IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 7, 2021, accepted May 3, 2021, date of publication May 6, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078095

Stochastic Modeling of Trees in Forest Environments

KARIM BEN ALAYA AND LÁSZLÓ CZÚNI
Department of Electrical Engineering and Information Systems, University of Pannonia, 8200 Veszprém, Hungary
Corresponding author: Karim Ben Alaya (benalaya.karim@virt.uni-pannon.hu)

This work was supported in part by the Széchenyi 2020 Program under Project EFOP-3.6.1-16-2016-00015, in part by the Hungarian Research Fund under Grant OTKA K 120367, and in part by the NVIDIA Corporation with Graphical Processing Units (GPUs) by the NVIDIA GPU Grant Program.

ABSTRACT Our article deals with the detection and model generation of complex objects with curvilinear parts, like trees, with stochastic relaxation. The proposed algorithm can rely on any initial estimation of object parts as a probability map (like those generated by Gaussian mixture models or neural networks) and can model the relation of randomly sampled parts resulting in a structural representation of whole objects. Semantic segmentation by convolutional neural networks or the pose estimation with deep learning of object parts can predict the possible areas or positions of interest, but in many cases, a higher representation of structures is needed for further (e.g., shape or connectivity) analysis. The model validation of such data is straightforward for objects with known structure (like the human body or other rigid things) but tough for such complex objects like trees in forest environments. In our approach, the possible configurations of structures are generated by a marked point process while the optimal state is achieved by a solver based on reversible jump Markov chain Monte Carlo dynamics. The model generator relaxation method itself is unsupervised, no training is required, and our analyses show it has satisfactory stability, regarding detection accuracy, against changing its parameters. Besides giving the theoretical background and algorithmic steps, we present numerical evaluations on three datasets: synthetic trees, another of natural images with different species of trees in various forest environments, and the third is of road maps. The analyzed examples show that our approach, contrary to previous thin line detectors, can handle thin and thick objects.

INDEX TERMS Parts-based object detection, image segmentation, curvilinear objects, marked point process, reversible jump Markov chain Monte Carlo dynamics.

I. INTRODUCTION

The detection, recognition, and pose estimation of objects of the real world are elementary computer vision problems. These three tasks can be solved independently, however, on many occasions, they heavily rely on each other. For example, in the case of non-rigid objects the possible pose of the parts is strongly limited, serving valuable information for detection. In several applications, we would like to generate the structural description of the parts of objects; think of aerial images of road networks, blood vessels of the liver or other organs in medical images, or the analysis of plant images. In all these cases, the structure can be described by curvilinear or piecewise linear parts, but since they appear in cluttered environments, occlusion, noise, and changes in scale, size, or pose can decrease the performance of the available recognition methods.

Object recognition techniques can be categorized in many ways, one popular grouping is as follows:

1) Learning-based recognition: It works mostly with training, far from explicit (shape, textural) models, and has the ability to use semantic labeling. Typical representations of this class are the deep neural network (DNN) based methods [1], bag of words (BoW) [2] approaches, and the cascade filters of Viola and Jones [3].
2) Image invariance methods are based on matching a set of image patterns (e.g., brightness levels), which ideally uniquely determine the objects being searched for [4]–[6].
3) Model-based recognition: explicit, high-level modeling of objects, which may include parts and their relations, f.e., [7]–[10], [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

In our paper, we concentrate on specific types of objects: structures built-up from linear or close to linear parts like trees are under investigation. Thus, we have a weak initial assumption about the objects' model structure but have no clue for the location, orientation, size, and the number of different parts of similar appearance. Component-based or parts-based object detection techniques emphasize the detection of components, making it possible to process images of non-rigid or occluded objects, handling significant changes in viewpoint, or tackling the influence of noise.

Semantic segmentation or pose estimation by DNNs can inherently code the inner structure of objects but to obtain a high-level explicit representation, model validation is required. For example, while in [12], a state-of-the-art pose estimator, the most probable positions of object corners with the vectors towards the centroid are estimated by a neural network, the final validation of the pose is done through the old PnP algorithm [13]. Unfortunately, for such objects as thick and thin branches of trees, this validation is not straightforward since there is no specific model for the possible locations of trunks and branches. Our proposed approach differs from those above and has the following main characteristics:

- We use no traditional templates; circles are placed to different parts of the objects as markers.
- The position and size of circles are determined by the underlying probability map showing the chance that the location belongs to the object. Thus, large variability in size is possible.
- The flexible relation of the parts is established by edges connecting the circles.
- There are no explicit global rules to define the connections of circles, only local optimization is applied to achieve a final representation.
- Optimization is carried out by a marked point process (MPP) [14] driven by reversible jump Markov chain Monte Carlo (RJMCMC) [15] dynamics followed by some post-processing steps finalizing the models.
- Besides the graph representation of the objects, a coherent pixel-wise segmentation map is also generated.

Our main contribution is a method that can build structural (graph) models of various curvilinear structures in cluttered environments based on probability assumptions. Its advantage that it is not limited to thin lines, size variations are handled by the MPP, and besides, the generated graphs code the segmentation map of the image. As we will show, it is robust to parameter settings and can be applied not only for trees but to other piece-wise linear structures. In Figure 1 we illustrate a road map from aerial view, its initial DeepLabv3+ [16] segmentation, and our model built upon it. In this example, thin lines are modeled with vertices and edges. Our experimental section will show how the proposed method behaves on objects composed of not only thin but also thick lines.

In the next section, we overview related works to understand the motivation of our proposed model, then in Section III we describe its details. In Section Framework of Applications, we show how it can be utilized for the generation of curvilinear object models, like trees in forest environments; the discussion part is in Section V. Finally, we conclude our findings in Section VI.

II. RELATED WORKS

Object detection and recognition are vast research areas in computer vision; we can only highlight some papers related to parts-based approaches and the detection of curvilinear objects. The discussion of some articles of these two fields is essential to understand our motivations and see how our solution implements both areas' functionality.

We start with some classical parts-based techniques: in early methods the parts of the human face were modeled [7]–[9], then more loose connections were expected for the parts of pedestrians [10], or even more general structural (periodical) patterns were assumed when aerial views of tree crowns were processed [17]. We also discuss methods for the detection of curvilinear objects applied in the medical field [18]–[21], [22].

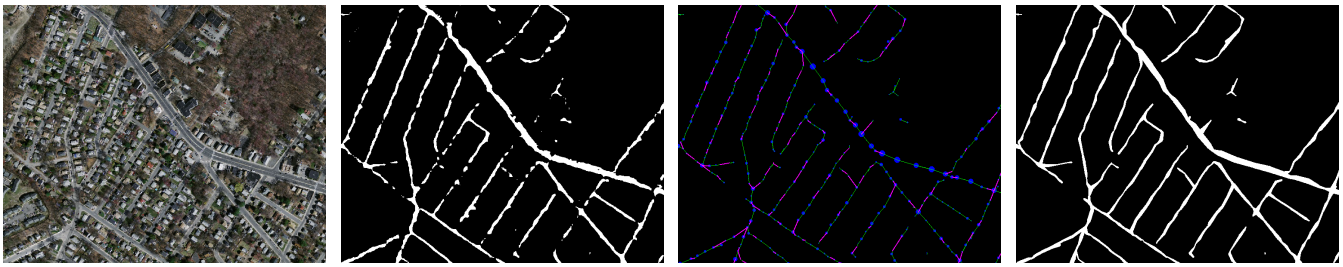
Part-based detection approaches can be categorized by several aspects:

- What kind of deformable templates or filters are used: while the templates are limited in variability, some distortions of features can be handled during matching.
- How the relation of parts is handled: from BoW approaches, where there is a very weak spatial relation of the parts, through constellation [23], star-shape [24], tree [25], k-fan [26], hierarchical [27] to sparse flexible models [17], [28].

The general ideas for parts-based recognition appeared very early. In [7] face detection was achieved with global templates, which consisted of the fitness of local features and spring forces between some of the candidate parts. The aim is to minimize the overall cost by finding the correct parts resulting in weak spring forces and a good appearance fit. They solved the problem via dynamic programming with the so-called linear embedding algorithm. The approach is relatively simple and can be generalized with some limitations. Its weakness is that the parameters are very scale-specific, and the used local features cannot cope with a large variety of possible forms.

The proposed system of Shams and Spoelstra [8] uses a neural network to generate confidences for possible left and right eye regions, which are paired together to form all possible combinations. These pairings' confidences are weighted by their topographic suitability, which are then thresholded to classify the pattern.

