

Marked Point Process Model for Curvilinear Structures Extraction

Seong-Gyun Jeong, Yuliya Tarabalka, and Josiane Zerubia

Inria, Ayin team, 2004 Route des Lucioles, 06902 Sophia-Antipolis, France
{firstname.lastname}@inria.fr

Abstract. In this paper, we propose a new *marked point process* (MPP) model and the associated optimization technique to extract curvilinear structures. Given an image, we compute the intensity variance and rotated gradient magnitude along the line segment. We constrain high level shape priors of the line segments to obtain smoothly connected line configuration. The optimization technique consists of two steps to reduce the significance of the parameter selection in our MPP model. We employ Monte Carlo sampler with delayed rejection to collect line hypotheses over different parameter spaces. Then, we maximize the consensus among line detection results to reconstruct the most plausible curvilinear structures without parameter estimation process. Experimental results show that the algorithm effectively localizes curvilinear structures on a wide range of datasets.

Keywords: curvilinear structure extraction, marked point process, Monte Carlo sampling with delayed rejection, aggregation algorithm

1 Introduction

Curvilinear structures are widely observed in natural scenes. Thus, it is an important task to detect lines in many computer vision applications. For example, road network extraction algorithms [18, 30] have been developed for remote sensing. To find defects of the road pavement, an adaptive filtering and image segmentation algorithm has been proposed in [5]. For medical application, blood vessel detection [8, 25] aids diagnosis of disease. Localization of facial wrinkles [2, 16] provides visual cue of aging. However, these algorithms have a limitation of the use on different domains because the corresponding models of curvilinear structures have been specifically designed for their target applications.

Since linear structures correspond to image gradient information, image filtering with higher order derivatives [9, 15] is successful to grasp such image characteristics. Pixelwise segmentation for linear structure extraction measures the linearity for each pixel, and then sets up a threshold to remove out redundant outcomes [8, 5, 25, 30]. Supervised learning algorithm [3] has been proposed to find optimal convolution kernels for extracting linear features. Mathematical morphology operator [27] can enhance thin line structures based on shape information. However, criteria used for choosing the threshold values are ambiguous

if the image gradient information is corrupted by noise or rough textures. Tree-like representation [12, 29] has been recently proposed to automatically extract curvilinear structures. These algorithms initially define a set of seed points, and then grow branches based on a local tubularity measure [19]. However, the tree representation requires heavy computations, and a localization of seed points is crucial for the final result.

Curvilinear structures can be seen as a combination of small line segments. Sampling techniques with geometric priors have been exploited to detect multiple line segments in a scene [2, 16, 18, 22, 24, 28]. The *marked point process* (MPP) framework [6, 7, 20, 26] is helpful to enforce high level constraints on shape prior. However, the MPP model requires heavy formalization to interpret spatial distribution of the objects. Large number of parameters should be defined to describe the geometric shape of the objects (*modeling parameters*) and to control the relative importance of data and prior energy terms (*hyperparameters*). MPP modeling has been considered less practical to solve general problem because the performance is very sensitive for the selection of parameters. Although *stochastic expectation maximization* algorithm [4, 21] has been used to estimate modeling parameters, it exhibits both speed and scalability issues.

Although the contour grouping algorithms [1, 28] also examine image features corresponding to curves and lines, the goal is quite different from the curvilinear structures extraction techniques. The contour grouping algorithms seek closed contour lines to divide an image into meaningful regions. On the other hand, we look for multiple curvilinear structures, which are not necessarily closed, within a homogeneous texture. While the contours are associated with salient edges around objects boundaries, the curvilinear structures are subtle local image features in the same plane. Unlike [1], we cannot exploit global texture cues for the data energy term; therefore, an accurate design of the shape prior energy is essential to solve our problem.

In this paper, we propose a new MPP model for curvilinear structures extraction in a fully automatic way, where the performance is not biased by the hyperparameter selection. Indeed, our MPP model can detect wide types of input data without a sophisticated parameter tuning process. To fit in with a dataset, we analyze image gradients and homogeneity of intensities along the line segments. The prior energy is defined on local configuration to implement smooth connection among line segments (Sec. 2). To avoid the burden of hyperparameter selection, we first generate multiple candidates of the line configuration with different hyperparameter settings. Markov chain Monte Carlo sampler [11, 13, 14, 23] with delayed rejection scheme [14] is employed to optimize the proposed probability density function. Next, we combine the whole set of line candidates in a way that maximizes the consensus among line detection results (Sec. 3). Extensive experiments on various datasets including facial wrinkles, road cracks, DNA filaments, and blood vessels demonstrate the effectiveness of the proposed MPP model for extracting thin curvilinear structures (Sec. 4).

2 Marked Point Process Modeling

2.1 MPP Revisited

We briefly review the definition of MPP [20, 26] to provide a mathematical description of the proposed model.

