A Hidden Markov Model For Modeling and Extracting Vine Structure in Images

Ricardo D. C. Marin
Department of Computer Science
University Of Canterbury
Canterbury, Christchurch
ricardo.castanedamarin@pg.canterbury.ac.nz

Richard D. Green
Department of Computer Science
University Of Canterbury
Canterbury, Christchurch

***Abstract*—In this paper we propose a Hidden Markov Model for modeling and extracting vine structure from images. We built up from previous research to infer connectivity of cane segments extracted from binary images. We use skeletonisation and polylines to model cane segments and we use simulated annealing to optimize an energy function defined in terms of attributes observed for each connection. We formulate our proposed solution in the context of MAP inference which is a state-of-the-art framework for inference in computer vision. We show comparative results of our method against state-of-the-art methods used for the same tasks, and our model generalizes and improves precision over prior research.**

I. Introduction

In this paper we are interested in modeling and extracting vine canes from 2D images. See Figure 1 for a description of the parts that compose a vine. The aim of extracting canes in images is to aid a robot arm in the process of automatic pruning that uses vision information. More specifically, canes recovered in images from multiple perspectives can be used to rebuilt a 3D model of the vine and help the robot to move in space and decide which canes to cut [1].

The main contributions of this paper are firstly a novel model of vine structure using a Hidden Markov Random Field – HMRF, and secondly results of the extraction of vine structure from images using Maximum A-Posteriori – MAP inference.

Extracting canes from vine images is a hard problem given the many occluded regions and cane's overlapping that are present (see Figure 1). To overcome these issues, in this paper we subdivide canes into several cane segments that can be observed in images. Then we propose to use a probabilistic approach to perform inference of connectivity between these cane segments given evidence attributes such as angle, dislocation and thickness difference. The inference problem is formulated in the context of HMRF and MAP estimation, which are highly used in vision applications [2], [3]. Therefore, our solution can be seen as the application of state-of-the-art inference methods in computer vision to the problem of vine structure extraction.

This paper is structured as follows. Section II reviews the background theory relevant to our method, in particular Markov Models and MAP inference are introduced. Next, in Section III-B we propose a vine model using a HMRF. Here we formulate the vine structure extraction as an energy minimization problem, which is then solve using simulated annealing in Section IV. We show results of applying our method in comparison to bottom-up parsing [1] and Gibbs sampling [4] in Section V. Finally, Section VI provide a summary of our research and directions on how to extend it in the future.

978-1-5090-0357-0/15/$31.00 2015 IEEE

II. Related Research

A. Tree Structure Modeling

Tree structures modeling can be either purely topological or a full description of the growth of the plant. For example, in a Functional-Structural Plant Model-FSPM defined by Godin and Sinoquet [5], plant structure is able to simulate how the plant evolves over time taking into account physic-chemical processes involved in the growth of the plant and its interaction with the environment [6]. In our system however, the input vine images are considered to be always at a stage of pruning, and therefore there is no need to model evolution over time.

Therefore, our model for vine structure is pure topological, similar to branching structures, tree-graphs and axial trees [7]. In particular our HMRF model is similar to the method of Chen and Neubert [8]. Here, tree structure is modeled using a Markov Random Field–MRF, and the goal is to synthesize realistic trees from a sketch. However, the Markov graph is assumed to be a tree, since it is modeled from human input strokes. Also, the graph is defined in a direct mapping of graph nodes to branch points and graph edges to branch segments. Furthermore, differently to my method, the random variable defined at nodes is the depth of the end tips of branch segments, in order for the structure of the tree to be in three dimensions. Therefore, our methods and those of Chen and Neubert [8], though use the same theory of Markov Models, differ in both goal and implementation.

In summary, my approach can be regarded as a non-functional structure model of a vine, similar to axial trees [7], and such that it uses a MRF model similar to Chen and Neubert [8] to infer the best tree structure for an input vine image.