Yow and Cipolla [9] have also developed a component-based approach to detect faces. Their system categorizes potential features into candidate groups based on topographic evidence and assigns probabilities (that they are faces) to these groups. The probabilities are updated using a Bayesian network. If the final probability measure of a group is above a



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

FIGURE 1. Illustrating the applicability of our graph model generation on an image of a road network. (a): input (aerial view of roads), (b): image generated from probability map of DeepLabv3+, (c): graph model generated by the proposed MPP method, and (d): segmentation reconstructed from our graph model.

certain threshold, a ''detection'' is declared. The features are initially identified by using an image invariance scheme.

There are less strict requirements for the relation of parts in people detection since the larger objects can be more easily occluded, can easily vary their shape, and often appear in scattered outdoor environments. In [10] the performance of person detection systems on frontal and rear views was increased by a parts-based approach, which was also capable of solving occlusions. The parts of a body (face, legs, arms) were recognized by detectors processing the candidate regions by applying the Haar wavelet transform and then classifying the resultant data vector. Data classification is handled by several support vector machines (SVM) classifiers arranged in two layers. The component classifiers are quadratic SVMs, which were trained before use in the detection process. The raw output of these SVMs is a rough measure of how well a classified data point fits its designated class. The combination classifier (a linear SVM) processes scores received from the component classifiers to determine if the pattern is a person.

More concentrating on curvilinear or line-like structures, we can find approaches for hand-drawn sketch vectorization, blood vessel detection, or road map localization. Now, look closer at some of these, where typically thin structures are to be discovered in various applications. In [29] we find a correlation-filter-based approach for the vectorization of hand-drawn fashion sketches. Here the crucial step is the precise extraction of thin lines from sketches that are potentially very diverse. For this step, Pearson's correlation coefficient with multiple Gaussian kernels is applied, granting invariance to image contrast and lighting, making the extracted lines more reliable for vectorization. Although the method was tested with artificially added noise, the whole approach is applicable to almost binary drawings but not textured, colorful photos.

A highly referenced paper for curvilinear structure detection in medical images is [18], where a 3D line enhancement filter is developed to discriminate line structures from others. The filter uses the eigenvalues of the 3D Hessian matrix. Multi-scale integration is carried out by taking the maximum among single-scale filter responses. The application is illustrated by the segmentation and visualization of several modalities of medical images. This technique's robustness is over-fulfilled by the method proposed in [19], where a new curvilinear structure detector, called optimally oriented flux (OOF) is introduced. Unfortunately, in [19] only preliminary detection results are shown on phase-contrast magnetic resonance angiographic image volumes.

More applications and extension of the OOF detector can be found in [20], where the tree structure reconstruction is achieved by enforcing time consistency. Here curvilinear tree structures were evolving, such as road networks in 2D aerial images or neural structures in 3D microscopy stacks acquired in vivo. When processing the neural images, they use a local scale-space tubularity measure computed for every pixel applying the oriented flux cross section trace measure [19]. This way, it can be characterized how likely it is that a given spatial position lies on a centerline of a tubular structure of a given radius. The tubularity map was thresholded, and the highest tubularity points were selected iteratively, applying a non-maximum suppression mechanism. A number of tree roots are manually selected by a human operator. To enforce temporal consistency, all images of a sequence were processed simultaneously. They formulated the problem as a quadratic mixed integer program and demonstrated the additional robustness that comes from using all available visual clues at once instead of working frame by frame. Unfortunately, manual selection of object points cannot be applied in many cases in the above manner. In several applications, we can face many thick structures, so in our proposal, we will use neural networks to produce probability maps for the possible positioning of the object parts during an MPP-RJMCMC optimization technique (without any manual interaction).

Image noise, low contrast, or the identification of connectivity when one component bifurcates or two or more cross each other, raise interesting questions. A possible solution can be found in [21], where a novel curvilinear structural similarity measure, to guide a dominant-set clustering approach, is introduced. It considers both intensity and geometric properties in the representation of curvilinear structures locally and globally, and it groups curvilinear objects at crossover points into various connected branches by dominant-set clustering.

While for image segmentation, Markov random field [30] and conditional random field [31] techniques were widely used for a long time, their performance was overtaken by CNNs, and in general, they were not about graph-like representations of curvilinear structures. MPP approaches look more suited to fit spatial object models to observations. Either 2D or 3D data are used, models for man-built structures, cars, trees, or people can fit energy optimization methods. [32] describes two similar techniques where line segments are used for modeling line networks. While these models' application is limited to relatively thin objects, it is unclear how the proposed techniques behave on cluttered real-life data since no real-world examples are given, just a few simulations with various parameters. In [33] point clouds, obtained from lidars, are processed, and simple rectangle models are fit to cars. In [34] also plain rectangles are used as models for buildings and cars, but their alignment is considered hierarchically. Some hierarchy is already considered in another relevant paper [17], where an MPP-RJMCMC dynamics framework solves the problem of tree crown detection from remotely sensed data. The tree crowns are determined by their top positions and diameters using stochastic geometry. They are modeled by ellipses, represented by a Bayesian energy formula, containing both prior energy, incorporating prior knowledge of the plantation geometric properties, and a likelihood that fits the objects to the observed data. We can consider this approach an implicit part-based method where the tree plantation is the ''whole object'' and the individual crowns are the parts. However, since a global energy term (the Fourier transform of the image) is also involved, the MPP does not rely on only local interactions. Moreover, this approach, like most of the others, use an explicit shape-model (ellipse).

A more recent MPP model is described in [35] where the task was related to the analysis of fiber-reinforced composite materials: the detection of short and long fibers in microscopy images. Fibers had varying sizes, so short ones were modeled with ellipses, long ones with connected ellipses (called tubes). The tube model had a connection prior, to favor certain connections between the tubes, based on their mutual positional relationship. This MPP model could also be used for roads but may easily fail for composite objects of greatly varying sizes.

A 3D extension of this 2D tube model is in [36] where fibers are detected in X-ray tomography images. Due to the lack of labeled data, deep learning methods cannot be directly applied to this task, but the MPP model performs well. We can somehow consider this as a parts-based approach where longer fibers are associated with connected cylinders in the framework: the cylinder model has two spherical areas at its ends (the joint areas), each is used to define connection priors that encourage the connection of tubes that belong to the same fiber. The MPP detection process is accelerated through a growth kernel: an optimization that allows more birth of new tubes near the ends of tubes.

A difficult problem for the detection and quantification of properties of individual, nanometer-sized stress granules from intact tissues is introduced in [22]. The challenge comes from their varying size, shape, intensity, low signal-to-noise ratio, and often out-of-focus images. The MPP behind fits sets of shapes on the image plane and selects the ones matching best with the predefined object characteristics. While this model cannot handle complex and larger objects, it performs well to detect poorly contrasted ones of heterogeneous size and intensities.

Our first attempt to tree segmentation with an MPP is in [11] where our initial models were formulated. We have significantly improved our model definitions, and besides changing cost functions, we added post-processing steps to improve results. Our approach's advantage is its generality and that it can use any detector to generate probability maps to rely on. As output, it can generate the graph representation of the structures consisting of flexible parts, and consequently pixel-wise segmentation maps.

III. THE PROPOSED METHOD

From a given probability map as an input, which can be obtained through various ways, like Gaussian mixture models (GMM) [11] or convolutional neural networks (CNN) [16], [37], we seek to achieve a higher-level representation of tree-like structures by using an MPP formulation. Our approach is based on the idea, that due to the cylindrical shapes of the different parts of the studied objects, a set of connected circles (nodes) and their connections (edges) can be used to efficiently model each of those structures.

To keep the model formulation relatively simple and avoid a combinatorial, stochastic variational explosion, we initially decide to enforce only one connection for each node. Moreover, restricting the search space to the four nearest circles (in addition to the previously connected structural element), while investigating possible connections of each node, largely contributes to the acceleration of the already computationally expensive modeling scheme.

We propose a custom solver based on RJMCMC dynamics, which, by testing different MPP configurations and comparing their respective energies, will lead, through the process of energy minimization, to an energy minimum.

Finally, we define three post-solver steps, in which we investigate additional joints that failed to occur due to the limitations of our initial simplistic mathematical modeling. Adding those missing connections, we can connect different object parts.

A. MPP FORMULATION AND LOCAL ENERGY TERMS

A marked point process is defined as a point process, with a density function following the Poisson distribution [14]. In our work, the N points of the process are the centers $x_n \in [0, x_{max}]$ and $y_n \in [0, y_{max}]$ of the circles c_n whereas the marks are the different attributes (radius $r_n \in [r_{min}, r_{max}]$, connections to other circles). Each configuration X_i is therefore uniquely defined by the point distribution of it's centers of circles and the attributes carried by each of those objects.

