Forest point processes for the automatic extraction of networks in raster data

Alena Schmidt, Florent Lafarge, Claus Brenner, Franz Rottensteiner, Christian Heipke

Forest point processes for the automatic extraction of networks in raster data

Alena Schmidt[a,], Florent Lafarge[b], Claus Brenner[c], Franz Rottensteiner[a], Christian Heipke[a]

[a] *Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Nienburger Str. 1, D-30167 Hannover, Germany*
[b] *Titane Research Group, INRIA, 2004 routes des Lucioles, 06902 Sophia Antipolis, France*
[c] *Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Appelstr. 9a, D-30167 Hannover, Germany*

Abstract

In this paper, we propose a new stochastic approach for the automatic detection of network structures in raster data. We represent a network as a set of trees with acyclic planar graphs. We embed this model in the probabilistic framework of spatial point processes and determine the most probable configuration of trees by stochastic sampling. That is, different configurations are constructed randomly by modifying the graph parameters and by adding or removing nodes and edges to/ from the current trees. Each configuration is evaluated based on the probabilities for these changes and an energy function describing the conformity with a predefined model. By using the Reversible jump Markov chain Monte Carlo sampler, an approximation of the global optimum of the energy function is iteratively reached. Although our main target application is the extraction of rivers and tidal channels in digital terrain models, experiments with other types of networks in images show the transferability to further applications. Qualitative and quantitative evaluations demonstrate the competitiveness of our approach with respect to existing algorithms.

Keywords: Spatial point processes, RJMCMC, graphs, digital terrain models

[*] Corresponding author. Tel.: +49 511 762 2566; fax: +49 511 762 2483
Email addresses: alena.schmidt@ipi.uni-hannover.de (Alena Schmidt), florent.lafarge@inria.fr (Florent Lafarge), Claus.Brenner@ikg.uni-hannover.de (Claus Brenner), rottensteiner@ipi.uni-hannover.de (Franz Rottensteiner), heipke@ipi.uni-hannover.de (Christian Heipke)

1. Introduction

The automatic extraction of networks from images or lidar data is of great interest in various disciplines. For instance, in remote sensing data, roads and rivers represent networks. In medical data, neurons and vessels can be described as network. Knowledge about the appearance of these objects helps to update road maps, to make predictions in case of flood events and to provide information for automated diagnostic systems.

For the automatic extraction of objects, different techniques from the field of image analysis are employed, frequently with the objective to integrate a-priori knowledge into the model. The knowledge is often expressed in terms of probabilities and, thus, qualifies probabilistic strategies for object extraction. In image analysis, a well-known approach is the use of *Markov Random Fields* and *Conditional Random Fields* as introduced by [1] and [2]. Here, the image is represented by a graph whose nodes correspond to the pixels or segments; the edges indicate local context. Knowledge about the objects in the image may be integrated by favoring similar classes for pixels in a local neighborhood [3]. However, it is difficult to integrate more global constraints of the objects, e.g. concerning their shape.

In contrast, probabilistic model-based approaches express knowledge about the objects in a more holistic way. While this requires a careful modeling of the objects in order to stay general enough, it allows the integration of object characteristics beyond pixel- or segment-based relations. In this context, the method of spatial point processes has been shown to achieve good results in various disciplines, e.g. for object detection in remote sensing using raster data ([4], [5], [6], [7], [8]), point clouds ([9], [10], [11]) or for the object detection in terrestrial images ([12], [13]).

Such spatial point processes benefit from (1) their flexibility in integrating knowledge about the objects and their relation to other objects, (2) their variability of the number of objects which is not restricted in the sampling process, and (3) the determination of a global optimal object configuration.

In this paper, we aim to take advantage of these benefits in order to detect networks in raster data. The paper is structured as follows: First, state-of-the-art approaches dealing with network detection are reviewed (Section 1.1). Then, we describe the mathematical foundation of stochastic optimization with spatial point processes using a Reversible jump Markov chain Monte Carlo sampler coupled with simulated annealing (Section 2). In Section 3, we present the models employed for detecting networks of rivers and tidal channels in digital terrain models (DTMs), but beyond the detection of different types of networks in optical images. Our experimental setting and the results for different data sets are described in Sections 4 and 5, respectively. Finally, conclusions and perspectives for future work are presented in Section 6.

1.1. Related work

Several methods have been developed for the automatic extraction of networks in raster data. In the following, we focus on methods for the detection of (1) rivers and tidal channels - using 3D input data and integrating physical knowledge about the flow direction of water and (2) roads in remote sensing and vessels in medicine - using 2D images and integrating knowledge about the gray values and characteristics of the network.

River extraction. For some hydrological tasks, it is necessary to describe the flow of the water in the whole scene, e.g. in order to enable flood predictions. This is typically done with *flow routing* algorithms using DTMs in the form of raster data as input data, see [14] and [15]. Generally, these algorithms calculate the flow of water in the scene which is related to the *catchment area* of each pixel in the DTM. Considering that the movement is mainly driven by gravity and neglecting other influences such as the property of the materials and surface roughness, the flow of the water is only allowed to neighboring pixels with a lower height within these approaches. Two groups of *flow routing* algorithms can be distinguished: (1) *single flow* methods (e.g. *D8* [16], *R8* [17], *Kinematic*

Flow [18]) where the movement is restricted to one neighboring cell and (2) *multiple flow* algorithms (e.g. *TOPMODEL* [19], *FD8* [20], *MFD* [21]) where the flow of the water can be subdivided into multiple neighboring pixels with a lower height. An advantage of these methods is the computational efficiency; some of them are implemented in GIS software packages. However, they focus on steep terrain and are not fully transferable to hydrological applications with nearly horizontal terrain such as tidal channel networks as shown in [22].

The characteristics of tidal channels - structures similar to rivers in Wadden Sea areas - differ from those of normal rivers, because the flow direction changes four times a day due to the tides. As a consequence, specific methods for the automatic detection of these networks have been developed, mainly based on image analysis. For instance, [23] consider the height and the curvature of the terrain in DTM at different scales and combine the results in a threshold-based approach. The method detects most of the channels in the input data, but fails for some of the small channels and does not generate a completely connected network. In contrast, [24] develop an approach starting with low level image processing operations such as edge detectors which are subsequently processed in order to find a channel network. For that purpose, parallel edges corresponding to both channel borders are searched and median axes of the channels are determined and linked based on their directions. The authors also combine the approach with optical input data which, however, does not improve the results [25].

Network extraction with local strategies. A far larger number of network extraction approaches can be found in the field of road detection, see e.g. [26], and the diagnosis of networks and trees in medical data such as neurons or blood vessels, see [27] and [28]. Here, optical images or 3D image stacks are used as input data.

Some approaches solely consider local characteristics of the network such as geometric and radiometric features in a local neighborhood. These methods benefit from computational efficiency, but have the disadvantage that they are not robust against noise and fail in case of occlusions of the objects in the images. Among this group of approaches, some methods can be characterized as tracking approaches starting from some seed points and directions and expanding the network by iteratively adding points and paths. This is done using a *Kalman filter* strategy by [29] and [30] or by defining statistical tests concerning the width, direction and curvature of roads [31]. In [32], segments are gradually added to the network by analyzing their shape and deriving the dominant directions of streets. For medical data tracking [33] and [34] analyze features based on the Hessian matrix of the gray values in a local neighborhood or the straightness of detected lines in the network. If many streets or vessels are close to each other, a disadvantage of tracking approaches is that the network may be expanded in an incorrect way.

Many approaches define a set of rules for delineating road or vessel networks. Their general strategy is to detect parts of the network starting from an image segmentation ([35], [36], [37], [38]) or image classification ([39], [40], [41]) and to group these parts based on prior knowledge. In general, the knowledge is integrated in a heuristic way and requires the tuning of a large set of parameters for each scene.

Network extraction with global strategies. In contrast to local methods, global approaches evaluate paths between seed points in the entire input data set and optimize the network for the whole scene. Active parametric contour models (*snakes*) [42] first initialize a contour representing the network. This contour is deformed by internal forces (describing knowledge about the smoothness of the contour) and external forces (constraining the network to the data). For road networks the internal and external forces are described by constraints about the linearity and width of streets or about radiometric features in optical images [43] or SAR data [44]. [45] also allow the initialized network to expand by adding pixels in a local neighborhood in the case of similar features of these pixels. There are different possibilities to extend the parametrization of the contours, for instance by considering the width of the roads with *Ribbon snakes*

[46] or modeling the contour by a graph in *network snakes* [47] and, thus, benefit from the direct consideration of the network topology. However, the modeling by *snakes* requires a good initialization of the network and, thus, makes these methods more suitable to applications where an approximate contour is given or extracted in a preprocessing step.

Another group of global methods is based on minimum cost paths. By using graph structures, the topology is modeled explicitly. For road networks, one of the first approaches using minimum cost is the method of [48]. The results of different line detectors are combined and a score function indicating the probability belonging to the network is evaluated. The *A** algorithm is employed to find the optimal path, i.e. the one having the highest scores (equivalent to the minimum costs). The minimum cost path is also used by [49] and [50] in order to connect road segments from a pre-processing step or to find the best path in classified image data [51]. By evaluating all pixels along the edges, the methods based on minimum cost path can successfully bypass single noise pixels in the data. Moreover, these methods provide not only the geometry, but also the topology of the network. However, in general, all seed pixels have to be added to the graph, which may result in topological errors. Another strategy of network detection is to embed the problem in an image classification using graphical models based on *Markov Random Fields* or *Conditional Random Fields.* [52] and [53] first generate line segments by using different line detectors and topological filters. [54] or [55] calculate probabilities for each pixel or superpixel, respectively, for being part of the road network and, then, randomly choose pairs of pixels with high probabilities and connect them by optimal paths. By using higher order clique potentials global context is integrated. However, global constraints of the objects, e.g. concerning their shape, are difficult to integrate in such an approach.

Network extraction by stochastic approaches. The network detection by spatial point processes is used for road network detection for instance by [56]. The authors introduce a model consisting of line segments having a certain width.

Two line segments are considered to be connected if the smallest Euclidean distance between their endpoints is smaller than a predefined threshold. In the optimization process, the line segments are modified by changing their lengths and widths. Moreover, line segments are added to or removed from the configuration. For the evaluation of each object configuration, segments which are not connected to other segments on both sides are penalized. They are also penalized if they overlap, are too short, or enclose a small angle. The conformity of the object hypotheses with the input data is evaluated based on the homogeneity of the pixel gray values inside the segments and the gray value differences to the pixels outside the segment. [57] refine the method and avoid constant penalization terms by evaluating the segment configuration depending on the angle and the distance between two adjacent segments. An expansion of the model of [56] to 3D data is realized by [58] and [59], modeling the objects by cylinders instead of line segments.

The methods based on marked point processes mentioned above have in common that they require the definition of geometric constraints in order to regard segments as connected, and, thus, enforce the connectivity of the network. Moreover, they do not directly provide the topology of the network. In contrast, this is done in [5] by modeling the network as a planar graph. In the sampling process, the nodes representing junction-points are iteratively connected by edges, which correspond to line segments and may have different widths. The edges are evaluated based on the homogeneity of the gray values inside the segments and the gradient magnitudes on the segment borders. The authors penalize graph configurations with non-connected components, atypical intersection angles and atypical numbers of outgoing edges (both compared to training data). However, the applications are restricted to 2D image data, and the method does not integrate physical knowledge. Moreover, the graph in the approach of [5] contains cycles which does not seem appropriate as object model for our main target application: the extraction of river networks. In general, river networks are structured as tree, where two or more streams converge and form a bigger stream in the network.

1.2. Contributions

In this paper, we propose a new type of spatial point process for the detection of line networks. Similarly to [5], our spatial point process is designed to sample planar graphs. In our case, graphs are guaranteed to be acyclic so that the sampled structure is a tree or a set of trees. We call this process a *forest point process.* This specificity requires us to define adequate perturbation operators. Moreover, our model differs from the approach of [5] by the energy function which is adapted for the detection of river networks in our model. Here, we integrate physical knowledge about the network from 3D input data. In comparison to our previous work ([60], [61]), we present an enhanced model which is characterized by a reduced number of parameters, an extension of the energy model and an additional kernel. Our contribution is that (1) we develop a new spatial point process for capturing tree structures in images, (2) we take advantage of the 3D input data and, thus, facilitate the integration of physical knowledge about river networks, (3) we develop a hierarchical optimization strategy and (4) we show the transferability to river networks in different types of terrain, but also to different networks in image data.