Definition 1 (Spatial point process) *A realization of point process consists of an unordered set of points in a compact set $\mathcal{F} \subset \mathbb{R}^d$. A point process on \mathcal{F} maps from a measurable probability space $(\mathcal{F}, \mathcal{B}, \mu)$ onto the configuration space $\Omega = \cup_{n=0}^{\infty} \Omega_n$, where \mathcal{B} denotes σ -algebra of subset of \mathcal{F} , and μ is the Lebesgue measure. In other words, for all bounded Borel sets $B \subseteq \mathcal{B}$, the number of points falling in B is a finite random variable.*

Definition 2 (Marked point process) *In the MPP framework, each point is associated with additional information which describes a shape of the object. Specifically, we reconstruct curvilinear structures as smoothly connected line segments. Let $s_i = (\mathbf{x}_i, \mathbf{m}_i)$ be a line segment specifying its center point $\mathbf{x}_i = (x_i, y_i)$ in the image sites \mathcal{F} with a label of the length and the orientation $\mathbf{m}_i = (\ell_i, \theta_i)$, where the label is sampled from the mark space M with a probability measure μ_M . We now define a marked point process on $\mathcal{F} \times M$ as a finite random configuration $\mathbf{s} = \{s_1, \dots, s_n\} \in \Psi$.*

The probability distribution of the MPP is defined based on an image I and spatial interactions between line segments. Given an image, we look for an optimal configuration $\hat{\mathbf{s}}$ which maximizes the unnormalized probability density $f(\mathbf{s})$ as follows:

$$\hat{\mathbf{s}} = \operatorname{argmax}_{\mathbf{s} \in \Psi} f(\mathbf{s}) = \operatorname{argmin}_{\mathbf{s} \in \Psi} \sum_{i=1}^{\#(\mathbf{s})} U_d(s_i) + \sum_{i \sim j} U_p(s_i, s_j), \quad (1)$$

where $\#(\mathbf{s})$ is the cardinality of the configuration, and $i \sim j$ represents the symmetry relationship between interacting line segments s_i and s_j . U_d and U_p denote the data likelihood and the prior energy, respectively. In general, Monte Carlo samplers [10, 13, 14, 31] are employed in MPP models to maximize the proposed density function $f(\mathbf{s})$. Each state of a discrete Markov chain $(X_t)_{t \in \mathbb{N}}$ corresponds to a random configuration on the Ψ . The chain is locally perturbed by transition kernels, and is evolved to converge to the stationary distribution which is identical to the proposed probability density.

2.2 Data Likelihood

We define the data likelihood of the line segment s_i as a weighted sum of the rotated gradient magnitudes U_d^m and the intensity variance U_d^v along the line:

$$U_d(s_i) = \omega_d^m U_d^m(s_i) + \omega_d^v U_d^v(s_i), \quad (2)$$

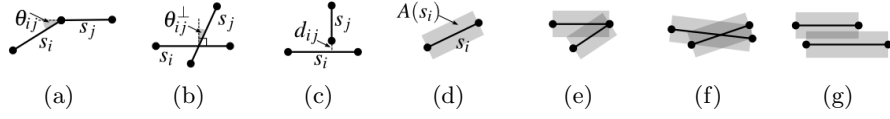


Fig. 1. Examples of the line configurations with different prior energies: (a)–(c) show preferable line configurations composed of aligned lines (a), almost perpendicular lines (b), and adjacent lines (c). (d)–(g) depict unfavourable line configurations which are penalized because of a singular segment (d), acute corner (e), overlap (f), and parallel (g), respectively.

where ω_d^m and ω_d^v are weighting coefficients corresponding to U_d^m and U_d^v , respectively.

We obtain the rotated gradient information by convolving the input image with steerable filters [9, 15]. Steerable filters are generated from a linear combination of basis filters. In this work, we use second-order derivatives of an isotropic Gaussian function as the basis filters. Let $g_{\theta_i}(\mathbf{x}; \sigma^2)$ be a steerable filter associated with an orientation θ_i and $\nabla I_{\theta_i} = g_{\theta_i} * I$ be its filtering response, which adaptively accentuates gradient magnitudes corresponding to the angle θ_i . Then, the gradient magnitude energy U_d^m is defined as

$$U_d^m(s_i) = \int_0^1 |\nabla I_{\theta_i}(\mathbf{p}_i(t))| dt, \quad (3)$$

where $\mathbf{p}_i(t)$ represents points on the line segment s_i . Note that $\mathbf{p}_i(t) = (1-t)\mathbf{u}_i + t\mathbf{v}_i$ is a function of the endpoints \mathbf{u}_i and \mathbf{v}_i with parameter $t \in [0, 1]$.

When the input image is heavily corrupted by noise or composed of uneven textures, observing gradient distribution often fails to detect linear structures. To ease this problem, we also measure the intensity variance along the line segment. This is because intensities are likely to be homogeneous, if pixels are laid on the same line. We can write:

$$U_d^v(s_i) = \frac{1}{\ell_i} \int_0^1 (I(\mathbf{p}_i(t)) - \mathbb{E}[I(s_i)])^2 dt, \quad (4)$$

where $\mathbb{E}[I(s_i)]$ denotes the intensity mean of the line segment s_i , and ℓ_i is the line length.

2.3 Prior Energy

In this section, we propose the prior energy to define spatial interactions on a local configuration. We want to obtain smoothly connected lines with a small curvature as a final solution. We compute the overlapping area $\Upsilon(s_i, s_j)$ to reject congestion of lines and the coupling energy states \mathbf{c}_{ij} to evaluate attraction between line segments (see Fig. 1). The prior energy $U_p(s_i, s_j)$ is defined as

$$U_p(s_i, s_j) = \Upsilon(s_i, s_j) + \mathbf{w}_p^T \mathbf{c}_{ij}, \quad \forall i \sim j, \quad (5)$$

where \mathbf{w}_p denotes a vector of weighting factors which control relative importance of each element in \mathbf{c}_{ij} . We assume that a line segment only correlates with the

other ones within a certain distance. Thus, a neighborhood system consists of pairs of line segments, such that their center distance is smaller than half the sum of their lengths. In other words,

$$i \sim j = \left\{ (s_i, s_j) \in \Psi^2 : 0 < \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \frac{\ell_i + \ell_j}{2} + \epsilon \right\}, \quad (6)$$

where ϵ denotes the marginal distance to be connected with each other.