Starting from the Bayes theorem and given an input image I, we express the density of each possible configuration as follows:

$$f(X_i) = f(X_i|I) = \frac{f_p(X_i)L(I|X_i)}{f(I)} \propto f_p(X_i)L(I|X_i) \quad (1)$$

where $f_p(X_i)$ stands for the prior density and $L(I|X_i)$ represents the likelihood of the data given in terms of a probability map [38].

By applying the logarithm on (1), we can bring the problem of maximizing configuration probabilities to the simpler task of local energy minimization of individual parts:

$$E(X_i) = -\log(f(X_i)) = \sum_{n=1}^{N} E(c_n) \quad (2)$$

with $E(X_i)$ as the energy of a particular configuration and $E(c_n)$ is the local energy of a part (node n). It follows that $E(c_n)$ has two main terms, the first is the data term (likelihood of observation O, $L(I|X_i)$ in (1)), the second is the prior ($P, f_p(X_i)$ in (1)):

$$E(c_n) = E_O(c_n) + E_P(c_n) \quad (3)$$

where

$$E_O(c_n) = w_l E_l(c_n) + w_{cn} E_{cn}(c_n) \quad (4)$$

and

$$E_P(c_n) = E_{no}(c_n) + w_s E_s(c_n) + w_{dcn} E_{dcn}(c_n) + w_{dn} E_{dn}(c_n) + w_{rs} E_{rs}(c_n) \quad (5)$$

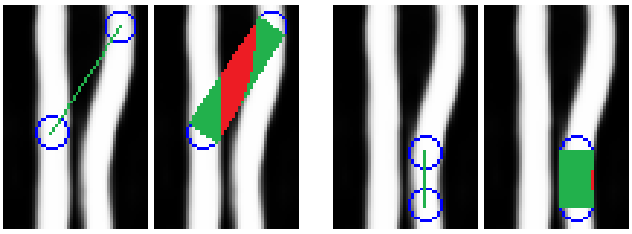
The individual energy terms (equipped with weighting constants), in the order of appearance, correspond to the following probabilities:

- $E_l(c_n)$: energy term based on the location of the circle; represents the likelihood probability calculated by averaging the values of the input probability map covered by the circle c_n.
- $E_{cn}(c_n)$: connection energy term expressing the probability of an edge originating from c_n; calculated by averaging the values of the probability map under the beam (a trapezoid-like structure) between the connected nodes (see Figure III-A).
- $E_{no}(c_n)$: non-overlap probability of the circle n.

$$E_{no}(c_n) = \begin{cases} 0, & \text{in the absence of an overlap} \\ \infty, & \text{otherwise} \end{cases}$$

The absence of overlap for a given circle c_n is determined if for all circles n' such as $n' \in [1, N] \setminus \{n\}$, the percentages of intersection for both n and n' are all below a fixed threshold $Th_{no} \in [0, 1]$.

- $E_s(c_n)$: represents size probability. Using a Gaussian distribution function to model the size of the expected objects (the width of the tubular parts), we set the mean of the distribution to the mean radius ($\frac{r_{min}+r_{max}}{2}$) while considering the standard deviation as another possible tuning parameter.
- $E_{dcn}(c_n)$: represents "length-of-connection" probability. As with size probability, we use a Gaussian function to model the distance distribution between the centers of each circle and its connection. We define $\frac{\sqrt{x_{max}^2+y_{max}^2}}{2}$ as the mean of the distribution with a tunable standard deviation.
- $E_{dn}(c_n)$: represents "distance-to-nearby" probability. It is defined similarly to the length-of-connection probability, except that it represents distances between each circle and the nearest circle to it, regardless of connectivity between these nodes.
- $E_{rs}(c_n)$: represents "radius similarity" probability. A Gaussian distribution function models the differences in radius between each circle and its connection. The mean is set to 0, and the standard deviation is left as a tunable parameter.



(a) (b)

FIGURE 2. High-energy (a) and low-energy (b): 2 different configurations with 2 circles: the colored trapezoid-like structure represents the connection probability, and can be decomposed to green (high probability pixels) and red (low probability pixels) areas.

B. RJMCMC SOLVER IMPLEMENTATION

In order to optimize the configurations of the previously detailed MPP modeling, we use RJMCMC dynamics as base reference for our custom solver. We incorporate essential optimization steps that can tackle our modeling's limitations while ensuring the exclusion of certain configurations that are, by design, incompatible with the solver.

Our usage of a dynamic set of circles makes the solution resilient to substantial variations in quantity and shape of objects to be modeled while at the same time ensuring the desired level of accuracy for the reproduced pixel-wise results.

Here we describe the main parts of the algorithm: first, the RJMCMC core block with local random walks is given, then in III-B2 the whole main loop is defined. Post-processing steps will be given in Subsection III-C.

1) RJMCMC CORE BLOCK

This part of the solver is implemented by directly applying our MPP model description and the standard definition of the RJMCMC algorithm [14], [15], using simulated annealing [39] and applying modified Metropolis dynamics [40] to reach better configurations.

From an initial configuration X_i, with an energy level E_i, each iteration of the core block algorithm follows the steps below:

1 Choose a movement kernel Q, according to a uniform probability.
2 Propose a new configuration X_{i+1}, with a global energy E_{i+1}, based on Q.
3 Compute $\alpha = \min(1, \frac{E_i^{\frac{1}{T_k}}}{E_{i+1}^{\frac{1}{T_k}}})$, the acceptance ratio of the move, with T_k, a slowly decreasing temperature over the solver's iterations.
4 Accept or reject the move, through a comparison of α to α_{min}:

$$\begin{cases} \text{accept } Q, & \text{if } \alpha \geq \alpha_{min} \\ \text{deny } Q, & \text{otherwise} \end{cases}$$

With the acceptance of Q, the model jumps from configuration X_i to X_{i+1}, whereas a denial forces the solver to remain on the previous configuration X_i.

We define the proposition kernel Q as the sum of two sub-kernels Q_1 and Q_2, each with its own set of movements. Any proposal of Q is therefore a combination of one element from Q_1 then another from Q_2, with respect to the stated order.

We implement the movements of Q_1 as follows:

- *Death and birth*: corresponds to the override, of a randomly selected circle $n \in [1, N]$, to its center coordinates $x_n \in [0, x_{max}]$, $y_n \in [0, y_{max}]$, and to all of its attributes (radius $r_n \in [r_{min}, r_{max}]$, connection to other circle).
  The newly drawn coordinates are set according to the Poisson distribution function, whereas the radius of the object follows the radius probability model previously mentioned in III-A.
- *Translation*: similar to the death and birth process, except the radius remains the same, while the coordinates of the center are reset as follows:

$$\begin{cases} x_{n_{new}} = x_{n_{old}} + x_{tr} \\ y_{n_{new}} = y_{n_{old}} + y_{tr} \end{cases}$$

  with x_{tr} and y_{tr}, 2 randomly chosen variables but constrained to the circle's size and to the boundaries resulting from the local nature of the movement.
- *Dilatation/erosion*: in contrast to the previous proposal, the coordinates of the center are unchanged, whereas the radius is reset according to the radius probability model mentioned in III-A.

We define sub-kernel Q_2 as a sequence of small movements and call it "shaking". This particular movement is a combination of translations, and dilatations/erosions, embedded into an RJMCMC-like algorithm and aimed at the minimization of an energy function, which in contrast to Eq. (3), is based only on the attributes size $E_s(c_n)$ and observation $E_l(c_n)$. As such, we are able to achieve locally optimal object positioning in each RJMCMC core block iteration, largely reducing the number of steps required by the solver to converge. Detailed implementation of this movement is given in Algorithm 1.

