Inference of Curvilinear Structure based on Learning a Ranking Function and Graph Theory

Seong-Gyun Jeong, Yuliya Tarabalka, Nicolas Nisse, Josiane Zerubia

▶ **To cite this version:**

Seong-Gyun Jeong, Yuliya Tarabalka, Nicolas Nisse, Josiane Zerubia. Inference of Curvilinear Structure based on Learning a Ranking Function and Graph Theory. [Research Report] RR-8789, Inria Sophia Antipolis. 2015. hal-01214932

HAL Id: hal-01214932

https://inria.hal.science/hal-01214932

Submitted on 15 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

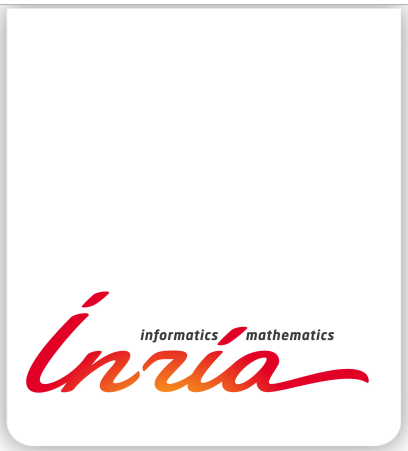
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inference of Curvilinear Structure based on Learning a Ranking Function and Graph Theory

Seong-Gyun Jeong, Yuliya Tarabalka, Nicolas Nisse, Josiane Zerubia

RESEARCH REPORT
N° 8789
October 2015
Project-Teams AYIN,COATI

ISSN 0249-6399 ISRN INRIA/RR--8789--FR+ENG



Inference of Curvilinear Structure based on Learning a Ranking Function and Graph Theory

Seong-Gyun Jeong*, Yuliya Tarabalka*, Nicolas Nisse*†, Josiane Zerubia*

Project-Teams AYIN,COATI

Research Report n° 8789 — October 2015 — 25 pages

Abstract: To detect curvilinear structures in natural images, we propose a novel ranking learning system and an abstract curvilinear shape inference algorithm based on graph theory. We analyze the curvilinear structures as a set of small line segments. In this work, the rankings of the line segments are exploited to systematize the topological feature of the curvilinear structures. Structured Support Vector Machine is employed to learn the ranking function that predicts the correspondence of the given line segments and the latent curvilinear structures. We first extract curvilinear features using morphological profiles and steerable filtering responses. Also, we propose an orientation-aware feature descriptor and a feature grouping operator to improve the structural integrity during the learning process. To infer the curvilinear structure, we build a graph based on the output rankings of the line segments. We progressively reconstruct the curvilinear structure by looking for paths between remote vertices in the graph. Experimental results show that the proposed algorithm faithfully detects the curvilinear structures within various datasets.

Key-words: Curvilinear structure extraction, Inference of structured data, Ranking learning, Graphical model, Shape simplification

* Inria, France
† Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France

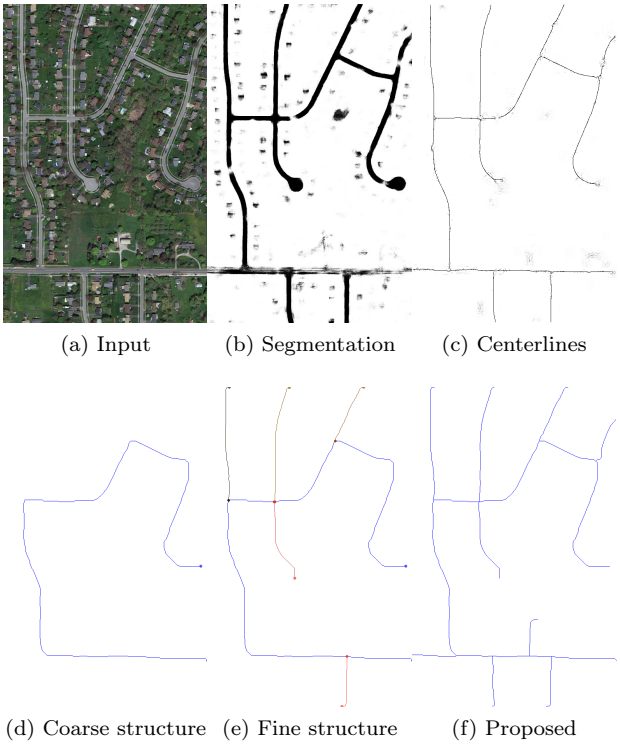
RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Inférence de structures curvilinéaires basée sur l'apprentissage d'une fonction de classement et la théorie des graphes

Résumé : Pour détecter des structures curvilinéaires dans les images naturelles, nous proposons un nouveau système d'apprentissage et un algorithme d'inférence de formes curvilinéaires abstraites, basé sur la théorie des graphes. Nous analysons les structures curvilinéaires comme un ensemble de segments. Dans ce travail, les poids de ces segments sont utilisés pour en déduire les caractéristiques topologiques des structures curvilinéaires. Structured Support Vector Machine est utilisée pour apprendre la fonction de poids qui prédit la correspondance de l'ensemble de segments donné avec les structures curvilináires sous-jacentes. Nous extrayons d'abord les propriétés curvilinéaires en utilisant des profils morphologiques. Pour déduire la structure curvilinéaire, nous construisons un graphe basé sur les poids calculés des segments. Nous reconstruisons progressivement la structure en calculant des chemins entre paires de sommets distants dans le graphe. Les résultats expérimentaux montrent que l'algorithme proposé détecte correctement les structures curvilinéaires dans le cas de données très variées.

Mots-clés : Extraction de structure curvilinéaire, Inférence de structure de données, apprentissage, modèle de graphes, simplification de forme



(a) Input (b) Segmentation (c) Centerlines

(d) Coarse structure (e) Fine structure (f) Proposed

Figure 1: Comparing with the segmentation ((b), [BRLF13]) and the centerline detection ((c), [SLF14]) methods, the proposed algorithm is able to organize the topological importance of the curvilinear structure. We represent different levels of detail in the latent curvilinear structure using the minimum number of pixels.

1 Introduction

Many computer vision algorithms have been proposed to analyze underlying curvilinear structures (also called line networks) of the image data. Although human can intuitively perceive the curvilinear structures, most of the previous works on the curvilinear structure modeling are developed for specific applications. For instance, [FNVV98] and [SAN+04] developed blood vessel detection algorithms to aid diagnosis of vascular diseases. Neuroimage scientists have attempted to analyze shape of the curvilinear objects in microscopic images ([BBC+11]). In remote sensing, road network extraction algorithms have been explored in order to interpret the geographical information ([HRF+07, LDZ05, VCB+10]). Furthermore, [CGMN10] formulated an image segmentation problem following the Markov Random Field (MRF) property to detect irregular shape of the road cracks.

In this paper we have an ambitious goal to automatically detect different types of curvilinear structure which is latent in natural images. It is a challenging task because the geometry of the line networks varies depending on the application. Also, a curvilinear structure could be hidden under the background textures, so that the information from an individual pixel often fails to interpret the topology. More specifically, curvilinear features corresponding to high gradient magnitudes are obtained via convolution filters ([FA91, Per95, JU04]); however the filtering responses are insufficient to discern linear structure from undesirable high-frequency

components due to the lack of shape interpretation. On the other hand, graphical models such as [GTFF10, TBA$^+$13, JTZ15] define spatial constraints in a local configuration to take the structural information into account. However, the models become over constrained to describe complex shaped line networks. Recently, machine learning algorithms have been involved to detect curvilinear structure. [BRLF13] applied a boosting algorithm to obtain an optimal set of convolution filter banks. [SLF14] developed a regression model to detect centerlines by learning the scale (width) of the tubular structures with the non-maximum suppression technique.

Assume that the entire curvilinear structure can be decomposed into many straight line segments. We learn a ranking function that predicts the shape similarity between individual line segments and the latent curvilinear structure. We propose an orientation-aware curvilinear feature descriptor for the learning system, which consists of the morphological profiles and steerable filtering responses. In this work Structured Support Vector Machine (SSVM) is employed to obtain weight coefficients for the proposed feature descriptor using the training dataset. To infer the structural information, we build an undirected and weighted graph based on the output rankings of the line segments. We then reconstruct the coarse curvilinear structure by exploring paths of vertices maximizing their distance in the graph. Unlike the previous approaches such as segmentation of the curvilinear structure ([BRLF13]) and centerline detection ([SLF14]), the proposed algorithm can provide the topological importance level of the curvilinear structure (see Fig. 1). Our experiments demonstrate that the proposed algorithm localizes the curvilinear structure with a high accuracy comparing to the state-of-the-art methods.

1.1 Related work

Curvilinear structure extraction: In early vision, researchers design convolution filters to separate curvilinear structure from the background texture ([FA91, Per95, JU04]). The main idea behind such filter design is to create simple line shape templates in order to extract features showing high gradient magnitudes with a locally consistent orientation. However, these image filtering responses are unable to discern linear structures from undesirable high-frequency components, *e.g.,*noise and edges. Enhancement filtering (EF) algorithm proposed by [FNVV98] analyzes the eigenvalues of the Hessian matrix of the image to evaluate local tubularity score. Optimally Oriented Flux (OOF) measures the amount of gradient flow at the boundaries to find the continuous linear structure ([LC08]). Morphological operator also highlights the curvilinear structure by collecting pixels according to the recursive structural similarity ([TA07]).

On the other hand, the graphical models exploit image-based evidence with geometric shape constraints to improve the detection performances. For instance, tree structure can be used to reinterpret complex line networks. [GTFF10] looked for a path between singular points corresponding to intersections of the latent curvilinear structure. [TBA$^+$13] formulated a large linear programming problem to constrain the diverse cases of the local interaction on the line networks.

Furthermore, curves can be approximated by multiple straight line segments. Stochastic models specify the distribution of line objects with pairwise interaction terms ([LDZ05, JTZ15]). Reversible Jump Markov chain Monte Carlo sampler proposed by [Gre95] has been involved to optimize the probability density. However, due to the sensitivity of the parameter setup, the stochastic models are less practical for a large amount of varied datasets.