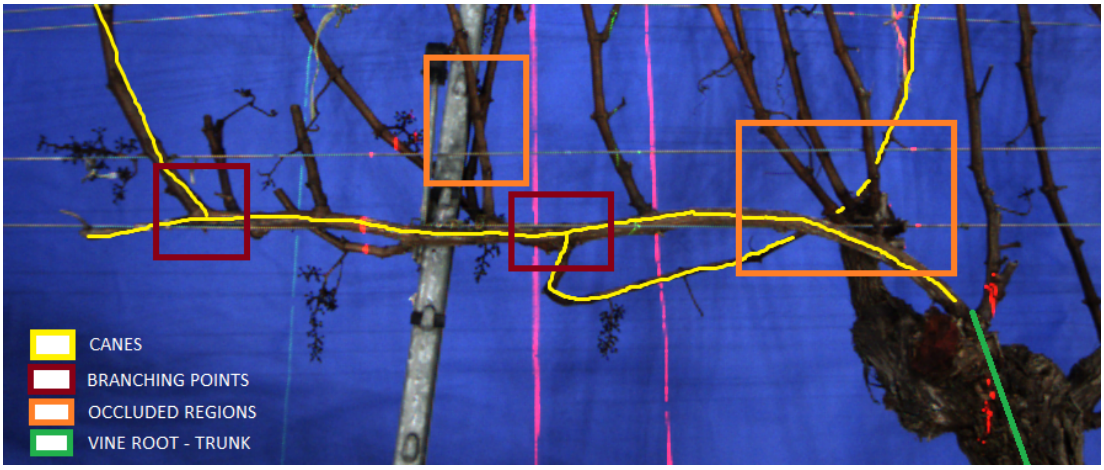

Fig. 1. Vine nomenclature: Examples of canes are shown in yellow. A cane grows from the vine root, shown in green, or from other canes at branching points, shown in red. Some cane parts may be occluded by other canes and/or scene components. Examples of regions with occlusions and overlapping of canes are shown in orange.

B. Vine Structure Extraction From Images

Our research can be regarded as further development of the method of Marin et al. [4] and Botterill et al. [1]. In Marin et al. [4], the author used a similar approach to infer vine structure using Gibbs sampling and MAP estimation. Our methods can be regarded as a generalization of this, where any Monte Carlo method can be used, as well as any energy minimization technique. On the other hand, Botterill et al. [1] used a purely bottom-up approach to extract full canes using edge segments and machine learning. This method, though use well trained classifiers for deciding segments connections at any level of the hierarchy, is heuristic in the sense that there is definition of vine structure, and quantitative results are yet to be addressed.

III. HMRF Model For Vine Structure

Our HMRF vine model is defined on connections between cane segments. Therefore, in the following subsections we first discuss how to model and extract cane segments and then we proceed to formulate our HMRF model.

A. Cane Segments Extraction

Finding full canes in vine images is hard to do directly, given the complexity of overlapping and occluded regions. Therefore, in our approach we further divide each cane into parts we call *cane segments*, so full canes can be reconstructed by solving connections between them. Examples of cane segments are shown in Figure 2 and Figure 4. A cane segment is modeled using polylines with a value of cane thickness. The start and end points of a cane segment polyline may be referring to a starting point of a cane, a branching point, a end point of a cane, or a point where occlusion/overlapping starts or ends.

To find a set of cane segments for an input vine image we used skeletonisation in a way similar to [9], [10], [11]. Here, best-first-search or Djikstra's algorithm is used to build skeleton polylines as paths of minimal distance to a reference point. However, in our method instead of providing a single reference point, we iterate using depth-first-search from any non-visited skeleton pixel, and so we get different skeleton curves starting in different places. After this, we split each curve further at junction skeleton pixels– points with more than 2 skeleton neighbors. The output of this procedure is a set of skeleton polylines, and each of them is a list of connected skeleton pixels. Furthermore, we simplified all polylines by using the Ramer-Douglas-Peucker algorithm with parameter $\epsilon = 1.0$. This allowed us to get fewer control points for representing the original polyline and keep its shape unchanged [12]. Finally, each simplified polyline is a cane segment in our system.

