Proceedings of the Sixth IASTED International Conference
Visualization, Imaging, and Image Processing
August 28–30, 2006, Palma de Mallorca, Spain
ISBN Hardcopy: 0-88986-598-1 / CD: 0-88986-600-7

COLOR IMAGE SEGMENTATION USING A SELF-INITIALIZING EM ALGORITHM

Dana Elena Ilea and Paul F. Whelan
Vision Systems Group, School of Electronic Engineering
Dublin City University, Glasnevin, Dublin 9
Ireland
{danailea, whelanp}@eeng.dcu.ie

ABSTRACT
This paper presents a new method based on the Expectation-Maximization (EM) algorithm that we apply for color image segmentation. Since this algorithm partitions the data based on an initial set of mixtures, the color segmentation provided by the EM algorithm is highly dependent on the starting condition (initialization stage). Usually the initialization procedure selects the color seeds randomly and often this procedure forces the EM algorithm to converge to numerous local minima and produce inappropriate results. In this paper we propose a simple and yet effective solution to initialize the EM algorithm with relevant color seeds. The resulting self-initialised EM algorithm has been included in the development of an adaptive image segmentation scheme that has been applied to a large number of color images. The experimental data indicates that the refined initialization procedure leads to improved color segmentation.

KEY WORDS
Color segmentation, EM, initialization, and diffusion filtering

1. Introduction

Statistical partitioning of color images into meaningful regions is the goal of many image analysis algorithms. In this regard, the need for automated techniques to extract the color information is of particular interest as the segmentation process will reduce the large number of color components existing in the initial image into a reduced number of components in the segmented result, which are strongly related to the image objects [1,2,3]. The area of color image analysis is one of the most active topics of research and a large number of color segmentation techniques have been proposed. Color segmentation techniques that have been proposed include histogram-based segmentation [4], probabilistic space partitioning and clustering methods [5,6,7], Markov random field and simulated annealing [8].
In this paper we have developed a color extraction technique based on probabilistic partitioning of the color space using the Expectation-Maximization (EM) algorithm [9,10,11]. The EM space partitioning algorithms are convergent and aim to optimize the partitioning decisions based on an initial set of Gaussian Mixture Models (i.e. attempts to identify the optimal partitions by maximizing the likelihood of the input data). A proper starting condition is important, otherwise the algorithm will be forced to converge to numerous local minima and produce erroneous decisions [10,12,13,14]. The most common initial stage for space partitioning algorithms consists of a simple procedure that selects the cluster centres randomly from input data [10,14,15]. This approach is far from optimal and does not eliminate the main problem faced by the space partitioning algorithms, namely converging to local minima. In addition the segmentation results returned by the EM algorithm will be different any time the algorithm is executed. To circumvent the problems caused by the random procedure, some authors use the result of the K-Means algorithm to set the initial cluster centres. The advantage of using this strategy is minimal since this does not assure that the K-Means algorithm will not be trapped itself in local minima decisions [11,14]. The experimental results indicate that this solution is not any better than the random initialization procedure. In this paper we propose to perform the selection of the cluster centres by extracting the dominant colors that are used to initialize the EM algorithm from the color histograms.
The devised initialization procedure is generic and it can be used in conjunction with other space partitioning techniques such as K-Means or fuzzy clustering. It useful to note that other implementations that address the problem of initialization the space partitioning techniques are documented in the computer vision literature and relevant approaches include the work detailed in [10,12,13,14]
To further refine the color extraction results we have developed a post-processing step that adaptively merges the image regions resulting from the EM algorithm in order to obtain optimal segmentation. To eliminate the errors caused by image noise we initial filter the data with a diffusion-based smoothing operator and accurate results have been obtained when the initial data is immersed in noise.

