Binary Image Scanning Algorithm for Cane Segmentation

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

***Abstract*—In this paper we present a novel approach to vine cane segmentation. We present a formal definition of a vine cane segment and describe an algorithm to find an approximation of a full cane segmentation for a 2D vine image. We have called the algorithm *Binary Image Scanning* - BIS, since it scans a binary image collecting vine sections of vine pixels in both directions, horizontal and vertical. The two scans are processed to form cane segments and are merged to a full cane segmentation. We show results of our cane segments to match skeleton curves obtained with morphological thinning, and we compare both methods to match ground truth data of vine images. Our method achieves better precision than thinning skeletonization, which from prior research is known to be the best skeletonization method to model vines [1], [2]. We also discuss ideas to further extend our findings in the future.**

I. Introduction

In this paper we describe a new method to perform the segmentation of canes of grape vine binary images (see in Figure 1). The segmentation is to be used to infer the 2D tree structure of the vine. The 2D structure is to be used by a vine pruning robot to reconstruct 3D model of vines.

Modeling and extracting each cane segment separately will allow us to observe and analyze the vine from a high level vision perspective. For instance, in Botterill et al. [3] we are currently researching bottom up/top down methods to parse vine images and recover the underling 2D structure. Here the terminals of the parsing tree for a given input image would correspond to the cane segments or furthermore to the pixels that form up these segments. In [3] the authors reported that the method could be improved by detecting directly cane segments. Therefore, here we are interested in defining what a cane segment is and in developing an algorithm to extract them directly given an input image. The algorithm should be also consistent between different vines, and efficient enough to be used on image sequences.

Our main contributions on this paper are firstly, a formal definition of cane segments which is related to skeleton curves; secondly a constructive algorithm that find approximately all cane segments of an input vine binary image; and finally comparative results of our canes segments versus skeleton curves to match ground truth vine data.

The rest of this document is structured as follows. Section II reviews other research on vine modeling and or methods that are relevant to this topic. The main contributions of this paper as described before are presented in Section III. We also show results of applying our method to match ground truth data in comparison to skeletonization in Section IV. Finally, Section V presents a conclusion, limitations and improvements of our method in the future.

II. Related Research

This research is related to segmentation on binary images and skeletonization methods. In particular, cane segmentation has been researched before in Marin et al. [4] and Botterill et al. [3]. In [4] the idea is to automatically cluster together all vine pixels that conform to one cane by learning a mixture of Gaussians with a custom split-and-merge expectation maximization algorithm. The method works well in recognizing different cane parts. However, these cane regions in the same output segmentation are not consistent with a common definition. This means some recognized canes may include overlapping sections or holes in between, while many others do not. Instead, our scanning algorithm construct approximations to cane segments that conform to a formal definition of cane segment (Section III-A). This ensures the consistency of the output which we think is a desirable characteristic for using the output as primitives in a grammar, for inferring the vine structure.

In [3] the authors showed promising results of bottom-up parsing of cane segments. These segments are extracted by simplifying edge polylines using the Ramer-Douglas- Peucker algorithm [3]. Here a hierarchical bottom-up representation is carried from cane edge segments, to full cane contours, then to cane parts, and finally to complete canes. The method makes use of SVM, to help in deciding which two pairs of parts should be joined or not. Results were reported to be limited to the training procedure [3]. Future research on this technique points into looking at alternative primitives. Our research can be seen as an alternative method to compute cane segments directly. These cane segments could be incorporated into an existing bottom up/ top down vine parser and then are useful for recovering 2D vine structures on images.

Our binary scanning algorithm and our cane segment definition are related to medial representations and skeletonization methods [5], [6]. Our resulting cane segmentation output can be seen as a medial representation of parts of the vines, plus information of width, orientation and segment assignments per pixel. In contrast, skeletonization methods do not have any other information apart from the skeleton points and sometimes distance to border. Nevertheless, skeletons are a natural representation of articulated object shapes, and so, have

