

A Mobile-Oriented Hand Segmentation Algorithm Based on Fuzzy Multiscale Aggregation

Ángel García-Casarrubios Muñoz, Carmen Sánchez Ávila,
Alberto de Santos Sierra, and Javier Guerra Casanova

Group of Biometrics, Biosignals and Security, GB2S, Centro de Domótica Integral
Universidad Politécnica de Madrid
{agarcia, csa, alberto, jguerra}@cedint.upm.es

Abstract. We present a fuzzy multiscale segmentation algorithm aimed at hand images acquired by a mobile device, for biometric purposes. This algorithm is quasi-linear with the size of the image and introduces a stopping criterion that takes into account the texture of the regions and controls the level of coarsening. The algorithm yields promising results in terms of accuracy segmentation, having been compared to other well-known methods. Furthermore, its procedure is suitable for a posterior mobile implementation.

1 Introduction

Nowadays, one area of interest which is undergoing a continuous development is Biometrics [1]. Within the field of Biometrics, one of the techniques most widely used is related to the recognition of a person through their hand [2], since, as fingerprint, this physical characteristic is different for each person. Then, in order to carry out this identification, it will be necessary to separate first the required object (hand) from the background of the image. In that regard, image segmentation is the branch of image processing theory [18] that studies different techniques with the main aim of separating an object from the background within a digital image [3]. Specifically, this article presents an image segmentation algorithm aimed at segmenting hand images acquired by a mobile phone, being part of a whole biometric technique oriented to mobile devices for daily applications (bank account access, pin codes and the like).

In the literature, there are a wide number of algorithms that address the segmentation problem from different mathematical approaches [4]. One family of segmentation algorithms that have experienced a great development in recent years are those based on multiscale aggregation [5]. These methods try to segment an image by finding the objects directly, that is, finding the regions that compose the image instead of focusing on detecting the possible existing edges. Moreover, recent results obtained by these algorithms have shown improvement compared to other methods [11], like the Normalized Cuts method [12] and Mean-Shift method [13].

Within the family of multiscale aggregation methods, there are also a wide range of algorithms that differ in the type of mathematical operations applied to image pixels. The algorithms based on Segmentation by Weighted Aggregation (SWA [6]) have shown good results by means of similarities between intensities of neighboring pixels

[7] and with measurements of texture differences and boundary integrity [8]. Other approaches include more complicated operations, such as Gradient Orientations Histograms (GOH [9]), or more straightforward grouping techniques based on the intensity contrast between the boundary of two segments and the inside of each segment [10]. Moreover, thanks to its definite structure, SWA methods may be even used in conjunction with other approaches [20].

Even though our algorithm presents a structure based on SWA methods, there are several differences that must be taken into account:

- SWA algorithms identify each segment by one representative node [9], so that all the pixels which form the segment are strongly connected to that representative node. However, in our algorithm segments are identified by a number of characteristic measures. These measures gather information of each segment by computing the average intensity, variance and position centroid.
- SWA methods start the aggregation process by selecting a few seed nodes [9], satisfying the condition that all pixels are strongly connected to at least one seed node. In contrast, our algorithm builds the first scale by the direct formation of the first segments, so that they are formed by a few pixels similar to each other.
- SWA methods transfer information between scales through the interpolation coefficients p_{ik} , where p_{ik} represents the probability that node i belongs to the segment represented by the node k [19]. However, in our algorithm there is no need to compute any interpolation coefficient, because the segments in each scale collect the information of the segments from the previous scales. In addition, we use Delaunay triangulation [14] to include spatial information, so that it provides in each scale the segments that are more likely to be grouped. This considerably reduces the computational cost, since the number of weights to calculate is much smaller than the number of interpolation coefficients used in SWA methods.

2 Problem Statement

Given an image I containing $M \times N$ pixels, a graph $G = (V, E, W)$ is constructed, where nodes $v_i \in V$ represent pixels in I , edges $e_{ij} \in E$ connect pixels v_i and v_j according to a defined neighborhood system (namely a 4-connected structure) and weights w_{ij} are associated with each edge e_{ij} , indicating to what extent two nodes v_i and v_j are similar. This implementation proposes the use of fuzzy logic for w_{ij} definition.

