Using Color and Edge Information in an Unsupervised Quadtree-based Segmentation Algorithm*

Adolfo Martínez-Usó, Filiberto Pla, Pedro García-Sevilla

Departamento de Lenguajes y Sistemas Informáticos, Universidad Jaume I
Campus Riu Sec, 12071 Castellón, Spain
{auso, pla, pgarcia}@lsi.uji.es

Abstract. There exist many different approaches for image segmentation. Most of these segmentation techniques were motivated by specific application purposes. In this article, we present an unsupervised segmentation algorithm through a multiresolution approach which uses both color and edge information in order to segment fruit images. In addition, we use a Quadtree structure that speeds up the whole procedure. The algorithm has been tested for segmenting typical images in order to compare our results with the ones provided by other color segmentation methods. It has also been applied to fruit images in order to segment the different areas of the fruit surface. Due to the unsupervised nature of the method, it can adapt itself to the huge variability of colors and shapes of the regions in fruit inspection tasks.

1 Introduction

Many segmentation techniques are available in the literature, and some of them have been widely used in different application problems [3,5], which mainly differ in the criterion used to measure the similarity of two regions and in the strategy applied to guide the segmentation process.

In split-and-merge techniques there are several criteria for splitting or merging. In our algorithm, we have paid great importance to an efficient strategy to solve the problem. The work we present here develops a robust criterion by means of color and edge information and an accurate strategy as the basis for the whole segmentation process that guides all the algorithm steps. In this sense, the algorithm expands its processes and constructs its structures and, when needed, makes use of the criteria to decide how to deal with the image regions.

The main motivation of this work was to obtain a method able to segment images of fruits for their classification according to the defects found on their surface. The work reported here presents some advances and improvements to the preliminary results obtained from a previous work presented in [6].

* This work has been partly supported by grants DPI2001-2956-C02-02 from Spanish CICYT and IST-2001-37306 from the European Union.

This approach accomplishes the following requirements for the image segmentation:

- ✓ An unsupervised method would be needed, due to manifold variables and the huge variability of situations which can arise in the images of fruits. Thus, any prior knowledge should be avoided for the segmentation procedure.
- ✓ The segmentation method has to be mainly based in color and edge criteria.
  - Color information: in order to segment color regions and due to hue/saturation variations, not only gray level information should be used.
  - Edge information: in order to define the segmented region boundaries as accurately as possible, the process must include edge information [1].

For meeting the above mentioned requirements, a Quadtree (QT) structure has been chosen to support the developed method. A QT is commonly used to split a given frame into blocks of different sizes since it enables an efficient representation of the resulting decomposition. Moreover, well-known algorithms are available for searching and neighboring operations over this sort of trees [8]. Image segmentation will be conditioned and orientated by the image representation adopted, that is, by the QT.

It is quite obvious that, from the described requirements, color and edge information would define the particular criterion to be used in the segmentation procedure. However, there is a difficult decision about which is the best color space to represent a given problem. In this way, the $L^*a^*b^*$ color space is used, in which the perceptual difference between two points is proportional to their distance.

The rest of the paper is structured as follows: in Section 2, we introduce the QT structure and explain the criterion which has been used to allow the construction of the QT. In Section 3, we formulate the problem of using color information and propose a method to solve it, while in Section 4 we present the edge criterion which will be used. In Section 5, we explain the segmentation strategy and summarize the algorithm used to solve the problem. Section 6 provides some experimental results. Finally, some conclusions are drawn in Section 7.

2 Generating a Quadtree Image

The QT structure allows to divide an image within a complete tree representation, including neighboring information. This spatial information can be further used by a merging strategy which joins the QT leaves using color and edge information.

To calculate the QT representation of an image, the procedure starts by considering the whole image in the root node. If the segmentation criterion is not fulfilled in the region representing this node, then the region is divided into four children regions. This process is recursively applied to each subregion until the criterion for a node is accomplished for each leaf.

Obviously, the QT process will produce subregions of different sizes represented in a tree structure, which may range from a single pixel to the whole image.

This structure has been extensively used in works such as the one by Wilson and Spann [9]. One of the most recent applications of this data structure can be found in

[1] where, through a methodology of multiresolution based on a QT, information of borders is combined with a region growing strategy to obtain the image segmentation on gray images.

In this work, we have used a similar strategy but using a different criterion to compare two regions and extending its application to a color environment. Moreover, the algorithms described in Samet [8] have been used in the QT structure for the implementation of the neighborhood operations.