In order to evaluate an overlapping area between line segments, we dilate the line segments with a three pixel-radius disk, and then count up the number of pixels falling in the same image site. Suppose that we have a set of points $A(s_i)$ which is a dilated version of the line segment s_i , and $n(A(s_i))$ denotes the number of pixels in $A(s_i)$. As shown in Fig. 1 (e)–(g), we penalize a configuration $\{s_i, s_j\}$, when a portion of the overlapping area is greater than 10% of $\min\{n(A(s_i)), n(A(s_j))\}$. However, almost perpendicular line segments are excluded from this penalty. The criteria for rejection are then given as

$$\Upsilon(s_i, s_j) = \begin{cases} 0 & \text{if } \theta_{ij}^\perp < \tau, \\ 0 & \text{if } \frac{n(A(s_i) \cap A(s_j))}{\min\{n(A(s_i)), n(A(s_j))\}} < 0.1, \\ \infty & \text{otherwise,} \end{cases} \quad (7)$$

where $\theta_{ij}^\perp = \frac{\pi}{2} - \theta_{ij}$ represents an angle difference between s_i and the perpendicular line of s_j , τ is the maximum angle difference for segments to be aligned.

The coupling energy states \mathbf{c}_{ij} of the lines are composed of the singularity, connectivity, curvature, and perpendicularity:

$$\mathbf{c}_{ij} = [1, \varphi(d_{ij}, \epsilon), \varphi(\theta_{ij}, \tau), \varphi(\theta_{ij}^\perp, \tau)]^\top, \quad \varphi(u, v) = \min\{0, (u/v)^2 - 1\}, \quad (8)$$

where d_{ij} denotes the minimum distance from endpoints of s_i to a point on the line s_j , and θ_{ij} is the angle difference between line segments. The function $\varphi(u, v)$ tests a firmness of the coupling state u by comparing with the given tolerance value v .

The weighting factors $\mathbf{w}_p = [\omega_p^s, \omega_p^c, \omega_p^a, \omega_p^r]^\top$ can be derived from their role in the prior energy. Specifically, ω_p^s penalizes birth of a single line segment in the final configuration; hence its value is affected by the average gradient magnitude and the noise level of the input. ω_p^c encourages adjacent segments within ϵ to become connected. ω_p^a promotes segments being aligned with a small curvature in the final configuration. ω_p^r supports perpendicularly approaching line segments. Although the selection of \mathbf{w}_p values is critical for the performances of the MPP model, it is hard to estimate the coefficients because of hidden dependencies among them.

2.4 Monte Carlo Sampler with Delayed Rejection

We employ the *Reversible jump Markov chain Monte Carlo* (RJMCMC) sampler [13] to obtain an optimal line configuration which maximizes the probability

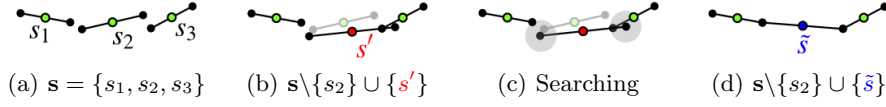


Fig. 2. Given configuration (a), if a line segment s' proposed by LT kernel is rejected (b), the delayed rejection kernel searches for the nearest extremes in the rest of line segments (c). An alternative line segment \tilde{s} , which enforces connectivity, will be proposed by interpolation of the retrieved points (d).

density function. The RJMCMC sampler is an iterative method that locally perturbs a current configuration \mathbf{s} with a transition kernel. The transition kernel consists of multiple sub-transition kernels, namely, *birth-and-death* (BD) and *linear transform* (LT). A new configuration \mathbf{s}' is proposed according to the transition kernel, given by

$$\xi(\mathbf{s}, \mathbf{s}') = \sum_m p_m \xi_m(\mathbf{s}, \mathbf{s}'), \quad (9)$$

where p_m denotes a probability to choose m -th type of sub-transition kernel $\xi_m(\mathbf{s}, \mathbf{s}')$. For each sub-transition kernel, the detailed balance condition [13] is required to ensure the reversibility of the Markov chain. Acceptance ratio $\alpha_m(\mathbf{s}, \mathbf{s}')$ is compared with a stochastic value $\text{rand}[0, 1]$ to take a new configuration into account. The RJMCMC sampler is coupled with the *simulated annealing* (SA) algorithm [17] to secure the convergence of the Markov chain via relaxation parameter T (temperature); the temperature gradually decreases as the iteration goes on. To compute an acceptance ratio of the transition kernel, we use a density $f(\mathbf{s})^{1/T}$ instead of $f(\mathbf{s})$. The acceptance ratio is

$$\alpha_m(\mathbf{s}, \mathbf{s}') = \min \left(1, \frac{\xi_m(\mathbf{s}', \mathbf{s}) f(\mathbf{s}')^{1/T}}{\xi_m(\mathbf{s}, \mathbf{s}') f(\mathbf{s})^{1/T}} \right). \quad (10)$$

