, \boldsymbol{\theta}_m)$ . We aim at merging neighboring segments by evaluating the gain of description length when fitting a new model to the joined set of points. Assume that we already found models  $\widehat{\boldsymbol{\theta}}_m$  and  $\widehat{\boldsymbol{\theta}}_{m+1}$  using the points  $\mathcal{X}_m$  of segment  $m$  and  $\mathcal{X}_{m+1}$  of segment  $m + 1$ , respectively. We propose the points of both segments to belong to a joined segment, thus,  $\mathcal{X}_{m,m+1} = \mathcal{X}_m \cup \mathcal{X}_{m+1}$ . Again, we select the best model, for this potentially merged segment, by minimizing the BIC

$$\widehat{\boldsymbol{\theta}}_{m,m+1} = \underset{\boldsymbol{\theta}_{m,m+1}}{\operatorname{argmin}} \text{BIC}(\mathcal{X}_{m,m+1}, \boldsymbol{\theta}_{m,m+1}). \quad (5)$$

The gain of description length is given by the difference between the joint description length using the model  $\widehat{\boldsymbol{\theta}}_{m,m+1}$  obtained with the merged segments  $\mathcal{X}_{m,m+1}$  and the sum of descriptions lengths of both previous models  $\widehat{\boldsymbol{\theta}}_m$  and  $\widehat{\boldsymbol{\theta}}_{m+1}$

$$\begin{aligned} \Delta \text{BIC}_{m,m+1} &= \text{BIC}(\mathcal{X}_{m,m+1}, \widehat{\boldsymbol{\theta}}_{m,m+1}) \\ &- \left( \text{BIC}(\mathcal{X}_m, \widehat{\boldsymbol{\theta}}_m) + \text{BIC}(\mathcal{X}_{m+1}, \widehat{\boldsymbol{\theta}}_{m+1}) \right) \quad (6) \\ &= \Omega_{m,m+1} - (\Omega_m + \Omega_{m+1}) + U_{m,m+1} \ln(N_m + N_{m+1}) \\ &\quad - (U_m \ln N_m + U_{m+1} \ln N_{m+1}). \end{aligned}$$

If  $\Delta\text{BIC}_{m,m+1} > 0$  the description length of the merged segment is shorter than the description length of two separate segments; thus, they should be merged to reduce the overall complexity.

To evaluate the whole set of segments, we proceed in a greedy manner. After initializing all segments by their best models in terms of description length, we propose all neighboring segments to be merged and select the according best model. From all neighbored pairs which have a positive gain of description length, we select the one with the largest gain. We update the segmentation, such that one break point is removed. We iterate this process until there are no more merging proposals with positive gain of description length.

Finally, we inspect the estimated variance factors  $\hat{\sigma}_0^2$  of the finally merged segments. If they deviate more than 10% from  $\sigma_0^2$ , the initial variance of observations was too optimistic or too pessimistic, respectively. Thus, we considered restarting the process, using  $\sigma'_0 = \hat{\sigma}_0 \cdot \sigma_0$ . Hence, the final segmentation adapts to the given data and to the specific characteristic of each road segment. We refer to this variant of generalization as *adaptive* version.

Note that the resulting partitioning may deviate from the original junctions, as in the adjustment procedures the geometric elements are not restricted to any particular points. However, the covariance matrices for junction points – introduced at the beginning of Sec. 2 – could be re-weighted in order to prevent the according points changing their positions, in terms of their residuals. This is part of our future work.

## 5. RESULTS

To evaluate the accomplished work, we considered the dataset from the inner city of Munich, Germany. Given several aerial panchromatic images enriched by near infrared channel, a digital surface model and an orthophoto were calculated using the method of Rothermel et al. (2012). The resolution of the orthophoto was around 0.2 m. To perform classification, we first computed the digital terrain model by a standard procedure, which comprises extraction of several ground points followed by a spline interpolation and is described in (Bulatov et al., 2014). Then we excluded right away the set of *forbidden pixels* with implausible values of relative elevation and NDVI. Finally, we extracted some regions for training and evaluation procedure. Besides, we used stripes computed from pairs of nearly parallel lines in orthophoto to suppress the noise stemming from vehicles, traffic signals, etc.: If at least a certain percentage of pixels belongs to the road class, all other non-forbidden pixels are also assigned to the street class. There are still many mis-classifications in this difficult dataset. However, especially by choosing regions for training data extraction, we made sure that road pixels are extracted as correct as possible in the regions around the traffic roundabouts since this is where we want to demonstrate the performance of our algorithm.