In recent years, machine learning algorithms have been favoured for designing optimal filter banks to extract curvilinear features ([RL12, BRLF13]). While a threshold value should be set to reconstruct the centerline of the curvilinear structure from the filtering scores, it is difficult to find the value which would be suitable for different types of applications. To address this issue, [SLF14] proposed a regression model of distance transform to predict the scale of the curvilinear structure and localize centerlines.

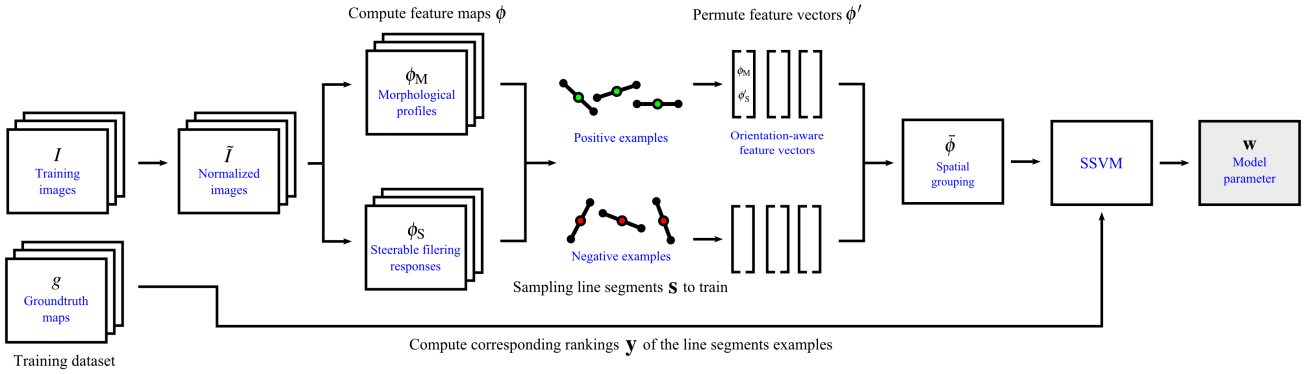


Figure 2: Overview of the curvilinear feature descriptor generation (Sec. 3) and ranking learning system (Sec. 4)

Learning structured information: We want to extract the structured information of the line networks that appear within homogeneous background textures. Since the information embedded in a single pixel is limited to infer the latent structure, Markov Random Fields ([GG84, Bes86]) or Conditional Random Fields (CRF, [LMP01, KH06]) based models have been developed to enforce the label consistency on the pre-organized output space. *e.g.,* neighborhood pixels are assumed to have the same label with the probability when minimizing the cost function. However, the topology of the curvilinear structure composed of line objects is too intricate to be applicable with such design approaches. It is also impossible to specify the topology of the complex line networks with a few parameters. Instead, we employ a machine learning framework, SSVM proposed by [TJHA05], to obtain undiscovered distribution which contains the structured relation of input-and-output pairs. The SSVM is often comparable with the graphical models which contain the pairwise structure. Therefore, the SSVM framework is favoured to solve computer vision problems, such as object segmentation by [BYVG11, LLSF12, KYNK14], pose estimation algorithm by [YR11], and multi-class classification by [MBZT12]. In this work, we use the SSVM framework to train a ranking function that predicts the compatibility with tokenized line segments and the latent curvilinear structures. Finally, the structured output rankings of the line segments are used to estimate the local orientation and to reconstruct the latent curvilinear structures as an ensemble of high ranked line segments.

1.2 Contributions

To apply a learning system, we need an accurately annotated ground truth for training datasets, where the ground truth is manually labeled to classify pixels on the curvilinear structure by human experts. It is difficult to secure enough quantity of training datasets in that the manual segmentation of curvilinear structure is time consuming work. To address this problem, we evaluate a line segment using the shape similarity measure whether it belongs to the latent curvilinear structure. We minimize corruptions of the learning process due to the erroneous annotations.

We also study the graph theory for shape simplification of the complex curvilinear structure. Although the centerline is able to encode the scale of the curvilinear structure, it is insufficient to interpret the topological level of the curvilinear structure. The structured output rankings of the line segment are used to draw the latent curvilinear structure in the order of topological importance. Thus, we create a weighted graph to reconstruct the curvilinear structure with

consideration for such topological order. We search for the longest path which minimizes the cumulative rankings of the line segments on the path. Recall the Fig. 1 to compare with the result of curvilinear structure segmentation ([BRLF13]), centerlines detection ([SLF14]), and proposed method.

The main contributions of this work are summarized as follows:

- We propose an orientation-aware curvilinear feature descriptor for the ranking learning system;
- We learn a ranking function to predict the correspondence between line segments and curvilinear structure;
- We reconstruct the curvilinear structure with topological order based on graph theory; and
- We test and validate the proposed algorithm on numerous datasets containing micro and macro scales of curvilinear structure in the nature.

The rest of the paper is organized as follows: Sec. 2 provides the outline of our method. Sec. 3 proposes an orientation aware curvilinear feature descriptor. Sec. 4 explains how we learn the ranking function. Sec. 5 develops a graphical model for inference of curvlinear structure. Sec. 6 shows experimental results on different types of datasets. Finally, Sec. 7 concludes this work.

2 Overview

In this section, we define notations to represent curvilinear structures, and provide an overview of the proposed algorithm. Assume that an image I contains a curvilinear structure. We denote the latent curvilinear structure $g : I \mapsto \{0, 1\}$ for any pixel $\mathbf{x}$:

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is on the curvilinear structure,} \\ 1 & \text{backgrounds.} \end{cases} \tag{1}$$

This function is also employed to create a ground truth map, which is manually labeled, for the machine learning system and the performance evaluations.

Instead of classifying each pixel whether it belongs to the curvilinear structure, we gather line segments that agree with the latent curvilinear structure of the given image data. Hence, the proposed algorithm aims to find a set of line segments $\mathbf{s} \subseteq \mathcal{S}$ highly corresponding to the curvilinear structure. We define a line segment $s \in \mathcal{S}$ as tuple of the center coordinate $\mathbf{x}$, fixed length ℓ, fixed thickness τ, and quantized orientations θ. That is $s = (\mathbf{x}, \ell, \tau, \theta) \in I \times \mathbb{R}^2 \times A$, where A is set of elements in $[0,\ 2\pi[$.

In this work, to train a ranking function with SSVM, we propose a novel curvilinear feature descriptor $\phi : \mathcal{S} \mapsto \mathbb{R}^N$ that assigns a feature vector $\phi(s) \in \mathbb{R}^N$ to each line segment s, where N denotes the dimension of the feature space. Specifically, the feature descriptor consists of the morphological profiles ϕ_{M} to detect expected length of the structure and the steerable filtering responses ϕ_{S} to accentuate directional properties (Sec. 3.1). To take into account the directional information of each line segment, we propose an orientation-aware feature vector $\phi'(s)$ that permutes the elements in the steerable feature vector of $\phi_{\mathrm{S}}(s)$ according to orientation θ of the given line segment s (Sec. 3.2). Also, we design a spatial neighborhood system for the line segments to improve the consistency of output ranking values (Sec. 3.3). A model parameter $\mathbf{w} \in \mathbb{R}^N$ is obtained via machine learning system (SSVM), which determines the relative importance of elements in the curvilinear feature vectors. The output rankings of line segments $y \in \mathcal{Y}$ are associated to the plausibility for each line segment s to belong to the ground truth map g. For example, a higher ranking is assigned the given line segment if it fits well to the latent curvilinear structure (Sec. 4). Fig. 2 summarizes the process how we learn a model parameter of the proposed curvilinear feature descriptor with the given training dataset.

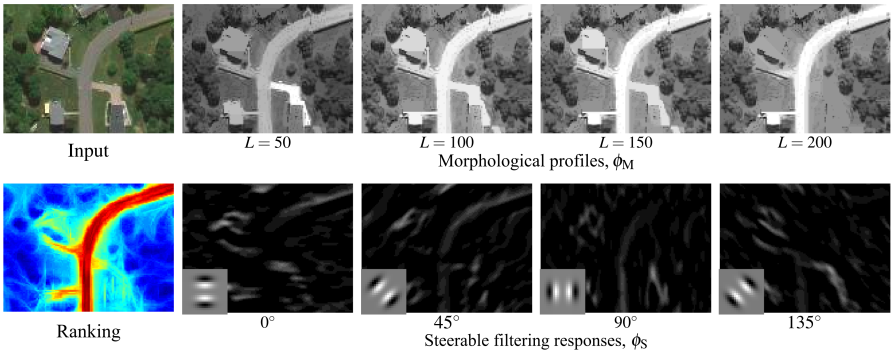


Figure 3: Example of the feature maps of the morphological profile ϕ_M with different length parameters $L \in \{50, 100, 150, 200\}$ and those of the steerable filtering responses ϕ_S with variable orientations $\theta \in \{0°, 45°, 90°, 135°\}$. The model parameters, which define the relative weights of those feature maps, are obtained via the ranking learning system. The higher ranking scores (denoted by red color) correspond to the latent curvilinear structure.

With the rankings of the line segments, we build an undirected and weighted graph $G = (V, E)$ to infer the curvilinear structure with a set of elementary paths on the graph G. Finally, we iteratively reconstruct the curvilinear structure by collecting paths between remote vertices in the graph (Sec. 5).

3 Orientation-Aware Curvilinear Feature

3.1 Feature extraction

This section is devoted to compute the curvilinear feature descriptors that are used for the inputs of the ranking learning system. The evidence of being the curvilinear structure is measured by the homogeneity of the pixel intensities and the directional responses on the path. We extract curvilinear features using the morphological profiles $\phi_M : \mathcal{S} \mapsto \mathbb{R}^4$ and the steerable filter responses $\phi_S : \mathcal{S} \mapsto \mathbb{R}^{24}$. Fig. 3 depicts an example of the curvilinear feature maps.