The process of finding cane segments works independent of the skeletonisation method used. There are not assumptions about whether the skeleton is one-pixel-wide. However, accuracy with respect to the true medial axis of the vine binary map is desired, since the skeleton points will be used to build up the cane segments. We used the Zhang-Suen thinning algorithm [13], which is easy to implement, gives the accuracy needed and is fast enough for our system.

B. HMRF *Vine Structure Model*

In general, a HMRF model in the context of computer vision can be designed using the following ideas [2]:

1) Build a Markov Graph. Decompose a target image into a graph where nodes are pixels or group of pixels, and edges are relations between them.
2) Define a *hidden* random variable x_i at each node. This corresponds to the information one is trying to infer, so it is not observable directly.
3) Define an *evidence* random variable z_i at each node. They correspond to measurements one can make at each node (e.g. color information at a pixel).
4) Build a joint probabilistic model for both random variables x_i and z_i defined in the two previous steps.

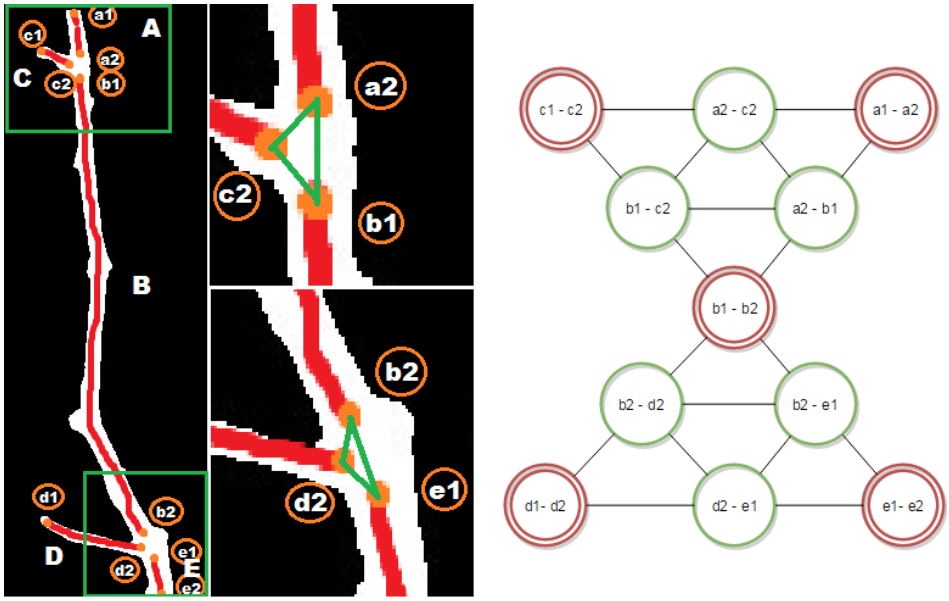

Fig. 2. Markov graph of the vine structure model. On the left, cane segments are shown in red, with end points shown in orange. Connection candidates between the orange points are shown as green lines. On the right, a node in the Markov graph correspond to a connection between either points in the same cane segment (double red circles), or points of different cane segments (single green circle). Edges in this graph are set on nodes that share a point in common (e.g. $e_1 - e_2$ is connected to $b_2 - e_1$ because they share the point e_1).

