Real Time Image Segmentation Using an Adaptive Thresholding Approach*

P. Arques, F. Aznar, M. Pujol, and R. Rizo

Department of Computer Science and Artificial Intelligence
University of Alicante
{arques, fidel, mar, rizo}@dccia.ua.es

Abstract. The aim of image segmentation is the partition of the image in homogeneous regions. In this paper we propose an approximation based on Markov Random Fields (MRF) able to perform correct segmentation in real time using colour information. In a first approximation a simulated annealing approach is used to obtain the optimal segmentation. This segmentation will be improved using an adaptive threshold algorithm, to achieve real time. The experiment results using the proposed segmentation prove its correctness, both for the obtained labelling and for the response time.

Keywords: Region Segmentation, Adaptive Thresholding, Simulated Annealing, Real Time, Markov Random Field.

1 Introduction

Image segmentation is the process that divides an image into a set of disjointed regions whose features, such as intensity, colour, texture, etc. are similar. Image segmentation can be considered as a labelling task with a set of homogeneous states and a discrete set of labels, therefore, the Markov Random Field theory can be applied to characterize the problem of trying to find an optimal solution.

In general, segmentation algorithms are based on two important criteria to be considered: the first is the homogeneity of the region and the second is the discontinuity between adjacent, disjointed regions. Although there are a wide variety of image interpretation techniques that are well treated in [1], [2], it is a very difficult task to satisfy all the properties for the optimal set of segmented regions. Normally, the resulting segmented image depends on a set of predetermined threshold values, so algorithms frequently fails to merge regions that must be separated or fails to split the regions that need to be together. These problems are due to the fact that the information about region uniformity or about discontinuity between different regions is not well integrated into segmentation algorithms. Therefore, it is interesting to incorporate features which are robust to some distortion level, so that solid results are produced.

* This work has been financed by the Generalitat Valenciana project GV04B685.

R. Marín et al. (Eds.): CAEPIA 2005, LNAI 4177, pp. 389–398, 2006.

Due to the large number of pixels on which the MRF is to be defined, it usually takes a long computional time to obtain optimal labels, which makes the MRF model difficult to apply to real scene domains. Most current approaches to segment colour images [3] [4] do not consider the processing time to apply it in real time systems. Image segmentation is a previous process to recognize objects. In our case, object recognition will be integrated in an autonomous agent to recognize objects, so real time in the segmentation process is needed.

The MPMT algorithm [5], [6], is used to obtain a preliminary segmentation and subsequently a MRF model in the segmented regions set is defined, so that the number of states that should be considered is significatively reduced.

To measure the region uniformity several first order statistics such as intensity average and variance are considered. Next, a border process will be introduced to reflect the measure of discontinuity along the common border between two adjacent regions. Before applying the optimal segmentation algorithms a process to remove isolated pixels is executed.

The process of searching for an optimal segmentation is time consuming so, it is a bottleneck for real time systems. In a first approximation, optimal labelling is proposed by minimizing an energy function using the simulated annealing algorithm.

In a second approximation, named Adaptive Thresholding, similar results were obtained for the optimal segmentation, but the processing time is much shorter. Therefore, this process could be used in real time systems.

2 Previous Segmentation

For the preliminary segmentation step, the Modified Partition Mode Test, MPMT [5], [6], has been chosen due to two main reasons: ease of implementation and it provides the necessary information to begin use of the MRF model. The segmentation algorithm based on modified partition mode test MPMT divides the input image into subareas (called windows). The MPMT algorithm is applied to assign a label to each window. A unique partition mode is assigned to each window, and also the windows are chosen so that they overlap. In this way, each pixel is covered by several different windows. The implemented algorithm uses overlapping 2x2 windows as basic subareas. The basic segmentation of subareas is done assigning one of the 12 partition modes to each window (See figure 1).

Nevertheless, the presegmentation MPMT algorithm provides an oversegmentation by nature (it only scans the image once, while the division into regions is

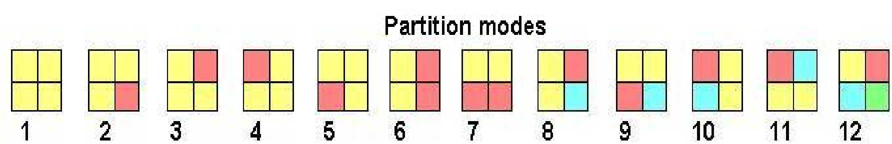

Fig. 1. Twelve partition modes used in the algorithm

done). Sometimes, this over-segmentation also produces some isolated points in the image. Isolated points add noise and complexity, due to the fact that they dramatically increase the number of regions.

