Fast Semi-supervised Segmentation of in Situ Tree Color Images

Philippe Borianne[1] and Gérard Subsol[2]

[1] CIRAD - AMAP, Montpellier, France
philippe.borianne@cirad.fr
[2] LIRMM – CNRS, University Montpellier 2, France
gerard.subsol@lirmm.fr

Abstract. In this paper we present an original semi-supervised method for the segmentation of in situ tree color images which combines color quantization, adaptive fragmentation of learning areas defined by the human operator and labeling propagation. A mathematical morphology post-processing is introduced to emphasize the narrow and thin structures which characterize branches. Applied in the L*a*b* color system, this method is well adapted to easily adjust the learning set so that the resultant labeling corresponds to the accuracy achieved by the human operator. The method has been embarked and evaluated on a tablet to help tree professionals in their expertise or diagnosis. The images, acquired and processed with a mobile device, present more or less complex background both in terms of content and lightness, more or less dense foliage and more or less thick branches. Results are good on images with soft lightness without direct sunlight.

Keywords: Semi supervised segmentation, in situ tree color image, image labeling, L*a*b* color system.

1 Introduction

Trees are important for the ecology and economy of the modern cities. Not only, a tree protects the structure of soils by limiting the erosion and the flooding risk but it is also a natural air conditioner which regulates temperature and ambient humidity. But today the health of tree is widely threatened by the effects of climate change [3]. It is then essential to perform a regular diagnosis of urban trees to assess their health. In practice, this is done by visually tracking external trauma symptoms as dead branches, discontinuities in shape and density of the crown, detachment of bark or presence of parasites. These analyses are now more and more based on digital photographs taken in situ.

Some image-processing software have been developed to help the expert. The most finalized software is the UrbanCrown application developed by the Forest Service of the US Department of Agriculture [16]. This program estimates the transparency of the tree crown from several digital images. It has however two drawbacks: it imposes drastic acquisition conditions, both in terms of light exposure and distance to target,

A. Elmoataz et al. (Eds.): ICISP 2014, LNCS 8509, pp. 161–172, 2014.

and it requires an important operator supervision for crown segmentation which is based on active contours. Some research software tools are also available: the most dedicated one is the ImageJ plug-in Croco [10] but it requires already segmented images of the tree. The segmentation has to be performed manually by using digital imaging software to erase the background and isolate the crown to analyze.

Some research has been done on the specific problem of segmenting a tree in natural images [15], in particular for 3D modeling applications [14]. The algorithms are based either on hypotheses about the tree structure (e.g. branches linked to a trunk with leaves around) or they require some operator feedback in order to tag the different tree parts. In fact, general methods to segment different superposed objects or to delineate the background in color images, as for example, SIOX [5], fail to isolate a textured tree crown in natural images, especially when artifacts (as power transmission lines or buildings) are present in the background.

Our long-term project aims to develop and embark on a standard mobile device (as a 10" tactile tablet) a method to assist the evaluation of the decline of an urban or peri-urban tree from an in situ tree photography. We think that it is impossible in practice to impose specific conditions of exposure, lighting or contrast during image acquisition. The segmentation method of the tree in the image should then be very robust with respect to acquisition parameters, tree diversity and background artifacts. We assume that the operator can guide the process by tagging some pertinent elements in the image but this implies that the segmentation method must be very fast (no more than a couple of seconds) in order to have an efficient interactive loop between the application and the operator. Our goal is the segmentation in 3 classes: "leaves" which will give information about the crown shape and transparency, "wood" to quantify the homogeneity and the quantity of the visible branches and of the trunk and the rest of the image which will be called "background".

In a first time, we evaluated a segmentation algorithm based on "novelty selection" which is presented as a semi-supervised and fast approach and which was tested on a tree image in [11]. It is divided in two stages: reduction of the dynamic range of colors by aggregating close colors, and labeling of the reduced color image based on some tags given by the user. The reduction of the color dynamic range is performed by a fixed-width clustering based on the average distance between the neighborhoods of the image pixels. Despite the use of an AdaBoost machine learning algorithm [13], this method remains too computationally expensive to be embarked on a mobile device. Moreover, the introduction of neighborhoods smoothes the local color variations and thus prevents from detecting small homogeneous areas while it is essential to evaluate crown transparency or to locate dead branches (see Section 3.1).