2. Theoretical background

2.1. Spatial point processes

Point processes belong to the group of stochastic processes, see [62] and [63]. A point process is a sequence of random variables X_t that takes values in a state space $\mathcal{X}$. In our application t represent the iteration in the sampling process. In object extraction, point processes are used to find the most probable configuration of objects in a scene given the data. A point is described by its location $v_i = (x_i, y_i)$. In a marked point process, a set of marks m_i, i.e. a multidimensional random variable describing an object of a certain type at position v_i, is added to each point. If we characterize an object $l_i = (v_i, m_i)$ by its location and mark, a marked point process can be thought of as a stochastic model of configurations of an unknown number n of such objects in a bounded region S (here: the DTM, thus the objects l_i exist in $\mathcal{X} = R^2$).

There are different types of point processes, each of them associated with a *measure* which assigns a numerical value to each suitable subset B of S and, thus, makes different realizations of point processes quantitatively comparable. For the Poisson point process - one of the fundamental point processes - the measure is given by the expected number of objects in each subset. The Poisson point process assumes a complete randomness of the objects and models the probability $p(n)$ for the number of objects by a discrete Poisson distribution

$$p(n) = \frac{\lambda(B)^n e^{-\lambda(B)}}{n!}. \tag{1}$$

In (1) the parameter λ - also called intensity parameter - corresponds to the expected value for the number of objects in the subset B. In practice, the assumption of complete randomness of the object distribution is often not justified, and more complex models are postulated instead. These models are described by a probability density function g with respect to a reference point process, which is usually defined as the Poisson point process.

In our approach, g is expressed through a Gibbs energy U under the form $g \propto exp - U$. In this way, we integrate interactions between adjacent objects by using Gibbs point processes, which are also applied e.g. in ([56], [64]). The Gibbs energy can be modeled by the sum of a data energy $U_d(X)$ and a prior energy $U_p(X)$:

$$U(X) = \beta U_d(X) + (1 - \beta) U_p(X) \tag{2}$$

where $X \in \Omega$ is the object configuration from the set of all configurations Ω and $\beta \in [0, 1]$ describes the relative influence of both energy terms. The data energy $U_d(X)$ measures the consistency of the object configuration with the input data. The energy $U_p(X)$ introduces prior knowledge about the object layout; our models for these two energy terms are described in Section 3.3. The optimal configuration $\widehat{X}$ of objects can be determined by minimizing the Gibbs energy $U(X)$, i.e. $\widehat{X} = \underset{X \in \Omega}{\arg\min}\, U(X)$. This is typically done by coupling a Reversible jump Markov chain Monte Carlo (RJMCMC) sampler and a simulated annealing relaxation.

2.2. Reversible jump Markov chain Monte Carlo Sampling

Markov chain Monte Carlo (MCMC) methods belong to the group of sampling approaches. The special feature of MCMC methods is that the samples are not drawn independently, but each sample X_t is drawn on the basis of a probability distribution that depends on the previous sample X_{t-1}. Thus, the sequence of samples forms a Markov chain in the space of possible configurations. If the number n of objects constituting the optimal configuration $\widehat{X}$ were known and constant, it could be determined by MCMC sampling ([65], [66]). RJMCMC is an extension of MCMC that can deal with an unknown number of objects and changes of the dimension of the parameters between two sampling steps [67]. The sampler proposes changes to the current configuration from a set of pre-defined types of changes. Each of the change types is associated with a density function Q_i called *proposition kernel.* Each kernel Q_i must be reversible, i.e. the inverse change must be possible [68]. At each iteration t, a kernel Q_i is chosen randomly according to a proposition probability p_{Q_i} which may depend on the kernel type. The configuration X is changed according to the kernel Q_i, which results in a new configuration X'. Subsequently, the Green ratio R ([67]) is calculated:

$$R = \frac{\pi(X')}{\pi(X)} \, \frac{p_{Q_i^r}}{p_{Q_i}} \, \frac{Q(X,h|X',h')}{Q(X',h'|X,h)} \left| \det\left(\frac{\partial f_{X \to X'}(X,h)}{\partial(X,h)} \right) \right| . \tag{3}$$

In (4), p_{Q_i} and $p_{Q_i^r}$ are the probabilities for choosing kernel Q_i and its inverse kernel Q_i^r, respectively. The kernel ratio $\frac{Q(X,h|X',h')}{Q(X',h'|X,h)}$ expresses the ratio of the probabilities for the change of the configuration from X' to X and from X to X'. In this context, possible changes in the parameter dimension are taken into account by introducing auxiliary dimension-matching variables h and h' [67]. They depend on a transition function f which maps one configuration to another one, $(X',h') = f_{X \to X'}(X,h)$. Its Jacobian determinant $\det\left(\frac{\partial f_{X \to X'}(X,h)}{\partial(X,h)}\right)$ is incorporated in equation 4. Both, f and h, have to be chosen depending on the application. $\pi(X)$ is the target function that has to be optimized. By using the Gibbs energy in our application, equation 3 becomes

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p_{Q_i^r}}{p_{Q_i}} \frac{Q(X,h|X',h')}{Q(X',h'|X,h)} \left|\det\left(\frac{\partial f_{X\to X'}(X,h)}{\partial(X,h)}\right)\right|. \tag{4}$$

We also coupled the RJMCMC sampler with simulated annealing. For that reason, the parameter T_t referred to as temperature ([69]) is introduced in equation (4). The sequence of temperatures T_t tends to zero as $t \to \infty$. Theoretically, convergence to the global optimum is guaranteed for all initial configurations X_0 if T_t is reduced (cooled off) using a logarithmic scheme. In practice, a geometrical cooling scheme is generally introduced instead because it is faster and usually still gives a good approximate solution ([70], [7]).

Following [65] and [66] the new configuration X' is accepted with an acceptance rate α and rejected with the probability $1-\alpha$, where α is computed from R using

$$\alpha = \min(1, R). \tag{5}$$

In practice, we sample a random number $\gamma \in [0,1]$. If $\gamma < \alpha$, the new configuration is accepted and X' becomes the new configuration ($X_{t+1} = X'$). Otherwise the old configuration is maintained ($X_{t+1} = X$). The four steps of (1) choosing a proposition kernel Q_i, (2) generating the new configuration X', (3) computing the acceptance rate α, and (4) accepting or rejecting the new configuration are repeated until a convergence criterion is achieved.

3. Extraction of line-networks

We use spatial point processes with RJMCMC sampling and simulated annealing as described in Section 2 to detect networks in raster data. First, we define the object representation of the networks (Section 3.1). Second, the types of change in the graph configuration which are applied during the sampling process are introduced (Section 3.2). Third, the energy function to be minimized during the global optimization is defined (Section 3.3).

3.1. Forest point process

Contrary to traditional marked point processes in which interactions between points are formulated in the energy through a neighboring relationship, we specify the adjacency of points in the space of marks directly. Each point of a configuration is thus associated to a set of point indices that gives the list of its neighboring points. This allows us to associate a unique adjacency graph to each configuration of points where the nodes of the graph correspond to the points and the edges correspond to pairs of neighboring points, as illustrated in Figure 1. As our target application is the extraction of rivers, we restrict the admissible topology to trees, i.e., undirected, acyclic graphs. Indeed we guarantee that water flows only in one direction by avoiding cycles in the graph. Simultaneously, we accept that we cannot detect two channels circling round an island. Note that this particular case rarely occurs in our application. Since our algorithm is applied to scenes which may contain multiple independent rivers, the output is a collection of one or more disjoint trees, which is commonly known as a forest [71].

We define by *forest point process* a point process where each point is associated to a set of point indices so that the adjacency graph of any configuration of points is a forest. In other words, we restrict the form of the adjacency graph to undirected acyclic planar graphs to guarantee that any configuration of points can be represented as a set of trees.

In our approach, we enrich the space of marks so that each point is also associated to a parameter specifying the width of corresponding edges. A point l is thus defined by the set of variables $l = (v, i_1, ..., i_k, w_1, ..., w_k)$ where $v = (x, y)$ refers to the position of the point, $i_1, ..., i_k$ are the indices of its k adjacent points, and $w_1, ..., w_k$ are the widths of the edges between the point and its adjacent points. When $k = 1$, the point is an extremity node. When $k > 1$, the point is a branch node that can be either associated to a change of direction ($k = 2$) or to a confluence of branches ($k > 2$), as illustrated in Figure 2.

Note that, when a pair of points are adjacent, they have an identical edge width in their set of marks. We denote by e the edge forming by two adjacent

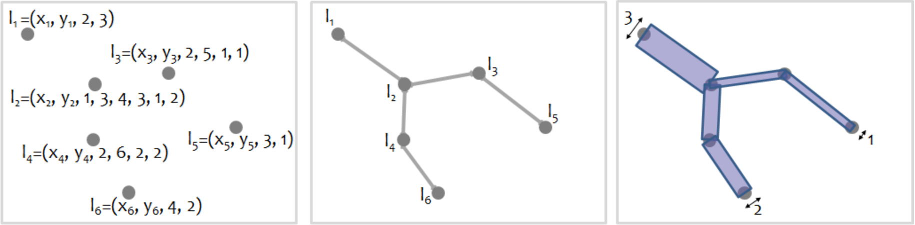

Figure 1: Forest point processes. (left) A forest point process generates configurations of points where each point l_i is associated with the index of its adjacent points (e.g. point index 2 and 5 for l_3) and their branch width (e.g. 1 for branch (l_2, l_3), and 2 for (l_4, l_6)). (middle) For any configuration, this formalism allows us to connect the points with a unique adjacency graph which is planar and acyclic. (right) The output network is a set of trees represented by connected line-segments, each line-segment having a specified width.

points in the adjacency graph.

3.2. Changes of the object configuration

In the sampling process, we modify the forest describing the network of objects in the raster data. For that purpose, potential changes (also referred to as perturbations) that can be applied as well as the corresponding kernels Q_i must be defined. We allow three types of changes: (1) *birth-and-death*: points are added to or removed from the current forest, (2) *modification*: the parameters of the points are modified, (3) *split-and-merge*: two points are split or merged which is done by merging the end nodes of two edges or by replacing their common end node by a new one for one of these edges while keeping the end node for the other edge unchanged. As mentioned in Chapter 2, each perturbation is required to be reversible, i.e. for each perturbation there has to exist an inverse perturbation; for instance, the death event is the inverse perturbation of the birth event. The corresponding kernel Q_i indicates the probability for a specific change to a new configuration.

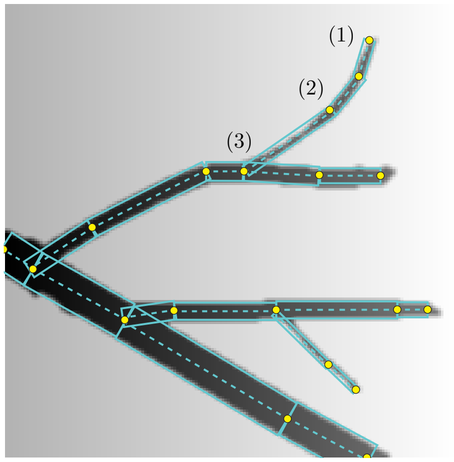

Figure 2: Types of nodes. The nodes correspond to either (1) extremities in the tree, (2) changes of direction, or (3) confluences of branches.

3.2.1. Birth-and-death

In the optimization, nodes and edges are added to or removed from the current forest, which is accomplished by the birth-and-death kernel Q_{BD}.