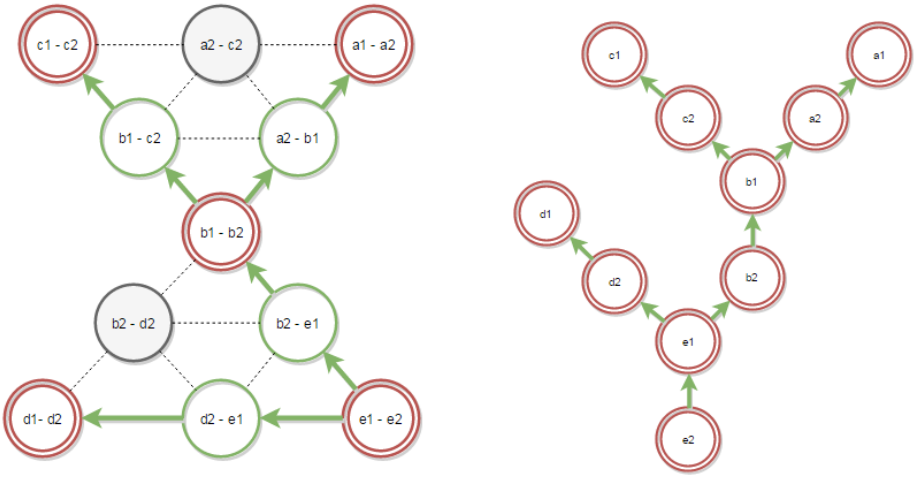

Fig. 3. Inferred structure for the vine in (a). Here I have two different visualizations of the structure graph. The *flow* of connections of the vine is shown as green arrows. On the left, the flow of the vine growing is implicit, since one cannot see directly if the node $e_1 - e_2$ is oriented as $e_1 \to e_2$ or $e_2 \to e_1$. On the right the flow is explicit, by taking the visualization on the left and factoring out the green nodes and expanding each red node into single points.

In the following we address each of these items separately.

*1) **Markov Graph For Vine Structure**:* The Markov graph of our model is summarized in Figure 2. Here cane segments are shown in red and their end points are shown in orange. The Markov graph is build by using as nodes *connections*. A connection is either a link between the two end points of the same cane segment (nodes in red), or a link between two end points of different cane segments (nodes in green). The notation for a connection between end points p and q is (p, q) and is symmetric in the sense that $(p, q) = (q, p)$. Edges in our Markov graph are built from connections that share one end point in common.

On the other hand, Figure 3 illustrates how the vine structure graph looks like after inference. The vine structure graph is different from the Markov graph. It is a tree, showing how end points of the cane segments are connected and flowing from one cane segment to another.

The procedure to build our Markov graph, given the set of skeleton cane segments is straight forward. The only thing that still needs to be specified is how to extract the set of connections between different cane segments, i.e, the green nodes in Figure 2. A brute force approach is to consider as connections for a point, all points of other cane segments. However, I can simplify the number of connections, by considering for each

end point, all end points of other cane segments that are at a fixed distance of D pixels.

*2) **Connection States– Hidden Variable $\boldsymbol{x}$:*** To define the hidden variables x_i associated to each node l_i in our Markov graph, we used a binary random variable:

$$x_i = x(l_i) = \begin{cases} 1 & \text{if connection } l_i \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

With this, instances of $\boldsymbol{x} = [x_1, x_2, ..., x_m]$ describe exactly which cane segments should be connected and which should not. This means different vine structure graphs can be achieved by shifting the values of x_i from connected (one) to unconnected (zero). We will call the hidden variable $\boldsymbol{x}$ a *configuration* of connections of all cane segments. Note that the number of possible configurations for m connections is 2^m. Since in average we have that $m > 100$, you can see why brute force search for the best configuration is not feasible.

*3) **Attributes for Connections – Evidence Variable $\boldsymbol{z}$:*** To define the evidence variables z_i associated to each node l_i in our Markov graph, we used a set of attributes $z_i = [\theta_i, w_i, d_i]$, and then $\boldsymbol{z} = [z_1, ..., z_m]$. The attributes are angle θ_i, thickness difference w_i, and separation d_i between the points of connection l_i.

To define the angle θ_i of a connection candidate l_i, we assigned to each end point of l_i a direction and used the angle between these directions. The direction at an end point of a polyline is computed as the average of directions between consecutive points in the polyline. The average is taken from an end point of the polyline until the end point in question.

Subsequently, the thickness of a polyline is computed as the distance between the two edge contours that are assigned to the polyline. In this way, the thickness difference w_i between the two polylines of the points of l_i, is simply the difference between their thickness values. To compute the edge contours assigned to the polyline, we iterate over all points of the polyline, and for each point, we selected the two closest edge points that are in different sides of the polyline curve.