In the next sections, we adapt the method to our application by using a classic color quantization to reduce the dynamic range of colors and by introducing a direct Euclidean distance between pixel colors for the labeling step. We also add a post-processing stage in order to emphasize thin structures as the branches. We particularly pay attention to the assessment of the method and we introduce an error function and a performance indicator to evaluate and tune the segmentation process.

2 Presentation of the Method

Our method consists of three independent stages: (1) reduction of the color image dynamic range, (2) the operator quickly and roughly tags some parts of the image which become learning areas for the labeling process which will then classify all the pixels of the image, and (3) post-processing to emphasize narrow and thin structures. We present below each stage and develop some aspects in order to implement and assess the method. We focus in particular on the choice of the color space and the robustness of the labeling stage.

2.1 Reduction of the Color Image Dynamic Range

Reducing the image dynamic range aims to convert the native 32-bit color image in an 8-bit color image in order to reduce the complexity of the labeling stage. Several methods – conversion, classification, quantization, etc. – using different criteria – mean, variance, entropy and ignorance [6] – have been proposed.

One of the most used is probably quantization ([18], [9]) which consists in reducing the number of colors while minimizing the influence on visual perception. But because the quantitative gap between two colors is not so straightforwardly correlated to visual perception, it may be difficult to define the pertinent criteria to be minimized. We selected the very standard Wu's method [18] where color coordinates are reoriented along the principal axes computed by PCA before a recursive partitioning, leading to a number of reduced colors which is set by the user. Faster methods based on k-means classification [7] are also known to give good results. Wu's method appears to be a good compromise between keeping small details as we want to evaluate crown transparency and computational time as we want to implement the algorithm on a mobile device.

2.2 Labeling of the Image

Each learning area delineated by the user is associated to one label: “leaves”, “wood” or “background”. The basic principle of the labeling method is simple:

1. All the image pixels which color belongs to a learning area take the corresponding label;
2. All the non-labeled pixels take the label of the learning area which has the closest mean color, according to a distance defined in the color space.

But, a same color may belong to several learning areas, and thus may lead to assign different labels to a pixel. Of course, it is inconceivable to require that the operator defines learning areas without any common color. This would mean either that the user delineates only very homogeneous areas which will give little valuable information for the labeling process or that the user spends a lot of time to decide which color must be assigned to which label. To solve this problem, we propose three different strategies:

S1 - Sorting the Learning Areas by Increasing Variance. Labeling is incrementally performed from the most to the less homogeneous learning area (LA). The algorithm is then:

1. ***remainingLA***=$\{LA_1 \ldots LA_n\}$; ***processedLA***=$\emptyset$
2. Sort all the LAs of the ***remainingLA*** set w.r.t. to their variance (which is defined in section 2.4).
3. Let $\boldsymbol{LA_{mv}}$ be the LA with the minimal variance. $\boldsymbol{s_{mv}}$ is its associated label and $\boldsymbol{C_{mv}}$ is the set of colors of the pixels of $\boldsymbol{LA_{mv}}$.
4. For each color $\boldsymbol{c}$ of $\boldsymbol{C_{mv}}$, assign the label $\boldsymbol{s_{mv}}$ to all image pixels of color $\boldsymbol{c}$.
5. ***remainingLA=remainingLA*** \ $\boldsymbol{LA_{mv}}$; ***processedLA=processedLA*** U $\boldsymbol{LA_{mv}}$
6. Remove all the colors of $\boldsymbol{C_{mv}}$ in the remaining LAs. Their color sets and their variance will then be modified.
7. Go to step 1 until the set ***remainingLA*** is empty.

But this algorithm leads to an aberrant labeling when a color belongs to an incorrect homogeneous learning area (cf. section 3.1).

S2 - Finding the Closest Learning Area. This is an improvement of the above strategy. Here, an image pixel is labeled according to the "closest" learning area which is the learning area which average color has the smallest distance to the pixel color (see section 2.4. for more details on the selection of the color system and on the associated distance).