2. The EM Framework

The EM algorithm is implemented using an iterative framework that attempts to calculate the maximum likelihood between the input data and a number of Gaussian distributions (Gaussian Mixture Models-GMM) [9,10,11]. The main advantage of this probabilistic strategy over rigid clustering algorithms such as K-Means is its ability to better handle the uncertainties during the mixture assignment process. Assuming that we try to approximate the data using M mixtures, the mixture density estimator can be calculated using the following expression:

$$p(x \mid \Phi) = \sum_{i=1}^{M} \alpha_i \, p_i(x \mid \Phi_i) \tag{1}$$

where x=[x_1, …, x_k] is a k–dimensional vector, α_i is the mixing parameter for each GMM and $\Phi_i=\{\sigma_i, m_i\}$, σ_i, m_i are the standard deviation and the mean of the mixture. The function p_i is the Gaussian distribution and is defined as follows:

$$p_i(x \mid \Phi_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{\|x-m_i\|^2}{2\sigma_i^2}}, \quad \sum_{i=0}^{M} \alpha_i = 1 \tag{2}$$

The algorithm consists of two steps, the expectation and maximization step. The expectation step (E-step) is represented by the expected log-likelihood function for the complete data as follows:

$$Q(\Phi, \Phi(t)) = E[\log p(X, Y \mid \Phi) \mid X, \Phi(t)] \tag{3}$$

where Φ(t) are the current parameters and Φ are the new parameters that optimize the increase of Q. The M-step is applied to maximize the result obtained from the E-step.

$$\Phi(t+1) = \arg\max_{\Phi} Q(\Phi \mid \Phi(t)) \text{ and }$$
$$Q(\Phi(t+1), \Phi(t)) \geq Q(\Phi, \Phi(t)) \tag{4}$$

The E and M steps are applied iteratively until the increase of the log-likelihood function is smaller than a threshold value. The updates for GMMs can be calculated as follows:

$$\alpha_i(t+1) = \frac{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))}{N} \tag{5}$$

$$m_i(t+1) = \frac{\sum_{j=1}^{N} x_j \, p(i \mid x_j, \Phi(t))}{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))} \tag{6}$$

$$\sigma_i(t+1) = \frac{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t)) \left\| x_j - m_i(t+1) \right\|^2}{\sum_{j=1}^{N} p(i \mid x_j, \Phi(t))}$$

$$\text{where } p(i \mid x_j, \Phi) = \frac{\alpha_i \, p_i(x_j \mid \Phi_i)}{\sum_{k=1}^{M} \alpha_k \, p_k(x_j \mid \Phi_K)} \tag{7}$$

3. Overview of the Algorithm

The main components of the color segmentation algorithm detailed in this paper are illustrated in Fig. 1.

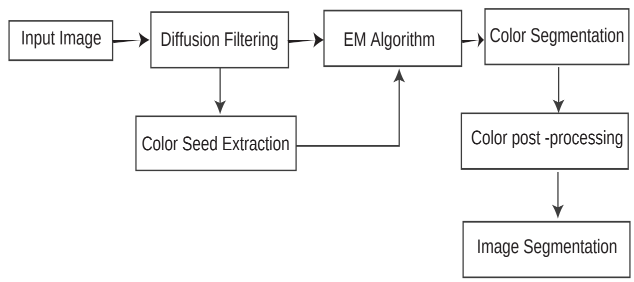


Fig. 1. Overview of the color segmentation algorithm.

The first step of the algorithm consists of filtering the color image with an anisotropic diffusion filter in order to improve the local color homogeneity and to reduce the level of image noise. The second step deals with the extraction of dominant colors from the filtered color image that are used to initialize the EM algorithm. The color segmentation result is further refined by joining the image regions resulting after the application of the EM algorithm based on the evaluation of the inter region variability. In this regard, we have developed a post-processing step that joins the image regions resulted from the EM algorithm based on a region adjacency graph and color similarity.

4. Diffusion-based Filtering

The image smoothing techniques can be broadly divided into two categories, linear and non-linear [16,17]. The standard linear smoothing filtering based on local averaging or Gaussian weighted spatial operators reduces the level of noise but this advantage is obtained at the expense of poor feature preservation (i.e. suppression of narrow details in the image). For this implementation we have developed a smoothing filtering scheme based on anisotropic diffusion where smoothing is performed at intra regions and suppressed at region boundaries

[18,19,20]. This non-linear smoothing procedure can be defined in terms of the derivative of the flux function:

$$u_t = div(D(|\nabla u|)\nabla u) \tag{8}$$

where u is the input data, D represents the diffusion function and t indicates the iteration step. The smoothing strategy described in equation (8) can be implemented using an iterative discrete formulation as follows:

$$I_{x,y}^{t+1} = I_{x,y}^{t} + \lambda \sum_{j=1}^{4} [D(\nabla_j I)\nabla_j I]$$

$$D(\nabla I) = e^{-\left(\frac{\nabla I}{k}\right)^2} \in (0,1] \tag{9}$$

where $\nabla_j I$ is the gradient operator defined in a 4-connected neighbourhood, λ is the contrast operator that is set in the range $0< \lambda<0.16$ and k is the diffusion parameter that controls the smoothing level. It should be noted that in cases where the gradient has high values the value for $D(\nabla I)\to 0$ and the smoothing process is halted.