In the case of a birth event, the position of the new node (x_j, y_j) is randomly chosen based on a *probability map* which assigns the probability for being part of the river network to each pixel in the DTM (see Section 3.4). Then, the node is added to the graph. In our model, each node has to be connected with at least one node in the forest. That is why we search for neighboring nodes in a local neighborhood r. We check if the connection to them does not result in a cycle in order to guarantee the acyclicity of the adjacency graph. Moreover, we check if the edges do not cross in the object space. From all neighboring nodes which fulfill these criteria, we randomly choose one and connect it to the new node by an edge (see Figure 3 top). If no neighboring points exist, the position of a second node $(x_{(j+1)}, y_{(j+1)})$ is randomly chosen, again, based on the *probability map* and the second node as well as an edge connecting both nodes and, thus, a new tree are added to the forest (see Figure 3 bottom). Then, we randomly choose a width for the new edge. Here, we assume a uniform distribution for all values in the range of the minimal and maximal width w_{min} and w_{max} which are given for each scene. In total, one new edge and one or two new nodes are added to the forest in the birth event. An example is illustrated in Figure 3. The kernel ratio for this setting can be reduced to the ratio of the probabilities $p(n)$ for the number of nodes in the forest which is given by the Poisson distribution in equation 1. In this way, the Gibbs point process is set in reference to the Poisson point process. Then, the kernel ratio of the birth event is

$$\frac{Q_{BD}(X,h|X',h')}{Q_{BD}(X',h'|X,h)} = \frac{p(n|n')}{p(n'|n)} = \frac{\lambda^{n'}e^{-\lambda}}{n'!} \cdot \frac{n!}{\lambda^{n}e^{-\lambda}} = \begin{cases} \frac{\lambda}{n'}, & n' = n+1 \\ \frac{\lambda^2}{n' \cdot (n'-1)}, & n' = n+2. \end{cases} \tag{6}$$

In equation 6 parameters denoted with an apostrophe belong to the new configuration, parameters without an apostrophe to the old configuration, i.e. n is the number of nodes in the old configuration X and n' is the number of nodes in the new configuration X'. λ is the expected value for the number of nodes (see equation 1). The absolute value of the Jacobian determinant is 1 (see Appendix) and, thus, the Green ratio (equation 4) for the birth event for the case $n' = n + 1$ is

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p_{QD}}{p_{QB}} \frac{\lambda}{n'} \tag{7}$$

where p_{QD} and p_{QB} are the probabilities for choosing the birth and the death kernel, respectively.

In the case of a death event, we randomly choose a node with only one incident edge and remove it from the forest. We also delete its connection to the forest and remove the incident edge. If the adjacent node is not connected to a further node, we remove it, too. In total, one edge and one or two nodes are removed from the forest in the death event.

Analogously to the birth event, only the probability for the number of nodes is required for the kernel ratio which is decremented by one or two in comparison to the old configuration. Thus, the kernel ratio for the death event is

$$\frac{Q_{BD}(X,h|X',h')}{Q_{BD}(X',h'|X,h)} = \frac{\lambda^{n'}e^{-\lambda}}{n'!} \cdot \frac{n!}{\lambda^{n}e^{-\lambda}} = \begin{cases} \frac{n'}{\lambda}, & n' = n-1 \\ \frac{n' \cdot (n'-1)}{\lambda^2}, & n' = n-2. \end{cases} \tag{8}$$

Again, the Jacobian determinant is 1. Consequently, the Green ratio for the death event is given by

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p_{QB}}{p_{QD}} \frac{n'}{\lambda} \tag{9}$$

for $n' = n - 1$.

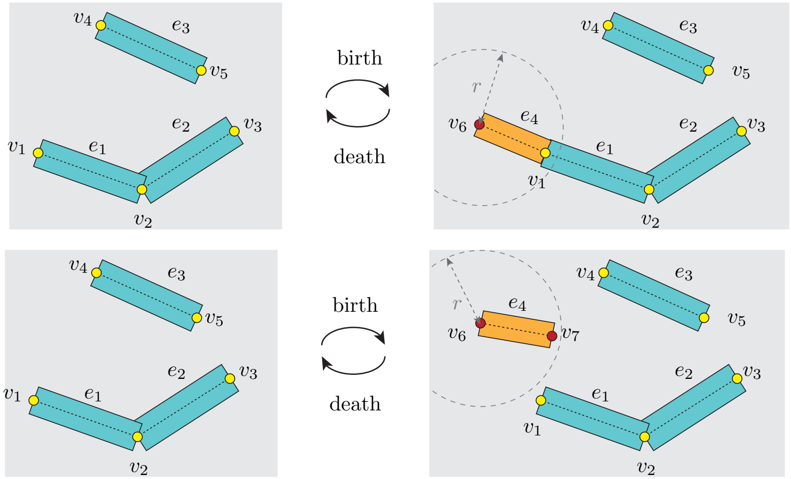

Figure 3: The initial example forest consists of three edges $\{e_1, e_2, e_3\}$ and five nodes $\{v_1, v_2, v_3, v_4, v_5\}$. Top: In the birth event node v_6 and edge e_4 are added to the forest. They are removed from the forest during the death event. Bottom: If neighboring nodes are not found for the new node v_6, a second node v_7 and the edge e_4 are added to the graph.

3.2.2. Modification

For the modification of the parameters of the forest, we distinguish between three types of perturbation:

(1) node translation

(2) edge width change

(3) edge connectivity change

We assume that all modifications are equally probable and randomly choose one of them. All types of modification are illustrated in Figure 4. For (1), we randomly generate a displacement vector within a local neighborhood. If the shift to the new position does not result in crossing edges, we change the node coordinates. For (2), we randomly change the width of an edge within a small range around the old width. For (3), we randomly choose a node and change its number of connections. This is done by adding or removing an edge to the forest. Both changes are equally probable, we randomly choose one of them. If we add a connectivity, similar to the birth event, we search for nodes in a local neighborhood r which can be connected to the chosen node without breaching the acyclicity or planarity of the adjacency graph. From all nodes which fulfill these criteria, we randomly choose one of them and add an edge connecting both nodes to the adjacency graph. If there does not exist a neighboring node, we do not change the configuration. For the reduction of the number of connections, we randomly choose a node and one of its incident edges. The edge is removed from the forest, if all nodes are still connected by another edge to the forest.

For the modification of the parameters, the Green ratio [1] is

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p^r_{QMo}}{p_{QMo}} \tag{10}$$

where p^r_{QMo} and p_{QMo} are the probabilities for choosing the modification kernel and its reversible kernel, respectively. Note, that all coordinates and width changes in these events are set to be equally probable and, thus, the Kernel ratio $\frac{Q(X|X')}{Q(X'|X)}$ is 1.

3.2.3. Split-and-merge

The third type of perturbation is a split-and-merge event where two nodes are split or merged.

[1] As the number of parameters does not change in this event, equation 10 is not a Green ratio in the proper sense, but rather the transition probability of the Metropolis-Hasting algorithm ([66]).

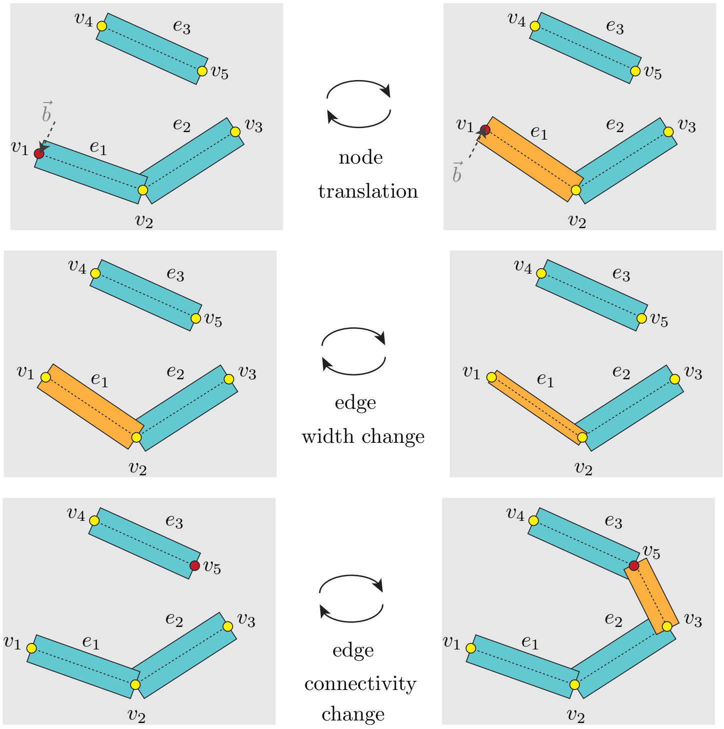

Figure 4: In the modification event, the parameters of the forest are changed. Modification (1): We change the coordinates of node v_1 of edge e_1. This is done by randomly generating a transition vector $\vec{b}$ in a local neighborhood. The inverse modification is the shift of node v_1 in the opposite direction. Modification (2): We randomly change the width of edge e_1. Modification (3): The connectivity of node v_5 is changed. Here, we add edge e_4 to the configuration and connect v_5 and v_3. In the inverse modification, the edge is removed from the forest.

In the case of a merge event, a node is randomly chosen from the forest. Then, nodes are searched in a local neighborhood. We check that merging two nodes does not lead to cycles or crossing edges in the forest. If more than one node fulfill these criteria, we randomly choose one of them. We set this node as new node for the edge and remove the chosen node (Fig. 5). The determinant of the Jacobian matrix for this setting is 1 and the kernel ratio is analogous to the birth event given by the ratio of the Poisson measure (equation 6). Then, the Green ratio of the split event is

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right)\ \frac{p_{QS}}{p_{QM}}\ \frac{n'}{\lambda}. \tag{11}$$

Conversely, in the case of a split event, a node with at least two incident edges is chosen randomly. Then, a local transition vector is randomly generated and a node of one of the incident edges is moved to the new position. A new node is added to the forest at this position. The Green ratio for this setting is

$$R = exp\left(-\frac{U(X') - U(X)}{T_t}\right) \frac{p_{QM}}{p_{QS}} \frac{\lambda}{n} \tag{12}$$

where p_{QM} and p_{QS} are the probabilities for choosing the merge kernel and the split kernel, respectively.

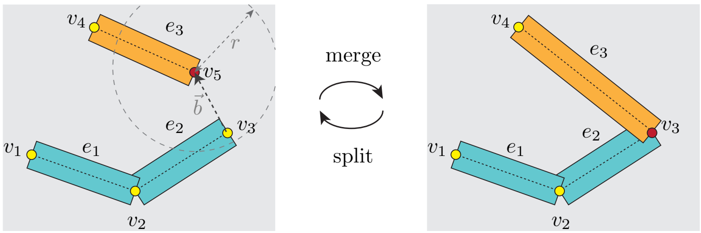

Figure 5: In the merging event, a node is randomly chosen, here v_5. It is merged with a node in a predefined neighborhood r, here v_3. Node v_5 is removed from the graph. Conversely, in the split event a node is randomly chosen, here v_3, and a transition vector $\vec{b}$ is randomly generated. A new node v_5 is added to the configuration.

3.3. Energy model

For the evaluation of each configuration we model a Gibbs energy U (equation 2) which is composed of a data energy U_d checking the consistency of the object configuration with the input data and a prior energy U_p where we introduce prior knowledge about the configuration.

3.3.1. Data energy

The model of the data energy is motivated by our main target application: the extraction of rivers and tidal channels in a DTM. These networks are characterized by locally lower heights than their surroundings. A typical profile of a

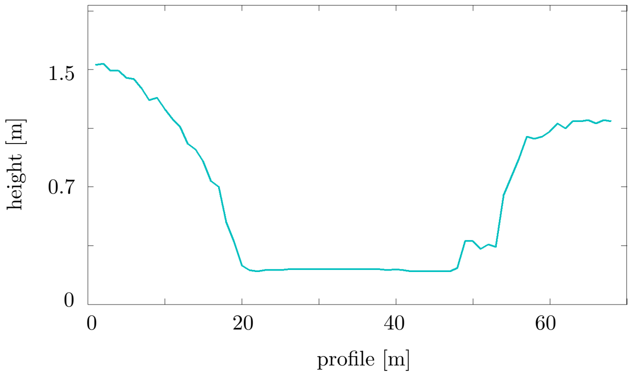

Figure 6: A typical example of profile of a tidal channel in our test site.

tidal channel is given in Figure 6. Consequently, rivers and tidal channels are assumed to have high DTM gradient magnitudes on each bank and homogeneous heights in between. We model the data energy by

$$U_d(X_t) = U_g(X_t) + U_h(X_t) \tag{13}$$

where each term represents one of the two assumptions about rivers.