3. For each color $\boldsymbol{c}$ of $\boldsymbol{C_{mv}}$:

   - Compute the distance between $\boldsymbol{c}$ and the average color of all the LAs belonging to the ***processedLA*** set (see the definition of the distance in section 2.4).
   - Let $\boldsymbol{LAm_d}$ the processed LA with the minimal distance and $\boldsymbol{s_{md}}$ its associated label.
   - Assign the label $\boldsymbol{s_{md}}$ to all image pixels of color $\boldsymbol{c}$.

If this strategy may avoid many aberrant labeling, it leads to some labeling instability, especially when learning areas are too heterogeneous (cf. section 3.1).

S3 - Fragmenting Heterogeneous Learning Areas. Here, the idea is to fragment a heterogeneous learning area into k homogeneous sub-areas. The fragmentation is produced by a k-means clustering [4] based on the distance between colors. The parameter k is incrementally increased until all the resulting sub-areas are homogeneous enough i.e. their variances are all inferior to a given limit.. These "homogeneous" sub-areas become new learning areas and are associated to the same label that the original learning area. Once all the learning areas have been fragmented, we apply the previous algorithm S2.

This fragmentation process stabilizes labeling by increasing the homogeneity of the learning areas and gives much better results (cf. section 3.1).

2.3 Post-processing to Emphasize Narrow and Thin Structures

We introduce a conditional morphology opening in order to strengthen the narrow and thin structures in the labeled image, especially the visible branches. The morphological opening operator is defined by the iterative application of *n* erosions (i.e. dilations of the complementary) followed by *n* dilations. This opening is conditional: its definition depends on the labels of pixels on which it is applied and of the complementary. If we note $\boldsymbol{p^*}$ the label of the pixel $\boldsymbol{p}$, the result $\boldsymbol{S_{p,b}}$ of the dilation on pixel $\boldsymbol{p}$ with respect to background label $\boldsymbol{b}$ and the 3×3 crossed-shape structuring element $\boldsymbol{S}$ will be given by:

$$\boldsymbol{S_{p,b}}\ \{\boldsymbol{s}+\boldsymbol{p} \mid \boldsymbol{s} \in \boldsymbol{S},\ \boldsymbol{s^*} \in \{\boldsymbol{b},\boldsymbol{p^*}\}\} \tag{1}$$

The erosion must strengthen the "wood" region $\boldsymbol{W}$ at the expense of the "leaves" region $\boldsymbol{L}$ in order to thicken the narrow structures corresponding to the branches. At the ith iteration, both classes are defined as following:

$$\boldsymbol{W_{i+1}} = \{\boldsymbol{p} \in \boldsymbol{W_i}\} \cup \{\boldsymbol{p} \in \boldsymbol{L_i} \mid \boldsymbol{S_{p,w}} \not\subset \boldsymbol{L_i}\} \text{ and } \boldsymbol{L_{i+1}} = \{\boldsymbol{p} \in \boldsymbol{L_i} \mid \boldsymbol{S_{p,w}} \subset \boldsymbol{L_i}\} \text{ with } \boldsymbol{W_0} = \boldsymbol{W}, \boldsymbol{L_0} = \boldsymbol{L} \tag{2}$$

On the contrary, the dilation must strengthen the "leaves" region at the expense of the "wood" region in order to compensate for the previous erosions which preserved visible wood.

$$\boldsymbol{L_{i+1}} = \{\boldsymbol{p} \in \boldsymbol{L_i}\} \cup \{\boldsymbol{q} \in \boldsymbol{W_i} \mid \boldsymbol{q} \in \boldsymbol{S_{p,w}}, \boldsymbol{p} \in \boldsymbol{L_i}\} \text{ and } \boldsymbol{W_{i+1}} = \{\boldsymbol{p} \in \boldsymbol{W_i} \mid \boldsymbol{p} \notin \boldsymbol{S_{q,w}}, \boldsymbol{q} \in \boldsymbol{L_i}\} \tag{3}$$

The "background" region is not affected by the modifications of the other two classes. The post-processing concerns only "leaves" and "wood" areas. In our experiments, we arbitrarily fixed *n* to 3 and discuss this below.

2.4 What Color System to Select?

Several papers present algorithms to compare color systems [6] as their relationship with the image content [2] in order to select the best color system according to a specific segmentation task. But, these methods are too complex to be efficiently implemented on mobile devices. So, for our application, we just analyze our method in the three standard color systems RGB, HLS and L*a*b* and identify the most adapted.