5. Automatic Color Seed Generation

5.1 Heuristic initialization

As it can be observed in equation (1), the EM algorithm requires to be initialized with the parameters for mixtures $\Phi_i=\{\sigma_i, m_i\}$, $i = 1\ldots M$. As the random selection of m_i from image data is not an appropriate solution, in this paper we developed a novel initialization procedure for the EM algorithm using the selection of dominant color components from the input image using the information contained in the color histograms. In this regard, we have constructed the histogram for each color channel where n_s is the number of pixels contained in the bin s. Next the histograms are partitioned linearly into R sections where $R > M$ is a fixed value and from each section is determined the bin that has the highest number of elements.

$$P_j = \arg \max_{j\in[1,R]} (n_s),\ \text{j = 1,…, M} \tag{10}$$

We continue with ranking the peaks obtained from each histogram of the color components in agreement with the number of elements. Finally, we form the color seeds starting with the histogram peak that has the highest number of elements. This process can be summarized by the following pseudo-code sequence:

1. Construct the histograms for each color channel.
2. Partition each histogram into M sections.
3. Compute the peaks in each section and rank the peaks, p_1, p_2,…, p_R where p_1 has the highest number of elements.
4. Start to form the color seeds for highest peak p_i:

   if($p_i\to$red) mark the pixels in the red channel and calculate the g_{mean} and b_{mean} for marked pixels from green and blue channels.

   if($p_i\to$green) mark the pixels in the green channel and calculate the r_{mean} and b_{mean} for marked pixels from red and blue channels.

   if($p_i\to$blue) mark the pixels in the blue channel and calculate the r_{mean} and g_{mean} for marked pixels from red and green channels.
5. Form the color seed and eliminate p_i from the list.
6. Repeat the steps 4 and 5 until the desired number of seeds has been reached.

5.2 EM initialization using color quantization

Another solution to initialize the parameters for mixtures $\Phi_i=\{\sigma_i, m_i\}$, $i = 1\ldots M$ with the dominant colors from the input image, consists of extracting the peaks from the color histogram calculated after the application of color quantization. For this implementation we applied linear color quantization [21,22,23] by re-sampling linearly the number of colors on each color axis [21]. The color quantization is performed as follows:

```
int ColorHistogram[noColors][noColors][noColors];
int  *levels;

//initialize the color histogram, compute the quantization levels
InitializeHistogram(ColorHistogram);
ComputeQuantizationLevels(levels, noColors);

for(int i=0;i<Image.RowNo();i++)
{
  for(int j=0;j<Image.ColNo();j++)
  {
      int R = (int)Image.Plane[0].Val[i][j];
      int G = (int)Image.Plane[1].Val[i][j];
      int B = (int)Image.Plane[2].Val[i][j];

      //compute the quantized value for each color component
      int quant_r = ExtractQuantizedValue(levels,noColors, R);
      int quant_g = ExtractQuantizedValue(levels,noColors, G);
      int quant_b = ExtractQuantizedValue(levels,no_colors, B);

      ColorHistogram[quant_r][quant_g][quant_b]++;
   }
}

int ColorPeaks[noMixtures];
//apply quick sort to extract the dominant colors
ExtractDominantColors(ColorPeaks,ColorHistogram);
```

The dominant colors contained in the image represented in the color space C are extracted by selecting the peaks from the color histogram as follows:

$$P_j = \arg\max_{C}(ColorHistogram),\ \text{j = 1,…, M} \tag{11}$$

Experimentally it has been observed that the EM initialization is optimal when the quatization levels are set to low values between 2 to 8 colors for each component (i.e. the quantized color image will have 8×8×8 colors (3 bits per each color axis) if the quantization level is set to 8). This is motivated by the fact that for low quantization levels the color histogram is densely populated and the peaks in the histogram are statistically relevant.