High gradient magnitudes at the borders: We determine the data energy from the DTM gradients at the edge borders by

$$U_g(X_t) = \sum_{e_j \in X_t} \left(c_1 - \sum_{m_j=1}^{2} \frac{1}{n_{m_j}} \sum_{k_j=1}^{n_j} \nabla^{\perp}_{DTM_k} \right). \tag{14}$$

In (14), $\nabla^{\perp}_{DTM_k} = \left\langle \nabla_{DTM_k}, m^{\perp}_{j,k} \right\rangle$ is the component of the DTM gradient at boundary pixel k_j in direction of the normal vector of the lateral boundary $m_j \in 1, 2$ of the edge e_j (see Fig. 7). The sum of the gradients is taken over the n_j pixels k_j along that border; all gradients have equal weights. We only take into account the two lateral borders m_1, m_2 of the segment, potentially corresponding to the river or channel banks. To ensure that the energy is only minimized, if the data term is above a predefined value, we introduce a constant c_1.

Homogeneity of the heights: Perpendicular to the direction of the edge,

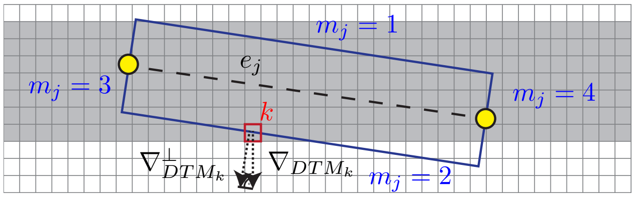

Figure 7: For the calculation of the data term, we take into account the DTM gradients of $m_j = 1$ and $m_j = 2$ and the homogeneity of the heights of $m_j = 3$ and $m_j = 4$ of each edge.

we analyze the DTM heights by

$$U_h(X_t) = p_h \cdot \sum_{e_j \in X_t} \max\left(0, -c_2 + \sum_{m_j=3}^{4} \sigma_z(m_j)\right), \tag{15}$$

where $\sigma_z(m_j)$ is the standard deviations of the heights along the perpendicular borders $m_j \in 3, 4$ of edge e_j in the forest (see Fig. 7). p_h is a weighting factor. We introduce a constant c_2 in order to be less susceptible to noise in the data. Since the data terms U_g and U_d are inconsistent with one another at the ends of the borders $m_j \in 3, 4$ where U_g forces the pixel to the pixels with the highest gradients and U_d requires homogeneous heights of the pixels, we do not take into account 5 % of the pixels on both ends of the borders $m_j \in 3, 4$ and calculate U_h only for the 90 % of the pixels in the middle of the borders.

Note, that despite developed for river networks, the data energy is principally transferable to any network in raster data that consists of homogeneous gray values and has strong gradients at the borders. If the gray values of the network are higher than the gray values of the background, we use the complementary image as input data.

3.3.2. Prior energy

Prior knowledge is integrated into the model in order to favor certain object configurations independently of the input data. We model the prior energy by three terms

$$U_p(X_t) = U_o(X_t) + U_s(X_t) + U_f(X_t) \tag{16}$$

where each term represents one of the characteristics of the network we intend to take into account in our model.

Non-overlapping segments: We penalize an object configuration in which the edges overlap. In this way, the accumulation of objects in regions with high data energy can be avoided. For that purpose, we calculate the overlap area a of all combinations of edges e_i and e_j and their relative overlap area $\frac{a}{a(e_i)}$ and $\frac{a}{a(e_j)}$. The corresponding energy term is calculated by

$$U_o(X_t) = p_o \sum_{i \neq j} \max \left(\frac{a}{a(e_i)}, \frac{a}{a(e_j)} \right), \tag{17}$$

whereby we only consider the larger relative overlap area of two edges e_i and e_j and p_o is a penalizing factor.

Graph connectivity: We want to obtain a configuration with only one tree and, thus, rate the connectivity of the nodes in the forest. For that purpose we penalize graph configuration consisting of two or more trees by

$$U_s(X_t) = p_s \cdot (n_{sg} - 1) \tag{18}$$

where n_{sg} is the number of trees and p_s is a penalizing factor.

Physical consistency of the network: For river networks, we take into account the flow direction of water by analyzing the heights of the nodes and edges of the forest. The general idea is to take into account that water does not flow uphill. We integrate this physical knowledge using two criteria. First, each node needs to be connected to exactly one node with a lower height value but can have connections to an arbitrary number of nodes with a higher height value - this means that we do not consider deltas or islands in our model. Second, each pixel on the medial axis of an edge has to show the same trend in the data, i.e. from a node to its adjacent node all height values have to increase or decrease; otherwise the edge is penalized. Note that we allow small deviations σ from this trend in order to be less susceptible to noise in the data. Thus, an edge is not penalized if the height of each pixel follows the trend of the edge within a tolerance of σ. Combining both criteria, the prior energy is modeled by

$$U_f(X_t) = \sum_{e_j \in X_t} p_f \cdot \left(n_{f_1} + \frac{n_{f_2}(e_j, \sigma)}{l(e_j)} \right). \tag{19}$$

In (19), n_{f_1} is the number of nodes in the forest violating the first criterion. For each edge e_j the number of pixels $n_{f_2}(e_j)$ violating the second criterion is counted and normalized by the length $l(e_j)$ of the edge.

Note that we do not use an explicit shape prior in our energy. However, as illustrated in Fig. 1, tree branches are modeled by rectangles whose width and length live in a bounded domain. This constitutes an indirect way to impose some prior shape knowledge without adding an additional term in the energy which would require more delicate parameter tunings: When a perturbation operator modifies, adds or removes a tree branch, the perturbation is not fully random, but driven by some restriction w.r.t. the shape.

3.4. Hierarchical approach and probability map

We develop a strategy for a hierarchical extraction of networks. For that purpose, we apply the method on the input data of the raster data with different resolution steps. We then search for the edges with the largest width in the lowest resolution, keep the results, reduce the *probability map* for the detected pixels and complete the network in the higher resolution by detecting edges with lower widths. For river networks this means, that we first detect the biggest streams in the network and then search for the smaller streams.

The *probability map* indicates the probability to be a node to each pixel in the raster data. We calculate the *probability map* based on the threshold for simple raster data features, e.g. the heights in the DTM or the gray values in the image. For river networks, we also convolve the DTM with derivatives of the Gaussian and calculate the terrain curvatures based on the second derivative in the direction perpendicular to the gradients in the DTM. We apply a simple threshold method and assign a probability of 1 to all pixels below / above this threshold while the probability is set to 0.01 for the other pixels.

4. Experimental setting

4.1. Experimental setup

We tested our method on different data sets and for different applications. First, the river and tidal channel network detection was evaluated. For that purpose, we applied our method for three DTMs. Synthetic data simulating an ideal tidal channel network were used in order to analyze the importance of each term in the energy function (Section 5.1). We also evaluated the influence of the generation of random numbers in the sampling process with this data sets (Section 5.2). The applicability to real data sets is shown based on two DTMs: a test site from the Wadden Sea with nearly horizontal terrain and a mountainous test site in Austria. On the one hand, we used this data for analyzing the influence of different parameter settings and, especially, for evaluating the influence of the integration of physical knowledge in the energy function (Section 5.3). On the other hand, we compared our results to those from a standard flow routing algorithm (Section 5.4). Finally, the transferability to different types of networks in images is shown (Section 5.5).

For the data sets of the Wadden Sea and Austria, we applied the hierarchical approach and searched for rivers in two resolution steps of the DTM. Moreover, for all experiments we used a geometrical cooling scheme in the simulating annealing process by reducing the temperature $T_t = T_{t0} \cdot (d_f)^t$ with a decreasing factor $d_f < 1$. We also implemented a logarithmic cooling scheme, which theoretically guarantees to achieve the global optimum and test it for the synthetic data set. Note that we do not implement a stop criteria for the optimization, but rather set the number of iterations sufficiently high.

4.2. Data sets

For the river detection, we tested our method using three DTMs. For some experiments, we only use a smaller section of theses scenes as input data. The sizes and grid size of the data sets are given in Table 1. The synthetic data simulates an ideal tidal channel network and does not include any noise. Moreover, the heights in the channels continuously decrease from the source to the mouth. The two real data sets are characterized by totally different terrain types. On the one hand, we extracted a tidal channel network from the Wadden Sea. Here, the terrain is nearly horizontal with height differences of 10 m and 1.7 m in the whole scene and the smaller section. In comparison to the synthetic data, the borders of the channels are less clear and the transition to land is smoother, especially for the small channels. On the other hand, we tested our approach on a river network in Vorarlberg, Austria. Here, the terrain is mountainous with height differences of nearl 700 m and 250 m for the whole scene or the smaller test site, respectively. For the synthetic data and the Wadden Sea data, we manually labeled the borders of the channels in the DTM and generated a binary mask of the network. Then, we used a contour-pruned skeleton approach [72] as implemented in Matlab [73] resulting in one pixel wide lines and defined the skeleton as centerlines of the channels. At the head of the streams where standard skeleton algorithms fail we manually digitized the centerlines. For the third data set orthoimages are available which helps to analyze the results. Moreover, reference data of the rivers centerlines are given for this test site. They include some piped river segments which we exclude from the quantitative evaluation.

For the detection of different types of networks in images, we tested our algorithm from the datasets of [9] that include images of river networks, leaf and retina vessels as well as regular textures as roof tiles (Figures 17 and 18). Apart from the roof tiles, all of these networks have a tree structure. The sizes of the data sets are given in Table 1. Quantitative comparisons with further approaches in the literature are available for these images.

4.3. Parameter setting

In our experiments, we set the parameters to values that were determined empirically based on our previous work [60]. Unless noted otherwise, they were kept fixed for all tests. In all experiments, we set the initial temperature in equation (4) to $T_{t=0} = 10$. The number of iterations was set to $t = 1 \cdot 10^6$ and $t = 10 \cdot 10^6$ for the synthetic and the real data, respectively. For the whole

Table 1: Data sets in our evaluation.

Data set (DTM)	**size of the scene** $[m^2]$	**grid size** $[m]$	$\Delta z[m]$
Synthetic data	170 x 170	1	1.7
Wadden Sea	8000 x 3500	2, 5, 10, 20	10.0
Wadden Sea (section)	1001 x 1001	1, 2	1.7
Austria	10000 x 7500	5, 10	700
Austria (section)	2500 x 2500	5, 10	250

Data set (image)	**size of the scene**
River	364 x 320
Leaf	651 x 364
Retina	365 x 378
Tiles	672 x 523

Austrian test site it was set to $t = 40 \cdot 10^6$. The decreasing factor in the cooling scheme was set to $d_f = 0.99999998$. The probabilities for choosing one of the changes and the probabilites of the kernels were set to be equally probable for all kernels (thus, the corresponding ratios in equations 7, 9, 10, 11 and 12 cancel to 1). The further parameters are listed in Table 2. In general, differences in the parameter settings can be explained by the varying appearance of the channels and rivers in the data sets and the size of the test sites.

4.4. Evaluation strategy

For the evaluation of the results we choose two different methods: a line evaluation for the river networks in the DTM and a pixel-wise comparison for the results of the networks in the images.

The line evaluation is performed based on the method of [75] by evaluating the edges (neglecting their width) in our graph with reference data of the channel centerline. This is done, first, by defining a buffer surrounding the reference centerline and the extracted edges in the graph, respectively, having a width of b_{ref} and b_{ext}. Then, we tap points on the edges at regular intervals and searched

Table 2: Parameter setting in the experiments. If the parameters vary with resolution of the input data, several parameters are listed.