Although in later segmentation processes these isolated point are labelled correctly, a new postprocess is performed (See table 1) to speed up the segmentation algorithm.

Table 1. MPMT Postprocess

```
ALGORITHM Postprocess MPMT
If Area(R_i)==1 then
   For all Neighbour(R_i)
      Calculate Distance(R_i,R_j) ∀j
      Choose Minimum Distance
      If Minimum Distance < Threshold then
         Label(R_i)=Label(R_j)
         Calculate
         (Avg_r(R_j), Avg_g(R_j), Avg_b(R_j))
         Area(R_j)++
      else
         label(R_i)=label(R_j)
         Area(R_j)++
end ALGORITHM
```

This algorithm introduces a higher definition in the object characteristics in the image. The postprocess performs a single scan for the complete image, selecting those regions made up of 1 pixel, in other words, isolated points; next the distance between each isolated point to all its neighbours is calculated and the minimum distance is chosen. At this point two cases need to be considered:

- **Point with a wrong label.** The distance between this isolated point and one of its neighbouring regions is smaller than the threshold used to join regions. In this case, the isolated point is labelled using the neighbouring region label and the average intensity of the region is calculated taken into consideration the value of the isolated point.
- **Isolated point.** The minimum distance between this isolated point and one of its neighbouring regions is greater than the threshold used to join regions. In this case the isolated point is labelled again using the nearest region label. It is supposed that this point is just noise.

3 Obtaining Regions Using Simulated Annealing

Simulated annealing algorithm is based on randomization techniques and it is related to iterative improvement algorithms [6]. This algorithm is frequently used to segment regions to obtain an optimal labelling given an energy function, as can be seen in [7], [8], [9]. Next, an energy function for segmentation by simulated annealing is presented. It is suitable for colour images.

For merging or splitting regions, similar colour criteria are applied in the definition of the energy function. In RBG colour space, the difference between two colours is obtained calculating the euclidean distance.

In our segmentation model, the energy function is defined as follows:

$$U(CE/R, F) = \sum_{c} V_c(CE/R, F) \tag{1}$$

where CE is the current label configuration, R is the region process (measure of the regions homogeneity) and F is the boundary process (measure of the discontinuity in the limits of adjacent regions).

The initial energy function is calculated by adding the intensity differences between adjacent regions.

Let $F = \{F_{red}, F_{green}, F_{blue}\}$ the set of region processes.

where F_i = average intensity in region i.

The spectrum features in region R_i could be defined as:

$F_i\{F_{red}, F_{green}, F_{blue}\}$

To achieve an optimal image segmentation the following restrictions are imposed:

- A segmentation on simple regions is caused by uniform spectrum features.
- In the common border between two different segmented regions strong discontinuities of intensity must exist.

Given a system of neighbourhoods on a lattice, we define a clique C as either a single site, or a set of sites of the lattice, in such a way that all the sites that belong to C are neighbours of each other.

So, the clique function [6] referring to region c is defined as:

$$\sum_{R_i \in C} \sum_{\substack{R_j \in C \\ R_j \neq R_i}} |F_i - F_j| \tag{2}$$

Function (2) calculates the initial energy of the adjacency graph. The frontier process is added to this equation, that is, the impact caused by merging two regions, in the individual process of each region and in the global process of the adjacency graph. The resulting energy function, the consequence of merging two regions, is defined as:

$$\sum_{R_i \in C} \sum_{\substack{R_j \in C \\ R_j \neq R_i}} |F_i - F_j| + NAdj_{i,j} |F_i - F_j| \tag{3}$$

Where $NAdj_{i,j}$ is the maximum value of adjacency of regions R_i and R_j.

Two regions should be merged only if their union decreases the global energy function, in other cases they should remain as independent regions.

The algorithm used to minimize the energy function is the *Simulated Annealing* algorithm [10], a robust method which allow us to perform an optimal image segmentation, as shown in figure 2, where *frontier segmentation* represents