3 Use of Color Information

Although many authors use partial information in a particular color space, it is evident that ignoring some color components causes a loss of relevant information that can be very important. Therefore, the presented approach uses the three planes simultaneously in the L*a*b* color space. In this color space the Euclidean distances between two colors are approximately proportional to the human perceptual difference between them.

In this work, two different color steps are used to decide whether two regions must be merged or not. The **first color step** is used only after the construction of the QT and it merges regions whose color differences are smaller than a given threshold. This first step is used to accelerate the segmentation process when the regions have an homogeneous distribution and it is applied firstly to the leaves of the QT and after that, to the regions obtained.

Once the first step is completed a **second color step** is used, and it will be applied iteratively until the final segmentation is obtained. Each region is considered as a set of pixels which follow a Normal distribution. To demonstrate this assumption we selected a sample of the 2,723,035 regions obtained from 12 real images and the Chi-Square test [7] was run on these regions. The results showed that 90.11% of the sample regions followed a Normal distribution with a probability of 95%. So, the average value μ and standard deviation σ in the L* plane are calculated for each region. When two regions are compared, the probability density function of each Normal distribution is computed and these distributions are compared according to the absolute value of their differences:

$$| N(\mu_A, \sigma^2_A) - N(\mu_B, \sigma^2_B) |$$

This measure provides a value 0 when the distributions of the values in the two regions are exactly the same, and 2 when they are totally different.

At the same time, planes a*b* are taken into account and the Euclidean distance between the average values of each region is used to measure the color difference between the two regions.

Now, the color criterion for the representation considers whether the regions have similar enough colors, according to some fixed thresholds. Note that, in our approach, the L* plane and the a*b* planes have the same importance and both criteria have to be fulfilled.

4 Use of Edge Information

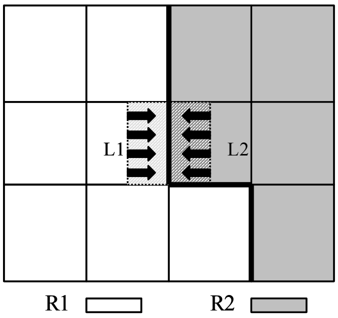

Fig. 1. Edge criterion

The use of edge information in the segmentation process allows to avoid the merging of regions when the color criterion is satisfied but there exists an edge between the regions. In this sense, gradient information is used to develop a boundary map. The gradient information is checked even when the color criterion is fulfilled and it is computed from the first plane of the Principal Component Analysis (PCA) obtained from the original RGB image. We also tried to construct it from the L* plane in the L*a*b* and from the first plane of the PCA obtained from the L*a*b* image, but this two options provided worse results.

Let R_1 and R_2 be two regions with a common boundary. Let L_1 and L_2 be two leaves of the QT that belong to R_1 and R_2 respectively, which share a linear segment of the common boundary between the two regions (see Fig. 1). From the leaves L_1 and L_2 we can compute the values $\sum_{half}\nabla_1$ and $\sum_{half}\nabla_2$ which correspond to sum of half of the values of the gradient magnitude in the perpendicular direction of the edge in the corresponding leaf (striped patterns in Fig. 1). For each leaf, the gradient values close to the boundary segment are taken into account. In this way, equation expression (1) is computed for each boundary segment.

$$FB = \frac{\left(\sum_{half}\nabla_1 + \sum_{half}\nabla_2\right)}{segment\_size} \tag{1}$$

Finally, we compute $(\sum FB)\,/\,N$ after all boundary segments have been processed, where N is the number of segments that form the whole boundary between the regions. If the final value obtained does not exceed a prefixed threshold, the two regions could be merged into one.

5 The Segmentation Process

As it was mentioned in the introduction, the segmentation method proposed is supported by a QT data representation of an image, and it uses color and edge information as particular criteria whenever the algorithm must take a decision.

In the first step of the procedure the QT is expanded until every square region is homogeneous according to an homogeneity criterion. The simplest homogeneity function is that for which all pixels in the QT leaves have the same color values. This will lead to a significant expansion of the QT and will produce an oversegmented representation that guarantees homogeneous regions at the beginning of the process.

Other homogeneity functions without such strict criteria would lead to a more compact QT, but at the cost of a loss of accuracy in the detected boundaries.

The oversegmented representation is input to the algorithm that starts by recursively merging neighboring leaves based on the first color step (Section 3). The criterion of this step establishes a threshold for merging leaves representing smooth regions. This stage is including just to make the QT more compact in order to make the whole process more efficient.

At a first moment, the algorithm works with leaves and, while the process go on, clusters of leaves are created. These clusters are what we called "regions" and will be represented as sets of leaves in the QT that meet some spatial constraints.