Before applying the filtering operations, we normalize the training and test images to remove the effects of various illumination factors:

$$\tilde{I} = \frac{1}{1 + e^{-\frac{I-\beta}{\alpha}}}, \tag{2}$$

where $\alpha = \max(I) - \min(I)$ represents the range of intensity values, and $\beta = \mathbb{E}[I]$ is the sample mean of the image.

To obtain the morphological profile, we apply the *path opening* operator ([TA07]) with different length parameters L. The path opening operator brightens pixels belonging to the linear shaped objects while it removes other types of structural elements. As the length parameter of

the path opening operator is increased, we obtain a map which emphasizes longer sequence of pixels in the image.

The steerable filtering responses are associated with the directional characteristics of the given line segment. Let $\mathbf{f}_\theta$ be a steerable filter that accentuates orientation θ. The steerable feature map is computed by the convolution of $\mathbf{f}_\theta$ and normalized image $\bar{I}$. We construct the baseline steerable filtering feature vector by stacking such steerable filtering responses with increasing order of orientation. That is

$$\phi_S(\mathbf{x}) = \left[\left(\mathbf{f}_{\theta_1} * \tilde{I}\right)(\mathbf{x}),\ \left(\mathbf{f}_{\theta_2} * \tilde{I}\right)(\mathbf{x}),\ \cdots, \left(\mathbf{f}_{\theta_k} * \tilde{I}\right)(\mathbf{x})\right]^\intercal, \tag{3}$$

where $\theta_1 < \theta_2 < \cdots < \theta_k$ and $k = |A|$ is the number of quantized orientations.

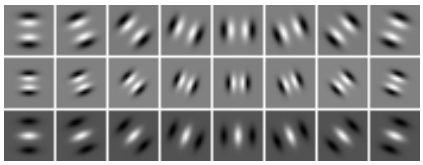


Figure 4: Steerable filter banks $\mathbf{f}_\theta$ to compute curvilinear feature maps

In this work, we use 4 different length parameters to create the morphological profiles and 24 filter banks to build the steerable feature maps. Specifically, the steerable filter banks consist of 3 basis kernels with respect to 8 different orientations. Fig. 4 shows the steerable filter banks which are employed in this work.

Finally, the feature vector of the given line segment $\phi(s)$ is generated by the concatenation of the feature vector of $\phi_{\mathrm{M}}(s)$ and $\phi_{\mathrm{S}}(s)$. We accumulate the values of the feature maps on the line segment feature and rescale those values according to length of the line segment ℓ:

$$\begin{aligned}\phi(s) &= \Big[\phi_{\mathrm{M}}(s),\ \phi_{\mathrm{S}}(s)\Big]^\intercal \\ &= \frac{1}{\ell}\sum_{\mathbf{x}\in s}\Big[\phi_{\mathrm{M}}(\mathbf{x}),\ \phi_{\mathrm{S}}(\mathbf{x})\Big]^\intercal.\end{aligned} \tag{4}$$

3.2 Permutation of feature maps

As the result of ranking learning system, we will obtain a model parameter $\mathbf{w}$ that determines the relative importance of each element in the feature vector. If the training set consists of a particular orientation, the corresponding parameter has a larger value than others. However, such situation can induce an over-fitting. To ease this issue, we also exploit the own directional information of the line segments. For this purpose, we permute the order of elements in the baseline steerable feature vector $\phi_{\mathrm{S}}(s)$ according to the orientation of the given line segment s.

Let $\mathbf{P}_{\theta_i}$ be a permutation matrix that shuffles the order of the feature vector according to the given orientation θ_i:

$$\begin{aligned}\mathbf{P}_{\theta_i} = \Big[\mathbf{e}_i,\ \mathbf{e}_{(i+n)\circ k},\ \mathbf{e}_{(i-n)\circ k},\ \cdots,\ \mathbf{e}_{(i+\lfloor k/2\rfloor)\circ k}\Big]^\intercal, \\ n = 1, 2, \cdots, \lfloor k/2\rfloor - 1,\end{aligned} \tag{5}$$

where $\mathbf{e}_i$ denotes a row vector with 1 in the i-th element and 0 in the others. We define a circular modulation operator $\circ$ which is given by:

$$i \circ k = ((i-1) \bmod k) + 1. \tag{6}$$

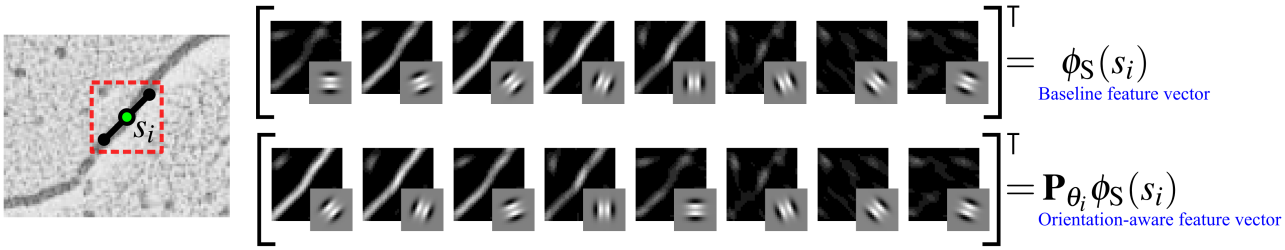


Figure 5: The baseline steerable feature vector $\phi_S(s_i)$ is created by steerable filtering responses $(\mathbf{f}_\theta * I)$ as the increasing order of orientation. To take into account the direction of line segment s_i, we multiply permutation matrix P_{θ_i} to the baseline $\phi_S(s_i)$.

For instance, $\mathbf{P}_{\theta_4}$ for $k = 8$ is defined as

$$\mathbf{P}_{\theta_4} = \begin{bmatrix} \mathbf{e}_4 \\ \mathbf{e}_5 \\ \mathbf{e}_3 \\ \mathbf{e}_6 \\ \mathbf{e}_2 \\ \mathbf{e}_7 \\ \mathbf{e}_1 \\ \mathbf{e}_8 \end{bmatrix} = \begin{bmatrix} 0, & 0, & 0, & 1, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 1, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 1, & 0 \\ 1, & 0, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & 0, & 1 \end{bmatrix}.$$

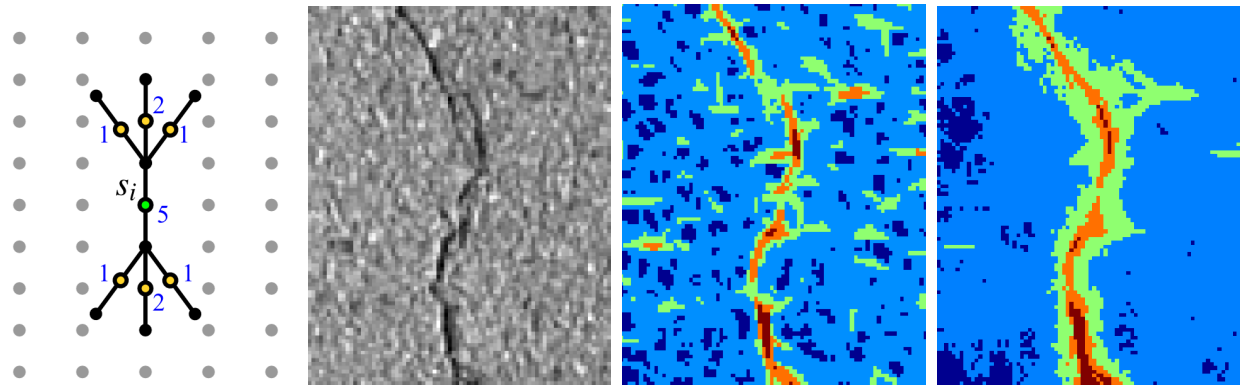
To compute the orientation-aware feature vectors $\phi'(s)$, we apply the permutation matrix $\mathbf{P}_\theta$ to re-order the elements of the baseline feature vector $\phi(s)$. For a better understanding, we give a toy example how to shuffle the baseline feature vector for the given orientation. Suppose that the orientation of line segments s_2 and s_5 are associated with θ_2 and θ_5, respectively. Then, the orientation-aware feature vectors $\phi'(s_2)$ and $\phi'(s_5)$ are given by

$$\phi'(s_2) = \left[\phi_{\mathrm{M}}(s_2),\ \mathbf{P}_{\theta_2}\phi_S(s_2)\right]^\intercal,$$
$$\phi'(s_5) = \left[\phi_{\mathrm{M}}(s_5),\ \mathbf{P}_{\theta_5}\phi_S(s_5)\right]^\intercal.$$

Now the elements of the orientation-aware feature vectors are aligned in the order of dominant directional response for the given line segments. Therefore, the model parameter $\mathbf{w}$ of the learning system properly works to measure compatibility of input line segments and the given image data. Fig. 5 compares the baseline feature vector to the orientation-aware feature vector for the given line segment s_i.

3.3 Spatial grouping of the features

To improve segmentation performance, the piecewise smoothness assumption on the image grid is often used, *e.g.,* 8-connected neighboring pixels have the same label with a high probability. While the thin line structure has the internal similarity, such prior model also encourages background textures to be classified as the same segment so that it produces discontinuities on the curvilinear structure. Instead, we propose a new topology to consider the neighborhood system of line segments. The proposed neighborhood system is designed to realize a smoothly connected curvilinear structure. Thus, to enhance the spatial coherence of the output rankings, the neighboring set consists of shifted line segments to the extendable direction and shifted-and-rotated



(a) (b) (c) (d)

Figure 6: The topology of the spatial feature grouping is depicted in (a), where the corresponding weighting factors $\omega_j, j \in \mathcal{N}_i$ of the neighboring line segments are denoted in blue colored numbers. The spatial grouping of the curvilinear features is able to improve the local consistency of the output rankings. Given image (b) is corrupted by irregular background textures. We visualize the output rankings: (c) is obtained without spatial feature grouping, whereas (d) shows the output rankings with the proposed spatial feature grouping.

line segments with a small curvature. (see Fig. 6 (a)). We define the spatial grouping operator $\bar{\phi}$ as:

$$\bar{\phi}(s_i) = \frac{\sum_{j \in \mathcal{N}_i} \omega_j \phi'(s_j)}{\sum_{j \in \mathcal{N}_i} \omega_j}, \tag{7}$$

where $\mathcal{N}_i$ denotes the neighboring set of the line segment s_i and ω_j is the corresponding weight factor (Fig. 6 (a) details the values of ω_j used in this work).