Parameter	**Synthetic data**	**Wadden Sea**	**Wadden Sea (section)**	**Austria**	**Austria (section)**
λ	50	500	500	4000	500
β	0.13	0.15	0.35	0.1, 0.04	0.1, 0.04
$r[px]$	16	20, 15	15	5, 6	5, 6
p_h	5	3	3	20, 3	35, 3
p_o	300	200	300	500, 400	500
p_c	100	80, 100	100	250, 150	100
p_f	50	50	50	50	50
c_1	50	150, 50, 20	50	0	0
c_2	4	4	4	8	8

for corresponding points perpendicular to the centerline within the buffer. The quality of the results is calculated by the *completeness* and *correctness* rate which analyze if every object in the scene has been extracted and if the objects are correct extracted, respectively. The combination of both rates is given by the *quality* and can be used for a ranking of different results. The quality criteria are calculated by

$$Completeness\ CP = \frac{reference\ points\ within\ b_{ext}}{sum\ of\ all\ points\ in\ the\ reference}, \tag{20}$$

$$Correctness\ CR = \frac{result\ points\ within\ b_{ref}}{sum\ of\ all\ points\ in\ the\ result}, \tag{21}$$

$$Quality\ Q = \frac{CR \cdot CP}{CR + CP - CR \cdot CP}. \tag{22}$$

In order to analyze the geometrical accuracy we calculate the Root Mean Square Error *RMS* of the distances d between all result points within b_{ref} and the corresponding reference points by

$$RMS_d = \sqrt{\frac{1}{n_p}\sum_{i=1}^{n_p} d_i^2}, \tag{23}$$

where n_p is the number of points in the result within b_{ref}.

For the images, ground truth data are available in the form of binary images labeling all pixels within the network. For a pixel-wise comparison, we evaluated all pixels within the segments of the edges (considering their width) and calculate the completeness and correctness rate as well as the quality of the results.

5. Results

5.1. Quality

Figure 8 shows how the object configuration in the synthetic data set evolves in the course of the sampling procedure, as edges are added to or removed from the configuration or their parameters are changed. After about $760 \cdot 10^3$ iterations the tidal channel network is completely covered by segments. The contours of the channels correspond well to the borders of the edge segments and the resultant graph is completely connected.

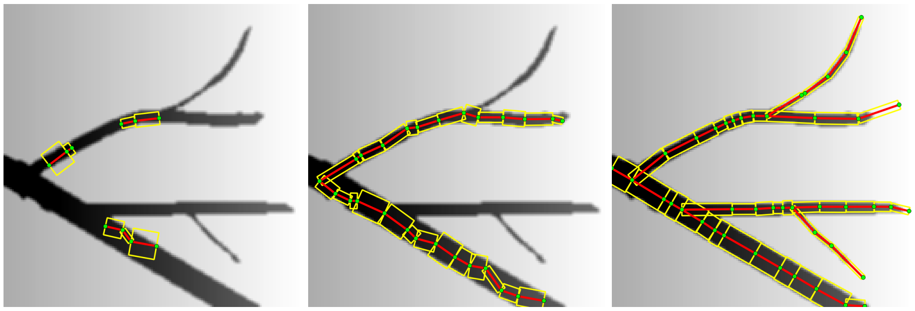

Figure 8: Results for the synthetic data set after $1 \cdot 10^2$, $1 \cdot 10^4$ and $1 \cdot 10^6$ iterations.

The quality parameters of the result in Figure 8 are given in Table 4. We also compared the logarithmic and the geometrical cooling in the process for this data set. For the logarithmic cooling, the completeness and correctness rates slightly differ, $CP = 98.8$ %, $CR = 94.0$ % in comparison to the geometrical cooling, $CP = 99.6$ %, $CR = 88.6$ %. The differences can be mainly explained by the difference of the edges at the streams' heads which are slightly beyond the channels borders in the second case. The RMS of the results is 0.66 m and 0.69 m at a grid size of 1 m.

We also analyzed the importance of each term in our model for the detection of rivers and tidal channels. This is done by successively excluding one of the terms. If we exclude the data term of the high gradient magnitudes at the channel borders ($p_g = 0$), we only detect some network parts (Fig. 9a). In general, the borders of the edges do not correspond to the channel banks. If we do not take into account the homogeneity of the heights perpendicular to the channels ($p_h = 0$), the quality decreases by 5.2 %. By excluding the penalization of overlapping areas ($p_o = 0$), nodes and edges accumulate near the medial axes of the channels where segments with similar widths are sampled at nearly the same positions (Fig. 9b). Thus, the graph does not represent the correct topology of the network. If we do not take into account the connectivity of the graph ($p_c = 0$), the sampling results in several trees which are not connected (Fig. 9c). Due to the data term, the segments still coincide with the borders of the channels rather well. However, we do not end up in a fully connected graph. If we exclude the term verifying the consistency of the flow directions ($p_f = 0$), the quality decreases by 6.2 %. In general, our results indicate that the chosen model seems to be suitable to describe the extraction problem of river networks.

5.2. Reliability

In order to exclude the influence of the generation of random numbers on the result, we repeated the geometrical cooling 100 times. Figure 10 shows the positions of the nodes in all 100 experiments. It can be seen that most of the nodes lie on the centerline of the reference data. In general, the scattering of the nodes positions is low. It is only high at the stream heads, which can be explained by the data term that is near $U_d = 0$ on the nearly horizontal

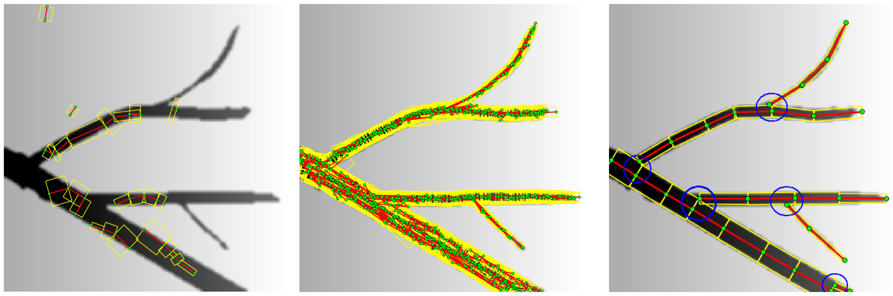

Figure 9: Results for the synthetic data set by excluding (left) the data term of the gradient magnitudes at the channel borders ($p_g = 0$), (middle) the penalization of the overlapping ($p_o = 0$) and (right) the evaluation of the graph connectivity ($p_c = 0$) where the blue circles indicate the parts of the network which are not connected.

terrain and, thus, does not increase the energy function. Moreover, we observe that the smallest channel is less often detected than the bigger ones. Gradient magnitudes at the borders are much lower than for the bigger channels and are therefore less often accepted in the sampling process. In order to calculate a precision of our result, we determine the standard deviation of all edge positions perpendicular to the channels at regular distance of 1 m for all edges in the 100 results. We only exclude the edge positions at the stream heads. In this way, we obtain a precision of 0.38 m on average that can be attribute to the variability of the random number generation.

We also analyze the energy curves in the experiments. Figure 11 shows the mean energy of all 100 experiments. In general, the performance can be considered as stable. The standard deviation of the energy decreases with increasing iteration. In the last iteration it is 2.6 % of the total energy which can be explained by the smaller channels which are not detected in all results. The number of nodes and edges varied by about 7 % in the final results of all experiments (see Table 3).

5.3. Applicability to different terrain types

We also evaluated our approach on real data: (1) a test site from the Wadden Sea with nearly horizontal terrain and (2) a mountainous test site in Austria.

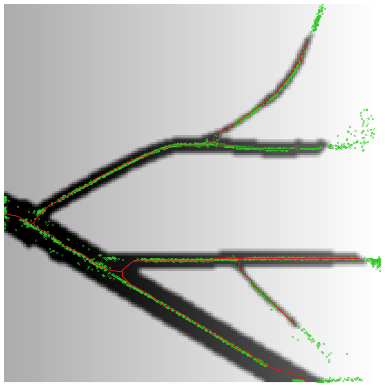

Figure 10: Reference centerline of the tidal channel in the synthetic data set (red) and the sampled node (green) for all 100 repetitions of the experiment.

Table 3: Variability of the results by repeating the experiment 100 times.

parameter	**mean**	**standard deviation**
number of nodes	37.6	2.6 (6.9 %)
number of edges	36.1	2.6 (7.2 %)
total energy U	-4720.4	126.3 (2.6 %)

For both data sets we used the hierarchical approach in different resolution steps and, first, detected the bigger streams in the data of lowest resolution and, then, searched for smaller streams in the higher resolution data. Note that we kept most parameters constant in these steps. The evaluation results for both scenes and two sections of them are given in Table 4. In general, the quality criteria are high. In the Wadden Sea (section) data set the completeness (CP), correctness (CR) and quality (Q) are $CP = 72.7$ %, $CR = 94.4$ % and $Q = 69.7$ %. All bigger channels are completely covered by line segments from the graph (Fig. 12). However, the lower order channels are not completely extracted. Here, the gradient magnitudes at the channels borders are much smaller than for the bigger channels and often show a smooth transition to land, so that the data term does not minimize the energy function in the same way. For most channels, the contours of the channels correspond well to the

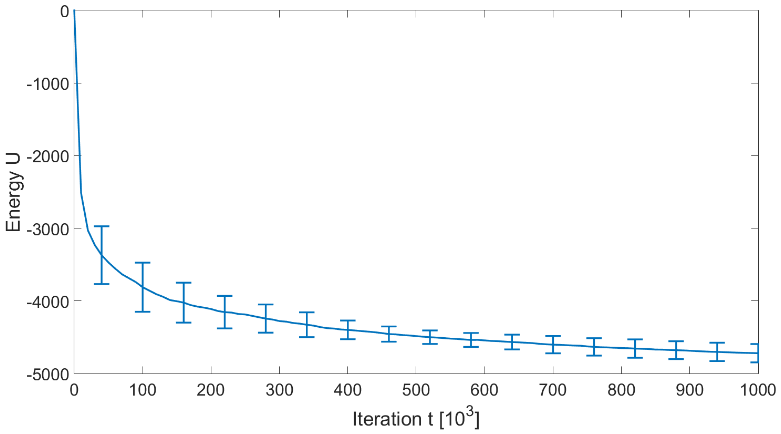

Figure 11: The energy is minimized during the sampling process and converges to the global optimum. The figure shows the mean energy of 100 experiments whereby the error bars represent the standard deviation in the respective iterations.

borders of the graph segments. The RMS of the position is 2.67 px at 1 m pixel size. For the whole Wadden Sea scene, several tidal channels are found in the different resolutions (Figure 13). However, some of the smaller channels are not detected which leads to a relatively low completeness rate of $CP = 50.6$ %. The correctness rate is higher, $CR = 78.2$ %. By using a pixelwise evaluation instead of the line evaluation described in Section 4.4 the completeness and correctness rates increase to $CP = 79.2$ % and $CR = 80.3$ %.

For the data set of Austria the first order stream is completely detected in the first hierarchical step as well. It is completely connected apart from one location where the riverbed is very wide and strongly curved (Fig. 14). We also extract many rivers of higher order which are difficult to see in the orthoimages. The accuracy results for this data set are $CP = 76.1$ %, $CR = 76.1$ % and $Q = 61.3$ %. The RMS of the position is 0.95 px with 1 m pixel size in the DTM. From the river network with a length of 247.3 km in the whole Austrian test site 156.1 km ($CP = 64.5$ %) are detected (Figure 15). The results show some misclassifications with terrain structures similar to river networks, e.g. valleys. Moreover, in the urban area on the left site of the scene some of the

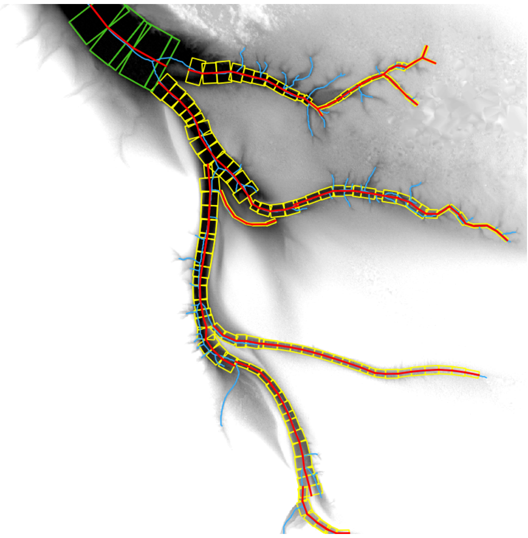

Figure 12: Sampling result for the Wadden Sea (section) area with the detected edges (red) of the graph its segments of the first hierarchical step (green) and the second hierarchical step (yellow). The blue line represents the reference centerline.