A further step consists of a merging strategy which looks for neighbor regions in the QT structure that can be merged due to their similarity.

During the fusion process the following premises are respected:

- ✓ Regions are arranged according to their size, giving more importance to the bigger regions and facilitating the merging of small regions with the big ones.
- ✓ Only regions with a common boundary can be merged. This will represent the spatial constraint in the merging process.

The merging strategy, only in its last stage, takes as a color criterion the second step described in Section 3. Furthermore, it takes as edge criterion the one described in Section 4.

Figure 2 shows the outline of the segmentation strategy used, which is basically an iterative process. In the iterative loop, each cluster is compared to all its neighboring clusters and, when the segmentation criterion is satisfied, the clusters are merged. Candidate merging clusters are searched through a list of clusters ordered by size, beginning with the biggest cluster, favoring initially the merging and formation of bigger regions that will act as seeds to further collect smaller regions. This clustering process stops when no other merging can be performed.

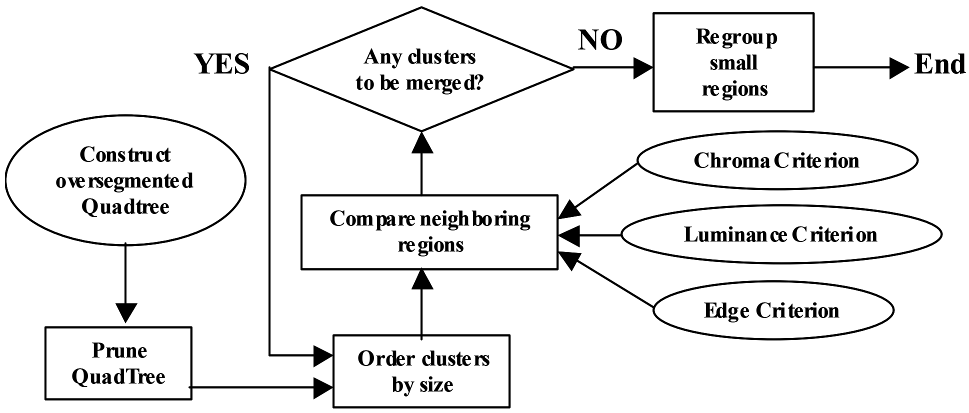

Fig. 2. Segmentation process flow chart

Finally, the algorithm looks for small regions by means of a threshold that determines which regions are small. Most of these small regions are due to the use of a

Gaussian filter and the gradient information. These two stages in the segmentation process produce blurred edges. So, a final step is included to regroup these small regions with their similar neighbors, giving more importance to the biggest ones.

The whole segmentation process can be summarized in the following algorithm:

1. Perform the Principal Component Analysis of the input image.
2. Construct the gradient image from the first PCA plane.
3. Generate L*a*b* image and convolve it with a Gaussian Filter.
4. Construct an oversegmented representation of the L*a*b* image (QT) using a color similarity criterion. The most restrictive one may be that each region has all its pixels with the same color. Create an ordered list according to region sizes.
5. Merge neighboring **leaves** with the same parent node using a color distance threshold. This merging process is not iterative (compact QT representation to speed-up the rest of the process).
6. Look for the first pair of neighboring **clusters** in the list that can be merged according to the first color merging step. The search is done beginning by the biggest cluster.
7. The color criterion is tested for each region and its neighbor regions by means of the second step described in Section 3. At the same time, the algorithm looks for border leaves between regions and checks whether there are high gradient values (edge criterion in Section 4). If both criteria are fulfilled, then the regions are merged.
8. If any clusters are merged, reorder the list of clusters.
9. Repeat steps 7 and 8 until regions became stable (iterative sequence that may be seen as a coarse-to-fine segmentation process).
10. Regroup small regions.

Note that each time that the algorithm modifies the QT representation by merging leaves or regions, there is a re-order process according to regions size. At the end of the procedure, every cluster in the list is represented by a set of leaves of the QT, which are labeled as a single resulting region.

6 Experiments and Results

We present some experimental results obtained with typical color images (Fig. 3) as well as with three fruit images (Fig. 4). These results have been obtained using always the same thresholds in the criteria. In addition, Table 1 presents a quantitative evaluation of the image segmentation results by means of the Liu and Yang [4] and Borsotti et al. [2] functions calculated on the RGB color space. These functions yield subjective values about the quality of the segmentation. The smaller these values, the better the segmentation.