The BD kernel changes the dimensionality of the current configuration \mathbf{s} by adding a new line segment or removing an existing line segment. When the birth kernel proposes a new configuration $\mathbf{s}' = \mathbf{s} \cup \{s\}$, the length and the orientation of the new line segment are uniformly sampled from the mark space $M = [\ell_{\min}, \ell_{\max}] \times [\theta_{\min}, \theta_{\max}]$, where ℓ_{\min} and ℓ_{\max} are the minimum and maximum length of the line segment, respectively. θ_{\min} and θ_{\max} denote the minimum and maximum orientation of the line segment, respectively. Note that we refuse a birth of the line lying on *singular points*, which have zero gradient magnitudes. On the other hand, the death kernel removes a line segment which is randomly picked from the current configuration. Thus, a new configuration $\mathbf{s}' = \mathbf{s} \setminus \{s\}$ is proposed by the death kernel. We compute the acceptance ratio of the birth kernel α_B and the death kernel α_D in the same way as proposed in [18], given by

$$\alpha_B(\mathbf{s}, \mathbf{s}') = \min \left(1, \frac{p_D}{p_B} \frac{\mu(\mathcal{F})}{\#(\mathbf{s}) + 1} \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right), \quad (11)$$

$$\alpha_D(\mathbf{s}, \mathbf{s}') = \min \left(1, \frac{p_B}{p_D} \frac{\#(\mathbf{s})}{\mu(\mathcal{F})} \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right). \quad (12)$$

Algorithm 1 RJMCMC sampler with delayed rejection

```

1: Initialize:  $X_0 \leftarrow \mathbf{s}_0$  (or  $X_0 \leftarrow \emptyset$ ),  $t \leftarrow 0$ ,  $T \leftarrow T_0$ 
2: while  $T > T_{\min}$  do
3:    $\mathbf{s} \leftarrow X_t$ 
4:   Choose a transition kernel  $\xi_m$  according to probability  $p_m$ 
5:   Propose a new configuration  $\mathbf{s}'$  with  $\xi_m(\mathbf{s}, \mathbf{s}')$ 
6:   if  $\alpha_m(\mathbf{s}, \mathbf{s}') > \text{rand}[0, 1]$  then
7:      $X_{t+1} \leftarrow \mathbf{s}'$ 
8:   else
9:     Propose an alternative segment  $\tilde{\mathbf{s}}$  based on  $\xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}})$ 
10:    if  $\alpha_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) > \text{rand}[0, 1]$  then
11:       $X_{t+1} \leftarrow \tilde{\mathbf{s}}$ 
12:    else
13:       $X_{t+1} \leftarrow \mathbf{s}$ 
14:    end if
15:  end if
16:   $t \leftarrow t + 1$ 
17:  Decrease the temperature:  $T \leftarrow T_t$ 
18: end while

```

The LT kernel chooses a line segment s randomly, and then modifies its model parameters: $s = (\mathbf{x}, (\ell, \theta)) \rightarrow s' = (\mathbf{x} \pm d\mathbf{x}, (\ell \pm d\ell, \theta \pm d\theta))$, where $d\mathbf{x}$, $d\ell$, and $d\theta$ denote changes of center position, length, and orientation, respectively. The LT kernel draws a new configuration $\mathbf{s}' = \mathbf{s} \setminus \{s\} \cup \{s'\}$. The acceptance ratio of the LT kernel is defined by

$$\alpha_{\text{LT}}(\mathbf{s}, \mathbf{s}') = \min \left(1, \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right). \quad (13)$$

The LT kernel can be extended by the delayed rejection scheme [14]. The main idea of the delayed rejection scheme is to give a second chance to a rejected sample point. The acceptance ratio of delayed rejection is defined by

$$\begin{aligned} \alpha_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) &= \min \left(1, \frac{\xi_{\text{LT}}(\tilde{\mathbf{s}}, \mathbf{s}') \xi_{\text{LT}}^2(\mathbf{s}', \tilde{\mathbf{s}}, \mathbf{s}) [1 - \alpha_{\text{LT}}(\tilde{\mathbf{s}}, \mathbf{s}')] f(\tilde{\mathbf{s}})^{1/T}}{\xi_{\text{LT}}(\mathbf{s}, \mathbf{s}') \xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) [1 - \alpha_{\text{LT}}(\mathbf{s}, \mathbf{s}')] f(\mathbf{s})^{1/T}} \right), \\ &\simeq \min \left(1, \frac{f(\tilde{\mathbf{s}})^{1/T} - f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T} - f(\mathbf{s}')^{1/T}} \right). \end{aligned} \quad (14)$$

where $\mathbf{s}' = \mathbf{s} \setminus \{s\} \cup \{s'\}$, $\tilde{\mathbf{s}} = \mathbf{s} \setminus \{s\} \cup \{\tilde{s}\}$, and $\xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}})$ is the transition kernel for the delayed rejection. In order to reduce the burn-in time, we add heuristics to design the delayed rejection kernel. When we propose an alternative line segment \tilde{s} , we look for the closest endpoints from both ends of s' , which is rejected from the first trial. The line segment \tilde{s} is generated by interpolation of the retrieved points; we force the connectivity of the neighboring segments, so that a probability of being accepted increases in terms of prior energy. Fig. 2 summarizes the process of the delayed rejection kernel, and Algorithm 1 provides the pseudo-code of the RJMCMC sampler with delayed rejection.