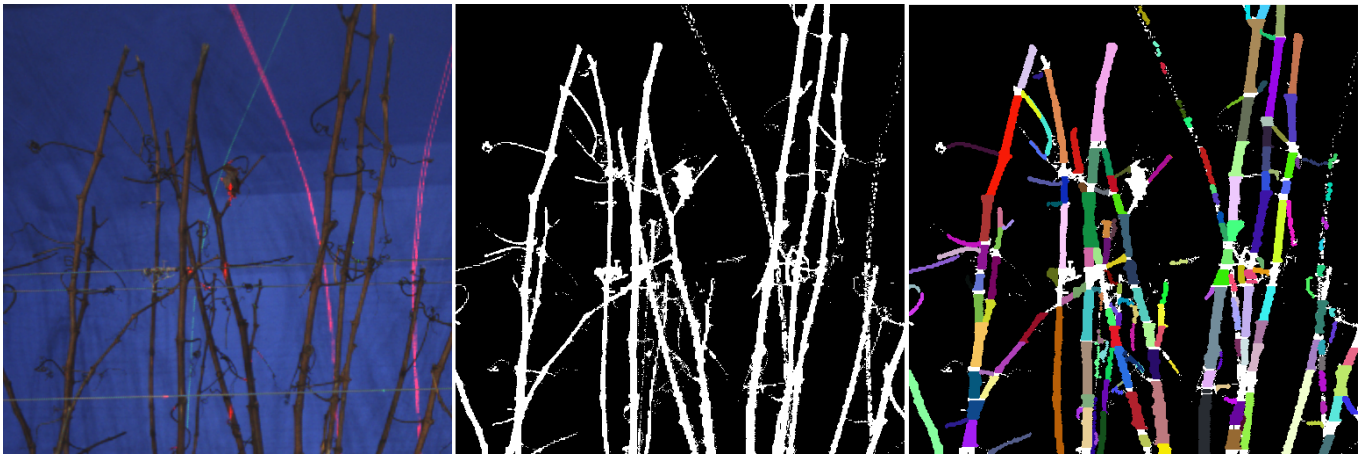


Fig. 1. From left to right: Input vine image, vine binary map, and result of cane segmentation with our BIS algorithm (see section III).

been used extensively in plants and trees modeling research. Approaches for constructing 2D skeletons from binary images are usually adaptation of morphological thinning algorithms, ridge finding, medial axis transform or hybrid methods [1], [7]–[9]. Gittoes et al. [1], [2] performed a quantitative analysis of the use of these methods for modeling vines. The authors found that skeletons that results from using the medial axis transformation are in general disconnected along the edges, while skeletons found by finding ridges are disconnected at branching points. Also, skeletons found with steerable filters are connected depending on the size of the filter. The author concluded that thinning skeletonization methods are the most suited for modeling vines given that connectivity is assured. However, thinning algorithms tend to create spurs that do not match the underlying vine structure, and so they cannot be used directly for modeling the vine hierarchy. In our research, skeleton curves are used to define cane segments. This definition of cane segment and our full cane segmentation scheme are described in the following sections.

III. Vine Image Segmentation

In this section we will describe a novel method to vine image segmentation. Figure 1 shows the output of our algorithm. We have as an input a vine binary map we will denote by I. This map is extracted from a full color vine image with methods that are out of the scope of this paper. We will start by giving a formal definition of the type of cane segments we want to find, and then we proceed to describe our algorithms for cane segmentation.

A. Preeliminaries

A vine consists of many individual branches called *canes*. A cane grows from the root of the vine and may have many branching points from which other canes grow. As shown in Figure 2, we can associate a skeleton curve to a given vine binary image [5], [6]. There are many ways in which skeleton curves can be defined and computed [6]. An ideal skeleton for vines should be insensitive to noise in the vine border, which often appear during the computation of the binary map. Also, the ideal skeleton is connected and must be composed of only three types of points: interior, junctions and end points, relating directly to the cane's hierarchy. Here we will assume we

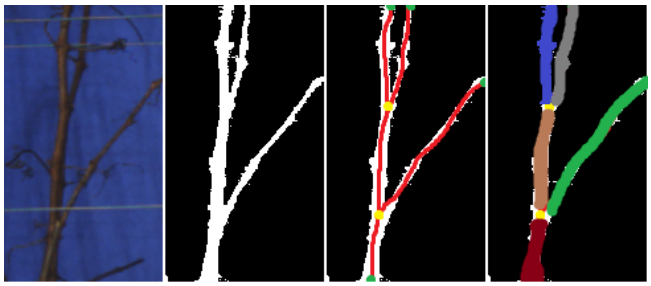


Fig. 2. Cane segments: From left to right, we have an input color cane image and its binary map. The ideal skeleton curve should match the vine topology and be composed of three types of points, interior (red), junctions (yellow) and end points (green). Finally, cane segments are shown in different colors.