Algorithm 1 Shaking: Q_2 (Part of Core Block)

Result: Optimal x_n, y_n, r_n of circle n.
$i = 1$;
compute energy:

$$E_{Sh}(c_n) = w_l^{Sh} E_l(c_n) + w_s^{Sh} E_s(c_n)$$

while $i < S_{max}$ **do**
  choose Q': x'_n, y'_n, r'_n with uniform probability;
  compute energy $E'_{Sh}(c_n)$;
  compute $\alpha' = \min(1, \frac{E_{Sh}(c_n)^{\frac{1}{T_{Sh}}}}{E'_{Sh}(c_n)^{\frac{1}{T_{Sh}}}})$;
  if $\alpha' \geq \alpha'_{min}$
  then
    accept Q';
  else
    deny Q';
  $i = i + 1$; decrease temperature T_{Sh};

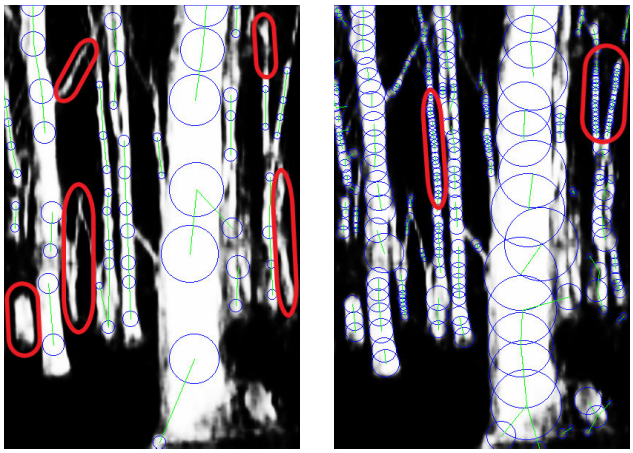
2) RJMCMC CUSTOM SOLVER

Using the previously defined MPP model and RJMCMC core block reveals two significant issues. The first question is related to the number of nodes, the other to the relatively simplified MPP modeling limitations. While the custom solver (the main loop) handles the first problem, the second will be dealt with post-solver steps described in Section III-C.

The massive variety from one image to another in the number, size, and shape of objects to be modeled makes it impossible to have a consistent structural quality using the same number of circles with each input data. As shown in Figure 3, if the number of circles exceeds the needs, generated graph structures may come-out with a significant number of "unnecessary" nodes. This problem becomes even worse if the number of circles is insufficient, ultimately leading to the non-discoverability of many tree-parts. The death-birth mechanism cannot estimate the correct number of circles since if larger areas remain uncovered, it does not result in higher energy states. Thus we aim to place and connect a given number of circles (N) in optimal positions then, after making some refinement steps, we increase N and rerun the RJMCMC core block. The number of such cycles and thus N is limited by a stopping criteria defined below.

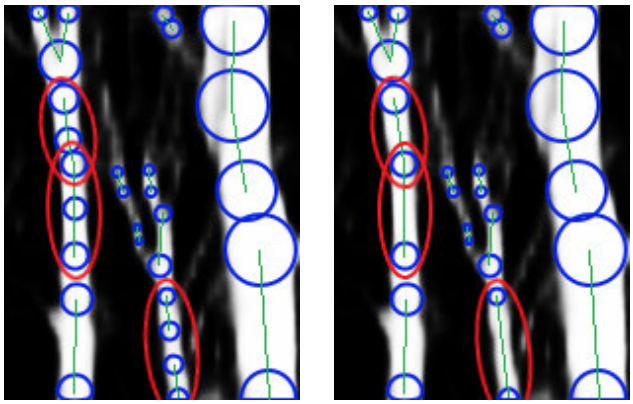
We propose the custom solver (detailed in Algorithm 2) based on the previously defined RJMCMC core block and coupled with the following new definitions:

- *Minimization procedure*: following the execution of the RJMCMC core block, we remove some nodes from the representation to achieve "simplifications" in the graph structures of the modeled objects. At first, we look for regular, linear structures, made of several aligned circles



(a) $N = 100$. (b) $N = 500$.

FIGURE 3. Demonstrating the importance of providing an adequate number of circles: (a) shows several undiscovered tree-parts, whereas (b) exposes excessive number of nodes for the modeling of relatively uniform vertical trunks.



(a) (b)

FIGURE 4. Illustrating the necessity of the minimization procedure: (a) shows tree-parts at the end of run of the RJMCMC core block, (b) is after a structure minimization procedure.

with roughly the same radius, in a connected chain. Such structures have the potential to be simplified by eliminating internal nodes, resulting in a configuration of fewer structural elements (with negligible pixel-wise precision loss). Then we evaluate the possibility of a simplification involving two separate sets of chains of linear structures. A good example is given in Figure 4, where on one tree branch we merge two separate sets of connected components, while on the other trunk we simplify the modeling of two distinct tree-part structures.

- *Locking function* $L(c_n)$: it allows further control over which node may be subject to a movement proposal, and which is "locked" and can only be updated in the minimization procedure.

$$L(n) = \begin{cases} 1, & \text{if the node is "locked"} \\ 0, & \text{otherwise (default value)} \end{cases}$$

The locking of some nodes depends on two "quality conditions":

1) $E_l(n) \leq Th_l$
2) $E_{cn}(n) \leq Th_{cn}$

with Th_l, Th_{cn}, the two "quality thresholds" ensuring that the given node is in good energy condition, locking prevents the core block from further investigations.

- *Stopping criteria*: since the custom solver follows an indefinite loop, a crucial step is needed to decide when to stop the model generation process. Therefore, we decide to enforce the stop if the RJMCMC core block could not find satisfying positions and connections to some nodes: not all nodes could be locked. Finally, we removed the latter from the representation.

Algorithm 2 Custom Solver (Main Loop)

Result: Configuration of N circles
$N = N_{init}$;
run $Q \mid Q = Q_1 + Q_2 \; \forall$ nodes c_n;
while *true* **do**
  run core block;
  run minimization procedure: $N = N_{opt}$;
  $\forall n \in [1, N]$:
  if $E_l(n) \leq Th_l$ *and* $E_{cn}(n) \leq Th_{cn}$ **then**
    $L(n) = 1$;
    add new circles;
    update N;
  else
    remove failing nodes and their connections;
    update N;
    break;

C. POST-SOLVER CONNECTIONS

Due to our initial incline towards the model of tree-like structures with a relatively simple MPP formulation, it becomes clear that such a minimalist approach will not be able to output complete models of trees. Limiting the number of connections per node helps to keep the complexity relatively low, but for many structures, we will most likely end up with fragmented representations of objects that will make the desired results if correctly linked together.
Thus, we define three separate post-solver steps, which are performed after Algorithm 2, and all can add new connections to reduce fragmentation.

1) POST-SOLVER PHASE 1

The proposed joints during this first post-solver phase are all based on the assumption of increased connectivity likelihood between close nodes, where the new additional areas, covered by the connections, have high likelihood. As such, we propose the cost function based on relative distance and "additional connection likelihood" (for illustration see Figure 5), as a reliable benchmark for the evaluation of the fitness of proposals through Algorithm 3 described below:

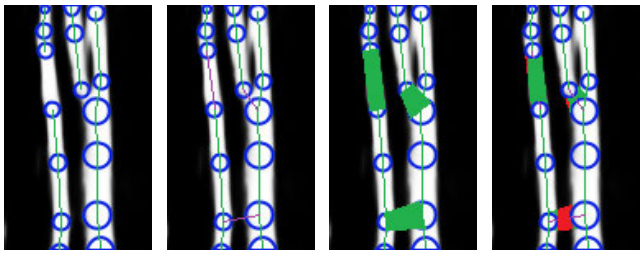


FIGURE 5. Comparing the additional connection probabilities of post-solver phase 1. (a): initial MPP configuration, (b): possible new connections, (c): implied areas highlighted, and (d): one connection rejected (in red) due to its low probability. The area in red in (d) has high cost since it has low probability $E_{l_{add}} = -\log(P_{l_{add}})$ expressing the likelihood of the newly covered area.

Algorithm 3 Post-Solver Phase 1

Result: Updated configuration
for $\forall\ (c_i, c_j),\ i \neq j$ **do**
  Compute cost of possible new connection:

$$C_{Ph1}(c_i, c_j) = w_{rd} \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{r_i + r_j} + w_{l_{add}} E_{l_{add}})$$

  if $C_{Ph1}(c_i, c_j) \leq Th_{Ph1}$ *and connection does not interact with others* **then**
    add connection between (c_i, c_j);

2) POST-SOLVER PHASE 2

Contrary to the first joint proposals, our second phase aims to create links between separated connected components (S_n, S_m). The proposed algorithm systematically investigates possible new connections with low energy and large area coverage. We use a cost function targeted towards the maximization of the area added by new links, by selecting the best join proposal among a given "additional connection likelihood" interval, within a steadily increasing acceptance threshold.

Detailed implementation of this step is given in Algorithm 4.

3) POST-SOLVER PHASE 3

Our final joint proposal, phase 3, is defined similarly to phase 2, with a cost function using the relative distance, absolute differences between the radius of the circles, and angle $\beta \in [0, \pi]$ describing the difference in orientation of the last edge of the connected parts.

As described in detail in Algorithm 5, we do not use data from the input probability map during this step, which in turn, makes our model description resilient to false negative noise (often persistent in CNN's prediction output due to heavy occlusion).