Finally, to define separation d_i of the end points of connection l_i, we computed the distance of one of the points to the line in the direction of the other point, and viceversa. We then defined d_i as the average of both distances. In this way, separation can be thought as well as a measure of dislocation between the center lines of the curves in question.

*4) **Probabilistic Model For $\boldsymbol{x}$ and $\boldsymbol{z}$:*** Here we are interested in modeling the posterior $P(\boldsymbol{x}|\boldsymbol{z})$ with the purpose of using MAP inference, and solve for the most likely state of connections between cane segments $\boldsymbol{x}^*$ given the observed set of attributes $\boldsymbol{z}$:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} P(\boldsymbol{x}|\boldsymbol{z}) \quad (2)$$

To model the posterior $P(\boldsymbol{x}|\boldsymbol{z})$ we make use of our Markov graph. First observe that cane segments connectivity decisions are dependent on each other. Whether a cane segment is connected to another should be dependent on the state of other connections made for the cane. This motivated us into using a HMRF for vine structure inference, since in this model we can catch implicitly the long-distance relations between x_i from the explicit local adjacency relations in the Markov graph [2]. Furthermore, assuming all x_i satisfy the local Markov property of being conditional independent of all others given its neighbors in the graph, the Hammersley-Clifford theorem [14] allows to write

$$P(\boldsymbol{x}|\boldsymbol{z}) = \frac{1}{Z(\boldsymbol{z})} e^{-E(\boldsymbol{x},\boldsymbol{z})} \quad (3)$$

where $Z(\boldsymbol{z}) = \sum_{\boldsymbol{x}} e^{-E(\boldsymbol{x},\boldsymbol{z})}$ is the normalization constant that makes $P(\boldsymbol{x}|\boldsymbol{z})$ a valid distribution[1]; and E is the energy function of the state $\boldsymbol{x}$ under observations $\boldsymbol{z}$. This representation of the posterior in Equation 3 is very convenient for the inference problem in Equation 2, because it translates the maximization of $P(\boldsymbol{x}|\boldsymbol{z})$ into the minimization of the energy function $E(\boldsymbol{x}, \boldsymbol{z})$. Furthermore, since $Z(\boldsymbol{z})$ does not depend on $\boldsymbol{x}$, it does not need to be included during optimization [2].

Therefore, instead of modeling the posterior $P(\boldsymbol{x}|\boldsymbol{z})$, our aim is to model the energy function $E(\boldsymbol{x}, \boldsymbol{z})$. In many vision problems, the energy function can be expressed as a sum of *likelihood* and *prior* terms [2], [3], [15]:

$$E(\boldsymbol{x}, \boldsymbol{z}) = \sum_i \Phi_i(x_i, z_i) + \sum_{c \in \mathcal{C}} \Psi_c(\boldsymbol{x}_c) \quad (4)$$

where $\mathcal{C}$ is the set of *maximal cliques* in the underlying Markov network— defined as a set of subgraphs of the Markov network that are fully connected, such that adding any other node to a subgraph will spoil its fully connectedness [15]. This expression is useful when modeling a problem using a HMRF, since it divides the energy into two terms that can be understood intuitively [2]. First a term that tell us how consistent/likely the current state variables x_i are according to the observations z_i— the unary potentials $\Phi_i(x_i, z_i)$; and second, a term that encapsulates the prior information one has about plausible states— the clique potentials $\Psi_c(\boldsymbol{x}_c)$. These potentials are defined for our vine model in the following.