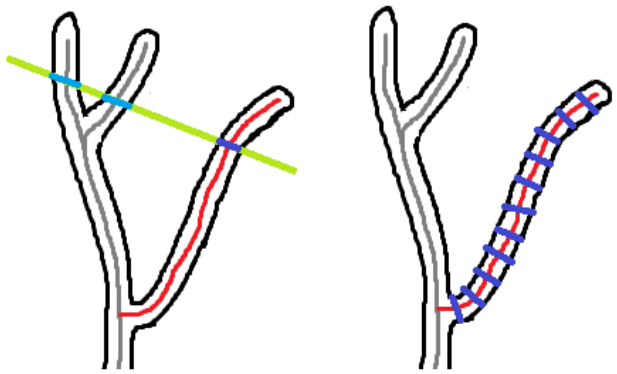


Fig. 3. Illustration of vine normal sections. Skeleton curve l is shown in red. The other portions of the skeleton that are not considered are shown in gray. $L^l_\perp(t)$ is shown in green, with $\mathcal{V}^l(t) = L^l_\perp(t) \cap V$ shown in blue. The dark blue represents the vine normal sections $V^l_\perp(t)$ of $l(t)$.

can compute such and ideal skeleton curve. Detailed research on this topic can be found on [6]. In our approach to cane segmentation, we further divide each cane into parts we call *cane segments*. A cane segment is a maximal part of a cane such that the skeleton curve restricted to this subset posses diameter variations bounded.

Formally, in continuous space $\mathbb{R}^2$ we will have skeleton curves l with parameter t. Here l denote a portion of the

skeleton that is composed only of interior points. A detailed study of local geometry of these curves can be found on Chapters 1 and 3 of [6]. In our method, we define a diameter variation function $r(t)$ using vine normal sections to l, as follows. Denote by $V \subset \mathbb{R}^2$ the set of vine points and by $L^l_\perp(t)$ the space generated by the normal vector $n(t)$ at $l(t)$: $L^l_\perp(t) = \{(l(t) + \lambda n(t) \in \mathbb{R}^2 / \lambda \in \mathbb{R}\}$. A collection of vine sections associated to l at t is the intersection

$$\mathcal{V}^l(t) = L^l_\perp(t) \cap V \tag{1}$$

The vine normal section $V^l_\perp(t) \subset V$ also associated to l at t is the connected component of the point $l(t)$ in $\mathcal{V}^l(t)$ (see Figure 3). We define $r(t)$ as the diameter of the set $V^l_\perp(t)$.

Now, we fix a value $T_r \in \mathbb{R}$ that we will call the *diameter variation bound*, and we define a *cane segment* S as a connected subset of vine pixels with a skeleton curve $l(t)$, with parameter t and $\bar{r}_S$ the mean diameter variation satisfying:

1) *Maximality:* There is no other cane segment $\hat{S}$ such that $S \subset \hat{S}$.
2) *Bounded Diameter Variations-BDV:* $|r(t) - \bar{r}_S)| \leq T_r$ for all t.

Cane segmentation on a vine image can now be stated as: Given a vine binary map I and a diameter variations bound T_r, find all cane segments S according to the definition written above. Observe that the set of cane segments on which we can divide a cane is relative to the value of T_r. A value of $T_r = 0$ will segment a cane into portions with constant radius. However, canes in images present variations in radius due to the presence of vine *buds*, depth variation, and due to overlaps with other canes. Furthermore, the binary vine map, which is found by background segmentation, will generally have aliasing artifacts at the vine boundary and this will imply small radius perturbations for sections of the same cane. In our experiments (SectionIV) we used $T_r = 0.25$ meaning we allow sections of a cane segments to be at most 25% of the mean diameter value. In the rest of this section we describe a method to approximate a full cane segmentation for an input vine binary image.

B. Binary Image Scanning Algorithm

We would like to find all cane segments S for a given vine binary image I and a fixed diameter variation bound T_r. For this, we could start by finding the skeleton curves in I and then use a segmentation technique such as K-means or Expectation Maximization to group together vine normal sections according to diameter variations along the skeleton. However, there are some problems with using this procedure. Firstly as researched in [6], skeletonization is hard to achieve consistently given its sensitivity to details in the boundary representation, that is, the same object could have different medial branching topology in function of small perturbations on the boundary. Secondly, skeletonization methods are not well suited for modelling vine canes [1], [2], since most of these methods produce disconnected outputs. Finally, algorithms that guarantee connected skeletons, like thinning methods, are biased to create spurs that does not match the underlying vine topology [2]. That said, generating a skeleton that match exactly a vine binary image I is a complex unsolved problem.

