An Accurate Hand Segmentation Approach using a Structure based Shape Localization Technique

Jose M. Saavedra[1,2], Benjamin Bustos[1] and Violeta Chang[1]
[1]*University of Chile, Department of Computer Science, Santiago, Chile*
[2]*ORAND S.A., Santiago, Chile*

Keywords: Hand Segmentation, Hand Localization, Color based Segmentation, Local Descriptors.

Abstract: Hand segmentation is an important stage for a variety of applications such as gesture recognition and biometrics. The accuracy of the hand segmentation process becomes more critical in applications that are based on hand measurements as in the case of biometrics. In this paper, we present a very accurate hand segmentation technique, relying on both hand localization and color information. First, our proposal locates a hand on an input image, the hand location is then used to extract a *training region* which will play a critical role for segmenting the whole hand in an accurate way. We use a structure-based method (STELA), originally proposed for 3D model retrieval, for the hand localization stage. STELA exploits not only locality but also structural information of the hand image and does not require a large image collection for training. Second, our proposal separates the hand region from the background using the color information captured from the training region. In this way, the segmentation depends only on the user skin color. This segmentation approach allows us to handle a variety of skin colors and illumination conditions. In addition, our proposal is characterized by being fully automatic, where a user calibration stage is not required. Our results show a 100% in the hand localization process under different kinds of images and a very accurate hand segmentation achieving over 90% of correct segmentation at the expense of having only 5% for false positives..

1 INTRODUCTION

Hand segmentation has become very important for many applications such as those related to vision-based virtual reality (Yuan et al., 2008), gesture recognition (Wachs et al., 2011), and biometric recognition (Huang et al., 2008; Yörük et al., 2006). Furthermore, the hand segmentation step turns more critical in cases where a depth analysis of the hand is required. This kind of analysis could imply getting some measures from the hand image. For instance, in the case of hand biometrics getting reliable hand measurements is essential. Commonly, hand-based biometrics require a special device to capture the hand properties limiting its usability. This problem could be solved if we could get hand measurements directly from a digital cam. Of course, this would require an accurate segmentation stage. Yoruk et al. (Yörük et al., 2006) proposed a method to carry out this process. However, in their work, they consider that the image background is dark and homogenous leading to a trivial segmentation stage.

Commonly, hand segmentation is carried out by skin color based techniques (Jones and Rehg, 2002; Lew et al., 2002). These techniques build a generic model (e.g. a statistical model) from a large collection of training skin images. The model will decide whether a pixel in the image is actually a hand point or not. An important study on this kind of model proposed by Jones et al. (Jones and Rehg, 2002) is based on a *Gaussian mixture model*. A critical drawback of a skin color based technique is its low performance under illumination variations. Moreover, this kind of technique performs poorly in presence of skin color-like regions. Figure 1 shows a segmentation result using the statistical model proposed by Jones et al. (Jones and Rehg, 2002) for skin segmentation.

Instead of using a generic color model one could use a model dependent on the user skin color leading to an adaptive approach. The adaptive term arises due to the fact that the color model will be adapted to the user skin color. This approach allows us not only to tackle the diversity of skin color but also to handle diverse illumination conditions.

In the adaptive segmentation approach, a hand portion called *the training region* needs to be previ-

Saavedra J., Bustos B. and Chang V..
An Accurate Hand Segmentation Approach using a Structure based Shape Localization Technique.
DOI: 10.5220/0004288703210326
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 321-326
ISBN: 978-989-8565-47-1

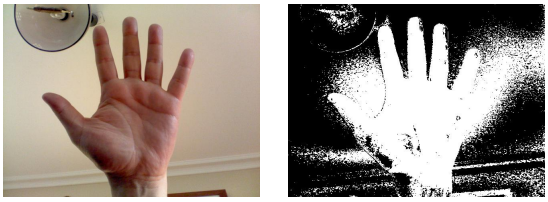

Figure 1: A low accurate segmentation result using a skin color based approach (Jones and Rehg, 2002). The white pixel on the right binary are the pixels detected as hand points.

ously marked on an input image. The pixels inside the marked region are used to build a color model that will be used to segment the rest of the hand by means of color similarity. Following this idea, Yuan et al. (Yuan et al., 2008) proposed an algorithm that makes up color clusters using a training region and then labels the clusters as hand or background depending on the size of each cluster. Finally, the image points are classified depending on what cluster they belong to. The problem with this algorithm is that it requires the user to mark the training region manually, an undesired task in automatic environments.

To get a training region automatically, we could locate the hand on the input image. Using this location we could mark an appropriate hand region, commonly using the location coordinates as the center of such a region.