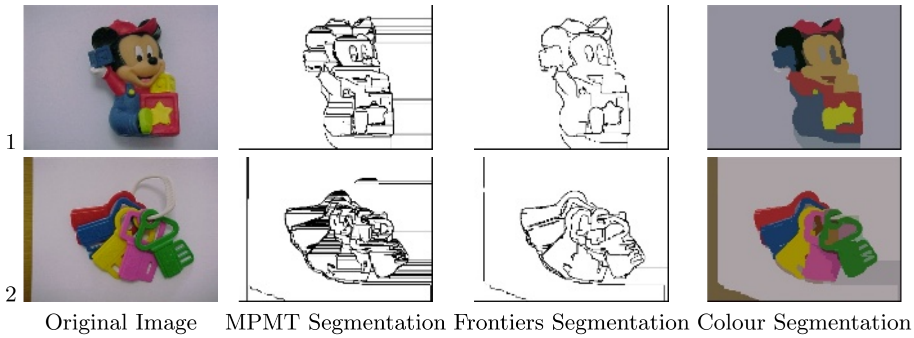

Fig. 2. Segmentation obtained using the proposed energy function

the object's frontiers when the algorithm is performed, and *colour segmentation* represents the same image as frontier segmentation but with the recuperation of colour of the objects detected by the algorithm.

4 Adaptive Thresholding

A natural way to segment an image into regions is using the thresholding technique, that is, dividing the image into bright and dark regions.

Thresholding generates binary images from grey level images, converting pixels which are below a certain threshold to 0 and those which are above the threshold to 1.

There are several approximations to automatically calculate an image threshold [12]. The *Otsu approximation* [13] assumes that the histogram is the addition of two gaussians. This threshold should minimize the weighted sum of the variance of each present object. It is supposed that the closer the gaussian to the real histogram, then the smaller the standard deviation.

The Otsu approximation has been modified to deal with RGB images. To obtain an optimal threshold, the average and the variance of each channel (red, green and blue) is used, among a set of images with similar illumination levels. In this way, the optimal threshold is the average value of standard deviation of each channel.

The threshold calculation consists of:

– Selecting a group of images (20 images have been used in our experiments) with the same luminosity as the image to be segmented. For each image the probabilities of each channel: red, green and blue, are calculated.

$$q_{channel}(t) = \sum_{i=0}^{nxm} P_{channel}(i) \tag{4}$$

where mxn are the image dimensions. The average of each channel probability is also calculated,

$$\mu_{channel} = \frac{1}{N}\sum_{t=0}^{N} q_{channel}(t) \tag{5}$$

where N is the maximum intensity value of each channel.

And finally, the standard deviation for each channel Red, Green and Blue is also calculated.

$$\sigma_{channel} = \frac{1}{N}\sqrt{\sum_{i=0}^{nxm}(P_{channel}(i) - \mu_{channel})^2} \tag{6}$$

- Once the data for all the samples are calculated, the threshold is obtained as the average of each channel standard deviation.

$$Threshold = \frac{1}{3 * p}\sum_{i=0}^{p}(\sigma^i_{red} + \sigma^i_{green} + \sigma^i_{blue}) \tag{7}$$

where p is the sample size and σ^i_{red} is the standard deviation for the red channel of the $i-$th image of the sample $i = 1..p$.

The value obtained after the process, supplies a suitable threshold to segment images in an adequate way.

Thanks to the use of real images the suitable threshold can be automatically calculated when the illumination conditions vary. As we are working with the standard deviation average of each channel, the actual property that is searched is the maximum difference of each channel.

4.1 Region Segmentation Using Adaptive Thresholding

The adjacency graph resulting from the MPMT algorithm and the isolated points removal, allow the system to obtain a certain label configuration and a neighbourhood definition. By using the adaptive thresholding algorithm a label configuration will be obtained using the previously defined threshold as a criteria for merging or dividing regions.

The *Adaptive Thresholding* algorithm consists of the following steps: taking the label configuration obtained from the MPMT pre-segmentation algorithm and the isolated points removal, then the adjacency graph must be scanned and the distance from one region to its neighbouring regions calculated. From the set of regions whose distance is lower than the calculated threshold, the smallest distance region is chosen. The adjacency graph is re-calculated and the process of region comparison continues. The process is repeated until all the regions with lower distance than the threshold are joined (See table 2).

Where CE is the label configuration obtained by the MPMT process, U is the calculated threshold for the images of this level of illumination and $Adj(R_i, R_j)$ is the next function:

$$Adj(R_i, R_j) = \begin{cases} 1 \text{ if regions } R_i \text{ y } R_j \text{ are adjacents.} \\ 0 \qquad\qquad\quad \text{in other case} \end{cases} \tag{8}$$

Table 2. Algorithm to merge regions

```
ALGORITHM MergingRegions
∀ R_i ∈ CE
∀ R_j ∈ CE so that R_j ≠ R_i
    If Adj(i, j) then
        Calculate Distance(R_i, R_j)
        Choose LowestDistance(R_i, R_j) so that Distance(R_i, R_j) < Threshold
            MergeRegions(R_i, R_j)
            Re − calculate CE
Repeat while CE is modified
EALGORITHM
```