Therefore, in this paper we propose a novel scanning algorithm to approximate cane segments. For this, we group vine normal sections according to radius and centre dislocations. The core idea of our method is that instead of using skeleton curves for finding these vines normal sections, we make use axis aligned lines in horizontal and vertical directions. We will see in Section IV experimental results showing that the cane segments found in this way will define curves that are approximations to the true skeleton curves. In summary, our method has two main stages. First we scan twice I to approximate cane segments in directions horizontal and vertical respectively; and second we use both segmentations and discard bad approximations according to a criterion we will define in Section III-B2. We will describe these two stages in the following.

1) Axis Aligned Vine Sections Scanning: In Eq. 1 we found a collection of vine sections associated to a skeleton curve l. However, skeleton curves that represent well canes are not available as explained in the introduction to Section III-B. Instead, we use axis aligned lines, and thus we find vine sections oriented horizontally and vertically respectively. After finding these vine sections, we can construct approximations to cane segments by grouping vine sections according diameter variations (See Figure 4). The description that follows is dedicated to horizontal oriented vine sections, being the vertical version analogous.

Denote by n_r and n_c the number of rows and number of columns of I respectively. In the following we proceed to work in the discrete domain of pixels of I. Taking $t \in \{1, 2, ..., n_r\}$ we define horizontal lines as

$$L_H(t) = \{(t, c) / c \in \{1, 2..., n_c\}\} \tag{2}$$

and analogous to Eq. 1 we can find the collection of horizontal vine sections at row t by

$$\mathcal{V}_H(t) = L_H(t) \cap V = \cup_{k=1}^{k_t} V^k_H(t) \tag{3}$$

where each $V^k_H(t)$ is a connected and horizontally oriented vine section, and k is an index for connected components of $\mathcal{V}_H(t)$. As before, the diameter $r^k_H(t)$ is defined as the diameter of $V^k_H(t)$. Figure 4 illustrates these definitions.

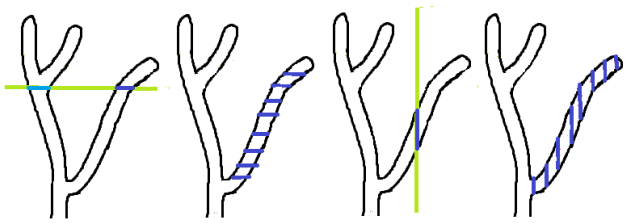


Fig. 4. Illustration of vine normal sections with axis aligned lines. $L^H_\perp(t)$ and $L^V_\perp(t)$ are shown in green, with $\mathcal{V}^H(t)$ and $\mathcal{V}^V(t)$ shown in blue. The dark blue represents the vine sections $V^k_H(t)$ and $V^k_V(t)$ respectively. Collecting vine sections aligned with the axis will allow approximating cane segments.

To construct approximations to cane segments, we aim to group together vine sections according to diameter variations. Suppose you have a group S of vine sections that satisfy the BDV condition of the cane segment definition. Then, we can continue growing S by inserting vine sections $V^k_H(t)$ that satisfy $|r^k_H(t) - \bar{r}_S| \leq T_r$, thus keeping the BDV property on

Data: Binary vine image I
Result: Array of cane segments S_H with horizontal vine sections

Initialize S_H to empty array;
for $t \leftarrow 1$ **to** n_r **do**
  Find $\mathcal{V}_H(t)$;
  foreach $V_H^k(t)$ *in* $\mathcal{V}_H(t)$ **do**
    if *r==1* **then**
      insert to S_H a new segment S with a single vine section $V_H^k(t)$;
    else
      compute $S^* = \min_{S \in S_H} \left\{ R(r_H^k(t), \bar{r}_S) \right\}$;
      set c^* to the center of the latest vine section in S^* at row $t-1$;
      if c^* *exist; and* $R(r_H^k(t), \bar{r}_S*) \leq T_r$*; and* $D(c_H^k(t), c^*) < T_d$ **then**
        insert cross section $V_H^k(t)$ to S^*;
      else
        insert to S_H a new segment S with a single vine section $V_H^k(t)$;
      end
    end
  end
end