For the localization problem, the Viola and Jones detector (Viola and Jones, 2002) may be used. The problem with this approach is that it requires large training image collection and it is time consuming for the training. Because a hand is a simple object with a well defined shape, an expensive training stage is unnecessary. Moreover, the localization problem may be carried out using a local structure-based approach exploiting not only locality information but also the structure of the hand shape.

Our contribution in this work is to present a very accurate hand segmentation technique composed of two main steps: (1) estimate the hand location on an image, and (2) separate the hand region from the background. For the localization stage, we use a local structure-based approach exploiting both structural and locality information of a hand. Structural information is related to the components forming a hand and locality information is related to the spatial relationship between these components. To this end, we use the STELA (StrucTurE-based Local Aproach) method proposed by Saavedra et al. (Saavedra et al., 2011). For the segmentation stage we extend the idea of Yuan et al. (Yuan et al., 2008) proposing strategies to compute the underlying parameters. In this case, we make up color clusters from a training region obtained directly from the localization stage (a manual localization is no longer required). For color representation we use only the chromatic channels of the L*a*b* color space as suggested by Yuan et al. (Yuan et al., 2008). The segmentation stage ends with a post-procesing phase to reduce imperfections caused by noise. An example of our results is shown in Figure 2 where the segmentation is specified by a blue contour.

Figure 2: A example of hand segmentation using our approach. The blue contour defines the segmented region.

The remaining part of this document is organized as follows. Section 2 describes the local structure based approach (STELA) which our proposed method relies on. Section 3 describes in detail the hand segmentation process. Section 4 presents the experimental evaluation, and finally, Section 5 discusses some conclusions.

2 STELA

STELA is a structure-based approach proposed by Saavedra et al. (Saavedra et al., 2011) for retrieving 3D models when the query is a line-based sketch. A STELA descriptor is invariant to translation, scaling, and rotation transformation. The main property of this approach is that it is based not only on the structural information but also on the locality information of an image.

For getting structural information, an image is decomposed into simple shapes. This method uses straight lines as primitive shapes which are named *keyshapes*. For getting locality information, local descriptors taking into account the spatial relationship between *keyshapes* are used. An interesting property of keyshapes is that these allow us to represent an object in a higher semantic level.

STELA consists of the following steps: (1) *get an abstract image*, (2) *detect keyshapes*, (3) *compute local descriptors*, and (4) *match local descriptors*.

1. **Abstract Image.** The abstract image allows us to reduce the effect of noise, keeping only relevant edges. To this end, STELA applies the *canny* op-

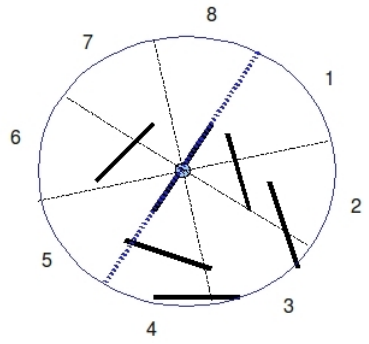

Figure 3: A synthetic representation of the partitioning to make up the STELA descriptor.

erator (Canny, 1986) over the image in order to have an edge map representation. Then, to approximate strokes, the method uses the *edgelink* operator (Kovesi, 2000) which returns a set of edge lists.

2. **Detecting Keyshapes.** Detecting simple shapes that compose a more complex object allows us to take into account structural information in the similarity measurement. STELA detects only straight lines which form an object. Lines can be detected easily and still maintain enough discriminative information. These lines are referred as *keyshapes* following the same idea of *keypoints* (Mikolajczyk and Schmid, 2004) even though *keyshapes* are more representative in terms of the object structure.

   Finally, the center of each line is taken as the representative point of each *keyshape*. In STELA, each *keyshape* L is represented as a 5-tuple $[(x_1,y_1),(x_2,y_2),(x_c,y_c),s,\phi]$, where (x_1,y_1) is the start point, (x_2,y_2) is the end point, (x_c,y_c) is the representative point, s is the line length, and ϕ is the corresponding slope.

3. **The Local Descriptor.** A local descriptor is computed over each *keyshape*. STELA uses an *oriented angular 8-partitioning descriptor*. Figure 3 depicts a graphical representation of this descriptor.