The idea of segmentation consists of dividing image I into T segments. Let $s_t \subseteq V$ (with $t = 1, 2, \dots, T$) be a segment gathering a set of pixels with similar properties in terms of w_{ij} . Concretely, the aim of this document consists of dividing the image I into $T = 2$ segments, distinguishing hand from background. However, sometimes it is difficult to achieve a result of $T = 2$ segments, and thereby it is important to isolate completely the hand in one only segment, even though the background may be formed by several segments.

Therefore, the problem is stated as follows: divide a given image I into segments s_t , with $t = 1, 2, \dots, T$, so that the hand is completely isolated in one segment from the

background. Next section 2.1 will explain how nodes from V are assigned to segments s_i through different scales.

2.1 The Algorithm

Before starting with the main procedure of segmentation, some considerations on image I must be stated. Image I is obtained from a mobile device (section 3) and it represents a color image constructed by the RGB color space [15]. However, since the purpose of our algorithm is segmenting hand images, it is evident that the use of techniques for human skin detection might be useful. In that regard, we propose the use of the Lab color space (abbreviation for *CIE 1976 L*, a*, b**) [16]. Empirically, we found that the best results are obtained with the use of layer b, so $I \leftarrow b$. Notice that the calculation of the layer b from a RGB image is a complicated process that is beyond the scope of this article, see [16] for more information about this conversion.

In Fig. 1 we present the steps followed in the construction of the algorithm's structure and they will be explained below.

1. If we consider $G^{[s]}$ the representation of the image in the scale s , then $G^{[0]} = I$. In this first scale, weights w_{ij} are calculated according to the following Gaussian-like fuzzy distribution (Eq. 1):

$$w_{ij} = \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right) \tag{1}$$

where I_i and I_j are the image intensities of I at positions i and j , and $\sigma > 0$ is a parameter responsible for accuracy in terms of segmentation, because if $\sigma \rightarrow \infty$, then $w_{ij} \rightarrow 1$. In our algorithm, for the sake of accuracy, this parameter is fixed to $\sigma = 0.01$.

Once the graph G is obtained, the aggregation algorithm proceeds by sorting every pair of nodes (namely v_i and v_j) according to their weight w_{ij} in descending order. In other words, first pairs (those with higher values regarding w_{ij}) are more similar, and therefore they deserve to belong to the same segment. Taking into account that an unassigned segment s_i is denoted by $s_i = 0$, the aggregation process is carried out until every node in V is assigned to a segment, according to Eq. 2.

$$(s_i, s_j) = \begin{cases} (s_q, s_q) & \text{if } s_i = s_j = 0 \\ (s_i, s_i) & \text{if } s_i \neq 0, s_j = 0 \\ (s_j, s_j) & \text{if } s_i = 0, s_j \neq 0 \end{cases} \tag{2}$$

where labels s_i and s_j are assigned to nodes v_i and v_j indicating to which segment they belong, according to three possible situations:

- If both nodes have not been assigned yet, then both nodes are assigned to the new segment s_q , being $q \leq M \times N$.
- If in a given edge e_{ij} , only one of the nodes has been assigned to a certain segment label (s_i or s_j depending on the case), then both nodes are assigned to that segment.
- If in a given edge e_{ij} , both nodes have been assigned to a segment label, then both nodes remain assigned to the corresponding segment.

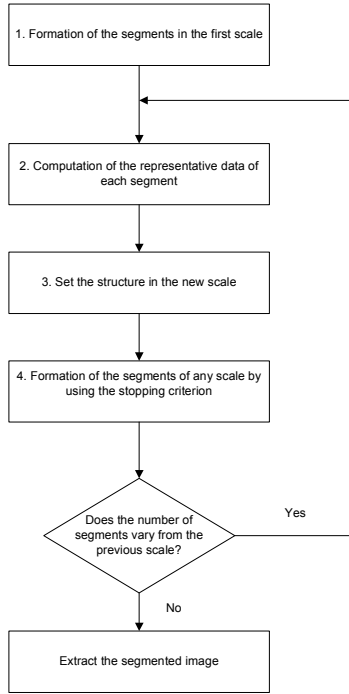


Fig. 1. Flow chart of the algorithm's structure

2. As it was mentioned in section 1, our algorithm does not identify each segment by a unique representative node. In contrast, it carries out that identification through measures that collect the characteristic information of each segment. Those measures are: position centroid ξ (Eq. 3), average intensity μ (Eq. 4) and variance σ^2 (Eq. 5). The centroid position is calculated based on the physical position of nodes within image I and is used later to find the neighbors of every segment. Considering that a node v_i has two components (x_{v_i}, y_{v_i}) , the centroid is computed as follows:

$$\xi = (\bar{x}_{s_i}, \bar{y}_{s_i}) \tag{3}$$

where x_{s_i} and y_{s_i} represent the coordinate locations of nodes within segment s_i .