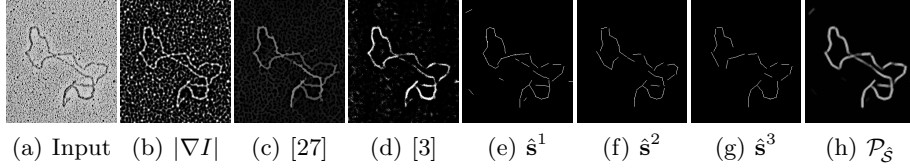


Fig. 3. Given the input image (a), we compute the gradient magnitude (b). Mathematical morphology operator, path opening [27], is applied on such gradient magnitude image (c). Linearity score of each pixel is drawn by the supervised feature learning algorithm [3] (d). We provide line hypotheses (e)–(g) associated with different hyperparameter vectors. Composition result (h) is equivalent to mixture probability density, and it highlights pixels corresponding to linear structures.

3 Curvilinear Structure Extraction via Integration of Line Hypotheses

While the MPP allows to design complex prior knowledge of the object distribution, its performance is very sensitive to the selection of modeling parameters and hyperparameters. For clarity, we note that the modeling parameters are related to the physical characteristics of the line segments (*e.g.*, range of length and orientation). The hyperparameters denote the weighting coefficients of energy terms (*i.e.*, w_d^m , w_d^v , and \mathbf{w}_p). The modeling parameters can be chosen empirically since the values are related to the image resolution (see Sec. 4); however, it is hard to estimate the hyperparameters via trial-and-error for different types of dataset. Our goal is to maximize the probability density without estimating hyperparameters.

3.1 Generation of K Line Hypotheses

Let $\mathbf{w} = [\omega_d^m, \omega_d^v, \omega_p^s, \omega_p^c, \omega_p^a, \omega_p^r]^\top$ be a hyperparameter vector which consists of the weighting coefficients of the proposed probability density. Suppose that we have K different hyperparameter vectors, $\mathbf{w}^1, \dots, \mathbf{w}^K$. For each hyperparameter vector, we substitute k -th hyperparameter vector \mathbf{w}^k into the proposed probability density $f(\mathbf{s}; \mathbf{w}^k)$. Then, we look for its optimal configuration $\hat{\mathbf{s}}^k$ via Monte Carlo sampler proposed in Sec. 2.4.

For a practical reason related to the implementation, we bound the values of \mathbf{w} . Specifically, we sweep the weighting coefficients of the prior energy \mathbf{w}_p according to the gradient magnitude and noise level of the input image. Let $\chi = -\ell_{\min} \times \mathbb{E}[\nabla I] + \text{Var}[I_{\sigma^2}]$ be a baseline to accept a new line segment into the current configuration without considering spatial interaction, where I_{σ^2} denotes a smoothed image using a Gaussian kernel with $\sigma^2 = \{1.5, 2.25, 3.5\}$. To reduce computation overhead, we fix the weighting factors of data likelihood energy as $\omega_d^m = -1$ and $\omega_d^v = 1$. We set $\mathbf{w}^1 = [-1, 1, \chi, 0.1\chi, 0.01\chi, 0.01\chi]^\top$, and gradually change χ by 10% of increments, *i.e.*, $\mathbf{w}^2 = [-1, 1, \chi_2, 0.1\chi_2, 0.01\chi_2, 0.01\chi_2]^\top$, where $\chi_2 = 1.1\chi$. In our experiments, we set $K = 15$ to create line hypotheses.

3.2 Combination of Line Hypotheses into a Probability Map

We now have a family of line hypotheses $\hat{\mathcal{S}} = \{\hat{\mathbf{s}}^1, \dots, \hat{\mathbf{s}}^K\}$ obtained from K different hyperparameter vectors. We jointly use the image data and the line hypotheses. More specifically, the final solution \mathbf{s}^* maximizes not only the probability density but also the consensus among line hypotheses. For each optimal configuration $\hat{\mathbf{s}}^k$, we compute a probability map \mathcal{P}_k of being a line in the image site. Then, we integrate K probability maps into a mixture density $\mathcal{P}_{\hat{\mathcal{S}}}$:

$$\mathcal{P}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in s_i^k, \\ \frac{1}{2} & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in A(s_i^k), \\ 0 & \text{otherwise,} \end{cases} \quad \mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \mathcal{P}_k(\mathbf{x}). \quad (15)$$

Fig. 3 compares image gradient magnitude, morphological filtering [27], supervised feature learning [3], line hypotheses, and the mixture density. Since the input image contains many high frequency components, its gradient also highlights non-linear structures in the background. While the morphological filter accentuates linear structures, its performance depends on the setting of path length. Supervised learning method requires high quality of a training dataset and corresponding ground truth images. Depending on the setting of hyperparameter vectors, the MPP model leads incomplete detection results as shown in Fig. 3. (e)–(g). We integrate line hypotheses of the proposed MPP model into a mixture density $\mathcal{P}_{\hat{\mathcal{S}}}$. The mixture density shows the consensus between line hypotheses in the sense that the pixels corresponding to line structures are more highlighted when compared to [3, 27].

We assume that the most promising hyperparameter vector draws a configuration which is more akin to the mixture density. We compute the *correlation-coefficient* (CC) between $\mathcal{P}_{\hat{\mathcal{S}}}$ and \mathcal{P}_k 's to analyze coherence of line detection results. That is

$$k^* = \operatorname{argmax}_{k=\{1, \dots, K\}} \operatorname{CC}(\mathcal{P}_{\hat{\mathcal{S}}}, \mathcal{P}_k), \quad (16)$$

$$\operatorname{CC}(\mathcal{P}_{\hat{\mathcal{S}}}, \mathcal{P}_k) = \frac{\sum_{\mathbf{x}} (\mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\hat{\mathcal{S}}}])(\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k])}{\sqrt{\sum_{\mathbf{x}} (\mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\hat{\mathcal{S}}}]^2 \sum_{\mathbf{x}} (\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k])^2}}, \quad (17)$$

where k^* represents the index of the most reliable hyperparameter vector.