6. Image Segmentation

The image segmentation method used for this implementation is based on a split and merge algorithm [24] that evaluates adaptively the color and texture information. The first step of the algorithm recursively splits the image hierarchically into four sub-blocks using the texture information extracted using the Local Binary Pattern/Contrast (LBP/C) method [25,26]. The splitting decision evaluates the uniformity factor of the region under analysis that is sampled using the Kolmogorov-Smirnov Metric (KSM). In this regard, the pairwise similarity values of the four sub-blocks are calculated and the ratio between the highest and lowest similarity values are compared with a threshold value (split threshold). The region is splitted if this ratio is higher than the split threshold. This process continues until the uniformity level imposed by the split threshold is upheld or the block size is smaller than a predefined threshold value (for this implementation the smallest block size has been set to 16×16). During the splitting process two distributions are computed, the LBP/C distribution that defines the texture and the distribution of the color labels computed using the color segmentation algorithm detailed in Sections 2 to 5 that samples the color information.

The second step applies an agglomerative merging procedure on the image resulting after splitting in order to join the adjacent regions that have similar color-texture characteristics. This procedure calculates the merging importance (MI) between all adjacent regions in the split image and the adjacent regions with the smallest MI value are merged. Since the MI values sample the color-texture characteristics within the image, for this implementation we devised an adaptive scheme that is able to locally adapt to the image content by evaluating the uniformity of the color distribution. In this regard if the color distribution is homogenous (it is defined by one dominant color) the weights w_1 and w_2 are adjusted in order to give the color distribution more importance. Conversely, if the color distribution is heterogeneous the texture will have more importance. The calculation of the weights employed to compute the MI values for merging process (see equation 12) is illustrated in equations (13 to 14).

$$MI(r_1, r_2) = w_1 * KSM(TD_1, TD_2) + w_2 * KSM(CD_1, CD_2) \quad (12)$$

where r_1, r_2 represent the adjacent regions under evaluation, w_1 and w_2 are the weights for texture and color distributions respectively, KSM defines the Kolmogorov-Smirnov Metric, TD_i is the texture distribution for region i and CD_i is the color distribution for region i. The weights w_1 and w_2 are calculated as follows:

$$K_i = \frac{\arg\max_C (CD_i)}{N_i}, \; K_i \in (0,1] \text{ and } i = 1,2 \quad (13)$$

where $\arg\max_C (CD_i)$ is the bin with the maximum number of elements in the distribution CD_i, N_i is the total number of elements in the distribution CD_i and C is the color space.

$$w_2 = \frac{\sum_{i=1}^{2} K_i}{2} \text{ and } w_1 = 1 - w_2 \quad (14)$$

where w_1 and w_2 are the texture and color weights employed in equation (12). The merging process is iteratively applied until the minimum value for MI is higher than a pre-defined merge threshold.

The resulting image after the application of the merging process has a blocky structure since the regions resulting from the splitting process are rectangular. To compensate for this issue the last step of the algorithm applies a pixelwise procedure that exchanges the pixels situated at the boundaries between various regions using the color information computed using the color segmentation algorithm detailed in Sections 2 to 5. For more details about the image segmentation algorithm the reader can refer to the following paper [24].

7. Experiments and Results

The first tests were conducted in order to evaluate the influence of the initialization procedures detailed in Section 5 on the overall performance of the color segmentation algorithm (outlined in Fig. 1) when compared with the results returned by the random initialization procedure.

The self-initialized EM algorithm has been applied to a large suite of images and some experimental results are illustrated in Figs. 2 and 3. In order to evaluate only the influence of the initialization procedures for these tests no diffusion filtering is applied and the number of mixtures for the EM algorithm was set to 10. These tests are particularly useful to evaluate the optimal level of color quantization when the EM algorithm is initialized using the procedure detailed in Section 5.2.