   Having a keyshape L as reference, this descriptor works as follows:

   - Create a vector h, containing 8 cells. Initially, $h(i) = 0$, $i = 1 \ldots 8$.
   - Let L be the reference keyshape represented as :
     $$L = [(x_1,y_1),(x_2,y_2),(x_c,y_c),s,\phi'] \quad (1)$$
   - Let $f_r : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ be a rotation function around the point (x_1,y_1) with rotation angle $\beta = -\phi$.
   - Let $(\hat{x}_c,\hat{y}_c) = f_r(x_c,y_c)$ be the normalized version of (x_c,y_c).
   - For each keyshape $Q \neq L$ represented by $[(x'_1,y'_1),(x'_2,y'_2),(x'_c,y'_c),s',\phi']$.
     - Get $(\hat{x}'_c,\hat{y}'_c) = f_r(x'_c,y'_c)$.
     - Determine the corresponding *bin* of $(\hat{x}'_c,\hat{y}'_c)$ on the oriented partitioning scheme.
     - $h(bin) = h(bin) + s'/MAX\_LEN$, where MAX_LEN is the length of the abstract image diagonal.
   - Finally, $h(bin) = \dfrac{h(bin)}{\sum_{i=1}^{8} h(i)}$, $bin = 1 \ldots 8$.

4. **Matching.** An image is treated as a set of descriptors. This set captures the object shape. The matching problem may be regarded as an instance of the *Bipartite Graph Matching*. STELA resolves the assignment problem applying the Hungarian Method. The cost of matching two local descriptors p and q could be thought of as the distance between p and q. In this way, the less similar the descriptors are, the more expensive the match becomes. As the STELA descriptor is a probability distribution, it uses the χ^2 statistic test as distance function.

   For estimating the pose transformation, this method uses the Hough Transform, where each candidate match must vote just for three parameters (scale, translation in x-axis and y-axis). The set of parameters with the highest score is retained. This set of transformation parameters characterizes the estimated pose. Only the matches which agree with the estimated pose are retained for the next process, the others are discarded.

   Finally, the dissimilarity between two images is computed as the average cost of the matched descriptors. The cost of the unmatched descriptors is set to 1.

3 PROPOSED HAND SEGMENTATION METHOD

We divide our approach in two stages, the first one corresponds to the hand localization, where STELA is applied to estimate the location of the occurrence of a hand. After this, we extract a training region from the center of the located region and proceed to segment the hand using the color information given by the training region. Finally we carry out a post-processing stage to make our result more robust to environment conditions like illumination or skin color-

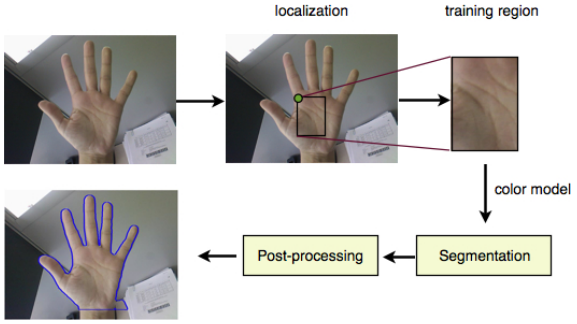

Figure 4: Proposal framework.

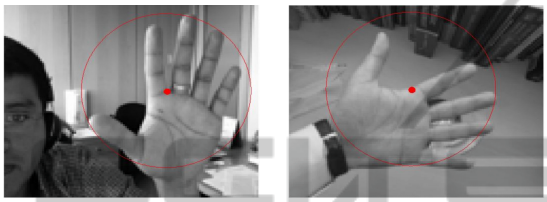

Figure 5: Two examples of hand localization.

like regions near to the hand region. A framework of our proposal is presented in Figure 4.

3.1 Hand Localization

Unlike the machine learning approach for detecting objects, our approach only requires a simple hand prototype. In our implementation we use a 80×80 hand prototype image.

We use STELA to estimate the hand location in an image. Since we are interested in detecting a hand shape we could make STELA faster reducing the image size. We resize the input image to a 100×120 image. To appropriately determine the location of a hand, we apply the sliding window strategy. Each region inside a 80×80 window is compared with the hand prototype by STELA. The center of the window keeps the dissimilarity value between the windowed region and the prototype.

Let d_{ij} be the STELA dissimilarity value for each pixel (i, j) in the resized input image. We define D as a set of points where the dissimilarity value is close to the minimum dissimilarity value (mdv). That is:

$$D = \{(i, j) : d_{ij} - mdv \leq 0.1, i = 1 \ldots 100, j = 1 \ldots 120\},$$

Finally, the occurrence of a hand is located in the centroid (i_c, j_c) of D only if the $mdv \leq TH_h$. This means that if we have $mdv > TH_h$ the method will report a *hand not found* message. In our experiments we set $TH_h = 0.65$. If a hand is located, the centroid (i_c, j_c) needs to be rescaled to have the real location. Two examples of hand localization are shown in Figure 5 where we notice that the proposed method works even when the hand undergoes rotation variations.