The definition of the color system variance depends on the definition of the brightness or luminance which may be complex and questionable. For our application, we chose to normalize the three coordinates (*x*,*y*,*z*) in the range [0,255] and to consider the sum of three dependent discrete random variables. The variance is then given by:

$$\mathrm{Var}_{x+y+z} = \frac{\sum_i (x_i-\bar{x})^2}{n} + \frac{\sum_i (y_i-\bar{y})^2}{n} + \frac{\sum_i (z_i-\bar{z})^2}{n} + \frac{2\sum_i (x_i-\bar{x})(y_i-\bar{y})}{n} + \frac{2\sum_i (x_i-\bar{x})(z_i-\bar{z})}{n} + \frac{2\sum_i (y_i-\bar{y})(z_i-\bar{z})}{n} \quad (4)$$

For the distance in RGB and L*a*b* systems, we use the Euclidean distance on the 3 coordinates which is considered as well adapted. In L*a*b* system, the Euclidean distance is also noted ΔE_{76}. For the HLS system, the distance between two colors *a* and *b* is usually given by:

$$|a,b|_{HLS} = \sqrt{min(|a_H - b_H|, 255 - |a_H - b_H|)^2 + (a_L - b_L)^2 + (a_S - b_S)^2} \quad (5)$$

2.5 How to Quantify Error and Performance?

Assessing the quality of segmentation is a complex problem, even if we have some results obtained by an expert [17]. In our application we will use simple evaluation parameters, as we do not look for a very precise and fully automatic segmentation but for a fast process which can be modified by the operator.

The first parameter is the global pixel-to-pixel error *E* between two images *A* and *B* of size *I*×*J*. It is defined by counting the pixels of *A* which are differently labeled than the corresponding pixel in *B*:

$$E = \frac{\sum_i^I \sum_j^J \min\,(1, |A_{ij} - B_{ij}|)}{I \times J} \quad (6)$$

where A_{ij} (resp. B_{ij}) is the label of the pixel of position (*i*,*j*) in the image *A* (resp. *B*). *E* varies between 0 and 1. If *A* is the segmentation given by our method and *B* is the expert segmentation, *E* should be as small as possible.

The second parameter is based on (i) the precision P_L which measures the capability of the method to minimize the over-segmentation of the image, and (ii) the recall R_L which measures the capability of the method to minimize the default of segmentation [12]. These indices are given by

$$P_L = \frac{card\ (TP_L)}{card\ (TP_L) + card(FN_L)} \ and\ R_L = \frac{card\ (TP_L)}{card\ (TP_L) + card(FP_L)} \quad (7)$$

where TP_L (True Positives) are the *L*-labeled pixels of *A* also labeled *L* in *B* whereas FP_L (False Positives) are the *L*-labeled pixels of *A* differently labeled in *B* and FN_L (False Negatives) are the *L*-labeled pixels of *B* differently labeled in *A*. The performance *P* (also called F-measure) of the method is then defined by the harmonic mean of the precision and the recall. This is a normalized coefficient which tends towards 1 when the method gives a result consistent with the expert one.

3 Results and Discussion

Different experimentations have been carried out and discussed to better understand the limits of our method. Algorithms were implemented in Java language, without any

CPU or GPU parallelization, before to be embedded on an Android-based mobile device. The tests were performed on a tablet (Samsung Galaxy Tab 2 with a 1 GHz NVidia® Tegra) and on a standard desktop (with a 3 GHz Xeon® W3550 processor). Segmentation of a color image of 2,048×1,536 pixels takes about 35 s on the tablet and about 7 s on the desktop.

3.1 Comparison of the Labeling Strategies

The 3 different labeling strategies were tested on a reference color image of a tree presented in [11]. The learning areas are two large heterogeneous areas, the blue one for "background" and the green for "leaves" (see Fig. 1.a). All the processing is performed in the RGB color system. The "novelty selection" algorithm described in [11] leads to a very compact segmentation (see Fig 1.b). In particular, all the holes in the tree crown disappear: the feature vector for each pixel was chosen to be a 5×5 image patch centered at the pixel, and over the three color components, resulting in a 75-dimensional feature vector.