In Fig. 2, we show two fragments of the dataset with classification result, the extracted polylines (chains are omitted), and the content of the shapefile obtained from a publicly available source Geofabrik (2017). These images show on the one hand the achievable accuracy of our street extraction module in comparison with the ground truth and on the other hand, the problematic of the wriggled road courses, which we will improve next. Thus, we will show in Sec. 5.1 the process of chain forming and in Sec. 5.2, we assess the results of generalization.



Figure 2: Detailed view of the classification results, where non-road class is emphasized by golden color, the input polylines, in blue, and the “ground truth” represented by the content of the OpenStreetMap shapefile, in red. Here, main and auxiliary roads are depicted by solid and dashed lines, respectively.

### 5.1 Chain Forming

We show in Figs. 3-4 the performance of both strategies for dissimilarity searching, namely by means of RANSAC outliers of the circle-fit function and deviations in partial dominant directions. The polylines not belonging to any chain (equivalently, to chain of cardinality one) will be denoted from here on as *side roads*. They are omitted in Fig. 3 and marked by thin cyan lines in Fig. 4. We see that the strategy based on fitting circles is better suitable for searching circular regions than comparing partial dominant directions. Thus, the circle in Fig. 4, right, has been correctly determined. However, in general, the function based on partial dominant direction tends to identify more natural street courses, which is best visible in Fig. 3, right, otherwise chains formed by straight lines become more easily interrupted. For regions not reasonable for generalization, such as foot paths, highlighted in Fig. 4, right, both methods exhibit rather short and senseless chains. Since only direct neighbors are considered for chain forming, situations where small segments appear between two junctions are undesirable, as well and should be avoided by means of DBSCAN. It remains to say that circle-fit + RANSAC is

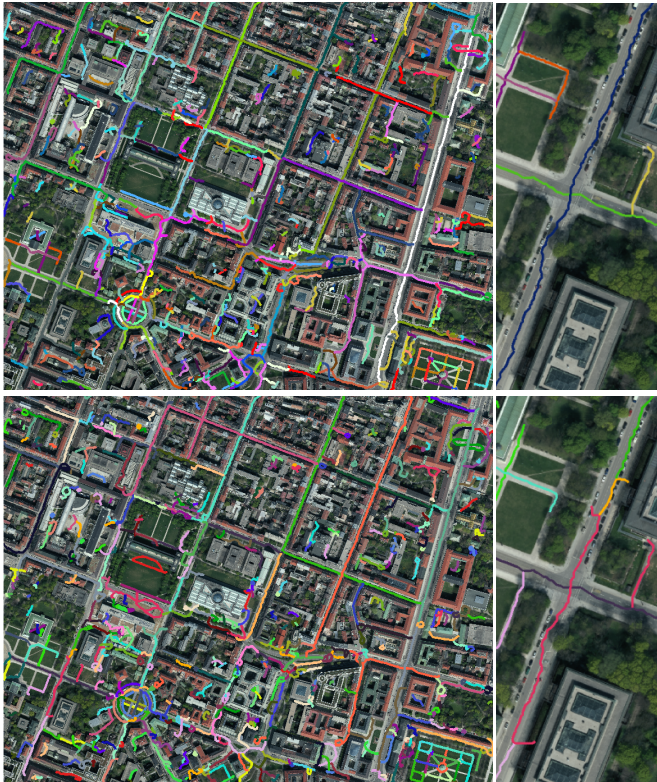


Figure 3: Results of chain forming with neighborhood searching function based on partial dominant directions (top) and on circle-fit + RANSAC (bottom), visualized for the complete dataset (left) and a for long street (right). The chains are given random, arbitrary colors. Side roads were omitted.

a slightly more time consuming strategy than the histogram analysis needed for estimation of partial dominant directions (around 20% of time) and the main problem of the latter method is made up by noisy assignments of directions to the individual segments.