3.2 Hand Segmentation

After locating the hand, we extract a 150x100 region where the upper left corner corresponds to the hand location point (see Figure 4). A great number of pixels inside this region must correspond to the hand. We will refer to this region as the *training region*.

We proceed to make up color clusters. In this way, each pixel of the training region must fall within one of the built clusters. We represent the image by the L*a*b* color space. We only use the chromatic channels $a*$ and $b*$ similiar as the proposal of Yuan et al. (Yuan et al., 2008). Though we have conducted experiments using different color spaces, we got better results using L*a*b*.

Each color cluster q needs to keep two values, the first one corresponds to the number of pixels falling inside it (N_q), and the second one is a representative point of the cluster (this is not necessarily a real pixel), expressed in terms of its corresponding a* b* color information. We represent this point as $[r_q^a, r_q^b]$. The representative pixel corresponds to the average of the a* b* components of all pixels falling inside q.

We make up the clusters following this given algorithm:

- $SoC = \emptyset$ (set of clusters)
- $N = 0$ (number of clusters)
- For each pixel p in the training region
  - Let [a,b] the corresponding a*b* color information of p.
  - $m = min_{1 \leq q \leq N}(L_2([a,b],[r_q^a, r_q^b]))$
  - $q^* = argmin_{1 \leq q \leq N}(L_2([a,b],[r_q^a, r_q^b]))$
  - If $m < TH$, update $[r_q^a, r_q^b]$ and increase N_q by 1.
  - Otherwise, add a new cluster w to SoC using p, increasing N by 1. In this case, $[r_w^a, r_w^b] = [a,b]$ and $N_w = 1$.
- return SoC

In the previous algorithm $TH = 0.01$ and L_2 corresponds to the Euclidean distance.

After building the set of clusters SoC, we segment a hand starting from the pixel corresponding to the hand location point expanding the process to the rest of the pixels using the *breadth first search* strategy through the pixels detected as hand point. This strategy avoids detecting objects that are far from the hand region, minimizing the false positive points. The algorithm determines what cluster a pixel p must belong to. An appropriate cluster $q\star$ for the pixel p must satisfy two criteria:

1. The number of pixels in $q\star$ must be greater than TH_N. Clusters with few pixels may correspond to the background.
2. Considering only clusters satisfying the first criterion, the cluster $q\star$ corresponds to that with minimum distance (md) between p and each cluster representative point. Here, the distance function is the *Euclidean distance*.

If $md > TH_D$ the pixel is marked as background, otherwise it is marked as a hand point.

In our implementation TH_N is the median of the cluster sizes. $TH\_N = median(N_1, \ldots, N_N)$. In the case of TH_D we conduct a different strategy. The main idea is that TH_D has to be computed depending on the distances between hand points from the training region. Therefore TH_D is a distance that allows us to discard the 10% of the training region points with higher distance value whit respect to the cluster they belong to.

3.3 Post-processing

After the segmentation stage we have a binary representation where detected hand points are set to 1 and background points are set to 0. To have an accurate segmentation we apply two post-processing operations. First, the method discards a point detected as hand point if the number of hand points in a local region around the point is less than 50% of the local region size. In this case, we use a 21x21 local region. Second, the method applies morphological operations to fill holes in the hand region (Soille, 1999).

4 EXPERIMENTAL EVALUATION

In this section we show results of the automatic hand segmentation using our proposal. To test our approach, we used a collection of 25 640×480 images containing a hand captured with different kinds of illumination. In terms of pixels, our collection is composed of 5,761,985 hand pixels (positive set) and 18,084,503 non-hand pixels (negative set). We compare our result against a renowned skin color model, specifically we use as baseline as proposed by Jones et al. (Jones and Rehg, 2002).

A good tool to assess the performance of our segmentation results is the ROC curve. In this way, we have a mask indicating the hand region for each image. ROC curve analysis takes into account the area under the curve (AUC) as a quality measure. The higher the AUC value is, the better the quality of our method is. Furthermore ROC curves show us the cost

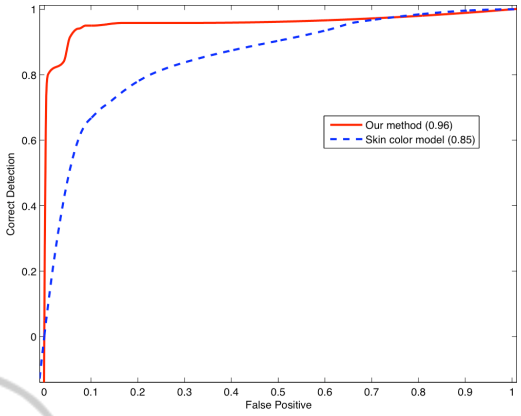

Figure 6: ROC curves comparing our method with the skin color model proposed by Jones (Jones and Rehg, 2002). The AUC value is indicated in the legend.