Where $R = \{R(r), R(g), R(b)\}$ represents the regions processes set and R_i is the average value of colour of region i. The spectral features of region R_i should be defined as: $R_i = \{R_i(r), R_i(g), R_i(b)\}$

The distance between two adjacent regions is defined as:

$$Distance(R_i, R_j) = \sqrt{(R_i(r) - R_j(r))^2 + (R_i(g) - R_j(g))^2 + (R_i(b) - R_j(b))^2} \quad (9)$$

Where $LowestDistance(R_i, R_j)$ is the process to calculate the lowest distance between region i and all its neighbours, $MergeRegions(R_i, R_j)$ is the process to calculate the values of the new region features, $Re - calculate\ CE$ is the process which modifies the adjacent graph with the new region.

5 Experimentation

A comparison between the two approaches has been carried out. This comparison is based on the resulting number of regions and the computational time required to execute each proposed algorithm.

Several kind of images have been chosen, figures 2 and 3 show coloured objects with well-defined shapes, figure 4.1 shows a real outdoor image with a traffic sign

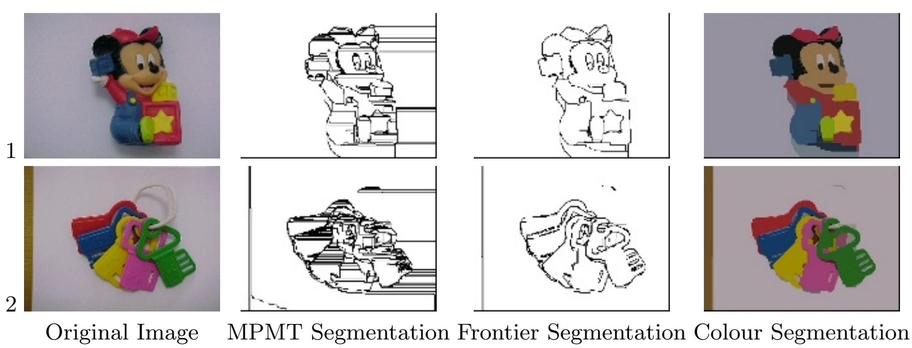

Fig. 3. Segmentation using the adaptive thresholding algorithm

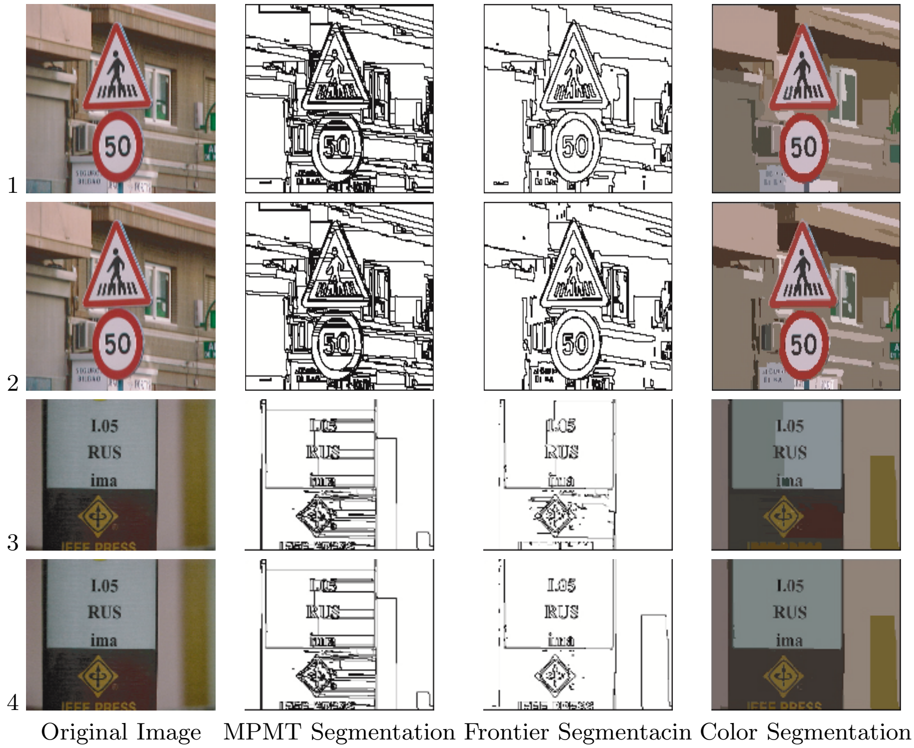

Fig. 4. Image Segmentation

Table 3. Two segmentation approximation comparison. The tests were conducted with a Pentium 4 Processor, with 512 RAM MB, and 2.4 GHz.