*– **Likelihood Potentials** Φ_i:* The likelihood potentials Φ_i are defined from the probability of connections given the observations $P(x_i|z_i)$:

$$\Phi_i(x_i, z_i) = x_i(1 - P(x_i|z_i)) + (1 - x_i)P(x_i|z_i) \quad (5)$$

Observe that each potential Φ_i can be understood as a penalty of a connection that was not made. The higher the probability of connection the less energy it will add when the connection $x_i = 1$ is set in the configuration. If the connection is not set, i.e $x_i = 0$, then it will add high energy values, unless the probability of connection is small. This will guide energy optimization techniques to look into configurations with high probabilities of connections and keep unconnected those with low probabilities. To model the conditional probabilities $P(x_i|z_i)$ we use Bayes rule and the likelihoods $P(z_i|x_i)$, similar to [16]:

$$P(x_i|z_i) = \frac{P(x_i)P(z_i|x_i)}{P(x_i)P(z_i|x_i) + (1 - P(x_i))P(z_i|\neg x_i)}. \quad (6)$$

[1] $Z(\boldsymbol{z})$ is also known as the *partition function* [3].

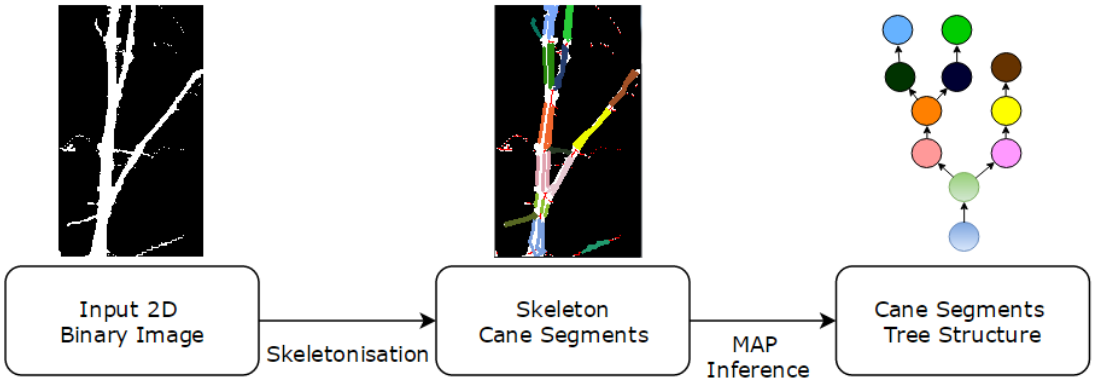

Fig. 4. Pipeline of the vine structure extraction system. Different colors represent different cane segments. We start with an input 2D binary image and build skeleton cane segments from the skeleton of the vine. Then we infer a tree structure using MAP inference over the connectivity of the cane segments.

In turn, to model the likelihoods $P(z_i|x_i)$ we used two normal distributions for the binary values of x_i of connected ($x_i = 1$) and unconnected ($x_i = 0$) connection states:

$$P(z_i|x_i) = [\mathcal{N}_0(z_i; \mu_0, \Sigma_0)]^{1-x_i}[\mathcal{N}_1(z_i; \mu_1, \Sigma_1)]^{x_i} \quad (7)$$

These are commonly known as *class conditional density functions* [17], since the density $\mathcal{N}_v(z_i; \mu_v, \Sigma_v)$ of each class v is modeled separately by a respective distribution, which is a normal in this case. The parameters of mean μ_v and covariance matrices Σ_v for each class can be learned in a supervised way from sets of connected ($v = 1$) and unconnected ($v = 0$) candidate connection samples.

– ***Prior Potentials*** $\Psi_c(\boldsymbol{x}_c)$*:* We defined the prior potentials in terms of penalty values on the number of connections of any single end point of a polyline. We penalized the maximum number of connections at a single end point to be at maximum three, with one of the connections being a cane segment itself. This is done to simplify our model of connectivity, so a cane segment can branch into maximum two other canes segments. Mathematically, denoting by $\boldsymbol{x}_p = [x_{p_1}, x_{p_2}, ..., x_{p_n}]$ the clique build from all candidate connections at end point p, this penalization can be written as a function g_M with penalty constant $\lambda_M > 0$:

$$g_M(\boldsymbol{x}_p) = \lambda_M \cdot [\max\{3, \sum_{i=1}^{n} x_{p_i}\} - 3]. \quad (8)$$

Observe that the sum term counts the number of connections candidates in the clique c that are set connected by setting $j_{c_i} = 1$, so if this number is less than our maximum number of connections allowed of three, then g_M is zero. On the other hand, when this number exceeds the limit, the penalty of λ_M is applied to the excess of number of connections. In this way, we have $\Psi_c(\boldsymbol{x}_c) = g_M(\boldsymbol{x}_c)$.