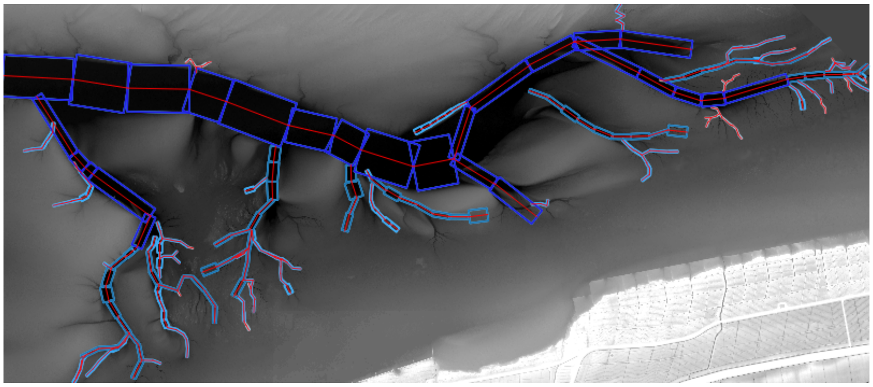

Figure 13: Results for the whole Wadden Sea scene. With increasing brightness of the blue segments, the resolution of the input data increases.

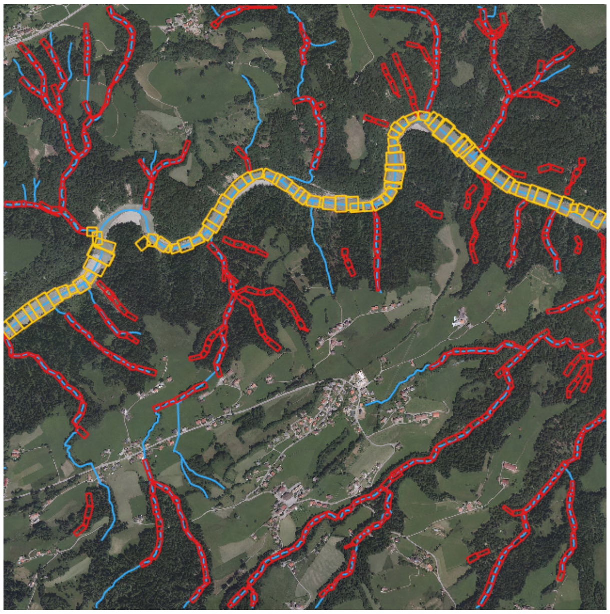

Figure 14: Segments of our approach of the first (yellow) and second hierarchical step (red) and the reference network (blue) for the data set of Austria with an orthoimage in the background ($\beta = 0.04$).

rivers are not detected.

For all scenes, we observe that the number of extracted rivers increases by increasing the weight β for the data term. Simultaneously, the correctness rate is reduced. These results confirm our model: By integrating prior knowledge about the network with a higher weight (lower value for β) the correctness rate increases. In contrast, the completeness rate increases by increasing the weight β and, thus, giving more weight to the data term.

We also analyze the influence of the prior term of the flow direction U_f on the results. By excluding this term, the quality of the results decreases by 0.5 % for the Austrian test site and by 1.1 % for the Wadden Sea scene (see Table 5). The correctness increases by 3.0 % and 1.7 %. The results show that the integration of 3D knowledge – which differentiates our approach from other marked point process such as [5] – helps to improve the correctness of the detected network.

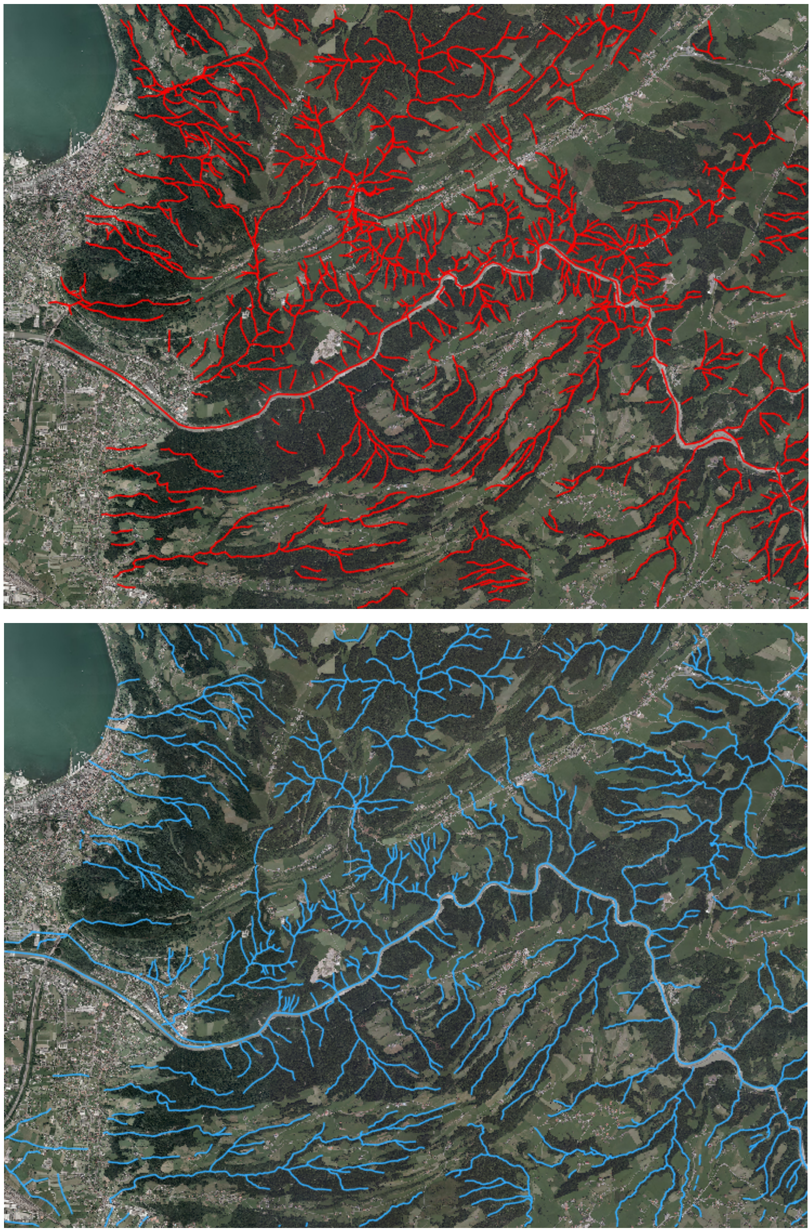

Figure 15: Results (top) and reference (bottom) for the whole Austrian test site ($\beta = 0.04$).

Table 4: Accuracy results (completeness CP, correctness CR, quality Q) by using buffer b for our method (fpp) with varying parameter settings for all test sites.

Data set	**Method**	**DTM**	**b**	**CP**	**CR**	**Q**	**RMS**
		[m]	**[m]**	**[%]**	**[%]**	**[%]**	**[px]**
synthetic data	fpp (log)	1	3	98.8	94.0	92.9	0.66
	fpp (geom)			99.6	89.0	88.6	0.69
Wadden Sea (section)	fpp, $\beta = 0.3$	1, 2	10	70.1	96.1	68.9	2.75
	fpp, $\beta = 0.4$			72.7	94.4	69.7	2.67
Wadden Sea	fpp, $\beta = 0.15$	2, 5, 10, 20		50.6	78.2	44.3	0.34
Austria (section)	fpp, $\beta = 0.04$	5, 10	15	76.1	76.1	61.3	0.95
	fpp, $\beta = 0.05$			82.2	64.4	56.5	0.95
Austria	fpp, $\beta = 0.04$	5, 10		61.4	68.4	47.9	1.04
	fpp, $\beta = 0.05$			64.5	63.1	46.9	1.05

Table 5: Analysis of the input of the prior term U_f concerning the flow direction.

Data set	**Method**	**CP** [%]	**CR** [%]	**Q** [%]	**RMS** [px]
Wadden Sea (section)	with U_f	72.7	94.4	69.7	2.67
	without U_f	70.3	92.7	66.6	2.56
Austria (section)	with U_f	76.1	76.1	61.3	0.95
	without U_f	78.4	73.1	60.8	0.97

5.4. Comparison with standard flow routing algorithms

We also compared our method with a standard flow routing algorithm and applied the *Multiple Flow Direction* approach as implemented in GRASS-GIS (r.stream.extract, [76]). This method was also analyzed in detail for a region of the same Austrian data set by [77]. In general, flow routing algorithms are computationally efficient; they calculate the flow of the water by exclusively analyzing height differences of neighboring cells in the DTM. That is why the flow routing algorithm outperforms the forest point process regarding the computing time, with computing times of a few seconds in comparison to a few minutes of our approach for the test sites shown in Figure 12 and 14. The quantitative and visual comparison for both test sites are given in Table 4 and Figure 16. We selected the following parameter setting for the flow routing algorithm: we set the minimum segment length to 100 m (similar to [77]) and evaluated different settings for the area of the catchment area C_a: $C_a = 7500\ m^2$, $C_a = 12500\ m^2$, $C_a = 25000\ m^2$ (the latter is similar to [77], the other values are chosen to be smaller as our test site is smaller). In general, the completeness rate decreases and the correctness rate increases by increasing the catchment area.

The visual comparison of the results in Figure 16 shows that the flow routing algorithm gives a description of the river centerlines in the form of zigzag lines. In contrast, the approximation by line segments in the forest point process results in straight lines which represent the river centerlines in a better way and, thus, give a higher geometric accuracy of the results. For the Wadden Sea test site, the geometric accuracy in our results is higher by a factor 1.6 ($RMS_{fpp} = 2.67\ px$, $RMS_{fr} = 4.35\ px$). For the Austrian test site the difference is lower, $RMS_{fpp} = 0.95\ px$ vs. $RMS_{fr} = 1.01\ px$. In contrast, the flow routing algorithm delivers better results than the forest point process concerning the connectivity of the network. Whereas all rivers are connected in the results of the flow routing algorithm, our results show few gaps, especially in those parts of the network where a small river flows into a bigger one. In order to overcome this problem, an extension of our model would be required, e.g. by calculating the prior energy term depending on the width of the line segments.

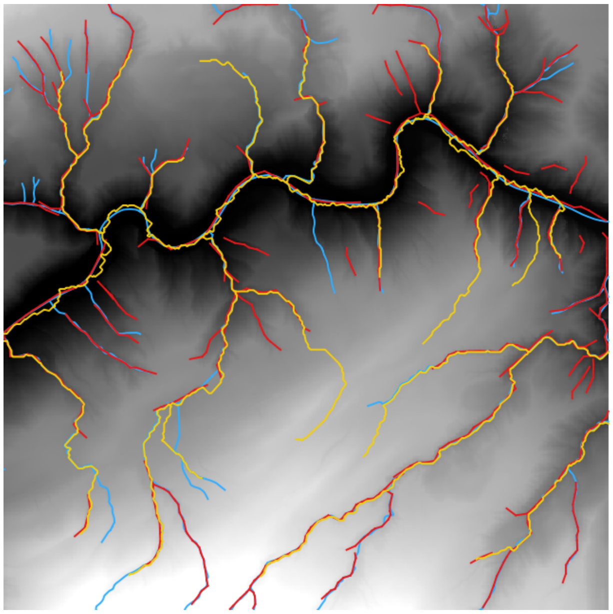

Figure 16: The extracted edges of our approach (red) compared to the reference network (blue) and the flow routing result, $C_a = 25000\ m^2$, (yellow) with the input DTM.

However, in regard to the correctness and completeness rates, our approach outperforms the flow routing algorithm for both test sites. Especially for the Wadden Sea test site, our correctness rate is much higher ($CR_{fpp} = 94.4$ %) in comparison to the flow routing algorithm ($CR_{fr} = 75.9$ %). For the Austrian test site the improvement in the correctness rate is lower, $CR_{fpp} = 76.1$ % in comparison to $CR_{fr} = 75.1$ %. However, we achieve a significant higher completeness rate, $CP_{fpp} = 76.1$ % in comparison to $CP_{fr} = 58.9$ %. Especially at the stream heads and near to the image borders, our approach achieves better results than the flow routing algorithm, which can be explained by the fact that the catchment area for these pixels is outside the DTM borders.