## 5.2 Generalization

We processed once the chains and once the side roads with the adaptive and non-adaptive version of generalization. In what follows, the differences between the results as well as their dependencies on the underlying function for neighborhood search will be analyzed. We observe in Fig. 5 that by using the adaptive version of generalization, straight line segments often tend to be approximated by groups of circle arcs while circle arcs are likely to be approximated by ellipses. This is due to the fact that model selection tries to solve the trade-off between precision and model complexity. Down scaling the assumed accuracy of observations leads to smaller residuals and more complex geometries. We can also see that the gap between the endpoints of a neighboring ellipse and a line is often smaller than in case of circles. Coming to the comparison of underlying function for neighborhood search, we see that after the approach based on partial dominant directions, several circle arcs belonging to the traffic roundabout in Fig. 5, top, are lost, however not many, and that the whole circle could not be recognized in Fig. 5, third row. Application of circle-fit + RANSAC allows extracting this circle completely (within the non-adaptive approach). As a disadvantage, we can see in Fig. 5, second row, some hallucinated circle and ellipse arcs. Additionally as described in previous section, long straight streets are sometimes interrupted and the slopes of the single regression lines are not identical.



Figure 4: Detailed view of chain forming in case for two traffic roundabouts, using partial dominant directions (top row) and RANSAC (bottom row).

Probably, in order to get rid of some erroneous circle and ellipse arcs, we should – apart from drastically improving the classification result – consider clothoids instead of ellipses. They are known to be an essential part of design of road geometries, they have less degrees of freedom than ellipses and they would perfectly fit in our model estimation and selection procedure from Sec. 2 and 4.

To demonstrate the advantages of fusion with respect to the previous approaches, Fig. 6 shows the results of application once of the generalization module based on multiple primitives (circle-fit-based, non-adaptive) but *without fusion*, visualized by dashed red straight lines and yellow circle arcs and once the result of polyline-wise Douglas-Peucker algorithm modified by Bulatov et al. (2016b), shown by blue line segments. As expected, the latter approach extremely compresses the number of vertices and the junction positions remain fixed, yielding sometimes slant road courses. This usually does not happen with red lines, since these result, basically, from a regression procedure. However, junction positions are not fixed anymore. Also, the former method tends to recognize, where possible, circle arcs, which sometimes make the road course more realistic, but sometimes stem clearly from the noise. Besides, because there are usually not enough observations in a single polyline, these circle arcs were more difficult to recognize than with chain-wise method visualized in Fig. 5 and the traffic roundabouts are not recognized that clearly. Summarizing, both alternatives do a fair job when it comes to generalize straight lines, however, in order to identify circular segments, polyline fusion seems to be indispensable.

## 6. CONCLUSIONS AND OUTLOOK

This work aimed to identify and calculate geometric primitives, such as straight line segments, circle and ellipse arcs within complicated road networks. The instances of these road networks are chains formed with raw polylines which have been identified as neighbors using geometric and topological similarities. Two possibilities in finding these similarities are to compare the partial dominant directions or to check the percentage of inliers for RANSAC with the circle-fit function taken as a basis. In both