Our first strategy leads to an aberrant segmentation (see Fig 1.c). The "leaves" learning area, which is more homogeneous, is processed before the "background" one, and all the sky pixels are then labeled as "leaves" which creates the green traces on the blue. The second strategy does not succeed in keeping background coherence (see Fig 1.d). In fact, the grass color is closer of the average color of the "leaves" learning area than the "background" one which mixes green and blue pixels. The third strategy produces a good compromise which preserves the crown holes while labeling grass as "background" (see Fig 1.e). The residual green traces represent the fence pickets. In fact, the threshold of variance is too large to fragment the "background' learning area in enough precise sub-areas to prevent the picket colors to be closer to one of the "leaves" learning sub-areas.

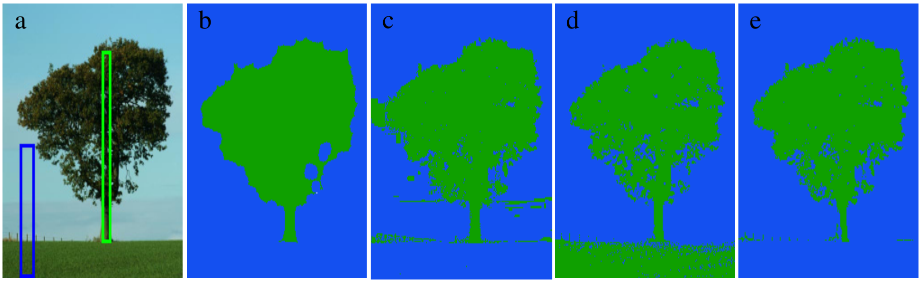

Fig. 1. Comparison of the different labeling method on a 200×221 color image: a- Reference image from [11] with two learning areas labeled as "leaves" (in green) and "background" (in blue); b- Result from [11]; c- Strategy "Sorting the learning areas by increasing variance"; d- Strategy "Finding the closest learning area"; e- Strategy "Fragmenting homogeneous learning areas" with a maximal variance threshold equal to 20. The three labeling strategies were applied with a reduction of the dynamic range to 512 colors.

3.2 Comparisons of the Color Systems

Figure 2 shows the incidence of the color system on the result of the segmentation. In the RGB system (see Fig 2.c), the border of leaves is classified as “wood” due to the high saturation of the color in the peripheral areas. This segmentation default is reduced in the L*a*b* system (see Fig 2.d), leading to a result closer to the expert segmentation (see Fig 2.b).

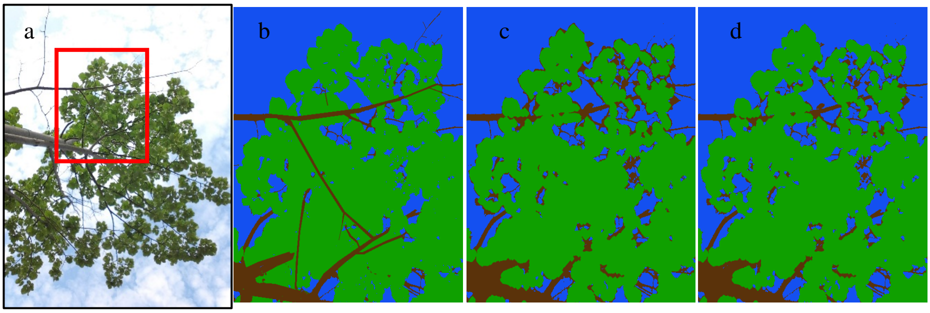

Fig. 2. Color image segmentation: a- 2,048×1,536 pixel color image; b- The image manually segmented in 3 classes by an expert; c- Automatic segmentation in the RGB system. d- Automatic segmentation in the L*a*b* system. In both cases, the third labeling strategy was used with a reduction of the dynamic range to 256 colors and a variance threshold equal to 20.

