Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Color Image Segmentation based on Automatic Derivation of Local Thresholds

Ety Navon[1], Ofer Miller[1], and Amir Averbuch[1]

[1]School of Computer Science, Tel-Aviv University
Tel-Aviv, 69978, Israel. millero@post.tau.ac.il

Abstract. In this paper a new method for color image segmentation is presented. The proposed algorithm divides the image into homogeneous regions by derivation of local thresholds via local information. The algorithm contains two main steps. First, the watershed algorithm is applied on the image gradient magnitude. Its results are used as an initial segmentation for the next step, which is region merging process. During that process regions are merged and local thresholds are derived one-by-one at different times by analyzing local characteristics of the regions. Every threshold refers to specific region and defines it as ífinal regioní (non-mergeable). Thus, regions are handled separately; some regions grow while others were already defined as ífinal regionsí. The significant use of local information improves the quality of the segmentation result. Experimental results have demonstrated the efficiency of the proposed method. The algorithm is found to be reliable and robust for different images.

1 Introduction

Image segmentation is a process of dividing an image into (non-overlapping) homogeneous regions with respect to a chosen property. It is an important task for variety of application such as object extraction, image compression, objects tracking and objects recognition. The existing segmentation methods 1, 2 can be roughly divided into four different groups: Histogram-based methods, boundaryñbased methods, region-based methods and hybrid-based methods.

Most of the *Histogram-Based* methods deal with gray level images, which are represented by one-dimensional histogram. The range of intensities is assumed to be constant and the histogram is considered as being a probability density function of a Gaussian. Then, the segmentation problem is reformulated as parameter estimation followed by pixel classification. For 3-D color space (RGB, HSV, etcí) some techniques were developed (for example,3).

The idea of *Boundary-based* methods is to search for pixels that lie on a region boundary. These pixels are called edges 4. An edge is characterized by a significant local change in image intensities. Edges are detected by looking at neighboring pixels. The basic assumption is that the change in pixels values between neighboring pixels inside a region is not as significant as the change in pixels values on the regions boundary. Then, the edges that represent a significant change are considered as a boundary between regions.

Region-based methods gather similar pixels according to some homogeneity criteria. They are based on the assumption that pixels, which belong to the same homogeneous region, are more alike than pixels from different homogeneous regions. The split-and-merge or the region-growing techniques 5 are examples for such method.

Hybrid-based methods aim at improving the segmentation result by combining the above methods. Many of the hybrid techniques (for example, 6 and 7) combine the region-based method with the boundary-based method. Some use the combination of the histogram-based with the region-based methods. The hybrid technique for segmentation is very common since it relies on wide information as global (histogram) and local (regions and boundaries).

This paper proposes a new hybrid method for color image segmentation that integrates edge- and region- based techniques. The algorithm segments the image by an automatic derivation of thresholds through an iterative region merging procedure, where local information is considered. Although thresholds are usually associated with histograms and thus with global analysis, the thresholds in the proposed algorithm are determined via local information. Since homogeneity relates to locality, analyzing local characteristic is pragmatically helpful. Significant details are considered more efficiently and thus the segmentation result is improved. The algorithm is composed of two main steps. Initially, the image is divided into a large number of small regions using the watershed algorithm. The Region Adjacency Graph (RAG) is the data structure we use to represent the partition of the image. The second step is an iterative process, in which regions are merged and thresholds are derived. The merging order is based on Kruskalís algorithm 10 for finding a minimum spanning tree (MST). During the merging process local characteristics of the regions are considered and analyzed in order to identify when inhomogeneous region are generated. Then, local thresholds are derived. The merges that produce inhomogeneous regions are canceled and the regions, associated with those thresholds, are identified as non-mergeable regions and considered as ìfinal regionsî. The number of thresholds and their values are known only when the process is terminated.

The rest of the paper is organized as follows. Section 2 describes the watershed algorithm. Section 3 describes the merging methodology. In section 4 the core process that derives the thresholds is proposed. Experimental results are given in section 5.

2 Initial Segmentation Using Watershed

At the first stage of the segmentation algorithm the watershed algorithm is applied. Its input is a gray-scale gradient image. Thus, we apply the application of Canny edge detection 8 on the input color image I to obtain its gradients image I_G. Since the watershed produces an over-segmentation due to its sensitivity to weak edges, its results are used as an initialization for the next merging process.