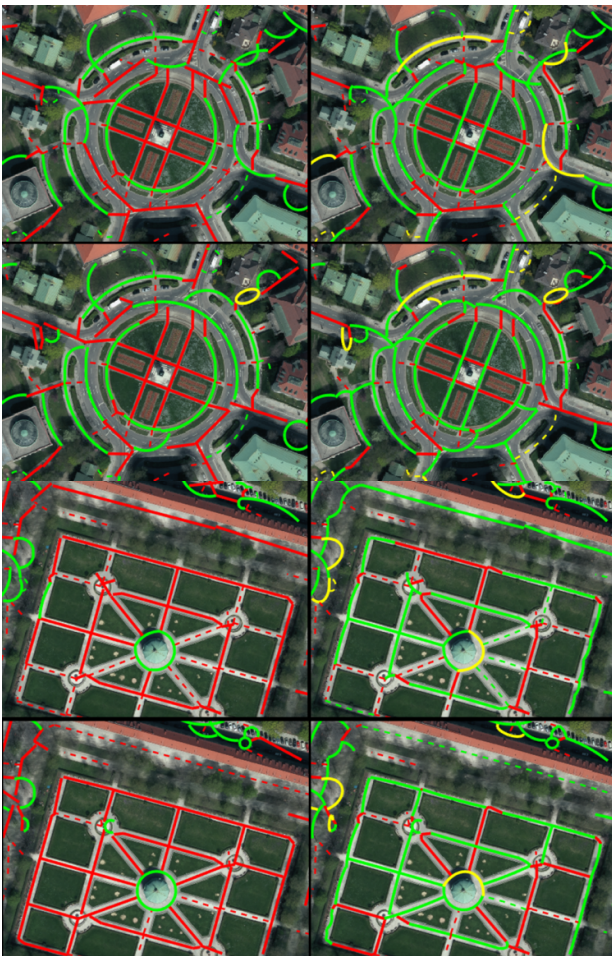


Figure 5: Detailed views of generalization results from two traffic roundabouts. The output of non-adapted and adapted generalization is shown, respectively, in the left and in the right images. The output of partial dominant directions and circle-fit + RANSAC is shown, respectively, in the top and in the bottom images. All elements stemming from a chain are highlighted with solid lines while the side roads are represented by dashed lines. We show the resulting circle and the ellipses arcs as well as the straight line segments by green, yellow, and red color, respectively.

cases, roads corresponding to both polylines are required to have similar width and to share a junction. Using a greedy approach based on model selection, we were able to identify the most of the important traffic roundabouts and street courses; it should also be mentioned that the whole generalization module has only one data-dependent parameter, namely the threshold  $t$ . Unfortunately, because of the noisy data and a lack of context information, it was not always possible to trace the whole circle arc. Besides, after applying the proposed procedure, the positions of junctions have been shifted, and their adjustment would in general either destroy the regular structures or require additional segments establishing connections. In order to restrict junction points to their positions during the generalization routine, we may change the covariance matrix of observations, such that these points get a high precision. This assumes a large enough number of observations to prevent the normal equation system from rank deficiency and will be a topic of our future work. Additionally, by luck, it is not critical in most applications since a street has a non-negligible width and so there remains some scope for a position of the junction. Instead,

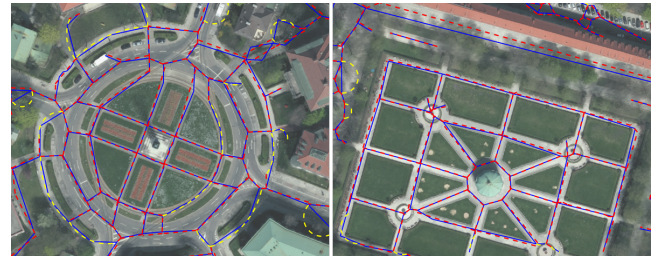


Figure 6: Detailed views of generalization results using polyline-wise (not chain-wise) alternative methods. Blue lines: Modified Douglas and Peucker (1973) algorithm, yellow and red dashed lines: Model selection approach from Sec. 4.

a smooth course of a road is very appealing for a simulation application (Bulatov et al., 2014) and the traffic roundabouts can be modeled appropriately. Besides, we wish to include in our future work a more thorough quantitative evaluation of results using ground truth data in form of shapefiles and more datasets.

## ACKNOWLEDGMENTS

We wish to thank Melanie Pohl for implementing the routines for estimation of partial dominant directions. We also wish to thank the reviewers for their fruitful comments.

This work has partly been supported by the DFG-Project FOR 1505 Mapping on Demand.