Similarly, average intensity is obtained by averaging the node intensities in v_{s_i} , as follows:

$$\mu_i = (\bar{I}_{v_{s_i}}) \tag{4}$$

where v_{s_i} indicates the set of nodes within segment s_i .

Finally, the variance provides information about the texture of the segment, as it takes into account its dispersion. The variance is calculated, based on previous Eq. 4:

$$\sigma_i^2 = \frac{1}{N_{s_i}} \sum_{i=1}^{N_{s_i}} (I_{v_{s_i}} - \mu_i)^2 \tag{5}$$

where N_{s_i} represents the number of nodes from V in segment s_i .

Therefore, from the second scale, a segment s_i is defined as a three-vector component $s_i = (\zeta_i, \mu_i, \sigma_i^2)$, facilitating in subsequent scales the recursive procedure, where the representative data of the segments in a scale s become the nodes in the new scale $s+1$.

3. Due to the aggregation process shown before, the 4-neighbor structure is not conserved, so it is necessary to build a new structure for each scale. This new structure is created by the Delaunay triangulation [17], which establishes a network based on the centroids ζ_i computed in the previous scale. Although the number of neighbors cannot be fixed for each node of the new scale, this algorithm provides a quick way to set every node's neighbors based on their proximity.
4. In order to calculate the weights associated to the new edges, the same Gaussian-like fuzzy distribution shown before (Eq. 1) is used, although in this case it computes the average intensities of the corresponding segments, μ_i and μ_j .

One problem that is encountered when using the aggregation process mentioned above is that it does not stop aggregating nodes until it reaches the end of the list of edges, which means that nodes connected by weak edges (at the bottom of the list) would be added. To solve this problem, we propose the introduction of a stopping criterion that takes into account the texture of the segments. So, if this stopping criterion is not satisfied by a pair of nodes, they will not be aggregated under the same segment. Thus, two nodes v_i and v_j of any scale will be aggregated only if the following condition is satisfied:

$$\sigma_{ij}^2 \leq \sqrt{\sigma_i^2 \cdot \sigma_j^2} + k \tag{6}$$

where σ_{ij}^2 represents the variance of the likely aggregate from nodes v_i and v_j , σ_i^2 and σ_j^2 represent the cumulative variances of the nodes v_i and v_j , and k is a constant set based on empirical results (section 4) that controls the level of coarsening, i.e. the number of remaining segments at the end of the process.

This condition will yield better results regarding segmentation, although k must be tuned to obtain $T = 2$ segments, since the algorithm get blocked in a point where no more segments are aggregated, and therefore $T \geq 2$, in general. As future work, this requirement must be refreshed dynamically in order to obtain $T = 2$ segment labels automatically.

3 Database Acquisition

An important aspect of a segmentation algorithm regards the data involved to validate the proposed approach. Since this document presents a segmentation algorithm oriented for mobile devices, it is obvious that the data must be acquired with the selected

device. In other words, the data acquisition consists of images acquired by an iPhone Mobile device. The dimensions of each image are 2 Mega-pixels and 1600 x 1200 pixels, with a resolution of 72 dpi. Furthermore, each image was taken with the hand open, considering a distance of 15-30 cm between the device and the hand. The database gathers 50 users of a wide range of ages (from 16 to 60) and containing different races. Moreover, the images were taken under uncontrolled illumination settings and hands were not required to be placed on a platform. The image acquisition procedure requires no removal of objects such as rings, bracelets, watches and so forth, so that the image is taken non-invasively. The database is publicly available at <http://sites.google.com/site/engb2s/databasehand>.

4 Results

The evaluation of a segmentation method is a difficult task. In Fig. 2 the reader may observe the degree of accuracy achieved when segmenting the hand from the background.



Fig. 2. Eight visual examples of the algorithm’s performance

Apart from the obvious way of a direct observation, some quantitative evaluation methods have been used in order to validate our algorithm.