2.1 The Watershed Algorithm

The watershed algorithm is a morphological tool to segment the image. The image is considered as a topographic relief. Each pixelís value (gray level) stands for the evaluation at that point. The algorithm defines catchment basins and dams. Each catchment basin, which is associated with a minimum *M*, is a set of connected pixels such that a drop of water falling from any pixel that belongs to this catchment basin, falling down until it reaches the minimum *M*. On its way down the drop passes only through pixels that belong to this catchment basin. Dams are watershed lines. They are pixels that separate different catchment basins.

We apply the Vincent and Soille 6 version of the watershed algorithm, which is based on immersion simulation. The surface is immersed from its lowest altitude until the whole image is flooded. First, the pixels are sorted. During the flooding process there are scanned by the sorted order and catchment basins are built. The output of the watershed is segmentation of I_G into a set of *n* non-overlapping regions R_i ,*i=1,...,n*. Since these regions are going to be merged during the next iterative process they are denoted by $R_i^{m_i}$, $i = 1,...,n,$, $m_i = 1,...,M_i$ where M_i is the number of merges of $R_i^{m_i}$.

2.2 The RAG Data Structure

The region adjacency graph (RAG) is the data structure that is used to represent the partition of the image. For the set of regions $R_i^{m_i}$, $i = 1,...,n,$, $m_i = 1,...,M_i$, the RAG is defined as an undirected graph *G=(V,E)* where *V*={1,2,Ö ,*n*} such that each region is represented by a node, and $e(i, j) \in E$ if $i, j \in V$ and $R_i^{m_i}$ and $R_j^{m_j}$ are adjacent. An example of an image that contains six regions with its corresponding RAG is illustrated in Fig. 1. Since the merging process is based on G , each edge is assigned a weight. The weight of an edge $e(i, j)$ is the value of the dissimilarity function $f\left(R_i^{m_i}, R_j^{m_j}\right)$, defined in section 3.1.

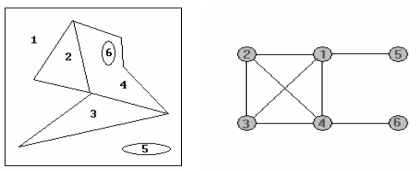


Fig. 1. Six partitions of the image and its corresponding RAG

3 Merging Methodology

Given the over-segmentation obtained by the first step, the main merging process of the algorithm based on the RAG is applied to produce the final segmentation.

3.1 Dissimilarity Measure Between Regions

To determine the merging order dissimilarity function between any two neighboring regions, $R_i^{m_i}$ and $R_j^{m_j}$, denoted by $f\left(R_i^{m_i}, R_j^{m_j}\right)$, is defined. The function is based on two components: color and edges. The Hue component of the HSV color space 9 is used for the color component since it less influenced by changes in illumination such as shade and shadow. The mean value of the hue component of a given region $R_i^{m_i}$ is denoted by $\mu_h\left(R_i^{m_i}\right)$. The gradient magnitudes are used as another source of local information for the second component. We denote by $\mu_G\left(R_i^{m_i}, R_j^{m_j}\right)$ the mean gradient between $R_i^{m_i}$ and $R_j^{m_j}$, which is calculated from the gradients among the shared pixels between the two regions in I_G. The dissimilarity function is defined as:

$$f\left(R_i^{m_i}, R_j^{m_j}\right) = w_1 \cdot d\left(\mu_h\left(R_i^{m_i}\right), \mu_h\left(R_j^{m_j}\right)\right) + w_2 \cdot \mu_G\left(R_i^{m_i}, R_j^{m_j}\right) \qquad \textbf{(1)}$$

where $d\left(\mu_h\left(R_i^{m_i}\right), \mu_h\left(R_j^{m_j}\right)\right)$ is the difference between $\mu_h\left(R_i^{m_i}\right)$ and $\mu_h\left(R_j^{m_j}\right)$ defined as:

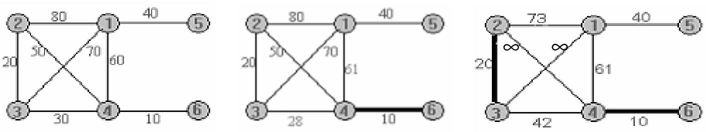
$$d(\mu_h(R_i^{m_i}), \mu_h(R_j^{m_j})) = \min\{|\mu_h(R_i^{m_i}) - \mu_h(R_j^{m_j})|, 360 - |\mu_h(R_i^{m_i}) - \mu_h(R_j^{m_j})|\} \qquad \textbf{(2)}$$

and w_1 and w_2 are predefined coefficients. The function is mostly based on the color space, thus, $w_1 >> w_2$.