## References

- Albano, A., 1974. Representation of digitized contours in terms of conic arcs and straight-line segments. *Computer Graphics and Image Processing* 3(1), pp. 23 – 33.
- Bulatov, D., Häufel, G. and Pohl, M., 2016a. Identification and correction of road courses by merging successive segments and using improved attributes. In: *SPIE Remote Sensing, International Society for Optics and Photonics*, pp. 100080V–100080V.
- Bulatov, D., Häufel, G. and Pohl, M., 2016b. Vectorization of road data extracted from aerial and UAV imagery. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* pp. 567–574.
- Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P. and Wernerus, P., 2014. Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry and Remote Sensing* 93, pp. 157–170.
- Chaudhry, O. and Mackaness, W., 2006. Rural and urban road network generalisation: Deriving 1:250,000 from OS MasterMap. *Proc. of 22nd International Cartographic Conference of ICA*.
- Douglas, D. H. and Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10(2), pp. 112–122.
- Fitzgibbon, A. W., Pilu, M. and Fisher, R. B., 1999. Direct least-squares fitting of ellipses. *PAMI* 21(5), pp. 476–480.

- Förstner, W. and Wrobel, B., 2016. *Photogrammetric Computer Vision - Statistics, Geometry, Orientation and Reconstruction*. 1 edn, Springer International Publishing.
- Geofabrik, 2017. OpenStreetMap-Shapefiles. [www.geofabrik.de/en/data/shapefiles.html](http://www.geofabrik.de/en/data/shapefiles.html). Accessed at: 2017-01-24.
- Günther, O. and Wong, E., 1990. The arc tree: An approximation scheme to represent arbitrary curved shapes. *Computer Vision, Graphics and Image Processing* 51, pp. 313–337.
- Hinz, S. and Baumgartner, A., 2003. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* 58(1), pp. 83–98.
- Ji, Q. and Haralick, R. M., 1999. A statistically efficient method for ellipse detection. In: *Proc. of International Conference on Image Processing (ICIP)*, pp. 730–734.
- Meidow, J., Beder, C. and Förstner, W., 2009. Reasoning with uncertain points, straight lines, and straight line segments in 2D. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(2), pp. 125–139.
- Mena, J. B., 2006. Automatic vectorization of segmented road networks by geometrical and topological analysis of high resolution binary images. *Knowledge-Based Systems* 19(8), pp. 704–718.
- Moore, A., Mason, C., Whigham, P. A. and Thompson-Fawcett, M., 2003. The use of the circle tree for the efficient storage of polygons. In: *Proc. of GeoComputation*.
- Patraucean, V., Gurdjos, P. and von Gioi, R. G., 2012. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In: *12th European Conference on Computer Vision ECCV 2012*, Lecture Notes in Computer Science, Vol. 7573, Springer, pp. 572–585.
- Porrill, J., 1990. Fitting ellipses and predicting confidence envelopes using a bias corrected kalman filter. *Image and Vision Computing* 8(1), pp. 37 – 41.
- Rosin, P. L. and West, G. A. W., 1995. Nonparametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(12), pp. 1140–1153.
- Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. SURE: Photogrammetric surface reconstruction from imagery. In: *Proc. LC3D Workshop*, pp. 1–9.
- Sander, J., Ester, M., Kriegel, H.-P. and Xu, X., 1998. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery* 2(2), pp. 169–194.
- Schwarz, G., 1978. Estimating the dimension of a model. *The annals of statistics* 6(2), pp. 461–464.
- Steger, C., 1998. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(2), pp. 113–125.
- Türetken, E., Benmansour, F., Andres, B., Pfister, H. and Fua, P., 2013. Reconstructing loopy curvilinear structures using integer programming. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1822–1829.
- Wegner, J. D., Montoya-Zegarra, J. A. and Schindler, K., 2015. Road networks as collections of minimum cost paths. *ISPRS Journal of Photogrammetry and Remote Sensing* 108, pp. 128–137.
- Wenzel, S., 2016. High-level facade image interpretation using Marked Point Processes. PhD thesis, Department of Photogrammetry, University of Bonn.
- Wenzel, S. and Förstner, W., 2013. Finding poly-curves of straight line and ellipse segments in images. *Photogrammetrie, Fernerkundung, Geoinformation (PFG)* 4, pp. 297–308.