3.3 Curvilinear Structure Extraction from Reduced Sampling Space

The line hypotheses span a configuration space $\mathbb{S} \subset \Psi$ which will be considered as a new sample space. Since the size of \mathbb{S} is significantly reduced compared to the original sample space Ψ , the optimization process becomes more tractable in terms of convergence time and detection accuracy.

We redefine the data likelihood energy by adding a new energy term as follows:

$$U'_d(s_i) = U_d(s_i) + U_d^h(s_i), \quad U_d^h(s_i) = \int_0^1 -\log \mathcal{P}_{\hat{\mathcal{S}}}(s_i(t)) dt, \quad (18)$$

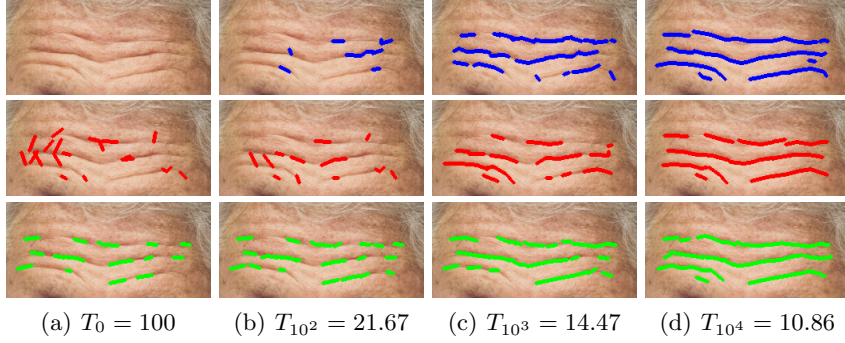


Fig. 4. We provide intermediate sampling processes when the temperature parameter T_t is decreasing. The results shown in first row are obtained without specifying seed segment. For the second row, we randomly set 20 seed segments and run the algorithm. For the third row, we initialize 20 line segments which are highly corresponding to underlying curvilinear structures. The algorithm converges toward almost the same solution regardless of the initial state.

where $U_d^h(s_i)$ quantifies the consensus among line hypotheses with respect to the line segment s_i . We stimulate the modified probability density over the reduced sample space \mathbb{S} with the most promising hyperparameter vector \mathbf{w}^{k^*} :

$$\mathbf{s}^* = \underset{\mathbf{s} \in \mathbb{S}}{\operatorname{argmin}} \sum_{i=1}^{\#(\mathbf{s})} U_d^h(s_i) + \sum_{i \sim j} U_p(s_i, s_j; \mathbf{w}^{k^*}). \quad (19)$$

4 Experiments

We test the proposed algorithm on a wide range of datasets: facial wrinkles, DNA filaments¹, road cracks, and retinas. The facial wrinkle images are collected on the Internet, and forehead areas are manually selected for the experiments. Test images of the defects on the road pavements and ground-truth are courtesy of Chambon *et al.* [5]. We use the DRIVE dataset [25] to test the proposed algorithm on retina images.

For all test sequences, we fix the modeling parameters as follows: ℓ_{\min} is set to 5 pixels and $\ell_{\max} = 20$ pixels. The orientation θ is varying from -90° to 90° with increments of 2° . The marginal distance of connected segments ϵ is fixed to 2 pixels, and the maximum angular difference of aligned segments τ is 30° . For the SA, the initial temperature T_0 is set to 100, and it follows the logarithm cooling schedule $T_t = T_0 / \log(1+t)$, where t denotes the number of the current iteration. We start the sampling process with the empty configuration. However, careful choice of initial segments can speed up the convergence of the algorithm (see Fig. 4). The computational time depends on the image resolutions; it takes less than a minute for the experimental images having 300×400 pixels, approximately. We use a PC with a 2.9 GHz CPU (4 cores) and 8 GB RAM.

¹ <https://www.biochem.wisc.edu/faculty/inman/empics/dna-prot.htm>

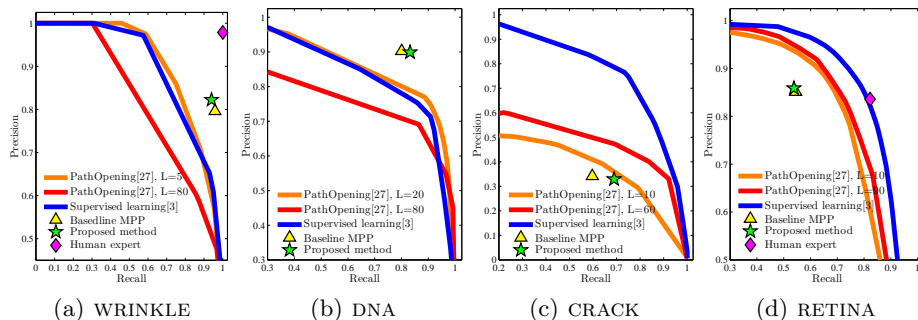


Fig. 5. Precision-and-recall curves for pixelwise segmentation of curvilinear structures using path opening operator [27] with different setups of length, supervised feature learning [3], baseline MPP, and the proposed method.

To compare the performances of the proposed method with the state-of-the-art techniques, we apply the path opening operator [27] on the gradient magnitude images by controlling the length parameters. For the supervised feature learning algorithm [3], we train 15 images for each dataset. In our experiments, we use the original implementations of path opening operator² and supervised feature learning algorithm³.