IV. FRAMEWORK OF APPLICATIONS

In order to probe the robustness of our model and ascertain the replicability of the achieved results, we run several tests

Algorithm 4 Post-Solver Phase 2

Result: Updated configuration
$Th_{Ph2} = 0$;
while $Th_{Ph2} \leq Th_{Ph2_{max}}$ **do**
  $C_{Ph2_{min}} = \infty$;
  for $\forall\ (c_i, c_j)\ s.t\ (c_i, c_j) \in (S_n, S_m)$, $n \neq m$, *and their connection does not interact with others* **do**
    compute $E_{l_{add}}(c_i, c_j)$;
    if $E_{l_{add}}(c_i, c_j) \leq Th_{Ph2}$ **then**
      compute cost of possible connection:

$$C_{Ph2}(c_i, c_j) = \frac{1}{A_{add}(c_i, c_j)}$$

      where $A_{add}(c_i, c_j)$ represents the area covered by the additional connection;
      if $C_{Ph2}(c_i, c_j) < C_{Ph2_{min}}$ **then**
        $C_{Ph2_{min}} = C_{Ph2}(c_i, c_j)$;
        memorize connection of (c_i, c_j);
  if $C_{Ph2_{min}} \neq \infty$ **then**
    accept memorized connection of (c_i, c_j);
  else
    $Th_{Ph2} = Th_{Ph2} + Th_{Ph2_{inc}}$;

on three different datasets. Two datasets are of trees in forest environments: first is synthetic, produced through rendering with Autodesk 3DS Max 2020, while the second is real, made from images taken by a forest engineer on various woodland sites. We chose the Massachusetts Roads Dataset as the third dataset to demonstrate our solution's extendability to similarly challenging problems.

The upcoming results are all based on using a probability map generated from DeepLabv3+ [16]. Although, as we mentioned earlier, other probability maps can be used, either from CNN [41] or GMM models.

We use the implementation publicly available at: https://github.com/tensorflow/models/tree/master/research/deeplab on an Ubuntu system with an Nvidia Quadro P6000 GPU and the following off-the-shelf settings:

- Backbone: Xception-65, with pretrained PASCAL VOC 2012 weights
- Atrous rates: 6, 12 and 18
- Output stride: 16
- Decoder output stride: 4

Finally, we ensure the same values on the previously mentioned tunable solver parameters while investigating, within each dataset, the structures of different objects.

The evaluation of our representation is done by the pixel-wise coverage. If the estimated structures well follow the objects and the radius of circles is also appropriately set, we assume that the generated binary masks fit the objects' area. The binary masks are formed by drawing filled trapezoids between two connected circles, defined by the connecting edge and the two perpendicular diagonals giving their bases. (In Section V we will show that this does not

TABLE 1. Statistical results of our MPP modeling on the synthetic trees dataset (with comparison to neural network output). For each of the 5 following benchmarks, the columns from left to right correspond to achieved performance after phase (Ph.) 1, 2, and 3 of the post-solver operations.

	DeepLabv3+ prediction map					MPP model														
	Accuracy	Precision	Recall	Cohen's kappa	F1 score	Accuracy			Precision			Recall			Cohen's kappa			F1 score		
						Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3
Average	95.25%	0.85	0.88	0.83	0.86	94.05%	94.84%	94.81%	0.88	0.86	0.85	0.78	0.84	0.86	0.78	0.82	0.82	0.82	0.85	0.85
Std Dev	1.39%	0.04	0.04	0.04	0.04	1.77%	1.39%	1.42%	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04

TABLE 2. MPP modeling performance on the real forest images dataset (with comparison to CNN prediction outputs). For each of the 5 following benchmarks, the columns from left to right correspond to achieved performance after phase (Ph.) 1, 2, and 3 of the post-solver operations.

	DeepLabv3+ prediction map					MPP model														
	Accuracy	Precision	Recall	Cohen's kappa	F1 score	Accuracy			Precision			Recall			Cohen's kappa			F1 score		
						Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3
Average	92.54%	0.86	0.87	0.81	0.86	90.75%	92.02%	91.93%	0.88	0.87	0.86	0.77	0.83	0.84	0.75	0.79	0.79	0.81	0.84	0.84
Std Dev	1.10%	0.04	0.03	0.02	0.01	0.93%	0.71%	0.92%	0.04	0.04	0.04	0.03	0.03	0.03	0.01	0.01	0.01	0.01	0.01	0.01

Algorithm 5 Post-Solver Phase 3

Result: Updated configuration
while $C_{Ph3_{min}} \leq Th_{Ph3}$ **do**
  $C_{Ph3_{min}} = \infty$;
  for $\forall\ (c_i, c_j)\ s.t\ (c_i, c_j) \in (S_n, S_m)$, $n \neq m$, *and their connection does not interact with others* **do**
    compute cost of possible connection:

$$C_{Ph3}(c_i, c_j) = w_{rd} \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{r_i + r_j} + w_{rad} \mid r_i - r_j \mid + w_{af} \tan(\beta)$$

    with w_{rd}, w_{rad} and w_{af}: coefficients for the relative distance, radius difference and angle flatness respectively;
    if $C_{Ph3}(c_i, c_j) < C_{Ph3_{min}}$ **then**
      $C_{Ph3_{min}} = C_{Ph3}(c_i, c_j)$;
      memorize connection of (c_i, c_j);
  if $C_{Ph3_{min}} \leq Th_{Ph3}$ **then**
    accept memorized connection (c_i, c_j);

always end in precise representation.) We compute accuracy, precision, recall, Cohen's kappa and F1 score to compare the binary masks of the ground truth maps to those generated by our graph models. Please note that the conventional accuracy is not always satisfactory to evaluate the results since often the background covers much larger areas than the trees themselves. Cohen's kappa correctly handles this problem of unbalanced classes.

A. APPLICATION ON THE SYNTHETIC TREES DATASET

In our first, synthetic-image dataset, we investigate the segmentation of a limited number of trees in forest environments. For this, we typically consider one, two, or three objects as foreground with a forest image showing other distant trees, the ground, leaves, and the sky as the background. We use 50 different images, split into a ratio of 70%/10%/20% for training, validation, and testing on DeepLabv3+. The CNN-generated 10 probability maps are run through our modeler to retrieve the graph-like description, including the parts' width (coded by the radius of circles).

As shown in Figure 6 and the statistical data given in Table 1, our method could successfully achieve a high-level representation of the tree-structures.

B. APPLICATION ON THE REAL FOREST IMAGES DATASET

Our second dataset is made of 16 manually annotated images of trees in different forest environments, representing the following species: beech, turkey oak, sessile oak, hornbeam, black alder, and black locust. We use 9 of those images, as basis for the training of the neural network, while the remaining 7 are left for testing.

It is important to mention that despite the apparently limited training sample, each individual image contains at least a dozen trees, bringing the total number of objects in the hundreds. Nevertheless, we extend the training set to 40 images by using transformations like flipping, rotation, and cropping.

A sample image, demonstrating the achieved result, is given in Figure 7 with detailed statistics in Table 2.

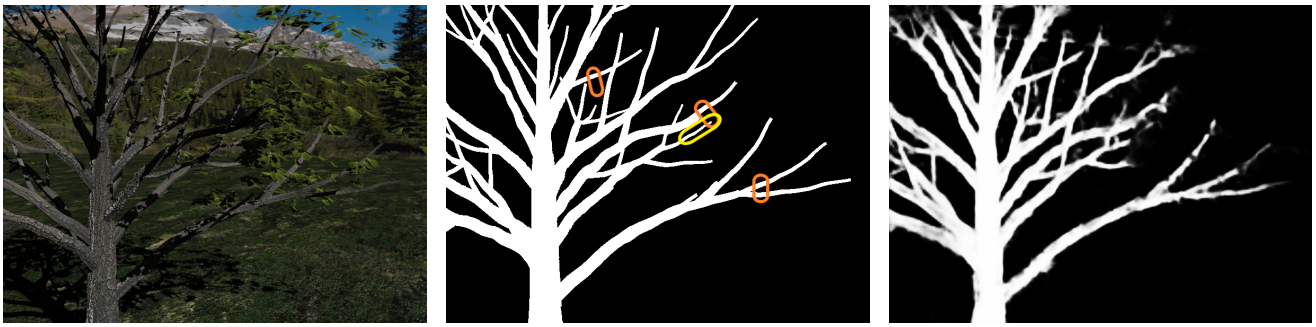
C. APPLICATION ON THE MASSACHUSETTS ROADS DATASET

For the third dataset, we explore the possibility of the generalization of our modeling technique by applying the same concept among other equally challenging computer vision task.