First, the performance of the segmentation has been assessed based on the ratio F-Measure [11] (Eq. 7), which compares our algorithm with ground truth segmentations. These ground truth segmentations were obtained by manually segmenting the images into two classes, foreground and background.

$$F = \frac{2RP}{R + P} \tag{7}$$

where R (Eq. 8) and P (Eq. 9) are the Recall and Precision values of a particular segmentation, according to [11].

$$R = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \tag{8}$$

$$P = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}} \tag{9}$$

In Fig. 3 (left) the results of the F-Measure for different images and the values of the constant k (stopping criterion) are provided. Notice the high scores achieved, validating the accuracy of the algorithm.

In addition, we have simulated the blurry effect that occurs sometimes when taking a photograph (Fig. 3 - right). For that, the original images were eroded by motion filters that approximate, once convolved with an image, the linear motion of a camera by L pixels, with an angle of θ degrees. Notice that, in spite of the erosion, the scores are over 96%. Moreover, if the original images are eroded by a Gaussian filter of variance $\sigma^2 \leq 20$, the scores are still over 90%.

Furthermore, it is provided a comparison with two well-known methods (Fig. 4), “Normalized cuts and image segmentation” (implementation in MATLAB available¹) [12] and “Efficient graph-based image segmentation” (according to our implementation) [10], resulting in an outperformance by our approach. It is remarkable that our algorithm allows the implementation of a light software program, which is proved by the fact that it performs considerably well with relatively large images (800x600 pixels computed in 36 seconds), whereas the software implementations of other algorithms, like Normalized Cuts, only perform well with smaller images (160x160 pixels in 13 seconds). In fact, computing a small image (160x160 pixels) with our method took only 4 seconds, meaning that our algorithm is over 3 times faster than the Normalized Cuts algorithm. The implementation of this algorithm has been carried out on a PC Computer @1.8 GHz, with MATLAB 7 R14. A mobile oriented implementation remains as future work.





Image	F-Measure	Constant k
	0.99 ± 0.013	0.003
	0.99 ± 0.016	0.003
	0.99 ± 0.011	0.001
	0.99 ± 0.012	0.001





Image	F-Measure	L	θ	Constant k
	0.99 ± 0.015	20	10	0.003
	0.97 ± 0.016	30	20	0.003
	0.96 ± 0.013	40	20	0.003
	0.97 ± 0.015	40	30	0.003

Fig. 3. Accuracy of the algorithm (left) and performance when eroding the original image with a filter that approximates the linear motion of a camera by L pixels, with an angle of θ degrees (right)

¹ MATLAB implementation at <http://www.cis.upenn.edu/~jshi/>

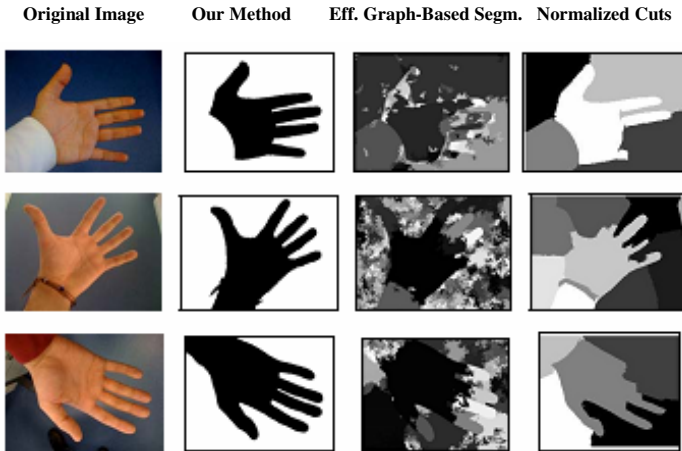


Fig. 4. Results of applying our method compared to other state of the art algorithms

5 Conclusions

We have introduced a multiscale algorithm for image segmentation and aimed for biometric purposes.

The algorithm uses a process of recursive aggregation in order to group the pixels and segments which share the most number of properties. In this article we have focused on the gray intensity and the Lab color space, but the structure of the method is designed to introduce as many features as the user may like, such as second order statistical measures. These properties are computed recursively by weights between pixels and segments, taking into account that fuzzy techniques are used for their conformation. Moreover, the algorithm is flexible enough to allow the user to introduce new fuzzy functions in order to calculate the weights.

We have also presented a stopping criterion that computes the texture of each segment and controls the level of coarsening by adjusting the parameter k .