5.5. Transferability to networks in images

We also applied our method to image data showing different types of line-network in images. In comparison to the river network detection, the gradients and the homogeneity of gray values (instead of heights) are calculated in the

Table 6: Comparison of the results of our approach (fpp) with a flow routing algorithm (fr) with varying parameter setting for the catchment area C_a.

Data set	**Method**	**CP [%]**	**CR [%]**	**Q [%]**	**RMS [px]**
Wadden Sea (section)	fpp	**72.7**	**94.4**	**69.7**	**2.67**
	fr, $C_a = 7500$	72.0	61.6	49.7	4.35
	fr, $C_a = 12500$	68.7	71.3	53.8	4.39
	fr, $C_a = 25000$	60.3	75.9	50.6	4.52
Austria (section)	fpp	76.1	**76.1**	**61.3**	**0.95**
	fr, $C_a = 7500$	**78.1**	63.5	53.9	1.01
	fr, $C_a = 12500$	68.6	73.1	54.8	1.01
	fr, $C_a = 25000$	58.9	75.1	49.3	1.07

data term. As we did not use a DTM as input data, the term concerning the physical consistency of the network was neglected or set to a low weight for these data sets. Figure 17 shows the results of our method from three different types of images. In general, major parts of the networks are detected by our approach. For the *retina* network, we end up in one connected tree. In the results of the *leaf* and *retina* image, some of the smaller network parts are missing which can be explained by low gradients at their borders. In the *tiles* image, the network is completely covered by edges of the forest. Contrary to the junction point process proposed by [5], our algorithm - constructed for trees - fails to recover correctly the junctions from the *tiles* image. Here, small gaps occur and the forest is composed of several trees (see Tables 6 and 7).

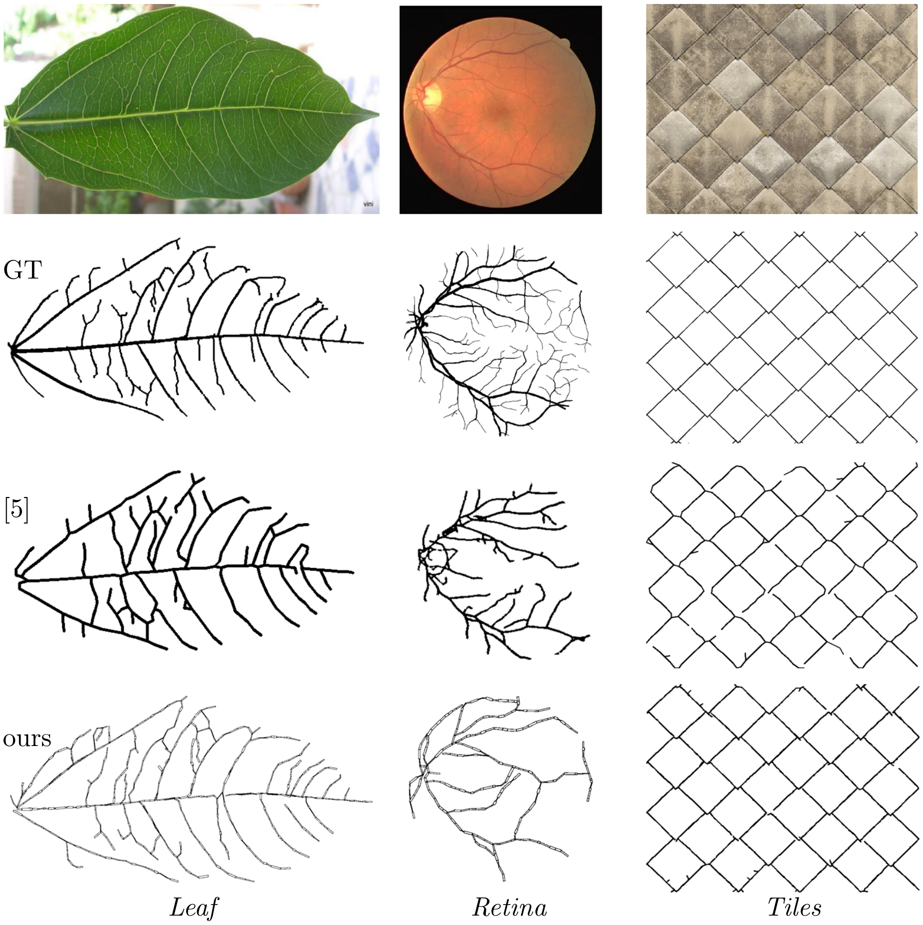

Figure 17: Comparisons with a junction point process [5] on different network images. Note that our tree-based network representation does not allow the junctions of the *Tiles* image to be recovered correctly.

Table 7: Quantitative comparison (correctness CR [%], completeness CP [%] and quality Q [%]) with a junction point process [5] from the different network images of Figure 17.

	Leaf			*Retina*			*Tiles*		
	CR	CP	Q	CR	CP	Q	CR	CP	Q
[5]	58.8	**70.6**	**47.3**	**60.4**	**54.4**	**40.1**	46.2	64.9	36.9
Ours	**69.5**	59.7	**47.3**	57.2	52.2	37.6	**65.1**	**87.6**	**59.6**

In comparison to other methods based on marked point processes or active contours, we achieve comparable results in terms or completeness, correctness and quality (see Table 8). Nevertheless, these methods are composed of many disconnected components as they do not use a graph structure to describe the network (see Figure 18). Here, our approach has advantages as it delivers the topology of the network by using a graph structure.

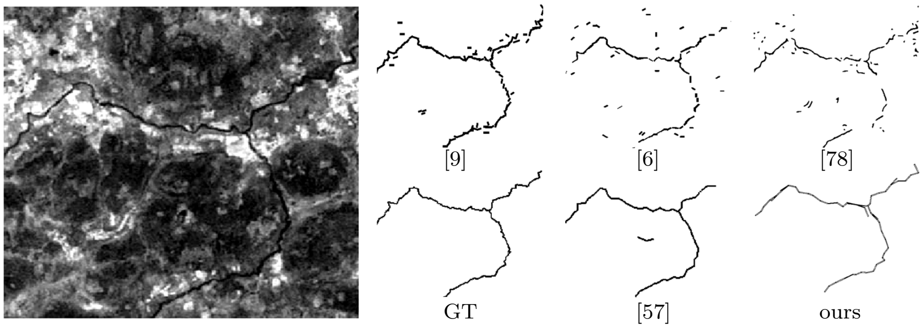

Figure 18: Comparisons with active contour and marked point process approaches.

Table 8: Quantitative comparison (correctness CR [%], completeness CP [%] and quality Q [%]) with active contour [78] and marked point process [57, 6, 9] approaches from the *river* image of Figure 18.

	Lacoste [57]	Rochery [78]	Lafarge [6]	Verdie [9]	Ours
CR	65.1	53.8	47.4	42.9	**66.6**
CP	70.0	49.9	55.0	**75.0**	61.4
Q	**48.2**	40.0	34.2	37.5	47.0

6. Conclusions

In this paper, we presented a stochastic approach based on marked point processes for the automatic extraction of networks in raster data. In our approach, the network is modeled as an undirected, acyclic graph which is iteratively constructed during the optimization process. The approach is evaluated on synthetic data and on two DTM derived from airborne lidar data. Our experiments show that for all data sets, the most relevant tidal channels and rivers are detected, apart from some smaller inlets. In most cases, we end up in one single tree or only a small number of trees. The quality is $> 60\%$ in all scenes. The geometric accuracy is < 1 pixel for the synthetic and the Austrian test site and < 3 pixel for the Wadden Sea data. We also developed a hierarchical detection strategy and searched for channels in different resolutions of the input data. In this way, the parameter space can be significantly reduced. The integration of 3D information in the model improves the correctness rates in all test sites. Our method outperforms a standard flow routing algorithm and achieves significantly higher accuracy results, especially in nearly horizontal terrain. We also proof the transferability of our approach to different types of networks in image data. Here, main parts of the network are detected by our approach. The results are competitive in accuracy in comparison to further approaches based on point processes in the literature.

In the future, we intend to determine parameter by training in order to be less sensitive to the experimental setting. We also plan to investigate on faster optimization schemes by parallelizing the sampling mechanism as proposed by [9] in case of Markovian energies, or by discretizing the energy formulation into a binary variable optimization problem [79]. Such extensions require, however, an entire remodeling of our energy formulation.

Acknowledgement

We would like to thank the State Department for Waterway, Coastal and Nature Conservation (NLWKN) of Lower Saxony for providing the Wadden Sea data and the State Office of Surveying and Geoinformation of Vorarlberg, Austria (http://www.vorarlberg.at/lvg), for providing the lidar and the reference data of Vorarlberg.

Appendix

In order to define the Jacobian determinant of the transition function in equation (4) from the initial configuration X to the new configuration X', the auxiliary variables h and h' are introduced. For instance, for the birth event in Figure 3 (top) where a new node (x, y) and the width w for the new edge are added to the configuration, three auxiliary variables are introduced. The further parameters of the initial configuration X remain. Subsequently, the transition can be written by

$$\left\{\begin{matrix} X \\ h_1 \\ h_2 \\ h_3 \end{matrix}\right\} \rightarrow \left\{\begin{matrix} X \\ x \\ y \\ w \end{matrix}\right\}. \tag{24}$$

Then, the Jacobian determinant for the birth event is given by

$$\det\left(\frac{\partial(X')}{\partial(X,h)}\right) = \det\left(\begin{array}{c|ccc} I & & & \\ \hline & 1 & & \\ & & 1 & \\ & & & 1 \end{array}\right) = 1. \tag{25}$$

For the death event in Figure 3, the transition can be described by

$$\left\{\begin{matrix} X' \\ x \\ y \\ w \end{matrix}\right\} \rightarrow \left\{\begin{matrix} X' \\ h'_1 \\ h'_2 \\ h'_3 \end{matrix}\right\}. \tag{26}$$

The Jacobian determinant is similar to equation 25.

Analogously, the event of *modification* and *split-and-merge* can be described. In both cases the absolute value of the Jacobian determinant is 1.

References

[1] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741.

[2] S. Kumar, M. Hebert, Discriminative Random Fields: A discriminative rramework for contextual interaction in classification, Proceedings of IEEE International Conference on Computer Vision 2 (2003) 1150–1157.

[3] S. Z. Li, Markov random field modeling in computer vision, Springer, Tokyo, 1995.

[4] C. Benedek, M. Martorella, Moving target analysis in ISAR image sequences with a multiframe marked point process model, IEEE Transactions on Geoscience and Remote Sensing 52 (4) (2014) 2234–2246.

[5] D. Chai, W. Förstner, F. Lafarge, Recovering line-networks in images by junction-point processes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, USA, 2013, pp. 1894–1901.

[6] F. Lafarge, G. Gimel'Farb, X. Descombes, Geometric feature extraction by a multimarked point process, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1597–1609.

[7] O. Tournaire, M. Brédif, D. Boldo, M. Durupt, An efficient stochastic approach for building footprint extraction from digital elevation models, ISPRS Journal of Photogrammetry and Remote Sensing 65 (4) (2010) 317–327.

[8] M. Ortner, X. Descombes, J. Zerubia, Building outline extraction from digital elevation models using marked point processes, International Journal of Computer Vision 72 (2) (2007) 107–132.

[9] Y. Verdié, F. Lafarge, Detecting parametric objects in large scenes by Monte Carlo sampling, International Journal of Computer Vision 106 (1) (2014) 57–75.

[10] H. Huang, C. Brenner, M. Sester, A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data, ISPRS Journal of Photogrammetry and Remote Sensing 79 (2013) (2013) 29–43.

[11] X. Descombes, J. Zerubia, Marked point process in image analysis, IEEE Signal Processing Magazine 19 (5) (2002) 77–84.

[12] W. Ge, R. T. Collins, Marked point processes for crowd counting, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 2913–2920.

[13] N. Ripperda, C. Brenner, Reconstruction of façade structures using a formal grammar and RjMCMC, in: Proceedings of the 28th DAGM Symposium, Berlin, Germany, Springer, Berlin Heidelberg, 2006, pp. 750–759.