Fig. 6 compares the output rankings obtained by individual feature vector and spatially grouped feature vector. When we learn a ranking function without the spatial feature grouping operation, the output rankings are less reliable due to the corrupted background texture. On the other hand, the output rankings with the spatial feature grouping increase the rankings of line segments corresponding to the latent curvilinear structures.

4 Learning

In this section, we briefly review the ranking learning system based on SSVM ([TJHA05]).

Our goal is to learn a function $h : \mathcal{S} \mapsto \mathcal{Y}$ that predicts structured output rankings of the line segments. For the setup of machine learning system, a training dataset $\mathcal{D} = \{(s_i, y_i)\}_{i=1}^{K}$ is required which consists of the input-and-output pairs. Assume that we have a list of line segments $\mathbf{s} = \{s_1, \ldots, s_K\} \subseteq \mathcal{S}$ and the relevant ranking values $\mathbf{y} = \{y_1, \ldots, y_K\} \subseteq \mathcal{Y},\ y_i \in [0,\ 1]$. The line segments $\mathbf{s}$ for training are independently sampled from the images of the training dataset. To determine the corresponding rankings $\mathbf{y}$ of the input line segments is a tricky part in this work. Recall that the function g that represents a binary map to classify pixels on the latent curvilinear structure. We use the binary map as the ground truth of the curvilinear structure. However, the ground truth g cannot encode the shape information of the line segments (length, orientation, and thickness). In other words, line segments proposed at the $\mathbf{x}$ could be irrelevant with the true curvilinear structure due to the shape information although a pixel is on the curvilinear

	0°	22.5°	45°	67.5°	90°	112.5°	135°	157.5°
F_1	0.56	**0.78**	0.51	0.30	0.27	0.22	0.19	0.26
y	0.44	**0.22**	0.49	0.70	0.73	0.78	0.81	0.74

Figure 7: Example of the evaluation of the given line segment s_i and corresponding ground truth patch g_i. Blue and red pixels show superfluous and missed areas, respectively, whereas green pixels paint the agreed area of line segment and ground truth.

structure by $g(\mathbf{x}) = 0$. We evaluate the shape dissimilarity between the given line segments s_i and the corresponding patch of the ground truth map g_i:

$$y_i = 1 - \texttt{score}(s_i, g_i). \tag{8}$$

Many options are available to compute the function `score`. In this work, we employ F_1 measure which is defined as

$$\texttt{score}(s_i, g_i) = F_1(s_i, g_i) = \frac{2P_iR_i}{P_i + R_i}, \tag{9}$$

where P_i and R_i denote *precision* and *recall*, respectively, which are associated with the line segment s_i. Fig. 7 shows an example that computes the shape similarity score of a line segment and the corresponding ground truth patch. If s_i lies on the true curvilinear structure g_i, the corresponding ranking value y_i is closer to 0; whereas s_i is on a non-curvilinear structure, y_i is set to 1.

We use a joint feature map of input-and-output pairs $\Psi(\mathbf{s}, \mathbf{y}) : \mathcal{S} \times \mathcal{Y} \mapsto \mathbb{R}^N$ to obtain structured output rankings. A scoring function $F : \mathcal{S} \times \mathcal{Y} \mapsto \mathbb{R}$ is defined as the inner product of the joint feature vector and a model parameter $\mathbf{w}$. For a given line segment, a prediction $\hat{\mathbf{y}} = h(\mathbf{s}; \mathbf{w})$ is performed by finding rankings that maximize the scoring function:

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} F(\mathbf{s}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\intercal \Psi(\mathbf{s}, \mathbf{y}). \tag{10}$$

We define the joint feature map $\Psi(\mathbf{s}, \mathbf{y})$ to obtain the decreasing order of rankings among input line segments:

$$\Psi(\mathbf{s}, \mathbf{y}) = \sum_{ij} y_{ij} \left(\bar{\phi}(s_i) - \bar{\phi}(s_j) \right), \tag{11}$$

where the ranking matrix y_{ij} indicates the pairwise order of the objects s_i and s_j in terms of shape dissimilarity measure (8). In other words,

$$y_{ij} = \begin{cases} +1 & \text{if } y_i > y_j, \\ -1 & \text{otherwise.} \end{cases}$$

The parameter of the learning system $\mathbf{w} \in \mathbb{R}^N$ is obtained by minimizing the following constrained objective function:

$$\min_{\mathbf{w}, \xi \geq 0} \ \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} + \frac{C}{K} \sum_{i=1}^{n} \xi_i \tag{12}$$

$$\text{s.t. } \forall \mathbf{y}', \mathbf{y}' \neq \mathbf{y} : \mathbf{w}^\intercal \Psi(\mathbf{s}, \mathbf{y}) \geq \mathbf{w}^\intercal \Psi(\mathbf{s}, \mathbf{y}') + \Delta(\mathbf{y}, \mathbf{y}') - \xi \tag{13}$$

where the first term of (12) penalizes the complexity of the solution (regularization) and the second term measures the relaxed loss of the system to fit the parameter into training data (regression). C controls the relative importance between two terms. $\Delta(\mathbf{y}, \mathbf{y}')$ is a loss function that counts the number of incorrectly ranked pairs. To solve this optimization problem, we use the cutting plane algorithm proposed in [TJHA05].

We have obtained a ranking function $h(\mathbf{s}; \mathbf{w})$ that arranges the line segments as the order of correspondence with the underlying curvilinear structure. Specifically, the linear combination of the model parameter $\mathbf{w}$ and the proposed feature descriptor $\bar{\phi}(s)$, $\hat{y} = \mathbf{w}^\intercal \bar{\phi}(s)$, is used to evaluate the line segment s. At each pixel grid, we create k line segment candidates, where k denotes the number of orientations. We then estimate the local orientation using the ranking function:

$$\hat{i} = \underset{i \in \{1,\ldots,k\}}{\operatorname{argmax}} \mathbf{w}^\intercal \bar{\phi}(s(\theta \leftarrow \theta_i)), \text{ then } \theta_{\max} = \theta_{\hat{i}}, \tag{14}$$

where $s(\theta \leftarrow \theta_i)$ denotes a candidate of the line segment which updates its orientation for θ_i whereas the rest of parameters (center locations, length, and thickness) are fixed. Moreover, the ranking function is employed to indicate a pixel on the curvilinear structures. For this purpose, the output rankings are computed at each pixel grid using the line segment maximizing local orientation responses by (14).

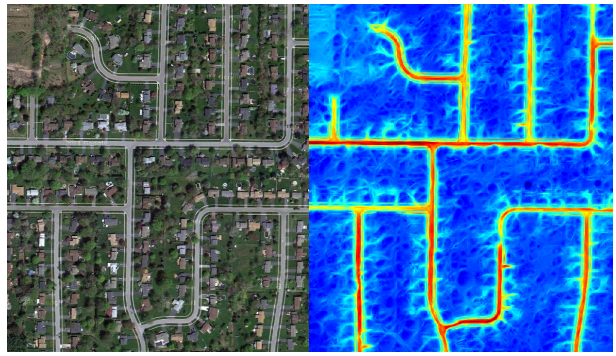
5 Inference

In this section, we aim to develop a graphical model to infer the topology of the curvilinear structure. The main motivation to represent the curvilinear structure as a graph is to use an optimal number of pixels to reconstruct the curvilinear structure without redundancies. Also, we want to see the topological features of the structure in different levels of detail.

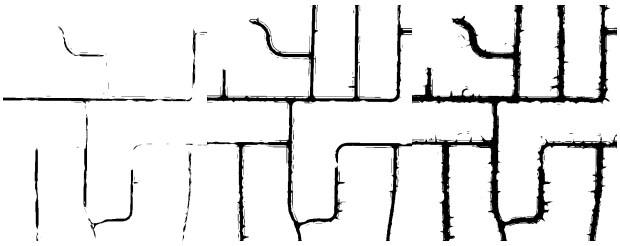
For the sake of convenience, we recall several terminologies related to graph theory before developing the proposed algorithm. We consider the graph $G = (V, E)$ where V is the set of pixels. Each pixel being given its image coordinates, and pixels $\mathbf{u} = (i, j)$ and $\mathbf{v} = (m, n)$ are adjacent if and only if $|i - m| \leq 1$ and $|j - n| \leq 1$. Moreover, we assign a weight for each edge $\{\mathbf{u}, \mathbf{v}\} \in E$ as $w(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \frac{\hat{y}_\mathbf{u} + \hat{y}_\mathbf{v}}{2}$, where $\|\mathbf{u} - \mathbf{v}\|$ denotes the Euclidean distance between pixel coordinates of $\mathbf{u}$ and $\mathbf{v}$. $\hat{y}_\mathbf{u}$ and $\hat{y}_\mathbf{v}$ are the output rankings at $\mathbf{u}$ and $\mathbf{v}$, respectively. A path P in the graph G is a sequence of distinct vertices such that consecutive vertices are adjacent. The length of P is the sum of the weight of its edges, and the distance $dist(\mathbf{u}, \mathbf{v})$ between two vertices $\mathbf{u}$ and $\mathbf{v}$ is the minimum length of a path from $\mathbf{u}$ to $\mathbf{v}$. The eccentricity $ecc(\mathbf{v})$ denotes the maximum distance from the node $\mathbf{v}$ to a node $\mathbf{u} \in V$, *i.e.,* $ecc(\mathbf{v}) = \max_{\mathbf{u} \in V} dist(\mathbf{u}, \mathbf{v})$. The diameter $diam(G)$ of G equals $\max_{\mathbf{v} \in V} ecc(\mathbf{v})$, *i.e.,* it is the maximum distance between two nodes in G.