In addition, one remarkable feature of the algorithm presented is the lack of interpolation matrix, which contributes to the lightness and fastness of the method. This lack of the interpolation matrix is in certain way compensated by the introduction of a Delaunay Triangulation algorithm, which is responsible for the assignment of neighbors in every scale. Moreover, we have introduced the concept of using the centroid of a segment as a way of representing its spatial properties and, therefore, allowing the Triangulation algorithm to create every new graph properly.

We have also introduced a new fast grouping technique that works with the representative properties of each segment. This algorithm receives the graph obtained by the Triangulation method and assigns every segment of the current scale to a new one of the next scale, taking into account that the weight between a pair of segments determines the likelihood of those segments to be grouped.

Finally, we have evaluated the efficiency of our quasi-linear algorithm and we have compared it to the “Normalized Cuts” algorithm and to the “Efficient-graph based segmentation” method, turning out that our algorithm drew better results.

It is remarkable that our algorithm allows the implementation of a light software program, which is proved by the fact that it performs considerably well with relatively large images (800x600 pixels computed in 36 seconds), whereas the software implementations of other algorithms, like Normalized Cuts, only perform well with smaller images (160x160 pixels in 13 seconds). In fact, computing a small image (160x160 pixels) with our method took only 4 seconds, meaning that our algorithm is over 3 times faster than the Normalized Cuts algorithm.

Acknowledgment

This research has been supported by the Ministry of Industry, Tourism and Trade of Spain, in the framework of the project CENIT-Segur@, reference CENIT-2007 2004. (<https://www.cenitsegura.es>).

References

1. Li, Y., Xu, X.: Revolutionary Information System Application in Biometrics. In: International Conference on Networking and Digital Society, ICNDS 2009, May 30-31, vol. 1, pp. 297–300 (2009)
2. Fong, L.L., Seng, W.C.: A Comparison Study on Hand Recognition Approaches. In: International Conference of Soft Computing and Pattern Recognition, SOCPAR 2009, December 4-7, pp. 364–368 (2009)
3. Shirakawa, S., Nagao, T.: Evolutionary image segmentation based on multiobjective clustering. In: IEEE Congress on Evolutionary Computation, CEC 2009, May 18-21, pp. 2466–2473 (2009)
4. Kang, W.-X., Yang, Q.-Q., Liang, R.-P.: The Comparative Research on Image Segmentation Algorithms. In: First International Workshop on Education Technology and Computer Science, ETCS 2009, March 7-8, vol. 2, pp. 703–707 (2009)
5. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Macmillan Publishing Ltd., Basingstoke (2006)
6. Son, T.T., Mita, S., Takeuchi, A.: Road detection using segmentation by weighted aggregation based on visual information and a posteriori probability of road regions. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, October 12-15, pp. 3018–3025 (2008)
7. Sharon, E., Brandt, A., Basri, R.: Fast multiscale image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, Proceedings, vol. 1, pp. 70–77 (2000)
8. Sharon, E., Brandt, A., Basri, R.: Segmentation and boundary detection using multiscale intensity measurements. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I-469 – I-476 (2001)
9. Rory Tait Neilson, B.N., McDonald, S.: Image segmentation by weighted aggregation with gradient orientation histograms. In: Southern African Telecommunication Networks and Applications Conference, SATNAC (2007)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Computer Vision* 59, 167–181 (2004)
11. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8 (June 2007)

12. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)
13. Comaniciu, D., Meer, P., Member, S.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 603–619 (2002)
14. Dyer, R., Zhang, H., Möller, T.: Delaunay mesh construction. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP 2007, Aire-la-Ville, Switzerland*, pp. 273–282. Eurographics Association (2007)
15. Vassili, V.V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: *Proc. Graphicon 2003*, pp. 85–92 (2003)
16. Hunter, R.S.: Photoelectric Color-Difference Meter. *Proceedings of the Winter Meeting of the Optical Society of America, JOSA* 38(7), 661 (1948)
17. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: *Computational Geometry: Algorithms and Applications*, 3rd edn., Springer, Heidelberg (April 2008)
18. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston (1992)
19. Meirav, G., Eitan, S., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003, Washington, DC, USA*, p. 716. IEEE Computer Society, Los Alamitos (2003)
20. Xiao, Q., Zhang, N., Gao, S., Li, F., Gao, Y.: Segmentation based on shape prior and graph model optimization. In: *2nd International Conference on Advanced Computer Control (ICACC), March 27-29, vol. 3*, pp. 405–408 (2010)