Algorithm 1: Horizontal Vine Sections Scanning

S. In our cane segments approximations we modify this by a relative diameter variations condition

$$R(r_H^k(t), \bar{r}_S) = \frac{|r_H^k(t) - \bar{r}_S|}{\max\left\{r_H^k(t), \bar{r}_S\right\}} \leq T_r \tag{4}$$

This allows S to grow relative to the average diameter of S. This condition alone is however, not enough for guaranteeing S is a cane segment, since unconnected sets can be formed. Therefore, we will impose two more connectivity conditions. The first one is that the candidate vine section $V_H^k(t)$ to insert to S should be at a row consecutive to some vine section of S. This ensures that vine sections are connected along the vertical direction. The second connectivity condition is a bound on the horizontal dislocation of the centers of the vine sections of S. The relative horizontal dislocation condition between the centers $c_H^{k_1}(t_1)$ and $c_H^{k_2}(t_2)$ of vine sections $V_H^{k_1}(t_1)$ and $V_H^{k_2}(t_2)$ respectively, is defined as

$$D(c_H^{k_1}(t_1), c_H^{k_2}(t_2)) = \frac{\left| x(c_H^{k_1}(t_1)) - x(c_H^{k_2}(t_2)) \right|}{\max\left\{ r_H^{k_1}(t_1), r_H^{k_2}(t_2) \right\}} < T_d \tag{5}$$

where $x(.)$ means the column value in I of the argument. This new condition does not have to be satisfied by all pairs of vine sections of S like the BDV condition, but instead, it must be satisfied only by pairs of vine sections that have consecutive rows in S. This second condition ensures consecutive vine sections are dislocated relative to each other at most by a value T_d. In our experiments we used $T_d = 0.25$ which means that consecutive vine sections have dislocation at most of 25% of the maximum diameter among both sections.

We can now develop an algorithm to approximate cane segments by grouping together vine sections of adjacent rows according to diameter variations and centre dislocations. This is an iterative procedure presented in Algorithm 1. We denote by S_H the array of all horizontal cane segments, initially empty. We start by finding $\mathcal{V}_H(1)$ and for each $V_H^k(1)$ we insert a new cane segment S in S_H composed of this single vine section. Then, for each of the subsequent rows t, we test whether the current $V_H^k(t)$ could be inserted to every segment S in S_H by using Eq. 4 and Eq. 5. If it can be inserted to more than one S in S_H, we insert it only to the one that minimizes the values of Eq. 4 and 5. If it cannot be inserted to any S, then a new cane segment S with this single vine section is inserted to S_H. The algorithm ends when all the rows have been processed.

2) Goodness of Segments: In this subsection we give a condition of goodness for segments S found using Algorithm 1. This is done by characterizing the shape of S. Denote by n_S the number of axis aligned vine sections of S, and $\bar{r}_S$ the mean diameter variation. We say that S is degenerated when