	MPMT		Simulated Annealing		Adaptive Thresholding	
Image	Labels	Time	Labels	Time	Labels	Time
2.1 and 3.1	280	209ms	62	1.285 ms	46	340 ms
2.2 and 3.2	488	247ms	97	4.517 ms	32	459 ms
4.1 and 4.2	1155	779ms	229	67.851 ms	67	2.393 ms
4.3 and 4.4	441	183ms	98	2.939 ms	54	398 ms

and figure 4.3 shows a book spine. Experimentation has been carried out using real images in JPEG format obtained by a digital camera.

Simulated annealing algorithm has been applied to figures 2.1, 2.2, 4.1 and 4.3 and adaptive thresholding algorithm has been applied to figures 3.1, 3.2, 4.2 and 4.4. Table 3 presents the comparisons in processor time and in the number of regions for each image. As shown, time needed is dependant on the complexity of the image.

As can be seen in the above images the optimal segmentation using the *simulated annealing* and the segmentation obtained by the *adaptive thresholding* approximation are similar. However, temporal costs are quite different. The adaptive thresholding algorithm considerably reduces temporal costs in the segmentation process, decreasing by up to 30 times, the processing time to obtain an optimal segmentation. Therefore, an adaptive thresholding algorithm could be more suitable to use in systems which need real time segmentation.

6 Conclusions

Image segmentation could be defined as an MRF problem. Most of todays colour image segmentation approximations do not consider the processor time for real time applications. In our case, we have an autonomous agent for recognizing objects. So that, segmentation is a previous process for object recognition and it should be executed in real time.

Starting form MRF theory to carry out a robust image segmentation, defining a neighbourhood system and an adjacency graph for images, two algorithms have been applied to obtain an optimal labelling.

The first algorithm uses a *simulated annealing* approach to minimize a proposed energy function for colour image segmentation. Using *simulated annealing* an optimal segmentation is obtained although it requires a longer processor time. Nevertheless, in real time systems, the optimal solution if often relaxed to obtain shorter temporal costs. So, a second approach based on an *adaptive threshold* criteria has been proposed to merge or divide regions. Although , this second algorithm can not guarantee the return of an optimal segmentation, it provides satisfactory results. Moreover, it is possible to use it in a real time segmentation, due to its low temporal cost.

Satisfactory results have been obtained proving the proposed algorithm validity. Future works will focus on object recognition using the presented segmentation approach.

References

1. Pal, N., Pal, S.: A review on image segmentation techniques. Pattern Recognition **26** (1993) 1294–1993
2. Muoz, X., Freixenet, J., Cufi, X., Marti, J.: Strategies for image segmentation combining region and boundary information. Pattern Recognition Letters **24** (2003) 375–392
3. Martinez-Uso, A., Pla, F., Garcia-Sevilla, P.: Color image segmentation using energy minimization on a quadtree representation. International Conference on Image Analysis and Recognition ICIAR'04. Porto (2004)
4. Kim, B., Shim, J., Park, D.: Fast image segmentation based on multi-resolution analysis wavelets. Pattern Recognition Letters **24** (2003) 2995–3006
5. Suk, M., Chung, S.: A new image segmentation technique based on partition mode test. Pattern Recognition **16** (1983) 469 – 480
6. Pujol, M.: Incorporacion de caracteristicas en la funcion de energia para segmentacin de imagenes usando campos aleatorios de Markov. PhD thesis, Departamento de Ciencia de la Computacion e Inteligencia Artificial. Universidad de Alicante (2000)

7. Arques, P., Pujol, M., Rizo, R.: Robust segmentation of scenes with colour mark. Frontiers in Artificial Intelligence and Applications. Artificial Intelligence Research and Develop **100** (2003) 149–159
8. Lievin, M., Luthon, F.: Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video. IEEE Transactions on Image Processing **13** (2004) 1–9
9. Luo, J., Guo, C.: Perceptual grouping of segmented regions in color images. Pattern Recognition **36** (2003) 2781–2792
10. Azencott, R.: Simulated Annealing. Parallelization Tecniques. John Wiley & Sons (1999)
11. Sahoo, P., Soltani, S., Wong, A., Chen, Y.C.: A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing **41** (1988) 233–260
12. Weszka, J.S., Rosenfeld, A.: Threshold evaluation techniques. IEEE Transactions on Systems, Man and Cybernetics **SCM-8** (1978) 622–629
13. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics **SCM-9** (1979) 62–66