Intuitively, to infer the topology of the curvilinear structure, we look for long shortest paths in the subgraph G' of G induced by the nodes with high rankings. More precisely, our algorithm computes a diameter of G', *i.e.,*a shortest path P with length $diam(G')$, this path P is added in the curvilinear structure inference, then the path P is contracted into a single node (equivalently, the weight of all edges of P become 0), and the process is repeated while the diameter of the subgraph is larger than pre-defined path length $\hat{\ell}$. The entire procedure is summarized in Algorithm 1.

The first part of the algorithm is to extract the subgraph G' from the whole set of pixels. This subgraph should contain the curvilinear structure and be relatively small for an efficient computation. For this purpose, from the training datasets, we first compute the average proportion ρ of pixels being part of the curvilinear structure. Then, the binary segmentation map is induced



(a) Input (b) Output rankings



(c) 4.16% (d) 8.72% (e) 14.78%

Figure 8: Binary segmentation maps are built depending on statistics of the output rankings.

by $\rho|I|$ pixels according to the output rankings, related to a high ranking. However, due to the noises and the choice of a small ρ, the binary segmentation map may be disconnected so that some parts do not reflect the connected components of the curvilinear structure. Also, the binary segmentation map only encodes the center coordinates of the line segments (see Fig. 8) while the output rankings contain the shape information of the line segments. To overcome these issues, we disjoint the binary segmentation map into line segments, and compute the dissimilarity scores (8) between these line segments and the corresponding patches of the binary segmentation map. We fill up areas corresponding to the line segments with those of dissimilarity scores. Specifically, for overlapping areas between adjacent line segments, we assign the average value of theirs (see Fig. 9). Thus, the subgraph G' is built upon a subset of pixels $V' \subset V$ which corresponds to the dissimilarity score map. We also update the weights of for each edge $\{\mathbf{u}, \mathbf{v}\} \in E$ in the subgraph as $w(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \frac{b_{\mathbf{u}}\hat{y}_{\mathbf{u}} + b_{\mathbf{v}}\hat{y}_{\mathbf{v}}}{2}$, where $b_{\mathbf{u}}$ and $b_{\mathbf{v}}$ denote the dissimilarity scores at $\mathbf{u}$ and $\mathbf{v}$, respectively.

The time-consuming part of the proposed algorithm is the computation of a diameter of G'. Rather than computing all pair distances (which requires a linear number of application of Dijkstra's Algorithm), we use an efficient heuristic algorithm called *2-sweep algorithm* proposed by [CDHP01]. The 2-sweep algorithm randomly picks a vertex $\mathbf{t}$ in G', then performs Dijkstra's algorithm from $\mathbf{t}$ to find a node $\mathbf{u}$ at the maximum distance from $\mathbf{t}$, *i.e.,* $dist(\mathbf{u}, \mathbf{t}) = ecc(\mathbf{t})$. Then, it computes (using Dijkstra's algorithm) a path from $\mathbf{u}$ to a node $\mathbf{v}$ at the maximum distance from $\mathbf{u}$. The length of the second path (from $\mathbf{u}$ to $\mathbf{v}$) is a good estimation of $diam(G')$. Note that this algorithm is able to compute the exact diameter if the graph has tree structure. Topologically speaking, most of latent curvilinear structures are very close to trees, so that the use of the 2-sweep algorithm is well adapted. For a better understanding, we schematically explain

Algorithm 1

Require: $G' = (V', E') \sim$ an undirected and weighted subgraph of the image G;
and $\hat{\ell} \sim$ a minimum length of the curvilinear structure
Ensure: $P \sim$ a set of vertices corresponding to the curvilinear structure
1: $P \leftarrow \emptyset$
2: `longest_path_length` $\leftarrow |V'|$
3: **while do**
4: Compute the longest path Q in G' using 2-sweep algorithm
5: `longest_path_length` $\leftarrow |Q|$
6: **if** `longest_path_length`$< \hat{\ell}$ **then**
7: break
8: $P \leftarrow P \cup Q$
9: $w(\mathbf{u}, \mathbf{v}) \leftarrow 0, \forall \mathbf{u}, \mathbf{v} \in Q$, All edge weights on the path Q become 0

the intermediate steps of the proposed curvilinear structure inference algorithm in Fig. 10.

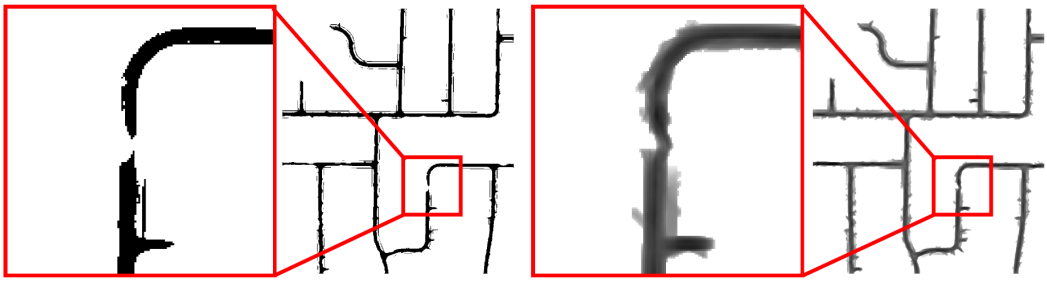
6 Experimental results

In this section, we first discuss the parameters of the proposed algorithm and datasets. We then compare the quantitative and qualitative results of the proposed algorithm and those of competing models proposed by [FNVV98], [LC08], [BRLF13], [SLF14], and [JTZ15].

6.1 Parameters and Datasets

The proposed algorithm requires few parameters to compute the curvilinear feature descriptor ϕ to define physics of the line segment (length ℓ and thickness τ), and to train a ranking function via SSVM. For the morphological profiles ϕ_M, we apply the path opening operator 4 times with different length parameters L. We linearly increase such length parameters by adding κ from the minimum length L^*. Thus, $L \in \{L^* + n\kappa | n = 0, 1, 2, 3\}$. In the case that the curvilinear structure appears darker than surrounding pixels, we invert the input images before applying the morphological operation. This is because the path opening operator assumes pixels on the curvilinear path to be brighter than neighboring pixels. For the steerable features ϕ_S, we perform convolution of the steerable filters with the normalized input images. The steerable filter banks consist of 3 basis kernels with $k = 8$ different orientations, where the size of the filters is 21×21 (see Fig. 4). To build the basis kernels, we superpose the elongated Gaussian kernels using different aperture and smoothing factors. The aperture value determines the space between the Gaussian kernels with respect to the minor axis. We control the aperture size of the Gaussian kernels as $\{6, 3, 1\}$ to take into account the thickness of the curvilinear structure. The smoothing factor of the outer kernels σ_o is set to 2.5 and those of inner kernels σ_i is set to 1.5. The elongation factor λ is fixed as 2.5 for all. Note that the steerable filters are symmetric with respect to the major axis. Also, we assume that the length of line segment should be larger than its thickness, $\ell > \tau$, and these two values depend on the dataset. For SSVM training, we sample 500 positive and negative query line segments, respectively, for every training image. C which controls the relative importance of regularization and regression terms in (12) is set to 0.01 for all datasets. We choose the set of parameters for the SSVM training via 3-fold cross validation which maximizes the average F_1 score of the training set.

We test our curvilinear structure model on the following public datasets:



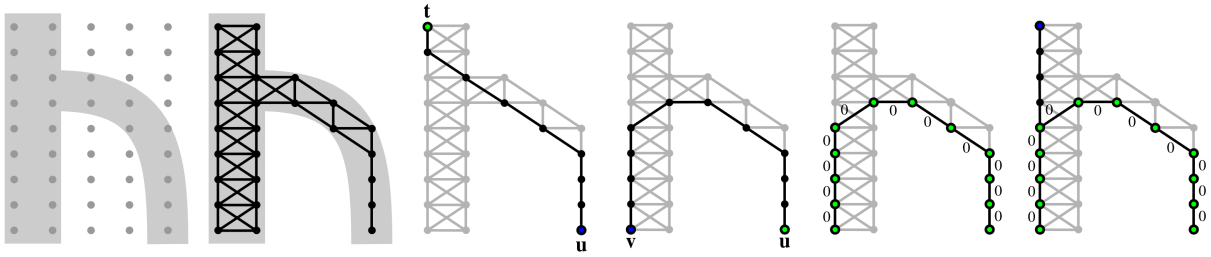
(a) Binary segmentation map (b) Dissimilarity score map

Figure 9: A binary segmentation map is created by thresholding output rankings at a deterministic value, where the value is obtained by the statistics of pixels on the curvilinear structure from the training dataset. While the structured output rankings contain the shape information of the line segments, we cannot exploit such useful information based on the binary segmentation map. The dissimilarity score map is generated by reaggregation of the binary segmentation map. For each pixel on the binary segmentation map, we introduce the shape information which is already embedded in the output rankings. Then, we compute dissimilarity scores of the disassembled line segments and the binary segmentation map. Finally, we fill areas corresponding to the line segments with the dissimilarity scores.

- **DRIVE** [SAN+04]: The dataset consists of 40 retina scan images with manual segmentation by ophthalmologists to evaluate the blood vessel segmentation algorithms. We use 20 images for the training and 20 images for the test, respectively. We build the morphological profiles with $(L^*, \kappa) = (40, 10)$ and use $(\ell, \tau) = (5, 3)$.
- **RecA** [JTZ15]: We collect electron microscopic images of RecA proteins on DNA which contain filament structure. We use 4 training images and 2 test images. We set $(L^*, \kappa) = (40, 10)$, and $(\ell, \tau) = (5, 3)$, respectively.
- **Aerial** [SLF14]: The dataset contains 14 remote sensing images of road networks. We select 7 images for the training and 7 images for the test, respectively. (L^*, κ) is set to $(50, 50)$ and each line segment consists of $\ell = 15$ and $\tau = 4$.
- **Cracks** [CGMN10]: Images of the dataset correspond to road cracks on the asphalt surfaces. We use 6 images to train and test the algorithms on different 6 images. Parameters for the morphological profiles (L^*, κ) are set to $(10, 5)$, and the physics of line segments (ℓ, τ) are set to $(5, 3)$.