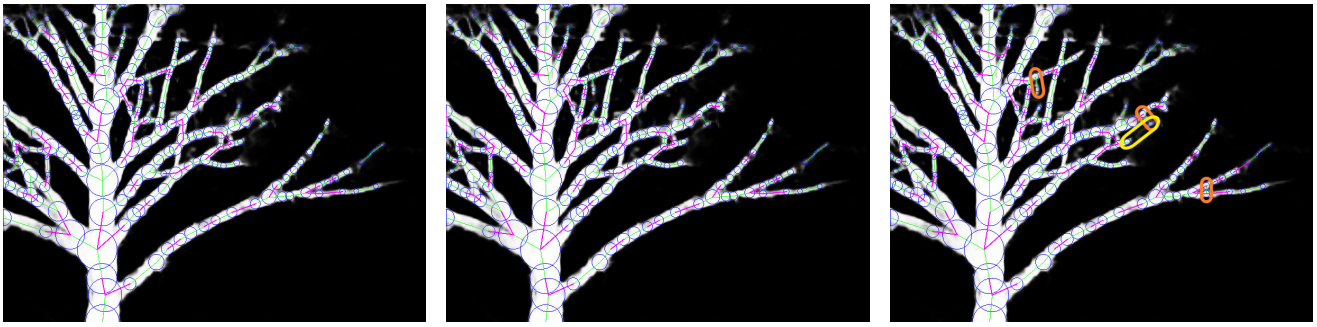
One of such similar problems can be the extraction (segmentation) of roads from satellite images. Thus, we decide to opt for the Massachusetts Roads Dataset, publicly available at https://www.cs.toronto.edu/ vmnih/data/, as a third application to model line-like structures.

We use the same split of data as mentioned by the authors of the link for training, validation, and testing on DeepLabv3+. Then the resulting probability maps are fed to our custom solver to retrieve the structures of different roadmaps.

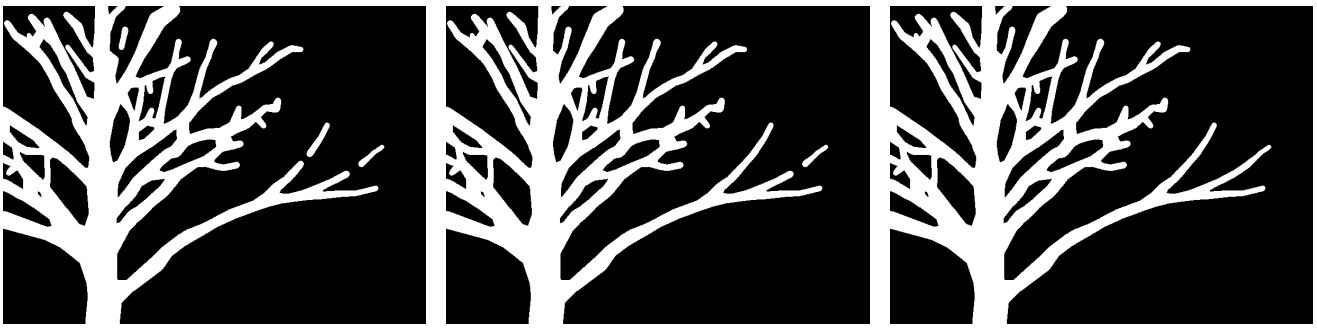
As with tree datasets, our method could now produce structural information of detected road parts, still giving good segmentation results via the trapezoid filling between nodes. Figure 1 shows examples of achieved results on a particularly challenging network of roads, demonstrating the ability of our



(a) Color image. (b) Ground truth. (c) DeepLabv3+ probability map.



(d) From left to right: achieved MPP model of a tree structure, after phase 1, phase 2 and phase 3 respectively.



(e) From left to right: corresponding pixel-wise coverage.

FIGURE 6. Structural modeling of a tree in a forest environment: notice the additional post-solver joints (magenta in (d)), which by successfully reconnecting individual tree-parts, are able to reconstruct the structure of the whole object. Nevertheless, statistical evaluation criteria are typically below the performance achieved by neural network's prediction outputs. The reasons for this are discussed in Section V.

approach to tackling ambiguous situations. Statistical details are given in Table 3.

V. DISCUSSION

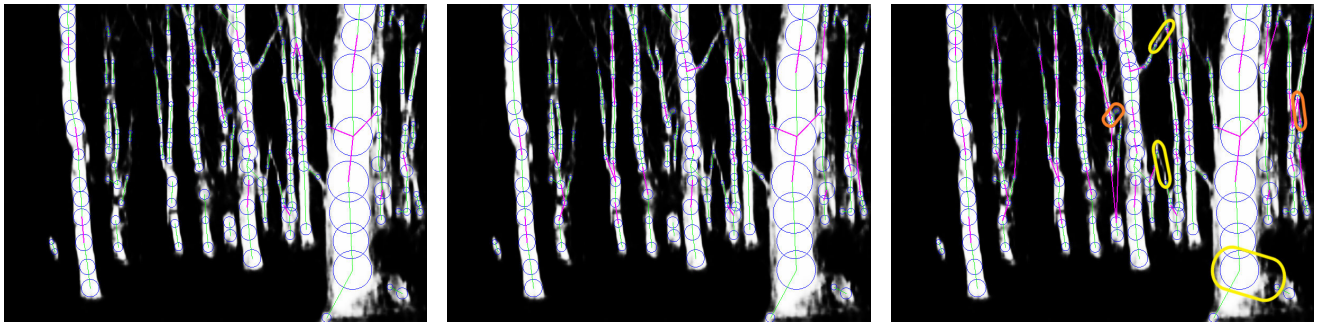
Achieved experimental results suggest that, on each of the previously applied datasets, our stochastic method was able to identify the structures of complex objects coherently. Comparing subfigures c and e in Figure 6 and Figure 7 clearly illustrate that the graph models could nicely represent the thin and thick branches and could connect, in many cases, the loose parts. As Tables 1 and 2 show, the tree models' average accuracy range from 91.93% to 94.81%, while these values for Cohen's kappa and F1 score are 0.79-0.82 and 0.84-0.85 respectively.

While our research objects are trees, we were curious how the algorithm performs on other types of structures specifically roads. A large difference is that roads are much thinner objects and DeepLabv3+, with off-the-shelf settings, tends to underestimate road pixels thus, accuracy is higher than for trees, but class-weighted Cohen's kappa is lower (Table 3). Despite the patchy and less accurate probability maps, our stochastic modeling solution still managed to output reasonable results (even slightly better than the raw data of DeepLabv3+).

One could note that on many occasions, detailed statistical figures of our retrieved segmentation fell slightly short in comparison to their CNN pixel-wise prediction map counterparts. This tendency can be explained by a multitude of factors, of which the four most important are:

- Precision loss due to the usage of regular shapes for the generation of segmentation maps: Object structure modeling is about to find a geometrical representation, that can describe some of the objects' properties while able to reproduce its shape decently. In our modeling of tree

(a) Color image. (b) Ground truth. (c) DeepLabv3+ probability map.



(d) From left to right: MPP model of tree structures, after each of the three post-processing steps.

(e) From left to right: reconstructed segmentation results from the achieved graph models of trees.

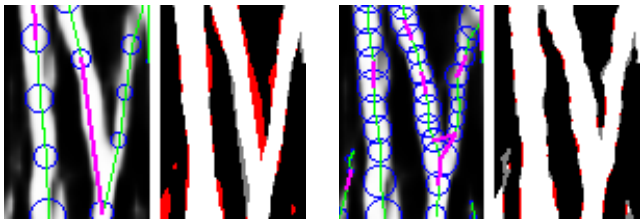
FIGURE 7. Structural modeling of trees in a forest environment: note how the additional, post-solver connections (magenta in (d)), are able to link between several pairs of tree-parts, previously disconnected by low probability map areas. The latter commonly occurs due to heavy occlusion and change in illumination conditions.

TABLE 3. Statistical performance of our MPP solution on the Massachusetts Roads Dataset (and in comparison to initial neural network output). For each of the 5 benchmarks below, the rows from left to right correspond to results after phase (Ph.) 1, 2 and 3 of the post-solver operations.