3.2 The Merging Order

The merging process is based on Kruskalís algorithm 10 for finding a minimum spanning tree (MST). Kruskalís algorithm generates the MST, denoted as *T*, from scratch by adding one edge at a time. Initially, the edges of the graph *G* are sorted in a non-decreasing order of their weights. Then, the edges in the sorted list are examined one by one and checked whether adding the edge that is currently being examined creates a cycle with the edges that were already added to *T*. If it does not, it is added to *T*. Otherwise, it is discarded. The process is terminated when *T* contains *n*-1 edges. At the end of the process *T* is the MST of *G*. We apply Kruskalís algorithm on *G* while considering the process itself. This process is the process that merges regions: Adding *e*(*i,j*) to *T* represents the merge of $R_i^{m_i}$ and $R_j^{m_j}$. Adding the edge with the minimum weight one-by-one to *T* is equivalent to the merge of the two most similar regions. When an edge is rejected because it creates a cycle in *T*, no merge is performed since its two regions have already been merged into one region. At the end, when *T* spans all the nodes, all the regions are merged into one region, and the merging process is terminated.

When a new region is generated by any merge, new information about this region and its surroundings exists while the previous information becomes irrelevant. In order to follow the local changes of the regions the updated information is used. Let $R_{ij}^{m_{ij}} = R_i^{m_i} \cup R_j^{m_j}$ be the new region that is generated by the merge of $R_i^{m_i}$ with $R_j^{m_j}$. For every region $R_v^{m_v} \in N\left(R_{ij}^{m_{ji}}\right)$, where $N\left(R_{ij}^{m_{ji}}\right)$ denotes the neighbors of $R_{ij}^{m_{ji}}$, $f\left(R_{ij}^{m_{ij}}, R_v^{m_v}\right)$ is recomputed and the corresponding edges in G and the sorted list are updated. Thus, the complete merging process is based on Kruskalís algorithm with the following modification: When an edge $e(i,j)$ is added to T and T has less than $n-1$ edges, $f\left(R_{ij}^{m_{ij}}, R_v^{m_v}\right)$ is calculated for the new region $R_{ij}^{m_{ij}}$ and its neighbors and the weights of its corresponding edges $\left\{e(i,k) \cup e(j,l) \middle| R_k^{m_k} \in N\left(R_i^{m_i}\right), R_l^{m_l} \in N\left(R_j^{m_j}\right), k \neq j, l \neq i\right\}$ are updated. *Parallel edges* are exceptional. Two edges, $e(i,j)$ and $e(u,v)$, are considered *parallel* if $R_i^{m_i}$ was already merged with $R_u^{m_u}$ (or with $R_v^{m_v}$) and $R_j^{m_j}$ was already merged with $R_v^{m_v}$ (or with $R_u^{m_u}$, respectively). Since only one of them may be added to *T*, one is assigned the cost of the dissimilarity function and the other is assigned "∞". "∞" indicates that no further consideration whether to add it to *T* is required. Fig. 2 illustrates an updating process. When the first edge $e(4,6)$ is added to T $e(3,4)$, $e(2,4)$ and $e(1,4)$ are updated. $f\left(R_i^{m_i}, R_j^{m_j}\right)$ for these edges is recalculated and the new values, ì61î, ì50î and ì28î are the new edge costs, respectively. $e(2,3)$ is the next edge that is added to T. Four edges have to be updated: $e(2,1)$, $e(2,4)$, $e(3,1)$ and $e(3,4)$. Since $e(2,1)$ and $e(3,1)$ become parallel, we assign to one of them the new value ì73î and the other edge is assumed to have "∞". The same is done for $e(2,4)$ and $e(3,4)$. The weight of $e(3,4)$ is ì42î and the weight of $e(2,4)$ is "∞".