IV. Vine Structure Inference

The previous section proposed a model for vine structure in binary images. In this section we apply this model to extract vine structure from images. The pipeline of our system is shown in Figure 4. Given an input 2D vine binary image, we use extract the cane segments as explained in Section III-A. After this is done, we build the Markov graph of Section III-B; find the set of all candidates connections l_i; find all attributes z_i corresponding to each connection candidate; and initialize the states of all connections x_i. We then aim to minimize the energy of Equation 4. In particular, we used simulated annealing for solving for the most likely state of connections $\boldsymbol{x}^*$. Simulated annealing is a non-greedy heuristic approach to energy minimization, that very often performs relatively better that other optimization techniques [18]. It is also used commonly for comparative studies [19], and so we believe it is important for our research.

TABLE I
Cane Structure Matching to Ground Truth

	Precision	Recall
80% Canes Overlap		
SA	0.155877	0.896615
Gibbs	0.09750	0.923147
Contour	0.55055	0.682525
50% Canes Overlap		
SA	0.163194	0.938701
Gibbs	0.0997294	0.94419
Contour	0.633948	0.78591
50% Canes Length Coverage		
SA	0.191369	0.188011
Gibbs	0.141376	0.303813
Contour	0.510073	0.390673

V. Comparative Results

We compared our method to the methods of Marin et al. [4] and Botterill et al. [1]. In Marin et al. [4], the author used a set of ground truth canes annotated manually on a set of real vine images, and compared its method to that of Botterill et al. [1]. The comparison methods were precision and recall of canes overlap and length coverage. Similarly, we used the same set of images of 45 frames and ground truth data, and assessed our method precision and recall to match ground

truth. Comparative results are shown in Table I. Similar to Marin et al. [4], a cane is considered correct if it overlaps with p% of a ground truth cane, or if it covers p% of the length of the ground truth cane. We used the same values of p = 80% and p = 50% for overlaps and p = 50% for length coverage. The results table shows the average precision and recall for all vine frames. Depending on the vine image, our method showed a maximum precision of ≈ 0.52 , and lowest precision of ≈ 0.05 with respect to the 50% for length coverage property, which is considered the best measure for evaluating vine canes extraction methods. In summary, our results showed an improvement of precision upon the method of Marin et al. [4] while maintaining recall similar. However, in terms of precision, the method of Botterill et al. [1] is still the best method to extract canes from images.

VI. Conclusions and Future Work

We have proposed a novel HMRF vine model to extract vine structure from images. The model used skeletonisation to build cane segments, and vine structure extraction was formulated as a MAP inference problem. Also, an energy model was constructed by penalizing the number of connections at end points and by learning two class conditional densities for modeling the probability of connecting a given pair of candidate points. Finally, we showed comparative results of our method against state-of-the-art methods used for the same tasks. Our method generalizes and improves precision while maintaining consistently recall in comparison to the method of Marin et al. [4].

Our method worked better on images where less noise from leafs regions or artifacts are present in the binary image. In general the connections at crossings of canes where solved by our method when only two canes where involved and when they had recognizable different directions. Other regions generated problems since crossing canes with similar direction are represented in a binary image with a single skeleton cane segment, and this unique cane segment is used for representing two different real cane segments. Therefore, we need to address in our model this case scenario. For this, first we need to be able to recognize these regions, then one solution could be to just duplicate the cane segment, or model the Markov graph differently at those regions.

We believe that our method can be improved to match the accuracy of the heuristic approach of Botterill et al. [1]. For this we can improve our energy model to include global information of currently connected segments at any given time during optimization. Our energy model is still making mainly local decisions, and that is a limitation of the current model. Another limitation of our method is that some cane segments are not considered even as candidates, because of our method of selecting the candidates is filtering points within a threshold distance. Since we don't want to consider every other possible point as candidate, we are researching other ways of selecting connection candidates.