To create subgraph G' for the curvilinear structure inference, we compute the average proportion of pixels ρ on the curvilinear structure by observing the training images. We find ρ values for DRIVE, RecA, Aerial and Cracks dataset as 8.63%, 4.99%, 8.16%, and 4.3%, respectively. The minimum path length $\hat{\ell}$ is adaptively set to 40, 30, 80, and 30 for DRIVE, RecA, Aerial and Cracks dataset, respectively, to obtain the best performance.

Fig. 11 shows sample images from the datasets. The topology of the latent curvilinear structure is varied while we compare it within the same dataset. Especially, Cracks dataset consists of difficult images due to the rough surfaced background textures. Also, the normalization operation (2) is necessary to remove irregular illumination factors on the datasets.



(a) (b) (c) (d) (e) (f)

Figure 10: Toy example of the proposed curvilinear structure inference algorithm: (a) input image contains a curvilinear structure which is denoted by gray color; (b) subgraph G' is induced from the structured output rankings; (c) and (d) show the intermediate processes of the 2-sweep algorithm starting from node $\mathbf{t}$ to find a diameter of the subgraph; (e) we assign 0 weight for all edges on the path; and (f) we repeat the process and add branches if the path length is larger than pre-defined length $\hat{\ell}$.

6.2 Evaluations

The proposed algorithm progressively reconstructs the curvilinear structure by adding a long path on the subgraph. Fig. 12 and Fig. 13 show the intermediate steps of the proposed graph inference algorithm for Drive and Aerial dataset, respectively. Unlike the previous models, the proposed algorithm is able to show different levels of detail for the latent curvilinear structure. Such information to visualize shape complexity of the curvilinear structure cannot be retrieved by tuning the threshold. In practice, a few number of iterations is required to converge the algorithm and each step to find a long path takes less than milliseconds for the computation. For the experiments, we use a PC with a 2.9 GHz CPU (4 cores) and 8 GB RAM.

We visually compare the performance of the proposed algorithm with the competing algorithms. Fig. 14, 15, 16, and 17 show the results from DRIVE, RecA, Aerial, and Cracks dataset, respectively. Although the proposed algorithm is the most suitable to show the topological importance level of the curvilinear structures, its intermediate result (dissimilarity score map) is comparable to the results of curvilinear segmentation algorithms.

For the quantitative evaluation, we provide precision-and-recall curve of the proposed algorithm (dissimilarity score map) and the state-of-the-art models in Fig. 18. Moreover, Table 1 summarizes the mean average precision (MAP) scores corresponding to Fig. 18 for each dataset. We plot the curves by adjustment of the threshold value to represent the curvilinear structure. The dissimilarity score map, which is an intermediate result of the proposed algorithm, is used to plot Fig. 18 and Table 1. The measure of true positive is sensitive for the misalignment; therefore, we consider surrounding pixels of the detection results as the true positive if a predicted point is falling into the ground truth with a small radius. To provide the average performance of each dataset, we compute the interpolated average precision values at the fixed recall values, where the method is detailed in [MRS08]. However, the evaluation with precision and recall is unable to measure the structural features. We also provide the average proportion of pixels to represent the curvilinear structures and the corresponding F_1 scores in Fig. 19 and Table 2. The graph shows that the proposed algorithm efficiently draw the curvilinear structures using smaller number of pixels than the other algorithms. The proposed algorithm achieved the best

F_1 scores for all datasets except the Drive dataset. It is because the proposed algorithm use the fixed thickness τ to describe line segments whereas the images of Drive dataset consists of blood vessels with varied thicknesses. The performance could be improved if we consider the thickness of the line segment as a learnable variable.

It is worth to mention the over-fitting problem in curvilinear segmentation task based on machine learning. Manual segmentation (ground truth) contains many errors around boundaries and minutiae components. Also, the number of training data employed in this work is relatively small due to the difficulties of making the accurate annotations. Specifically, since RecA dataset has only four training images, there is a high risk of over-fitting to the training set. It is remarkable that the proposed algorithm shows good performance for all datasets without over-fitting problem when compared to other learning based algorithms [BRLF13, SLF14].

7 Conclusions

This paper proposed a curvilinear structure reconstruction algorithm based on the ranking learning system and graph theory. The output rankings of the line segments corresponded to the plausibility of the latent curvilinear structure. Using an optimal number of pixels, the proposed algorithm provided different levels of detail during reconstruction of the curvilinear structure. More precisely, we learned a ranking function based on SSVM with the proposed orientation-aware curvilinear feature descriptor. The weighted average of curvilinear feature groups improved the consistency of the output rankings. In this paper, we invented a novel graphical model that infers the curvilinear structure according to the topological importance. The proposed algorithm looked for remote vertices on the subgraph which is induced from the output rankings. Across the various types of datasets, our model showed good performances to reconstruct the latent curvilinear structure with a smaller number of pixels comparing to the state-of-the-art algorithms.

Acknowledgments

The authors would like to thank David Coudert from Inria, France, for a fruitful discussion on graph search algorithms and Pierre Charbonnier from IFSTTAR, France, for the courtesy of road crack dataset.

References

[BBC+11] K. M. Brown, G. Barrionuevo, A. J. Canty, V. De Paola, J. A. Hirsch, G. S. X. E. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. A. Ascoli. The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2–3):143–157, 2011.

[Bes86] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society B*, 48(3):259–302, 1986.

[BRLF13] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua. Supervised feature learning for curvilinear structure segmentation. In *MICCAI*, pages 526–533, September 2013.

[BYVG11] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural SVM learning for supervised object segmentation. In *CVPR*, pages 2153–2160, June 2011.

[CDHP01] Derek G. Corneil, Feodor F. Dragan, Michel Habib, and Christophe Paul. Diameter determination on restricted graph families. *Discrete Applied Mathematics*, 113(2–3):143–166, 2001.

[CGMN10] S. Chambon, C. Gourraud, J.-M. Moliard, and P. Nicolle. Road crack extraction with adapted filtering and Markov model-based segmentation. In *VISAPP(2)*, pages 81–90, May 2010.

[FA91] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE TPAMI*, 13(9):891–906, September 1991.

[FNVV98] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *MICCAI*, pages 130–137, October 1998.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the bayesian restoration of images. *IEEE TPAMI*, 6(6):721–741, November 1984.

[Gre95] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):771–732, 1995.

[GTFF10] G. González, E. Türetken, F. Fleuret, and P. Fua. Delineating trees in noisy 2D images and 3D image-stacks. In *CVPR*, pages 2799–2806, June 2010.

[HRF+07] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE TGRS*, 45(12):4144–4157, 2007.

[JTZ15] S.-G. Jeong, Y. Tarabalka, and J. Zerubia. Marked point process model for curvilinear structures extraction. In *EMMCVPR 2015, LNCS 8932*, pages 436–449, January 2015.

[JU04] M. Jacob and M. Unser. Design of steerable filters for feature detection using Canny-like criteria. *IEEE TPAMI*, 26(8):1007–1019, August 2004.

[KH06] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 68(2):179–201, 2006.

[KYNK14] S. Kim, C. D. Yoo, S. Nowozin, and P. Kohli. Image segmentation using higher-order correlation clustering. *IEEE TPAMI*, 36(9):1761–1774, September 2014.

[LC08] M. W. Law and A. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *ECCV*, pages 368–382, October 2008.

[LDZ05] C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE TPAMI*, 27(10):1568–1579, October 2005.

[LLSF12] A. Lucchi, Y. Li, K. Smith, and P. Fua. Structured image segmentation using kernelized features. In *ECCV*, pages 400–413, October 2012.

[LMP01] J. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmetaing and labeling sequence data. In *ICML*, pages 282–289, 2001.

[MBZT12] A. Mittal, M. B. Blaschko, A. Zisserman, and P. H. S. Torr. Taxonomic multi-class prediction and person layout using efficient structured ranking. In *ECCV*, pages 245–258, October 2012.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[Per95] P. Perona. Deformable kernels for early vision. *IEEE TPAMI*, 17(5):488–499, May 1995.

[RL12] R. Rigamonti and V Lepetit. Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters. In *MICCAI*, pages 189–197, January 2012.

[SAN+04] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE TMI*, 23(4):501–509, April 2004.

[SLF14] Amos Sironi, Vincent Lepetit, and Pascal Fua. Multiscale centerline detection by learning a scale-space distance transform. In *CVPR*, pages 2697–2704, June 2014.

[TA07] H. Talbot and B. Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing*, 25(4):416–425, April 2007.

[TBA+13] E. Türetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. Reconstructing loopy curvilinear structures using integer programming. In *CVPR*, pages 1822–1829, June 2013.

[TJHA05] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, September 2005.

[VCB+10] S. Valero, J. Chanussot, J. A. Bendiktsson, H. Talbot, and B. Waske. Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Lett.*, 31(10):1120–1127, July 2010.

[YR11] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *CVPR*, pages 1386–1392, June 2011.