Table 1: Correct detection rate (CD) vs. false positive rate for our method and the skin color model proposed by Jones (Jones and Rehg, 2002).

CD	FP (Our method)	FP (Skin color model)
0.95	0.146	0.641
0.90	0.052	0.490
0.85	0.046	0.334
0.80	0.009	0.234

we have paid in terms of false positives when a high correct detection is desired. The ROC curve comparing our method with the skin color model is presented in Figure 6.

Our method achieves an AUC value of 0.96 as the skin color model only achieves 0.85. In addition to the overall good performance of our method, it is worth pointing out that our proposal achieves a correct detection rate over 90% at the expense of having only 5% for false positives. The skin color model results in 49% of false positives to achieve a comparable result. In Table 1 the relationship between false positive and correct detection is shown for both evaluated methods.

Figure 7 shows how well our method segments a hand in an image in comparison with the baseline method. Additionally, six examples of hand segmentation using our method are depicted in Figure 8.

Finally, we have to say that our method correctly locates a hand for different images. In our experiments we get 100% of correct localization. Therefore, any application requiring a hand localization step could take advantage of our proposal, hand tracking and hand biometrics are two potential applications.

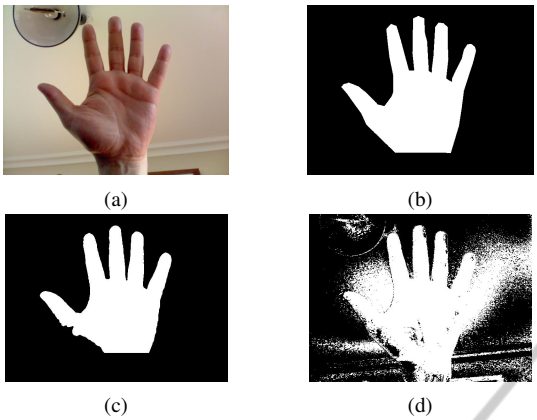

Figure 7: Hand segmentation comparison. (a) Input image, (b) target segmentation, (c) output using our method, and (d) output using the skin color model.

Figure 8: Examples of hand segmentation using our proposed approach.

5 CONCLUSIONS

We have presented a novel approach for the hand segmentation task. Our method estimates the hand location to capture a training hand region. Then, we propose an adaptive color model based on the image's skin color to segment the hand. This allows us to handle illumination changes and diverse skin colors. We compare our method with a skin color based model, achieving notable improvement in the accurate segmentation. One advantage of our method is that it segments the hand in an accurate way without requiring a lengthy time consumption for the training stage.

Additionally we have presented a novel approach for hand localization which could be used in other applications like hand tracking for instance.

We would like to extend our proposal in order to focus on the segmentation of other kinds of objects as well as that within the context of hand biometrics.

REFERENCES

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6).

Huang, D.-S., Jia, W., and Zhang, D. (2008). Palmprint verification based on principal lines. *Pattern Recognition*, 41.

Jones, M. J. and Rehg, J. M. (2002). Statistical color models with application to skin detection. *Int. Journal Comput. Vision*, 46.

Kovesi, P. D. (2000). MATLAB and Octave functions for computer vision and image processing. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

Lew, Y., Ramli, A., S.Y.Koay, Ali, R., and Prakash, V. (2002). A hand segmentation scheme using clustering technique in homogeneous background. In *Proc. of 2nd Student Conference on Research and Development*.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

Saavedra, J. M., Bustos, B., Scherer, M., and Schreck, T. (2011). STELA: Sketch-based 3d model retrieval using a structure-based local approach. Submitted to ACM-ICMR.

Soille, P. (1999). *Morphological Image Analysis: Principles and Applications*. Springer-Verlag Telos.

Viola, P. and Jones, M. (2002). Robust real-time object detection. *Int. Journal of Computer Vision*.

Wachs, J., Klsch, M., Stern, H., and Edan, Y. (2011). Vision-based hand gesture interfaces: Chall. and innov. *Communications of the ACM*.

Yörük, E., Dutağaci, H., and Sankur, B. (2006). Hand biometrics. *Image and Vision Computing*, 24:483–497.

Yuan, M., Farbiz, F., Manders, C. M., and Tang, K. Y. (2008). Robust hand tracking using a simple color classification technique. In *Proc. of The 7th ACM SIGGRAPH Int. Conf. on Virtual-Reality Continuum and Its Applications in Industry*.