(a) **(b)** **(c)**

Fig. 2. Updating edges during the construction of MST. (a) The source graph. (b) Adding $e(4,6)$ to T. (c) Adding $e(2,3)$ to T

4 The Need of Local Information

Since we consider the segmentation process to be a local operation we can assume that not all the local merges will be terminated simultaneously. The use of one global threshold is insufficient since different regions should be separated from their surroundings at different times during the process with different thresholds. However, when the image contains one object, which is homogenous in its colored texture, and so is its surrounding background, one global threshold is adequate. Since nearly all the images contain more than two homogenous regions more than one threshold is required. Moreover, it is obvious that it is difficult to predict whether one global threshold can handle a given input image. Fig. 3 illustrates the reason why local threshold are required. Fig. 3a is the source image. Fig. 3b is the result of the watershed algorithm. Fig. 3c is the result after using one global threshold t=20. Fig. 3d is the result after using global threshold, t=30. In Fig. 3c all the regions are homogeneous and can grow. In Fig. 3d regions such as the face and the sofa, which are considered visually as homogenous, are still over segmented, while the region, which is indicated by the yellow arrow, is inhomogeneous. Thus the construction of that region should be terminated at t<30 while the construction of the other regions can proceed.

The calculation of local thresholds will be based on local information, which is related to the regions and their surroundings, since regions are affected by their surroundings. The dependency between regions and their surroundings can be simply demonstrated. For example, the same yellow elliptic object exists in two images in Fig. 4. In the left image it is clearly seen and it is well separated from its background while the right image is almost invisible.

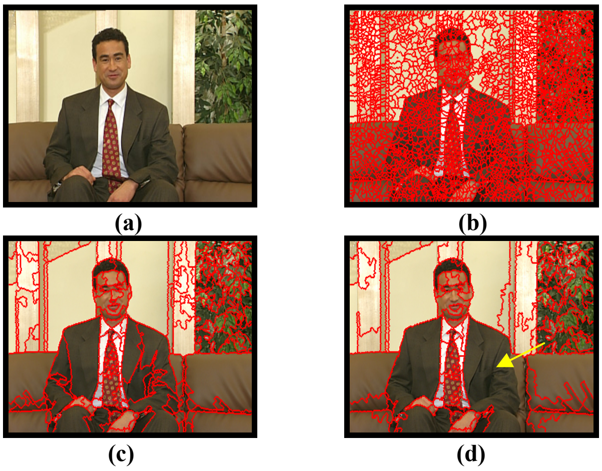


Fig. 3. (a) Original input image. (b) After the application of the watershed algorithm. (c) After the completion of the merging process using one global threshold t=20. (d) After the merging process using one global threshold t=30

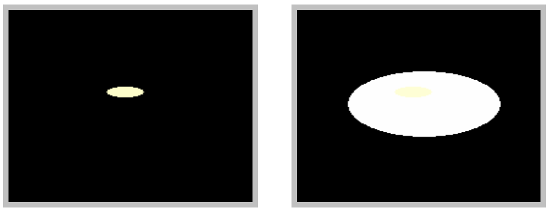


Fig. 4. The same bright elliptic object appears differently due to its different backgrounds

4.1 The calculation of Adaptive Local Thresholds

The automatic process that calculates local thresholds is based on local homogeneity changes of the regions during the merging process. These thresholds, that determine which regions in each time should not be merged, based on the following proposition, generate the final segmentation.

Proposition 1: A significant change in the homogeneity of a given region occurs during a merge that generates inhomogeneous region. At this merge, local threshold is determined.