3.3 Assessment of the Segmentation Method

In the previous section, the experiments showed that the best segmentation is probably produced by the third labeling strategy applied on the L*a*b* system. In this section, we compare different settings and strategies by varying the three following parameters: the color system (C), the quantization rate (Q) which will define the reduction of the color image dynamic range and the labeling strategy (L).

$$Variant = C\{RGB, HLS, L*a*b*\} + Q\{64,128,256\} + L\{S1, S2, S3\} \qquad (8)$$

Each variant can be identified by the decimal value of its ternary code: for example, the decimal code ‘19’ corresponds to the ternary code ‘201’ identifying the C{L*a*b*}+Q{64}+L{S2} variant.

Assessment of results is made on the 2,048×1,536 pixel color image presented in Fig. 2.a, by using the expert segmentation presented in Fig. 2.b. Two learning different areas are used here: the first noted BLA is only composed of 3 large heterogeneous learning areas (1 for each structure) whereas the second noted SLA is composed of 9 smaller homogeneous learning areas (3 for each structure).

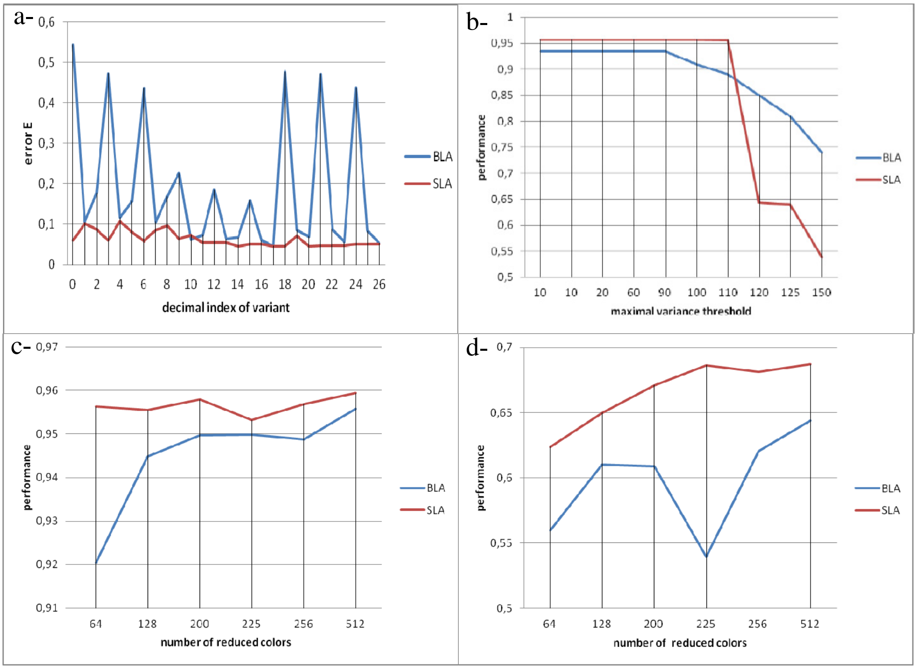

Fig. 3. a- Error with respect to the different variants; b- Incidence of the maximal variance threshold on the labeling of "leaves" label; c- Performance F for "leaves" label; d- Performance F for "wood" label

In Fig. 3.a, the important gap between the two curves BLA and SLA illustrates the sensitivity of the results with respect to the definition of the learning areas. The variants 0, 3, 6, 18, 21 and 24 give a very large error gap which excludes using the strategy "Sorting the learning areas by increasing variance". On the contrary, the minimal values of *E* is reached for the variants 11, 14, 17, 23 or 26 which correspond to the third labeling strategy "Fragmenting homogeneous learning areas". We will then select this strategy.

Fig. 3.b illustrates the influence of the maximal variance threshold on the segmentation results with the third strategy. Below a variance of 100, the performance of the segmentation remains high whatever the definition of the learning areas. So we will fix the maximal variance threshold to 20.

Fig. 3.c shows the incidence of the number of reduced colors on the segmentation result. The performance of the algorithm with the third strategy and the maximal variance threshold of can be considered as satisfying with only 128 colors. Performance is greater than 0.95 with 200 colors whatever the definition of the learning areas (even if better performances are obtained with SLA).

Fig. 3.d illustrates the relative difficulties of our method to segment the "visible branches" of the image which correspond to the "wood" label. The performance value is much lower than for the "leaves" areas, which illustrates two particular aspects:

- The incidence of the expert labeling accuracy on the performance factor.
- The post processing stage strengthens some isolated high saturated color pixels around the tree crown which unfortunately are labeled "wood". These pixels are then counted in the "False Negative" class and penalize the performance indicator.