As a future work, we are currently researching how to incorporate a direction attribute at every candidate connection. This will allow us to solve for cycles in the vine structure graph, which happens currently. Also, we are researching new energy potentials and we plan to evaluate several energy minimization techniques as well as Monte Carlo sampling methods to perform vine inference in the proposed model.

References

[1] T. Botterill, R. Green, and S. Mills, "Finding a vine's structure by bottom-up parsing of cane edges," in *Proceedings IVCNZ*, 2013.

[2] A. Blake, P. Kohli, and C. Rother, *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.

[3] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd ed. Springer Publishing Company, Incorporated, 2009.

[4] R. Marin, T. Botterill, and R. Green, "Gibbs sampling for 2d cane structure extraction from images," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, Feb 2015, pp. 461–465.

[5] C. Godin and H. Sinoquet, "Functionalstructural plant modelling," *New Phytologist*, vol. 166, no. 3, pp. 705–708, 2005. [Online]. Available: http://dx.doi.org/10.1111/j.1469-8137.2005.01445.x

[6] R. Sievanen, C. Godin, T. M. DeJong, and E. Nikinmaa, "Functionalstructural plant models: a growing paradigm for plant studies," *Annals of Botany*, vol. 114, no. 4, pp. 599–603, 2014. [Online]. Available: http://aob.oxfordjournals.org/content/114/4/599.abstract

[7] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. New York, NY, USA: Springer-Verlag New York, Inc., 1990.

[8] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang, "Sketch-based Tree Modeling Using Markov Random Field," in *ACM SIGGRAPH Asia 2008 Papers*, ser. SIGGRAPH Asia '08. New York, NY, USA: ACM, 2008, pp. 109:1—-109:9. [Online]. Available: http://doi.acm.org/10.1145/1457515.1409062

[9] B. Gorte, "Skeletonization of laser-scanned trees in the 3d raster domain." in *3D-GIS*, ser. Lecture Notes in Geoinformation and Cartography, A. Abdul-Rahman, S. Zlatanova, and V. Coors, Eds. Springer, 2006, pp. 371–380. [Online]. Available: http://dblp.uni-trier.de/db/conf/3dgis/3dgis2006.htmlGorte06

[10] A. Schilling and H.-G. Maas, "Automatic reconstruction of skeletal structures from TLS forest scenes," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-5, pp. 321–328, 2014. [Online]. Available: http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-5/321/2014/

[11] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Trans. Graph.*, vol. 26, no. 4, oct 2007. [Online]. Available: http://doi.acm.org/10.1145/1289603.1289610

[12] W. Shi and C. Cheung, "Performance evaluation of line simplification algorithms for vector generalization," *The Cartographic Journal*, vol. 43, no. 1, pp. 27–44, 2006.

[13] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984. [Online]. Available: http://doi.acm.org/10.1145/357994.358023

[14] P. Clifford, "Markov random fields in statistics," in *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, G. Grimmett and D. Welsh, Eds. Oxford: Oxford University Press, 1990, pp. 19–32.

[15] C. Wang, N. Komodakis, and N. Paragios, "Markov random field modeling, inference & learning in computer vision & image understanding: A survey," *Comput. Vis. Image Underst.*, vol. 117, no. 11, pp. 1610–1627, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2013.07.004

[16] T. Botterill, R. Green, and S. Mills, "A decision-theoretic formulation for sparse stereo correspondence problems," in *To appear in Proc. 3DV*, 2014.

[17] S. J. D. Prince, *Computer Vision: Models, Learning, and Inference*, 1st ed. New York, NY, USA: Cambridge University Press, 2012.

[18] P. Salamon, R. Frost, and P. Sibani, *Facts, conjectures, and improvements for simulated annealing*, ser. SIAM monographs on mathematical modeling and computation. Philadelphia: SIAM, 2002. [Online]. Available: http://opac.inria.fr/record=b1102464

[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 1068–1080, June 2008.