	DeepLabv3+ prediction map					MPP model														
	Accuracy	Precision	Recall	Cohen's kappa	F1 score	Accuracy			Precision			Recall			Cohen's kappa			F1 score		
						Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3	Ph.1	Ph.2	Ph.3
Average	96.11%	0.65	0.36	0.43	0.45	96.04%	96.09%	96.13%	0.68	0.66	0.65	0.27	0.32	0.34	0.39	0.41	0.44	0.41	0.42	0.45
Std Dev	2.27%	0.07	0.10	0.09	0.09	2.33%	2.31%	2.27%	0.07	0.08	0.08	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10

structures, we assume that individual parts' shapes are somewhat similar to 2D projections of standard cylinders. However, for most case scenarios with real-life trees, the irregular and typically non-linear variation among the width of trunks and branches leads to a state of a trade-off between the accuracy of the shape of the built models and the number of structural elements involved. Figure 8 graphically demonstrates this issue, while Figure 9 accounts for pixels lost due to this particular problem.

- Missing connections: Our formulation of the MPP model and the post-processing additional joints steps can more easily reflect star models than object structures with many loops. In both post-solver phase 2 and phase 3, we evaluate and attempt to set joints between two different sets of connected components. Furthermore, while this reconstruction of graph model structures performs well with disjoint and unoccluded tree-like objects, it becomes less optimal when those conditions are not satisfied, such as the case of



(a) $N = 15$. (b) $N = 40$.

FIGURE 8. 2 configurations with different number of circles, and their respective pixel-wise coverage comparison to the ground truth (grey standing for false positives and red for false negatives).

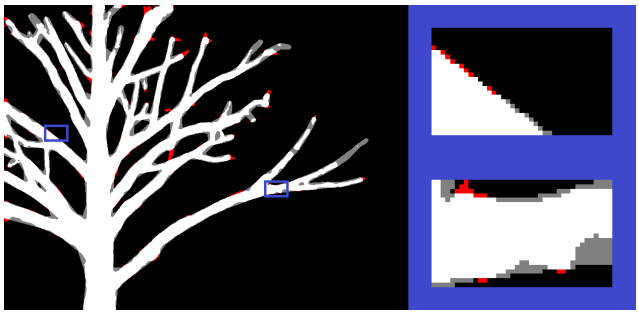


FIGURE 9. Comparing the pixel-wise coverage of a previously given MPP modeling sample to its respective DeepLabv3+ prediction map output: grey pixels represent false positives while red pixels are for false negatives. Notice the framed (in blue) and enlarged parts, incorporating precision loss pixels, and how they're easily distinguishable by a thin layer formed around external boundaries of modeled objects.

crisscrossing and inter-connecting tree branches (see highlights in yellow, in Figures 6 and 7). On the other hand, dropping this high-level constraint to connect nodes from two separate parts and opting to investigate additional connections among any two nodes cannot lead to good, yet consistent results, regardless of any combination of settings for the cost functions involved. An illustration is given in Figure 10, where we removed the "belonging to different sets" condition on previously achieved MPP results.

- False positives: Despite our tuning of different coefficients and parameters to better fit the specificity of each dataset (see Table 4 for these settings), false positives do still exist, as shown in orange in Figures 6 and 7.
- Performance of phase 3: The reader might question the necessity of phase 3, since in some cases (Table 1 and Table 2) accuracy decreases slightly (0.1%) compared to phase 2. Phase 3 adds new connections between object segments. Creating these new connections is relatively simple, does not consider likelihood; only spatial properties are considered as given in Algorithm 5. Also, consider that adding these new edges does not mean adding new nodes, i.e., the shape of branches are not precisely followed. All these facts increase the recall value but not necessarily increase precision. Recall and precision values of Tables 1-3 support this reasoning.

As per the definition and implementation details of Section III, and the previous explanations, our modeling

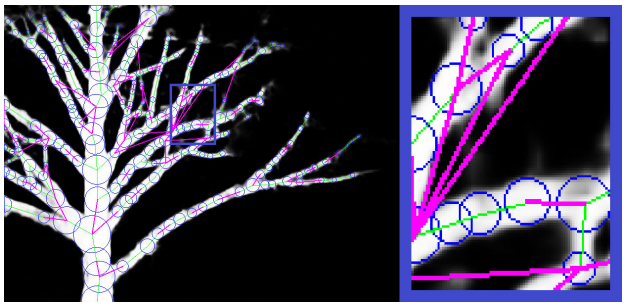


FIGURE 10. Resulting graph model structure, when both phase 2 and phase 3 were ran without consideration to the constraint on nodes, being part of different sets of connected components.

technique is not about the finding of an optimal segmentation for a fixed number of structural elements involved. Instead, the latter is an output, a consequence of the choice of a certain "quality" level, while describing the shapes of represented objects.

Nevertheless, the previously wrong assumption can still be used to illustrate the complexity of the encountered problem. As a simple example, considering an image of VGA size and 200, non-overlapping, one-pixel size "circles", the total number of all possible configurations accounting in this particular case is:

$$n_{cfs} = \frac{(640 \cdot 480)!}{((640 \cdot 480) - 200)!} \approx 2.85 \cdot 10^{1097} \tag{6}$$

At this point, even by using a fictionally efficient system, able to perform our structural modeling solution at a 5 GHz speed while investigating one configuration per clock cycle, the amount of time needed to check all of the configurations involved would therefore be:

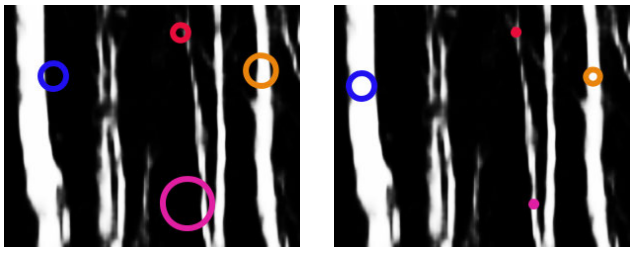
$$t_{cfs} = \frac{n_{cfs}}{(5 \cdot 10^9) \cdot (3.154 \cdot 10^7)} \approx 1.81 \cdot 10^{1080} \text{ years!} \tag{7}$$

Unfortunately, the problem gets even more significant with each of previously disclosed modeling options, as images are of a size of up to 800 × 600 pixels, circles have a varying radius, and possible connections and overlapping are permissible to a certain point.

As a result, we ditch the quest of a global optimum solution in favor of a local minimum, reaching a satisfactory model configuration in only a matter of few hours, with the exact processing time depending upon the number of iterations of the RJMCMC core block, the settings of the minimization procedure, and the quality thresholds.

While the introduction of the shaking movement helped the positioning ability, behind the drastic increase in convergence speed, led to a huge reduction in the search space of possible configurations. As demonstrated in Figure 11, the Q_2 part of the movement proposals acts within a given probability map area by constraining initially proposed circles' attributes. In turn, it infers robustness to our modeling approach by raising its tolerance toward a reasonable variation of the various parameters involved.

To monitor the sensitivity of different parameter settings, we introduced a 20% standard deviation Gaussian noise



(a) Before shaking. (b) After shaking.

FIGURE 11. Illustration of the inner working of shaking movements: 4 different circles are drawn before and after their second movement proposal part.

TABLE 4. Listing of the main RJMCMC custom solver settings, defined in Eq. 5 in Section III, and used in experiments for optimal experimental results.

Datasets	Th_{no}	w_l	w_s	w_{cn}	w_{dcn}	w_{dn}	w_{rs}	Core block iterations
Synth. trees dataset	0.4	7	2	4	0.5	0.1	0.2	200.000
Real forest dataset	0.2	6	1.5	4	1	0.2	0.3	300.000
Mass. R. Dataset	0.3	7	3	5	0.5	0.2	0.2	500.000

function, on each of the major RJMCMC custom solver parameters (listed in Table 4), while we reran our MPP structural modeling solution on the synthetic trees dataset, for another 100 sample results. The statistical performance was very similar, illustrated by a standard deviation of 0.14% for different pixel-wise accuracy and as low as 0.005 for both Cohen's kappa and F1 score.

VI. CONCLUSION

Semantic segmentation or pose estimation by DNNs can inherently code the inner structure of objects, but obtaining a high-level explicit representation requires model validation. For many types of objects (e.g. trees), the validation can be difficult due to cluttered background, various lighting effects, size changes, and branches' complicated structures. We presented a framework suitable for the detection, structural representation, and segmentation of curvilinear or piecewise linear objects. The proposed pipeline starts with a probability estimation of object locations then applies stochastic optimization via MPP, RJMCMC dynamics, and post-processing steps to create the graph representations. The model's advantages are that it can handle a mixture of thin and thick flexible curvilinear objects and connect separate parts. Besides introducing the technique's details as possible applications of the model, we analyzed its performance on two tree datasets, and a roadmap dataset. We found that the generated graph-like structural representations can also reproduce the ground truth pixel-wise segments with high fidelity. There are limitations of our study. First, there are always improved CNN architectures, so we could try some, aimed toward the specific segmentation of curvilinear and line-like structures exhibiting minimal output deterioration for when it comes to thin object parts (see RoadNet in [42]). Second, the possible extension of the real tree images dataset would support the analysis of robustness, but now, it was beyond our potential (instead we used the road map dataset). Third, the number of nodes is automatically determined. It would be reasonable to include a mechanism to balance between accuracy and complexity of graph-description by explicitly specifying the number of nodes.