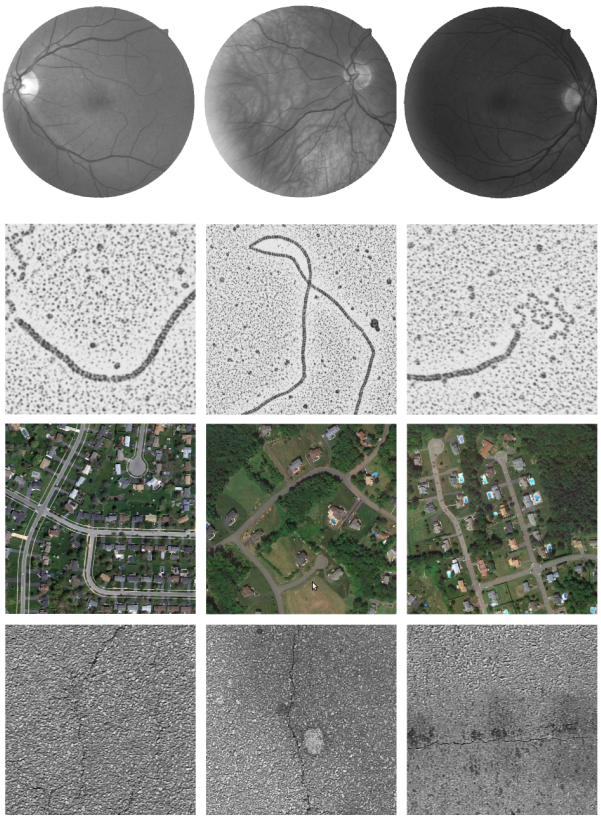
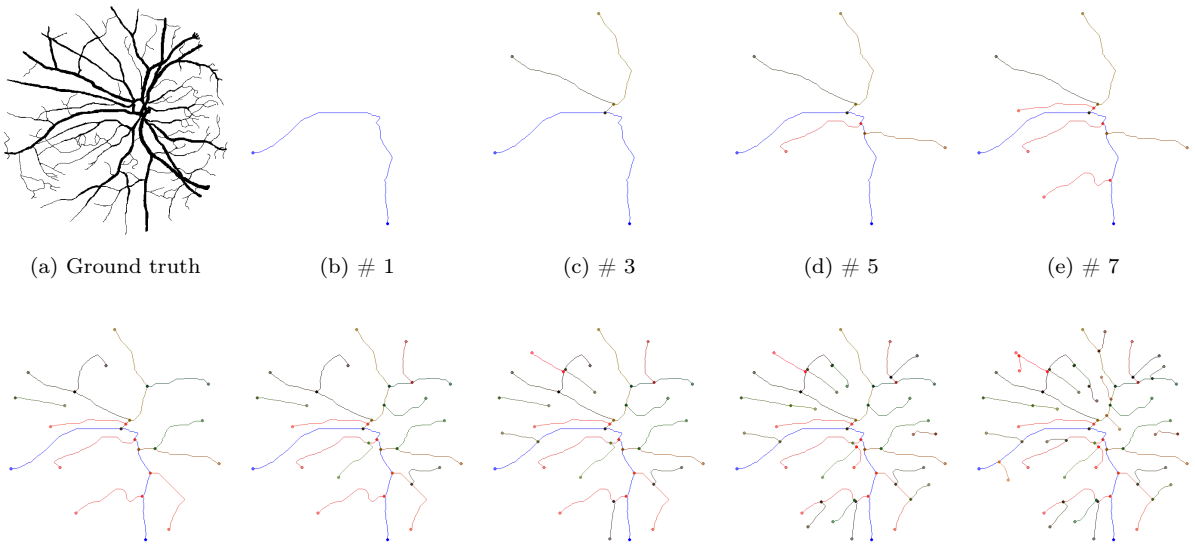


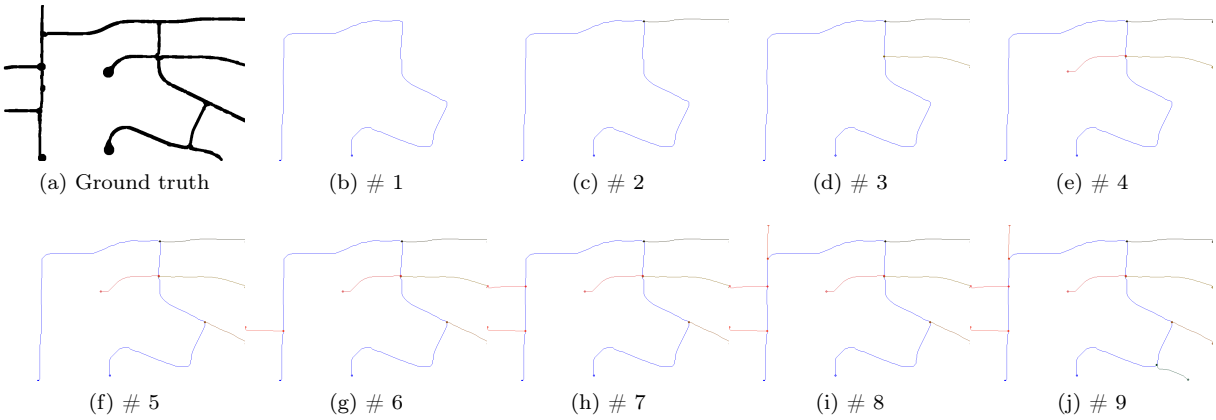
Figure 11: Sample images from the datasets employed in our work. Images are cropped to show the latent curvilinear structures clearly. **Top to bottom**: DRIVE, RecA, Aerial, and Cracks.



(a) Ground truth (b) # 1 (c) # 3 (d) # 5 (e) # 7

(f) # 12 (g) # 16 (h) # 20 (i) # 28 (j) # 36

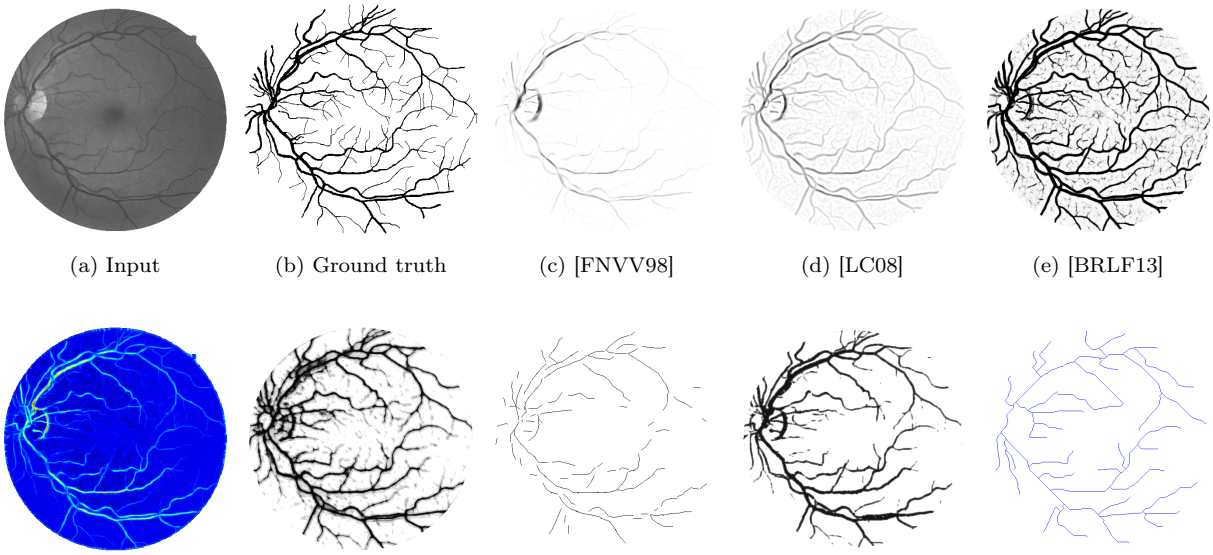
Figure 12: Intermediate steps of the curvilinear structure reconstruction for a retina image. We iteratively reconstruct the curvilinear structure according to topological importance orders. As the iteration goes on, detail structures (layer) appear.



(a) Ground truth (b) # 1 (c) # 2 (d) # 3 (e) # 4

(f) # 5 (g) # 6 (h) # 7 (i) # 8 (j) # 9

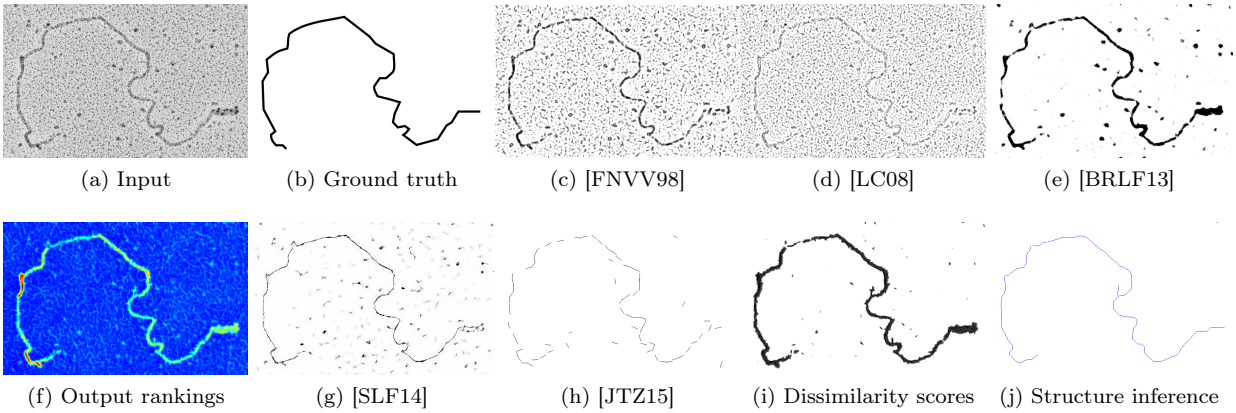
Figure 13: Intermediate steps of the curvilinear structure reconstruction for a road network image.