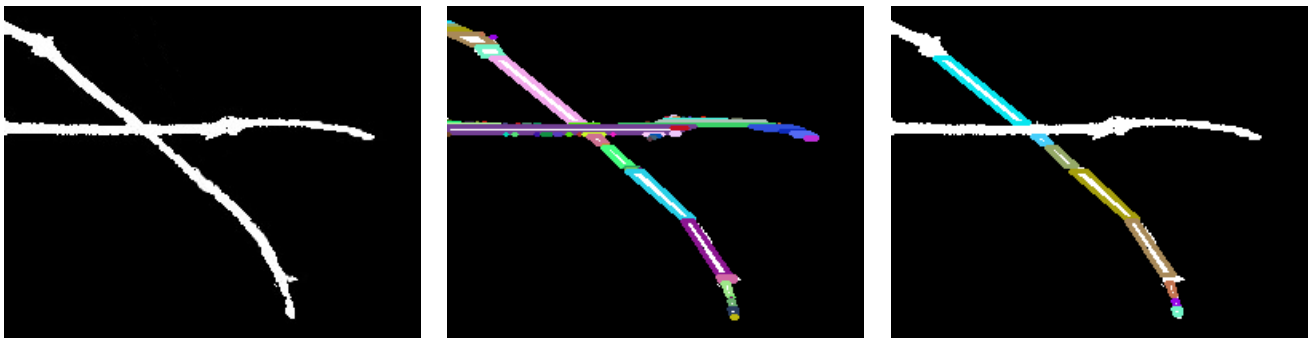
$$n_S < T_s \bar{r}_S \tag{6}$$

$$n_S < T_n \tag{7}$$

where T_s controls the proportion of segment's length to average thickness; and T_n that controls the minimum number of vine sections in S. Note that we thresholded the proportion of length to average thickness given that in general real cane segments posses this tubular shape restriction. Both T_s and T_n are constant parameters we tune in our system. In Section IV we show the values we used in our experiments. Figure 5 shows the result of discarding bad segments after using Algorithm 1.

3) Full Cane Segmentation: In the previous subsection we saw that by using Algorithm 1 in one axis, some segments S are degenerated according to Eq. 6. This means that after one scan, by discarding degenerated segments, some regions of the vine will not be associated to any segment, even though some of these regions may still be real cane segments (See Figure 5 (c) and (d)). Therefore, we designed a set of operations to perform full cane segmentation by using scans in horizontal and vertical directions and discarding degenerated segments. Figure 5 shows this process. First we scan in the horizontal direction and discard degenerated segments. Then we scan in the vertical direction the portion of the vine pixels that is not represented by the horizontal segments. Finally, we discard degenerated vertical segments to achieve a full cane segmentation.

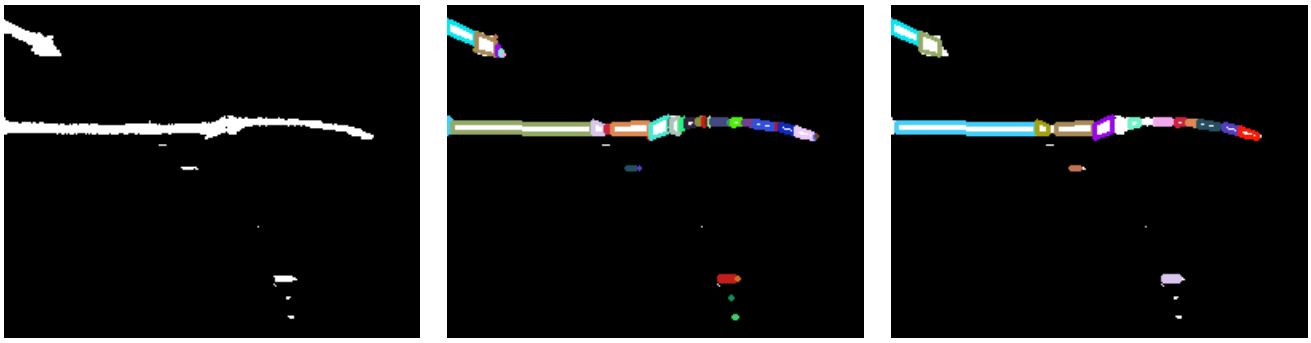
This way of merging scans in both directions ensures that non of the segments with overlap with each other. Also, note that in the final result still some portions of the vine pixels will not be associated to any segment. However, this portions now will conform parts of the vine that cannot be represented with either horizontal or vertical segments. We found that these portions are in general regions of complex occlusions between canes or noise of the binary map. In our system, we further include in our segmentation, connected components of these regions, and we called them *unassigned regions*. We included them on the segmentation output because we think they may be useful as primitives together with the cane segments, for using in a bottom up / top down vine structure parsing system.



(a) Input Binary Image.

(b) Horizontal scan cane segments.

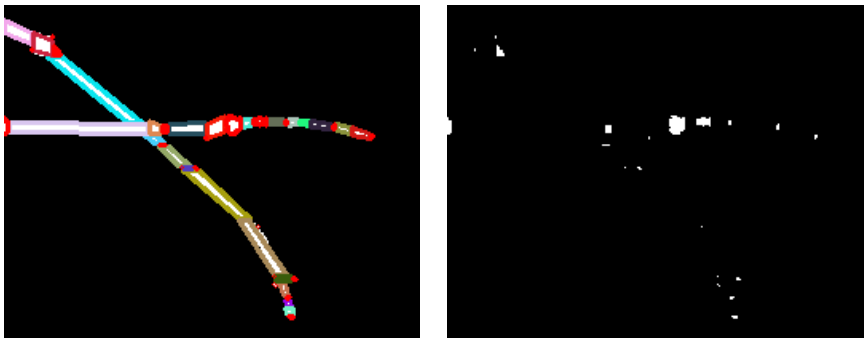
(c) Horizontal scan cane segments after discarding degenerated segments.