Fig. 5 shows the precision-and-recall curves for four test images. To obtain the curve of the comparison methods [3, 27], we tune thresholds on line detection results. The baseline MPP is selected from the line hypotheses among which it shows the best performance. The performances of the supervised learning algorithm are controlled by the quality of the training set; hence, it shows low performances on WRINKLE and DNA datasets, which are composed of noisy images with various sizes. In particular, the ground truth set of the WRINKLE dataset is based on subjective perception. While the morphology operator enhances linear structures on gradient magnitude images, it is required to specify the length of the linear structures according to the target applications. Since the pixelwise comparison fails to incorporate the geometry similarity with the ground-truth, the proposed algorithm shows lower scores on the CRACK and RETINA datasets. More specifically, the proposed algorithm detects slightly shifted lines for the CRACK image.

Fig. 6 compares the detection results of the proposed MPP model with the manually labeled image by human expert, morphology operator [27], supervised feature learning algorithm [3], and baseline MPP. For a fair comparison, we set the threshold values of the competing algorithms [3, 27] to obtain the closest recall scores to the proposed algorithm. Blue pixels denote perfectly matching regions as compared with the ground-truth. Green and red pixels show over-detected and under-detected results, respectively. The main strength of the proposed algorithm is that it ensures stable performances for all datasets without any parameter estimation procedure. The proposed algorithm extracts the most

² <http://hugues.zahlt.info/91.html>

³ <http://cvlab.epfl.ch/page-108936-en.html>

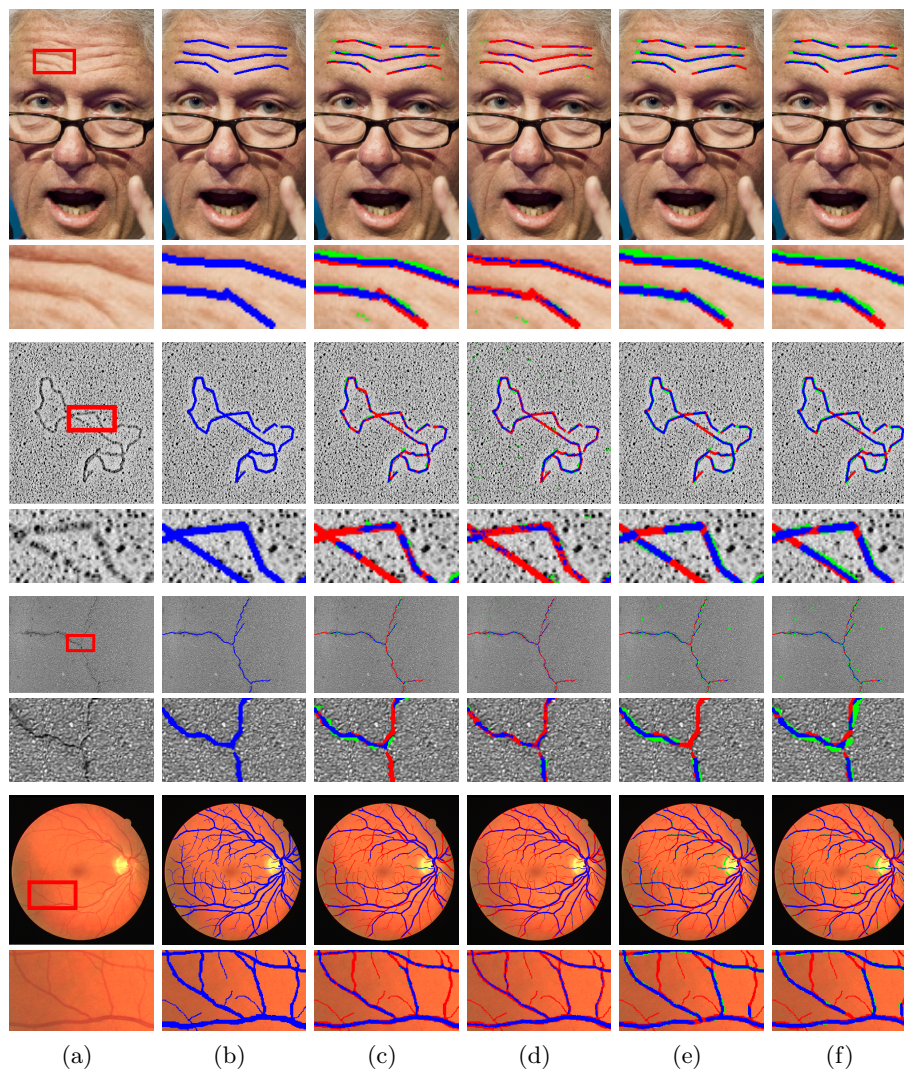


Fig. 6. We visualize the localization of the curvilinear structures on input images (a). We compare with the results of a manually labeled image by a human expert (b), morphological filtering [27] (c), supervised feature learning [3] (d), baseline MPP (e), and the proposed algorithm (f). Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to (b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground-truth. The name of the test images is from top to bottom: WRINKLE, DNA, CRACK, and RETINA.

salient line structures in the input image. On the other hand, the proposed algorithm suffers from under-detection when the width of the line structure is varying, for example, see the result for the RETINA. Such drawback can be overcome if we introduce an additional parameter for width of the line segment in our MPP model.