In the future, we plan to extend our technique to introduce the ideas of [21] to handle the parts' crossing. Further improvement of our modeling solution could be the usage of two probability maps: one for the pixel-wise estimation of tree-parts, while the other would reflect locations specifically belonging to graph nodes. A base structure for such node detection could be a DNN similar to [43].

REFERENCES

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C Van Esesn, A. A S. Awwal, and V. K. Asari, "The history began from AlexNet: A comprehensive survey on deep learning approaches," Mar. 2018, *arXiv:1803.01164*. [Online]. Available: http://arxiv.org/abs/1803.01164

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop Stat. Learn. Comput. Vis. (ECCV)*, Prague, Czech Republic, May 2004, pp. 59–74.

[3] M. Jones and P. Viola, "Fast multi-view face detection," Mitsubishi Electr. Res. Lab, Cambridge, MA, USA, Tech. Rep. TR-20003-96, Jul. 2003, vol. 3, no. 14, p. 2.

[4] P. Sinha, "Object recognition via image invariance a case study," *Invest. Ophthalmol. Vis. Sci.*, vol. 35, no. 4, pp. 1735–1740, Mar. 1994.

[5] J. Tian, G. Wang, J. Liu, and Y. Xia, "Chinese license plate character segmentation using multiscale template matching," *J. Electron. Imag.*, vol. 25, no. 5, Sep. 2016, Art. no. 053005.

[6] M. Lébl, F. Šroubek, J. Kautský, and J. Flusser, "Blur invariant template matching using projection onto convex sets," in *Proc. 18th Int. Conf. Comput. Anal. Images Patterns (CAIP)*, Salerno, Italy, Sep. 2019, pp. 351–362.

[7] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. COM-22, no. 1, pp. 67–92, Jan. 1973.

[8] L. Shams and J. Spoelstra, "Learning Gabor-based features for face detection," in *Proc. World Congr. Neural Netw. Int. Neural Netw. Soc. (WCNN)*, San Diego, CA, USA, Sep. 1996, pp. 15–20.

[9] K. C. Yow and R. Cipolla, "Feature-based human face detection," *Image Vis. Comput.*, vol. 15, no. 9, pp. 713–735, Sep. 1997.

[10] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, Apr. 2001.

[11] L. Czuni and K. B. Alaya, "Low- and high-level methods for tree segmentation," in *Proc. 10th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, Metz, France, Sep. 2019, pp. 189–192.

[12] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," Sep. 2018, *arXiv:1809.10790*. [Online]. Available: http://arxiv.org/abs/1809.10790

[13] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate O (*n*) solution to the P*n*P problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.

[14] X. Descombes and J. Zerubia, "Marked point process in image analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 5, pp. 77–84, Sep. 2002.

[15] P. J. Green, "Reversible jump MCMC computation and Bayesian model determination," *Biometrika*, vol. 57, pp. 97–109, Dec. 1995.

[16] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. 15th Eur. Conf. Comput. Vision (ECCV)*, Munich, Germany, Sep. 2018, pp. 801–818.

[17] G. Perrin, X. Descombes, and J. Zerubia, "A marked point process model for tree crown extraction in plantations," in *Proc. IEEE Int. Conf. Image Process.*, Genoa, Italy, Sep. 2005, pp. 661–664.

[18] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Med. Image Anal.*, vol. 2, no. 2, pp. 143–168, Jun. 1998.

[19] M. W. Law and A. C. Chung, "Three dimensional curvilinear structure detection using optimally oriented flux," in *Proc. 10th Eur. Conf. Comput. Vision (ECCV)*, Marseille, France, Oct. 2008, pp. 368–382.
[20] P. Glowacki, M. A. Pinheiro, A. Mosinska, E. Turetken, D. Lebrecht, R. Sznitman, A. Holtmaat, J. Kybic, and P. Fua, "Reconstructing evolving tree structures in time lapse sequences by enforcing time-consistency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 755–761, Mar. 2018.
[21] J. Xie, Y. Zhao, Y. Liu, P. Su, Y. Zhao, J. Cheng, Y. Zheng, and J. Liu, "Topology reconstruction of tree-like structure in images via structural similarity measure and dominant set clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 8505–8513.
[22] F. De Graeve, E. Debreuve, S. Rahmoun, S. Ecsedi, A. Bahri, A. Hubstenberger, X. Descombes, and F. Besse, "Detecting and quantifying stress granules in tissues of multicellular organisms with the *Obj.MPP* analysis tooll," *Traffic*, vol. 20, no. 9, pp. 697–711, Sep. 2019.
[23] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, USA, Jun. 2003, pp. 264–271.
[24] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 259–289, May 2008.
[25] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, Jan. 2005.
[26] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 10–17.
[27] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 710–715.
[28] G. Carneiro and D. Lowe, "Sparse flexible models of local features," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, Graz, Austria, May 2006, pp. 29–43.
[29] L. Donati, S. Cesano, and A. Prati, "An accurate system for fashion hand-drawn sketches vectorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 2280–2286.
[30] G. Celeux, F. Forbes, and N. Peyrard, "EM procedures using mean field-like approximations for Markov model-based image segmentation," *Pattern Recognit.*, vol. 36, no. 1, pp. 131–144, Jan. 2003.
[31] T. Lavergne and F. Yvon, "Learning the structure of variable-order CRFs: A finite-state perspective," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 433–439.
[32] X. Descombes, R. Stoica, and J. Zerubia, "Two Markov point processes for simulating line networks," in *Proc. Int. Conf. Image Process.*, Kobe, Japan, Oct. 1999, pp. 36–40.
[33] A. Börcs and C. Benedek, "A marked point process model for vehicle detection in aerial LiDAR point clouds," in *Proc. Ann. Photogramm., Remote Sens. Spat. Inf. Sci. (ISPRS)*, Melbourne, VIC, Australia, Aug./Sep. 2012, pp. 93–98.
[34] C. Benedek, "An embedded marked point process framework for three-level object population analysis," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4430–4445, Sep. 2017.
[35] T. Li, M. Comer, and J. Zerubia, "A connected-tube MPP model for object detection with application to materials and remotely-sensed images," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 1323–1327.
[36] T. Li, C. G. Aguilar, R. F. Agyei, I. A. Hanhan, M. D. Sangid, and M. L. Comer, "Connected-tube MPP model for unsupervised 3D fiber detection," *Electron. Imag.*, vol. 2020, no. 14, pp. 1–305, Jan. 2020.
[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[38] Z. Kato and J. Zerubia, "Markov random fields in image segmentation," *Found. Trends Signal Process.*, vol. 5, nos. 1–2, pp. 1–155, 2012.
[39] M. Pincus, "Letter to the editor—A Monte Carlo method for the approximate solution of certain types of constrained optimization problems," *Oper. Res.*, vol. 18, no. 6, pp. 1225–1228, Dec. 1970.
[40] Z. Kato, J. Zerubia, and M. Berthod, "Satellite image classification using a modified metropolis dynamics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, San Francisco, CA, USA, Mar. 1992, pp. 573–576.
[41] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *Int. J. Multimedia Inf. Retr.*, vol. 7, no. 2, pp. 87–93, Jun. 2018.
[42] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, and Y. Liu, "RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2043–2056, Apr. 2019.
[43] L. Ke, H. Qi, M.-C. Chang, and S. Lyu, "Multi-scale supervised network for human pose estimation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Munich, Germany, Oct. 2018, pp. 713–728.

KARIM BEN ALAYA received the Dipl.Ing. degree in industrial engineering from the National Institute of Applied Science and Technology, in 2016, and the M.Sc. degree from the National Engineering School of Tunis, in 2018. He is currently pursuing the Ph.D. degree in information science with the University of Pannonia. His research interests include image segmentation, scene understanding, and stochastic object modeling.

LÁSZLÓ CZÚNI received the Ph.D. degree in computer science from the University of Pannonia, Hungary, in 2001. He is currently an Associate Professor with the University of Pannonia. His research interests include image-based recognition, object detection, visual inspection, and applying those in engineering problems.