(d) Unassigned vine pixels after horizontal scan.

(e) Vertical scan cane segments.

(f) Vertical scan cane segments after discarding degenerated segments.



(g) Full Cane Segmentation .

(h) Unasigned Regions .

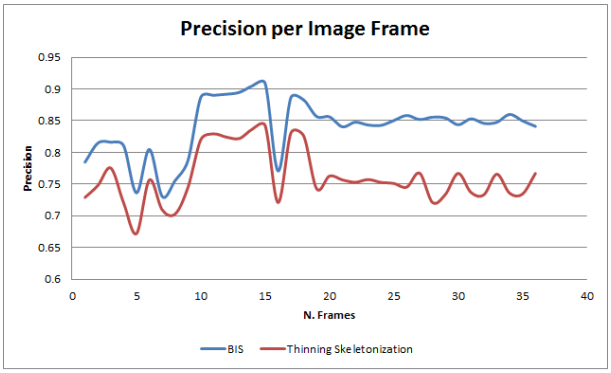
Fig. 5. Full Cane Segmentation Process: An input binary image (a) is scanned horizontally to get (b). The degenerated horizontal cane segments are discarded to achieve (c) and the vine pixels left unassigned (d) are scanned vertically (e). Then, discarding vertical degenerated segments (f) we merge both horizontal and vertical canes segments to achieve full cane segmentation (g). Unassigned regions of pixels after the merging are shown in (h). Parameter values for the scanning algorithm were: $T_r = 0.25$, $T_d = 0.25$, $T_n = 5$ and $T_s = 0.66$.

These unassigned regions together with a full segmentation are shown in Figure 5.

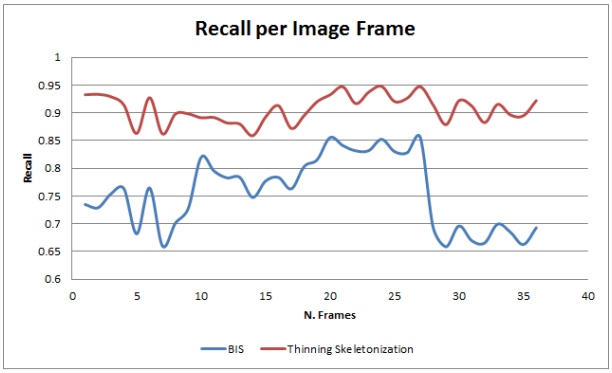
IV. Results

To evaluate our method we performed two experiments. Firstly we compared our scanning algorithm to skeletonization for matching manually annotated ground truth vine data. Secondly, we measured the error between our cane segments center points to skeleton curves. For all our experiments we used morphological thinning skeletonization, since as researched before in [1], [2] is the best skeletonization approach for modeling vines. Also, the set of parameters used for all our experiments was $T_r = 0.25$, $T_d = 0.25$, $T_n = 5$ and $T_s = 0.66$. These values were setup heuristically.

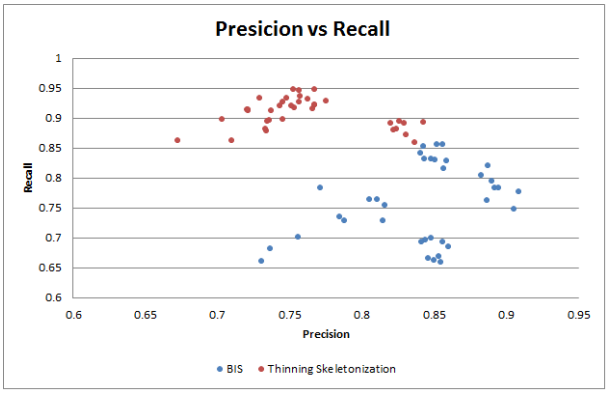
Ground truth data is composed of a set of real vine images that have canes manually annotated as polylines. Therefore, given a curve to represent the canes (found by either skeletonization or our method), we can measure precision and



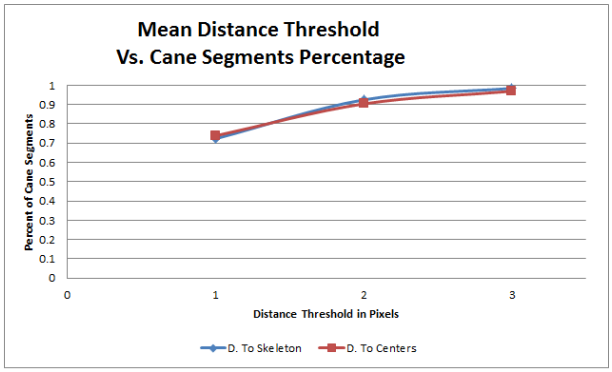
(a) Precision comparison per image frame.