Fluctuations in results with respect to the two definitions of learning areas show that a semi-supervised method is well adapted to the task. The human operator can interactively adjust the labeling by seeing quickly any progressive adjustment of the learning areas. The post-processing settings are hard to justify: there is no straightforward correlation between the significant size of branches and the number of morphological openings.

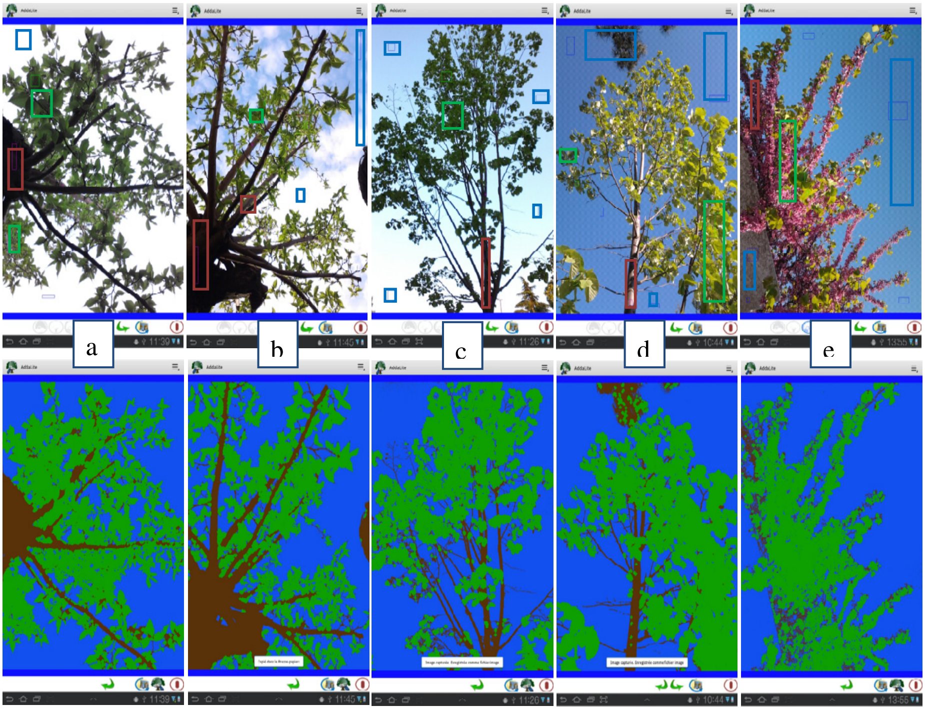

Fig. 4. Embarked semi-supervised segmentation: images are acquired and segmented from a Samsung Galaxy Tab 2. The accuracy and the quality of the result depend not only on the complexity of the image content but also on lightness conditions. Segmentation is satisfying when there is a good lightness without direct sunlight (a, b, c). Light reflection impacts directly the segmentation (d) as the saturated parts of the bark and of the leaves have too close colors to be distinguished. Notice that segmentation works well even in the case of a heterogeneous vegetation, including flowers and leaves (e).

3.4 Practical Examples

This experimentation consists in testing the method, on different in situ images with more or less complex background both in terms of content or lightness. All the images were directly acquired and processed with a Samsung Galaxy Tab 2 tablet.

The results are quite good on images with a soft lightness without direct or sharp sunlight (cf. Fig. 4.a, b, c). Direct sunlight produces reflections and shadows which induce some errors in segmentation: too weakly or strongly saturated colors become too light or dark to remain discriminative (cf. Fig. 4.d).

Several solutions are currently studied to suppress this limit: use of polarizing filters to reduce reflections, acquisition by a stereoscopic device or an infrared captor to increase foliage perception [8]. But the issue is much broader, this is in fact the discriminate power of colors which meets its limit. The separation of the different depth planes is delicate, especially to delineate the different trees which are present in an image. The introduction of a local texture analysis procedure could allow one to solve some configurations. An idea could be to add to color features a textural feature, for example based on Local Binary Pattern [1] which is easy to compute.