(a) Input (b) Ground truth (c) [FNVV98] (d) [LC08] (e) [BRLF13]

(f) Output rankings (g) [SLF14] (h) [JTZ15] (i) Dissimilarity scores (j) Structure inference

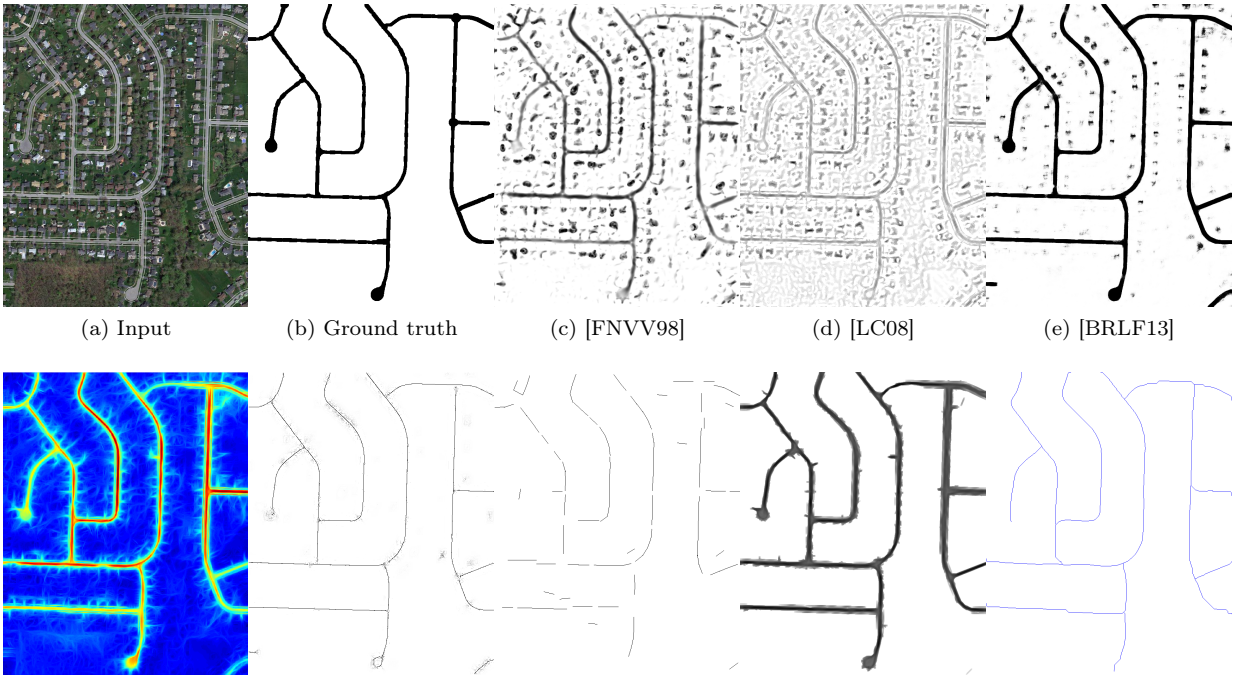
Figure 14: Experimental results on **DRIVE** dataset



(a) Input (b) Ground truth (c) [FNVV98] (d) [LC08] (e) [BRLF13]

(f) Output rankings (g) [SLF14] (h) [JTZ15] (i) Dissimilarity scores (j) Structure inference

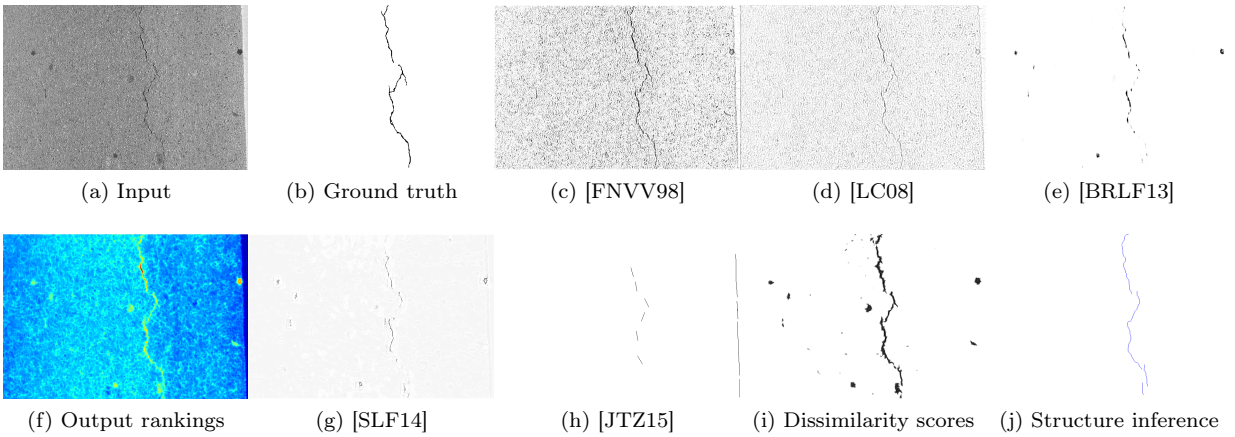
Figure 15: Experimental results on **RecA** dataset



(a) Input (b) Ground truth (c) [FNVV98] (d) [LC08] (e) [BRLF13]

(f) Output rankings (g) [SLF14] (h) [JTZ15] (i) Dissimilarity scores (j) Structure inference

Figure 16: Experimental results on **Aerial** dataset



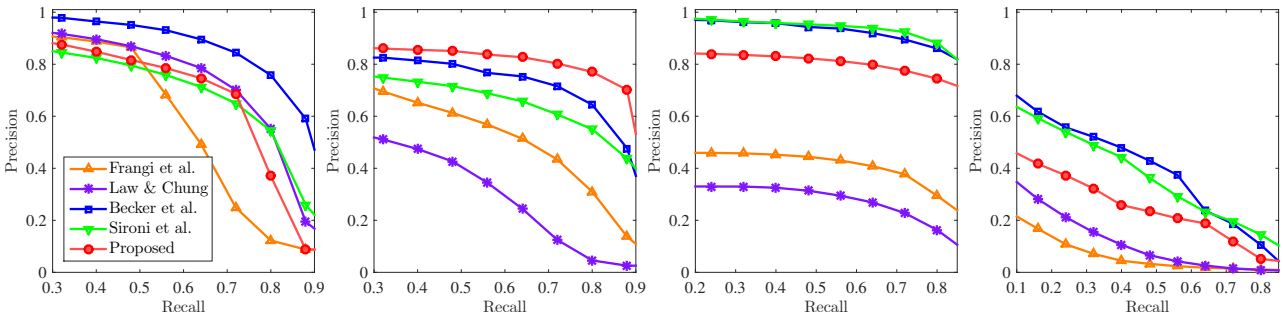
(a) Input (b) Ground truth (c) [FNVV98] (d) [LC08] (e) [BRLF13]

(f) Output rankings (g) [SLF14] (h) [JTZ15] (i) Dissimilarity scores (j) Structure inference

Figure 17: Experimental results on **Crack** dataset

Table 1: Mean Average Precision (MAP) scores of each dataset for quantitative evaluation. Boldfaced numbers are used to indicate the best performance in each test.

	[FNVV98]	[LC08]	[BRLF13]	[SLF14]	Proposed (dissimilarity scores)
DRIVE	0.61	0.73	**0.83**	0.68	0.67
RecA	0.55	0.33	0.70	0.64	**0.77**
Aerial	0.38	0.25	0.88	**0.89**	0.76
Cracks	0.074	0.14	**0.37**	0.36	0.25

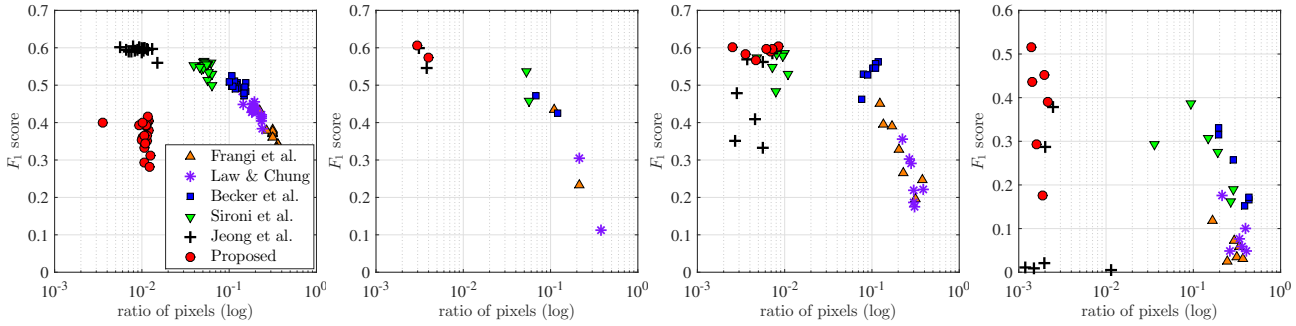


(a) DRIVE (b) RecA (c) Aerial (d) Cracks

Figure 18: Precision-and-recall curve of the curvilinear structure segmentation algorithms for each dataset. We plot the curve by controlling threshold values for each algorithm: [FNVV98], [LC08], [BRLF13], [SLF14] and dissimilarity scores of the proposed algorithm.

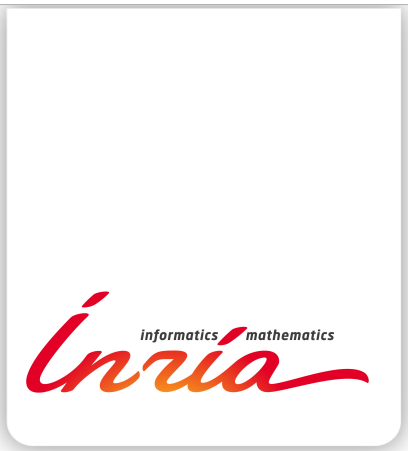
Table 2: F_1 scores and the average proportion of the pixels being a part of curvilinear structure (%) are computed. Boldfaced numbers are used to show the best F_1 score in each test. Gray colored cells in this table denote the algorithm using the minimum number of pixels to represent the curvilinear structure.

	[FNVV98]	[LC08]	[BRLF13]	[SLF14]	[JTZ15]	Proposed
DRIVE	0.33 / 37.09%	0.43 / 19.60%	0.50 / 12.67%	0.55 / 5.41%	**0.59** / 0.97%	0.36 / 1.07%
RecA	0.33 / 16.14%	0.21 / 29.82%	0.45 / 9.39%	0.50 / 5.47%	0.57 / 0.35%	**0.59** / 0.35%
Aerial	0.32 / 22.19%	0.25 / 29.51%	0.53 / 9.76%	0.55 / 0.83%	0.47 / 0.46%	**0.59** / 0.57%
Cracks	0.056 / 28.64%	0.085 / 33.00%	0.23 / 32.13%	0.27 / 17.12%	0.12 / 0.34%	**0.38** / 0.17%



(a) DRIVE (b) RecA (c) Aerial (d) Cracks

Figure 19: We analyze the relationship between the average proportion of pixels to draw the curvilinear structure and the corresponding F_1 score for each dataset.



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399