(b) Recall comparison per image frame..



(c) Unassigned vine pixels after horizontal scan.



(d) Vertical scan cane segments.

Fig. 6. $T_r = 0.25$, $T_d = 0.25$, $T_n = 5$, $T_s = 0.66$.

recall of this curve to match the ground truth polyline. In our experiments this matching is done per point in the curve. A point in a curve is correct if it is at most in euclidean distance dp pixels away from the ground truth polyline. Figure 6 (a), (b) and (c) shows precision and recall results with $dp = 5$ pixels, of both skeletonization and of our method curves which are formed as the center points of all vine sections of a cane segment. Precision and recall measurements per frame are shown in (a) and (b). We can see that our method achieves better precision in every frame compared to skeletonization. On the other hand our method has the worst recall, since on regions of cane occlusions our method generate unassigned regions which doesn't have associated any cane curve, and thus on these regions the ground truth polylines are not well matched with our method. Figure 6 (c) shows precision versus recall of all frames.

Finally, to assert the quality of our cane segments found by axis aligned vine sections, we matched our cane segments to skeleton curves. This was done per cane segment, and we measured the averaged euclidean distance of the centers of the vine sections of a cane segment to the skeleton points and vice versa. Figure 6 (d) shows results of the percent of cane segments that have average distance to skeleton curve (blue) and from skeleton to vine section centers (red), in less than 1 , 2, and 3 pixels respectively. We can see that with an error of at most 3 pixels, 99% of our cane segments center curves approximates skeleton curves.

V. Conclusion and Future Work

This paper described the Binary Image Scanning BIS algorithm that is able to perform cane segmentation of vine images. The algorithm constructs vine segments in horizontal and vertical directions that approximates real cane segments. We have showed results that our method has greater precision compared to morphological thinning skeletonization to match manually annotated ground truth vine data. Our method also obtained worse recall that skeletonization due to the vine overlapping regions that are found to be unassigned to any cane segment by our scanning procedure. Finally, we showed that the center curve of our cane segments is 99% similar to this kind of skeleton curve of the vine binary map,

A natural direction of research is to develop a framework to analyze the error in approximating the ground truth cane segments according to our definition. In our method, different segmentations can be achieved by changing the parameters T_r, T_d, T_n and T_s. Thus, such error analysis could help in determining the best set of parameters to obtain the best possible solution using the BIS algorithm.

REFERENCES

[1] W. Gittoes, T. Botterill, and R. Green, "Quantitative analysis of skeletonisation algorithms for agricultural modelling of branches," in *Image and Vision Computing New Zealand (IVCNZ)*, 2011.

[2] W. Gittoes, "Vine model fitting using a composite skeletonisation method," University of Canterbury, Tech. Rep., 2011.

[3] T. Botterill, R. Green, and S. Mills, "Finding a vine's structure by bottom-up parsing of cane edges," in *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, Nov 2013, pp. 112–117.

[4] R. Marin, T. Botterill, and R. Green, "Split-and-merge em for vine image segmentation," in *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, Nov 2013, pp. 270–275.

[5] A. Tagliasacchi, "Skeletal representations and applications," Simon Fraser University, Tech. Rep., 2012.

[6] K. Siddiqi and S. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[7] C.-H. Teng, Y.-S. Chen, and W.-H. Hsu, "Tree segmentation from an image," in *IAPR, Conference on Machine Vision and Applications*, 2005.

[8] Y. D. Lopez, L. D., "Modeling complex unfoliaged trees from a sparse set of images." vol. 29, no. 7, 2010, pp. 2075–2082.

[9] L. D. Lopez, D. Shantharaj, H. B. Lu Liu, and J. Yu, "Robust image-based 3d modeling of root architecture," in *Computer Graphics International*, 2011.