The identification of homogeneity is mainly based on color space. The hue component of the HSV color space is used here. Let $h(x,y)$ be the value of hue in location (x,y). The second order moment (the variance) of any region $R_i^{m_i}$ $i=1,...,n$ after its m_i-th merge, is defined as:

$$\sigma^2\left(R_i^{m_i}\right)=\frac{1}{\left|R_i^{m_i}\right|}\sum_{(x,y)\in R_i^{m_i}}\left(h(x,y)-\mu_h\left(R_i^{m_i}\right)\right)^2 \quad \textbf{(3)}$$

where $|R_i^{m_i}|$ is the size of $R_i^{m_i}$. The change in the homogeneity of $R_i^{m_i}$ after the m_i-th merge is defined to be

$$\Delta\sigma(R_i^{m_i})=|\sigma(R_i^{m_i})-\sigma(R_i^{m_i-1})|. \quad \textbf{(4)}$$

Let J_i be the set of K_i local maximums of $\Delta\sigma\left(R_i^{m_i}\right)$, $m_i=1,...,M_i$.

$$J_i=\left\{\left(m_i,\Delta\sigma\left(R_i^{m_i}\right)\right)\middle|\Delta\sigma\left(R_i^{m_i}\right)>\Delta\sigma\left(R_i^{m_i-1}\right)\&\Delta\sigma\left(R_i^{m_i}\right)>\Delta\sigma\left(R_i^{m_i+1}\right)\right\} \quad \textbf{(5)}$$

Although no statistical information on the image is given, local information on any region $R_i^{m_i}$ $i=1,...,n$ is obtained by $\Delta\sigma\left(R_i^{m_i}\right)$. Since the variance is a measure for homogeneity, the merges in J_i represent significant transitions of $R_i^{m_i}$ during the merging process. Given that the merging process begins with over-segmentation of homogeneous regions and the regions are merged until one region is left, every region becomes inhomogeneous at different merge operation. Hence, we argue that $R_i^{m_i}$ becomes inhomogeneous at the first local maximum in J_i that satisfies:

$$\Delta\sigma\left(R_i^{m_i}\right)>\beta \quad \textbf{(6)}$$

where β is the mean value of $\Delta\sigma\left(R_i^{m_i}\right)$ at every $m_i \in J_i$ defined as

$$\beta = \frac{1}{K_i} \cdot \sum_{m_i \in J_i} \Delta\sigma\left(R_i^{m_i}\right) \tag{7}$$

Due to the unique behavior of $\Delta\sigma\left(R_i^{m_i}\right)$ (see Fig. 5), the definition of β enables to reject the local maximums that refer to the merges in which $R_i^{m_i}$ is still homogenous. The three plots in Fig. 5 illustrate the behavior of $\Delta\sigma\left(R_i^{m_i}\right)$ of three different regions of the manís shirt (Fig. 3b), which reflects the changes in the homogeneity. The plots describe the values of $\Delta\sigma\left(R_i^{m_i}\right)$ as a function of the number of the merges. The arrow in each plot points to the first local maximum, among all the local maximums, that satisfies Eq. (6). In this merge, the shirt region is merged with another region (the bright background) and becomes inhomogeneous. Since the three regions were merged into one region, the identification of the merge that generates inhomogeneous region is equal and is independent on which region (among all the regions that compose the homogenous region, the manís shirt) we examine.

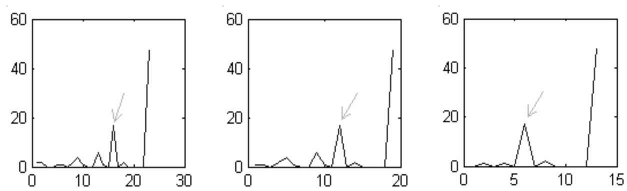


Fig. 5. The representation of $\Delta\sigma\left(R_i^{m_i}\right)$ of three different regions of the manís shirt (from Fig. 3b). The x-axis is the number of merges m_i. The y-axis is $\Delta\sigma\left(R_i^{m_i}\right)$. The arrow in each plot points to the first local maximum that satisfies Eq.(6) among all the local maximums

As was mentioned, an iterative process is applied in order to derive the thresholds. More precisely, any iteration obtains a single threshold. Let $s = 1,...,K$ be the index of the iterations number. K is currently unknown since the number of thresholds (iterations) is unknown. Let t_s be the threshold of the s^{th} iteration and let mt_s be the merge that is associated with t_s. During the s^{th} iteration regions are merged until one region is left except from ìfinal regionsî that are discussed bellow. For every region $R_i^{m_i}, i = 1,...,n$ we get from that process a map