4 Conclusion

This paper presents a semi-supervised method for the easy and quick segmentation of in situ tree color images. This work is motivated by the increasing need of quantitative analysis of the crown transparency for computer-aided diagnosis of the urban tree decline. In particular, the method must be enough accurate to evaluate the cumulative area of the crown holes.

The method is based on three independent stages: reduction of the color image dynamic range by quantization based on Wu's method, labeling of image pixels based on learning areas which are defined by the human operator and automatically fragmented in homogeneous sub-areas and post processing to emphasize arrow and thin areas.

The method has been tested with the RGB, HLS and L*a*b* color systems. The best results have been produced with the last system which allows one to define a distance between colors corresponding to the differences perceived by the human eye.

We defined an error and a performance indicator to measure the difference with respect to a segmentation performed by an expert. Experimentations have shown that reducing the native color image to only 256 colors is quite enough and that the method can deal with homogeneous or heterogeneous learning areas as in both cases, these areas being successively fragmented by k-means clustering to become homogeneous.

Our method is quick and easy to use. The semi -supervised method is well adapted to interactively adjust the learning areas in order that final labeling corresponds to the feeling of the human operator.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Trans Pattern Analysis and Machine Intelligence 28(12), 2037–2041 (2006)
2. Busin, L., Shi, J., Vandenbroucke, N., Macaire, L.: Color space selection for color image segmentation by spectral clustering. In: Signal and Image Processing Applications (ICSIPA), pp. 262–267 (2009)
3. Choat, B., Jansen, S., Bordribb, T.J.: Global convergence in the vulnerability of forests to drought. Nature 491, 752–755 (2012)
4. Forgy, E.: Cluster analysis of multivariate data: Efficiency vs interpretability of classifications. Biometrics 21, 768–769 (1965)
5. Friedland, G., Jantz, K., Rojas, R.: SIOX: Simple interactive object extraction in still images. In: Seventh IEEE International Symposium on Multimedia, December 12-14, p. 7 (2005)
6. Jurio, A., Pagola, M., Galar, M., Lopez-Molina, C., Paternain, D.: A comparison study of different color spaces in clustering based image segmentation. IPMU 2(81), 532–541 (2010)
7. Hu, Y.C., Su, B.H.: Accelerated k-means clustering algorithm for color image quantization. Imaging Science Journal 56(1), 29–40 (2008)
8. Hunt Jr., E.R., Hively, W.D., Fujikawa, S.J., Linden, D.S., Daughtry, C.S.T., McCarty, G.W.: Acquisition of NIR-Green-Blue Digital Photographs from Unmanned Aircraft for Crop Monitoring. Remote Sens. 2, 290–305 (2010)
9. Lin, W.J., Lin, J.C.: Color quantization by preserving color distribution features. Signal Process 78(2), 201–214 (1999)
10. Mizoue, N.: Croco: Semi-automatic Image Analysis system for crown condition assessment in forest health monitoring. Journal of Forest Planning 8, 17–24 (2002)
11. Paiva, A.R.C., Tasdizen, T.: Fast semi-supervised image segmentation by novelty selection. ICASSP, 1054–1057 (2010)
12. Powers, D.M.W.: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies 2(1), 37–63 (2011)
13. Seyedhosseini, M., Paiva, A.R.C., Tasdizen, T.: Fast AdaBoost training using weighted novelty selection. In: International Joint Conference on Neural Networks, pp. 1245–1250 (2011)
14. Tan, P., Fang, T., Xiao, J., Zhao, P., Quan, L.: Single image tree modeling. ACM Transactions on Graphics (TOG) 7(5), art. 108 (2008)
15. Teng, C.H., Chen, Y.S., Hsu, W.H.: Tree segmentation from an Image. IAPR Machine Vision Appl., 59-63 (2005)
16. Winn, M.F., Araman, P.A., Lee, S.M.: Urban Crowns: An assessment and monitoring tool for urban trees., Gen. Tech. Rep. SRS-135. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station (2011)
17. Warfield, S.K., Zou, K.H., Wells, W.M.: Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation. IEEE Trans. Med. Imaging 23(7), 903–921 (2004)
18. Wu, X.: Color quantization by dynamic programming and principal analysis. ACM Transactions on Graphics 11(4), 348–372 (1992)