5 Conclusions

We have developed a new MPP model to reconstruct curvilinear structures via vectorized line segments. For the data likelihood, the density function computes rotated gradient magnitude and intensity variance. Prior energies of the proposed MPP model define interactions of the local configuration in terms of coupling energy states and overlapping areas. We have presented a new optimization scheme which is not biased by the parameter selection in the MPP model. We used an advanced RJMCMC sampler with different hyperparameter vectors to obtain line hypotheses. The line hypotheses span a feasible sample space, so that the final solution interprets underlying curvilinear structures more faithfully. We have shown line detection results on a wide range of datasets, and compared the performances of the proposed method with morphological filtering [27], supervised learning [3], and baseline MPP method. The whole optimization process is friendly designed to the parallel implementation; thus, the computational time can be further reduced by applying the parallel Monte Carlo sampler [31]. We plan to extend our model for time-varying sequences in order to analyze the temporal changes of the linear structures. While the heuristically proposed values for modeling parameters detect lines in practice, it is one of our future research topics to generate an optimal parameter vector using learning methods.

References

1. Arbe ez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE TPAMI* 33(5), 898–916 (May 2011)
2. Batool, N., Chellappa, R.: Modeling and detection of wrinkles in aging human faces using marked point processes. In: *ECCV Workshops*. pp. 178–188 (Oct 2012)
3. Becker, C., Rigamonti, R., Lepetit, V., Fua, P.: Supervised feature learning for curvilinear structure segmentation. In: *MICCAI*. pp. 526–533 (Sep 2013)
4. Celeux, G., Chauveau, D., Diebolt, J.: Stochastic versions of the EM algorithm: an experimental study in the mixture case. *J. Statist. Comput. Simulation* 55(4), 287–314 (1996)
5. Chambon, S., Gourraud, C., Moliard, J.M., Nicolle, P.: Road crack extraction with adapted filtering and Markov model-based segmentation. In: *VISAPP(2)*. pp. 81–90 (May 2010)
6. Chatelain, F., Descombes, X., Lafarge, F., Lantuejoul, C., Mallet, C., Minlos, R., Schmitt, M., Sigelle, M., Stoica, R., Zhizhina, E.: *Stochastic geometry for image analysis*. Wiley-ISTE (2012)
7. Descombes, X., Zerubia, J.: Marked point process in image analysis. *IEEE SPM* 19(5), 77–84 (Sep 2002)

8. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: MICCAI. pp. 130–137 (Oct 1998)
9. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE TPAMI* 13(9), 891–906 (Sep 1991)
10. Gamal-Eldin, A., Descombes, X., Charpiat, G., Zerubia, J.: Multiple birth and cut algorithm for point process optimization. In: SITIS (2010)
11. Gilks, W.R., Richardson, S., Spiegelhalter, D.: Markov chain Monte Carlo in practice. Chapman & Hall/CRC (1995)
12. González, G., Türetken, E., Fleuret, F., Fua, P.: Delineating trees in noisy 2D images and 3D image-stacks. In: CVPR. pp. 2799–2806 (Jun 2010)
13. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4), 771–732 (1995)
14. Green, P.J., Mira, A.: Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika* 88(4), 1035–1053 (2001)
15. Jacob, M., Unser, M.: Design of steerable filters for feature detection using Canny-like criteria. *IEEE TPAMI* 26(8), 1007–1019 (Aug 2004)
16. Jeong, S.G., Tarabalka, Y., Zerubia, J.: Marked point process model for facial wrinkle detection. In: to appear ICIP (Oct 2014)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (May 1983)
18. Lacoste, C., Descombes, X., Zerubia, J.: Point processes for unsupervised line network extraction in remote sensing. *IEEE TPAMI* 27(10), 1568–1579 (Oct 2005)
19. Law, M., Chung, A.: Three dimensional curvilinear structure detection using optimally oriented flux. In: ECCV. pp. 368–382 (Oct 2008)
20. Lieshout, M.N.M.v.: Markov point processes and their application. Imperial College Press (2000)
21. Møller, J., Waagepetersen, R.P.: Statistical Inference and Simulation for Spatial Point Processes. Chapman & Hall/CRC (2003)
22. Predoehl, A., Barnard, K.: A statistical model for recreational trails in aerial images. In: CVPR. pp. 337–344 (Jun 2013)
23. Robert, C., Casella, G.: Monte Carlo statistical methods. Springer (2004)
24. Schlecht, J., Barnard, K., Spriggs, E., Pryor, B.: Inferring grammar-based structure models from 3D microscopy data. In: CVPR. pp. 1–8 (Jun 2007)
25. Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. *IEEE TMI* 23(4), 501–509 (Apr 2004)
26. Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic geometry and its applications. Wiley (1987)
27. Talbot, H., Appleton, B.: Efficient complete and incomplete path openings and closings. *Image and Vision Computing* 25(4), 416–425 (Apr 2007)
28. Tu, Z., Zhu, S.C.: Parsing images into regions, curves, and curve groups. *IJCV* 69(2), 223–249 (Aug 2006)
29. Türetken, E., Benmansour, F., Andres, B., Pfister, H., Fua, P.: Reconstructing loopy curvilinear structures using integer programming. In: CVPR. pp. 1822–1829 (Jun 2013)
30. Valero, S., Chanussot, J., Bendiktsson, J., Talbot, H., Waske, B.: 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 (Jul 2010)
31. Verdié, Y., Lafarge, F.: Detecting parametric objects in large scenes by Monte Carlo sampling. *IJCV* 106(1), 57–75 (2014)