$$L_i : \{1,...,M_i\} \rightarrow \{1,...,M\} \tag{8}$$

for every $m_i \in \{1,...,M_i\}$ to $m \in \{1,...,M\}$, where M is the total number of merges in the current iteration. $L_i(m_i) = m$ means that the m_i^{th} merge of $R_i^{m_i}$ is its m^{th} merge among all the M merges. For example, $m_i = 5$ and $L_i(m_i) = 27$ means that the fifth merge of $R_i^{m_i}$ is its 27^{th} merge among all the M merges of all regions. For every region $R_i^{m_i}$ let $m_i' \in J_i$ be the first local maximum that satisfies Eq. (6). The mt_s merge of the current threshold is defined as:

$$mt_s = \min_{i=1,\ldots,n} \{L_i(m_i')\}. \qquad \textbf{(9)}$$

mt_s refers to the first merge among all the merges in the current iteration that generates inhomogeneous region. Let $R_i^{m_i}$ and $R_j^{m_i}$ be the two regions that were merged at the mt_s merge. Since t_s prevents the merge of $R_i^{m_i}$ and $R_j^{m_j}$, all the merges from the final merge to the mt_s merge are canceled. This process is called a *regression process*. When the mt_s merge is reached and canceled during the regression process, $R_i^{m_i}$ and $R_j^{m_j}$ are considered as ì final regionsî, and denoted as R_i^* and R_j^*. They will remain unmerged. During the next iteration, the merging process proceeds and all the regions, except the ì final regionsî, are merged into one region, and the next threshold t_{s+1} will be derived similarly. The iterative process, which consists of merging process, derivation of local threshold and regression process, is terminated when no regions to be merged are left and only ì final regionsî exist. The î final regionsî generate the segmentation result.

5 Experimental Results

The following examples (Fig. 6 and Fig. 7) demonstrate the final segmentation results of three different images characterized by different color homogeneity. Whether only one threshold is required (Fig. 6) or more (Fig. 7) the algorithm can handle variety of images efficiently since it emphasizes the idea of local analysis: Local thresholds are derived adaptively by local analysis, which is significantly used through all the segmentation process. Moreover, this is an automatic process, which does not use any pre-defined parameters or thresholds, which are image-depended. The results demonstrate the efficiency of the proposed method.

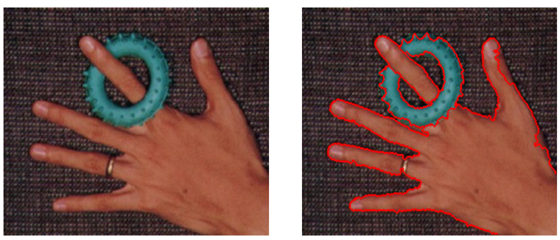


Fig. 6. Final segmentation results by derivation of one threshold

Fig. 7. Final segmentation results of three different images by derivation of different number of thresholds

References

1. Pal N., Pal S.: A review on image segmentation techniques. Pattern Recognition, Vol. 26 (1993) 1277ñ1294
2. Haralick R. M., Shapiro L. G.: Survey: Image segmentation techniques. Comput. Vis. Graph. Image Process*,* Vol. 29 (1985) 100-132
3. Sarabi A., Aggarwal J. K.: Segmentation of chromatic images. Pattern Recognition, Vol. 13, no. 6, (1981) 417ñ427
4. Marr D., Hildreth E.: Theory of edge detection, in Proc. R. Soc. Lond. B, no. 207 (1980) 187ñ217
5. Channg Y.-L., Li X.: Adaptive image region-growing. IEEE Trans. Image Processing, Vol. 3 (1994) 868-872
6. Vincent L. and Soille P.: Watershed in digital space: an efficient algorithm based on immersion simulation. IEEE Trans. Pattern Anal. Machine. Intell., Vol. 13 (1991) 583-598
7. Fan J., Yau D. K. Y., Elmagarmid A. K., Aref W.G.: Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans. Image Processing, Vol. 10 (2001) 1454-1466
8. Canny J.: A computational approach to edge detection. IEEE Trans. on Pattern Anal. Machine. Intell., Vol. 8 (1986) 679-698
9. Jain, A. K.: Fundamental of Digital Image Processing, prentice-Hall (1989)
10. Corman T., Leiserson C., Rivest R.: Introduction to Algorithms, the MIT press (1990)