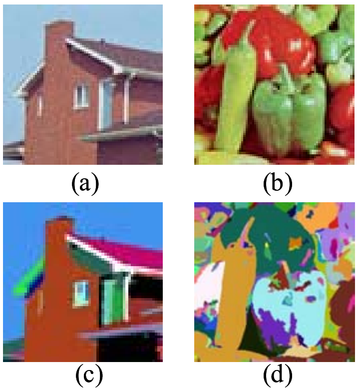

Fig. 3. (a) House (b) Peppers. (c), (d) Segmentation results displayed using randomly chosen colors

Table 1. Quantitative evaluation of image segmentation results

	Liu Function (F)	Borsotti Function (Q)
house	146	1138
peppers	685	5392
orange4a	4	35
oranges4b	114	956
orange4c	58	700

As it was mentioned in the introduction, the segmentation method presented has been aimed at characterizing images of fruits in visual inspection applications. The segmentations of the fruit images are used as input in a further process to characterize and classify each region in the fruit surface to identify and detect different types of defects and parts of the fruit.

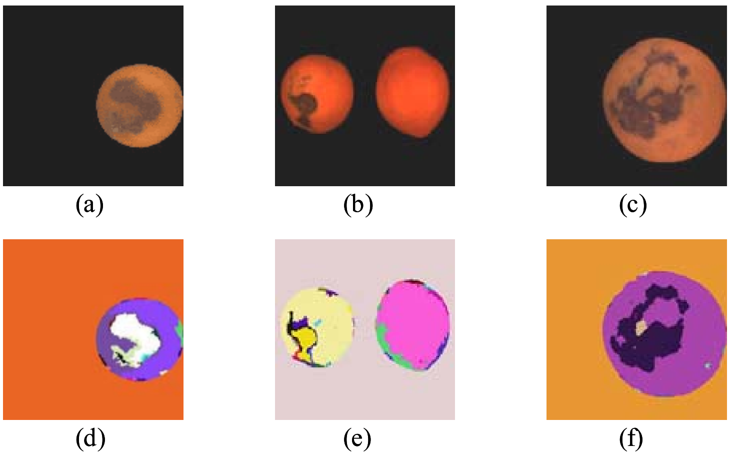

Fig. 4. (a), (b), (c) Oranges with stains on their surface. (d), (e), (f) Segmentation results displayed using randomly chosen colors

Figure 4 shows some examples of the results of the segmentation on different fruits and situations to be characterized. Note how the segmentation process has adapted to the regions of each image due to its unsupervised nature. For instance, figure 4a, 4b and 4c shows examples of fruits with various stains produced by the effect of rot and how the segmentations obtained, figure 4d, 4e and 4f, has found the different variations of the stains of the rotten zone. This will allow the extraction of region descriptors for their identification and classification.

7 Conclusions

In this paper, some accurate results of an unsupervised color segmentation algorithm have been presented.

The method has been applied to typical images and to fruit images in order to characterize the variety of regions appearing on the fruit surfaces. The results obtained show how the algorithm can adapt to the different situations and variability of color regions, being able to segment areas and locating the borders quite accurately.

The use of color and edge information combined with a QT representation of the image is the base of an unsupervised segmentation strategy that can locate color regions finding their contours in a satisfactory way.

References

1. Bhalerao, A., Wilson, R.: Unsupervised Image Segmentation Combining Region and Boundary Estimation. In Image and Vision Computing. Vol. 19, Num. 6, (2001) 353–386
2. Borsotti, M., Campadelli, P., Schettini, R.: Quantitative Evaluation of Color Image Segmentation Results. Pattern Recognition Letters, vol.19 (1998) 741-747
3. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. PR(34), No. 12, December (2001) 2259-2281
4. Liu, J., Yang, Y.-H.: Multiresolution Color Image Segmentation. IEEE Trans. on PAMI, 16(7) (1994) 689-700
5. Lucchese, L., Mitra, S.K.: Advances in Color Image Segmentation. Proc. of Globecom'99, Rio de Janeiro, Brazil, 5-9 Dec. (1999) 2038-2044
6. Martínez-Usó, A., Pla, F., García-Sevilla, P.: A Quadtree-based Unsupervised Segmentation Algorithm for Fruit Visual Inspection. LNCS 2652, Pattern Recognition and Image Analysis, F.J. Perales et al (Eds). Springer-Verlag. (2003) 510-517
7. Press, W.H., Teukolsky, S.S., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press. (1994)
8. Samet, H.: Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS. Addison-Wesley, (1990)
9. Wilson, R.G., Spann, M.: Finite Prolate Spheroidal Sequences and their Applications II: Image Feature Description and Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(2) (1988) 193-203