[14] S. Gruber, S. Peckham, Land-surface parameters and objects in hydrology, in: T. Hengl, H. I. Reuter (Eds.), Developments in Soil Science, Vol. 33, Elsevier, 2009, Ch. 7, pp. 171–194.

[15] J. P. Wilson, G. Aggett, D. Yongxin, C. S. Lam, Water in the landscape: a review of contemporary flow routing algorithms, in: Q. Zhou, B. Lees, G.-A. Tang (Eds.), Advances in Digital Terrain Analysis, Springer, Berlin, 2008, pp. 213–236.

[16] J. F. O'Callaghan, D. M. Mark, The extraction of drainage networks from digital elevation data, Computer Vision, Graphics, and Image Processing 28 (1984) 323–344.

[17] J. Fairfield, P. Leymarie, Drainage network from grid digital elevation models, Water Resources Research 27 (5) (1991) 709–717.

[18] N. Lea, An aspect-driven kinematic routing algorithm, in: A. J. Parsons, A. D. Abrahams (Eds.), Overland flow: Hydraulics and erosion mechanics, Chapman & Hall/CRC, New York, 1992, Ch. 16, pp. 147–175.

[19] P. Quinn, K. Beven, P. Chevallier, The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, Hydrological Processes 5 (1) (1991) 59–79.

[20] T. G. Freeman, Calculating catchment area with divergent flow based on a regular grid, Computers and Geosciences 17 (3) (1991) 413–422.

[21] P. Holmgren, Multiple flow direction algorithms for runoff modelling in grid based elevation models: An empirical evaluation, Hydrological Processes 8 (4) (1994) 327–334.

[22] B. Lohani, D. C. Mason, Application of airborne scanning laser altimetry to the study of tidal channel geomorphology, ISPRS Journal of Photogrammetry and Remote Sensing 56 (2) (2001) 100–120.

[23] S. Fagherazzi, A. Bortoluzzi, W. E. Dietrich, A. Adami, S. Lanzoni, M. Marani, A. Rinaldo, Tidal networks - Automatic network extraction and preliminary scaling features from digital terrain maps, Water Resources Research 35 (12) (1999) 3891–3904.

[24] D. C. Mason, T. R. Scott, H. J. Wang, Extraction of tidal channel networks from airborne scanning laser altimetry, ISPRS Journal of Photogrammetry and Remote Sensing 61 (2) (2006) 67–83.

[25] B. Lohani, D. C. Mason, T. R. Scott, B. Sreenivas, Extraction of tidal channel networks from aerial photographs alone and combined with laser altimetry, International Journal of Remote Sensing 27 (1) (2006) 5–25.

[26] H. Mayer, S. Hinz, U. Bacher, E. Baltsavias, A test of automatic road extraction approaches, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (3) (2006) 209–214.

[27] C. Kirbas, F. Quek, A review of vessel extraction techniques and algorithms, Computing Surveys 36 (2) (2004) 81–121.

[28] D. Lesage, E. D. Angelini, I. Bloch, G. Funka-Lea, A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes, Medical Image Analysis 13 (6) (2009) 819–845.

[29] G. Vosselman, J. de Knecht, Road tracing by profile matching and Kalman filtering, in: A. Gruen, O. Kuebler, P. Agouris (Eds.), Automatic extraction of man-made objects from aerial and space images, Birkhäuser, Basel, Boston, Berlin, 1995, pp. 265–276.

[30] S. Movaghati, A. Moghaddamjoo, S. Member, A. Tavakoli, Road extraction from satellite images using particle filtering and extended Kalman filtering, IEEE Transactions on Geoscience and Remote Sensing 48 (7) (2010) 2807–2817.

[31] D. Geman, B. Jedynak, An active testing model for tracking roads in satellite images, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1) (1996) 1–14.

[32] J. Hu, A. Razdan, J. C. Femiani, M. Cui, P. Wonka, Road network extraction and intersection detection from aerial images by tracking road footprints, IEEE Transactions on Geoscience and Remote Sensing 45 (12) (2007) 4144–4157.

[33] P. Chothani, V. Mehta, A. Stepanyants, Automated tracing of neurites from light microscopy stacks of images, Neuroinformatics 9 (2-3) (2011) 263–278.

[34] E. Bas, D. Erdogmus, Principal Curves as Skeletons of Tubular Objects, Neuroinformatics 9 (2) (2011) 181–191.

[35] A. Grote, C. Heipke, F. Rottensteiner, Road network extraction in suburban areas, Photogrammetric Record 27 (137) (2012) 8–28.

[36] P. Gamba, F. Dell'Acqua, G. Lisini, Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping,

and simple topological concepts, IEEE Geoscience and Remote Sensing Letters 3 (3) (2006) 387–391.

[37] C. Poullis, S. You, Delineation and geometric modeling of road networks, ISPRS Journal of Photogrammetry and Remote Sensing 65 (2) (2010) 165–181.

[38] J. B. Mena, J. A. Malpica, An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery, Pattern Recognition Letters 26 (9) (2005) 1201–1220.

[39] V. Mnih, G. E. Hinton, Learning to detect roads in high-resolution aerial images, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision - ECCV 2010, Vol. LNCS 6316, Springer, Berlin, Heidelberg, 2010, pp. 210–223.

[40] D. Marín, A. Aquino, M. E. Gegúndez-arias, J. M. Bravo, A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features, IEEE Transactions on Medical Imaging 30 (1) (2011) 146–158.

[41] C. Zhang, Towards an operational system for automated updating of road databases by integration of imagery and geodata, ISPRS Journal of Photogrammetry and Remote Sensing 58 (3-4) (2004) 166–186.

[42] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988) 321–331.

[43] A. Gruen, H. Li, Road extraction from aerial and satellite images by dynamic programming, ISPRS Journal of Photogrammetry and Remote Sensing 50 (4) (1995) 11–20.

[44] L. Bentabet, S. Jodouin, D. Ziou, J. Vaillancourt, Road vectors update using SAR imagery: A snake-based method, IEEE Transactions on Geoscience and Remote Sensing 41 (8) (2003) 1785–1803.

[45] Y. Wang, A. Narayanaswamy, B. Roysam, Novel 4-D open-curve active contour and curve completion approach for automated tree structure extraction, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1105–1112.

[46] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, H. Ebner, Automatic road extraction based on multi-scale, grouping, and context, Photogrammetric Engineering & Remote Sensing 65 (7) (1999) 777–785.

[47] M. Butenuth, C. Heipke, Network snakes: Graph-based object delineation with active contour models, Machine Vision and Applications 23 (1) (2012) 91–109.

[48] M. A. Fischler, J. A. Tenenbaum, H. C. Wolf, Detection of roads and linear structure in low resolution aerial imagery using a multi-source knowledge integration technique, Computer Graphics and Image Processing 15 (3) (1981) 201–223.

[49] M. Gerke, M. Butenuth, C. Heipke, F. Willrich, Graph-supported verification of road databases, ISPRS Journal of Photogrammetry and Remote Sensing 58 (3-4) (2004) 152–165.

[50] S. Hinz, A. Baumgartner, Automatic extraction of urban road networks from multi-view aerial imagery, ISPRS Journal of Photogrammetry and Remote Sensing 58 (1-2) (2003) 83–98.

[51] E. Türetken, F. Benmansour, P. Fua, Automated reconstruction of tree structures using path classifiers and mixed integer programming, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 566–573.

[52] F. Tupin, H. Maître, J.-F. Mangin, J.-M. Nicolas, E. Pechersky, Detection of linear features in SAR images: Application to road network extraction, IEEE Transactions on Geoscience and Remote Sensing 36 (2) (1998) 434–453.

[53] A. Katartzis, H. Sahli, V. Pizurica, J. Cornelis, A model-based approach to the automatic extraction of linear features from airborne images, IEEE Transactions on Geoscience and Remote Sensing 39 (9) (2001) 2073–2079.

[54] J. Montoya-Zegarra, J. Wegner, L. Ladicky, K. Schindler, Mind the gap: modeling local and global context in (road) networks, German Conference on Pattern Recognition (GCPR).

[55] J. D. Wegner, J. A. Montoya-Zegarra, K. Schindler, Road networks as collections of minimum cost paths, ISPRS Journal of Photogrammetry and Remote Sensing 108 (2015) (2015) 128–137.

[56] R. Stoica, X. Descombes, J. Zerubia, A Gibbs point process for road extraction from remotely sensed images, International Journal of Computer Vision 57 (2) (2004) 121–136.

[57] C. Lacoste, X. Descombes, J. Zerubia, Point processes for unsupervised line network extraction in remote sensing, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1568–1579.

[58] R. S. Stoica, V. J. Martínez, E. Saar, A three-dimensional object point process for detection of cosmic filaments, Journal of the Royal Statistical Society. Series C: Applied Statistics 56 (4) (2007) 459–477.

[59] B. W. Kreher, Detektion von Hirnnervenfasern auf der Basis von diffusionsgewichteten Magnetresonanzdaten, Dissertation der Fakultät für Angewandte Wissenschaften der Albert-Ludwigs-Universität Freiburg im Breisgau.

[60] A. Schmidt, C. Kruse, F. Rottensteiner, U. Soergel, C. Heipke, Network detection in raster data using marked point processes, ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B3 (2016) 701–708.

[61] A. Schmidt, F. Rottensteiner, U. Soergel, C. Heipke, A graph based model for the detection of tidal channels using marked point process, The In-

ternational Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-3/W3 (2015) 115–121.

[62] J. Møller, R. Waagepetersen, Statistical inference and simulation for spatial point processes, Chapman & Hall/CRC, Boca Raton, 2004.

[63] D. Daley, D. Vere-Jones, An introduction to the theory of point processes: Volume I: Elementary theory and methods, 2nd Edition, Springer, New York, Berlin, Heidelberg, 2003.

[64] C. Mallet, F. Lafarge, M. Roux, U. Sörgel, F. Bretar, C. Heipke, A marked point process for modeling lidar waveforms, IEEE Transactions on Image Processing 19 (12) (2010) 3204–3221.

[65] N. Metropolis, A. W. Rosenbluth, M. Rosenbluth, A. Teller, Equation of state calculations by fast computing machines, Journal of Chemical Physics 21 (6) (1953) 1087–1092.

[66] W. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.

[67] P. J. Green, Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (4) (1995) 711–732.

[68] C. Andrieu, N. de Freitags, A. Doucet, M. I. Jordan, An introduction to MCMC for machine learning, Machine Learning 50 (1-2) (2003) 5–43.

[69] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science, New Series 220 (4598) (1983) 671–680.

[70] P. Salamon, P. Sibani, R. Frost, Facts, conjectures, and improvements for simulated annealing, Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[71] J. Bondy, U. Murty, Graph Theory, in: Graduate texts in mathematics, Springer, New York, London, 2008.

[72] R. Strzodka, A. Telea, Generalized distance transforms and skeletons in graphics hardware, in: Proceedings of EG/IEEE TCVG Symposium on Visualization 2004, 2004, pp. 221–230.

[73] N. Howe, MATLAB implementation of contour-pruned skeletonization (2016).
URL `http://www.cs.smith.edu/{~}nhowe/research/code/`

[74] Y. Verdié, F. Lafarge, Efficient Monte Carlo sampler for detecting parametric objects in large scenes, Computer Vision - ECCV 2012 LNCS 7574 (2012) 539–552.

[75] C. Wiedemann, C. Heipke, H. Mayer, O. Jamet, Empirical evaluation of automatically extracted road axes, in: Empirical Evaluation Techniques in Computer Vision, 1998, pp. 172–187.

[76] J. Jasiewicz, M. Metz, A new GRASS GIS toolkit for Hortonian analysis of drainage networks, Computers and Geosciences 37 (8) (2011) 1162–1173.

[77] M. Vetter, G. Mandlburger, Modification of high resolution airborne laser scanning DTMs for drainage network delineation, Photogrammetrie, Fernerkundung, Geoinformation 1 (2014) (2014) 41–54.

[78] M. Rochery, I. H. Jermyn, J. Zerubia, Higher order active contours, International Journal of Computer Vision 69 (1) (2006) 27–42.

[79] T. T. Pham, S. H. Rezatofighi, I. Reid, T.-j. Chin, Efficient point process inference for large-scale object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2837–2845.