(a) (b)

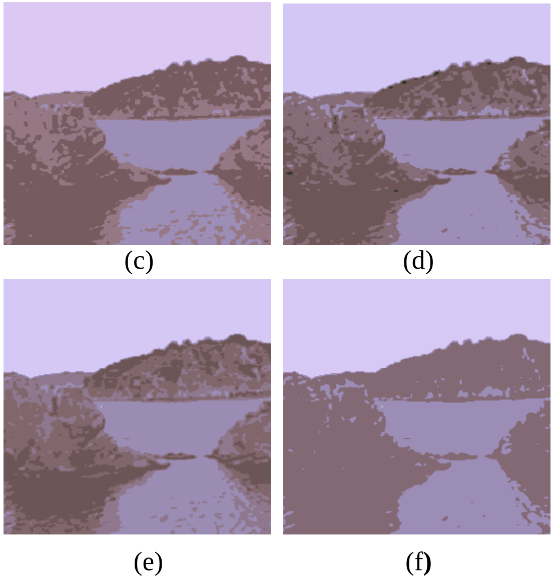


Fig. 2. Results of the color segmentation. Evaluation of the initialization procedure. (a) Input image. (b) Color segmentation result – Random initialization. (c) Color segmentation result – Heuristic initialization. (d,e,f) Color segmentation results – Color quantization. (d) Quantization level 4 - 2 bits per each color axis. (e) Quantization level 16 - 4 bits per each color axis. (f) Quantization level 64 - 6 bits per each color axis.

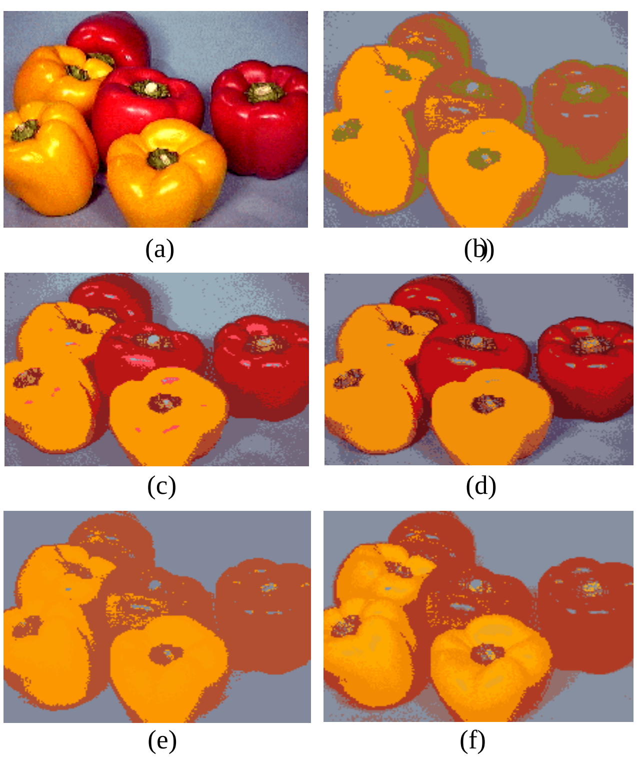


Fig. 3. Results of the color segmentation. Evaluation of the initialization procedure. (a) Input image. (b) Color segmentation result – Random initialization. (c) Color segmentation result – Heuristic initialization. (d,e,f) Color segmentation results – Color quantization. (d) Quantization level 4 - 2 bits per each color axis. (e) Quantization level 16 - 4 bits per each color axis. (f) Quantization level 64 - 6 bits per each color axis.

As Figs. 2 and 3 illustrate the optimal results are obtained when the EM initialization is performed using the technique based on color quantization presented in Section 5.2 (figures best viewed in color). An important parameter is the level of quantization and the experimental data indicates that optimal results are obtained when the quantization level is set to low values between 2 to 8 (maximum level is 256, 8 bits/color axis). If the quantization level is set to large values, the color histogram become sparse and this leads to a flat histogram with no apparent peaks that define the dominant colors. The performance of the EM algorithm when the heuristic procedure detailed in Section 5.1 is applied it is also significantly better than in the case when the mixtures Φ_i are initialized using the random procedure.

The second tests attempt to evaluate the performance of the developed color segmentation algorithm with respect to the parameter k required to be specified for diffusion filtering. The behaviour of anisotropic filtering is illustrated in equation (9) and it can be observed that the larger the value for k the more pronounced the smoothing. Thus, the value of this parameter should be increased when the color segmentation algorithm is applied to noisy images or images captured with low resolution.

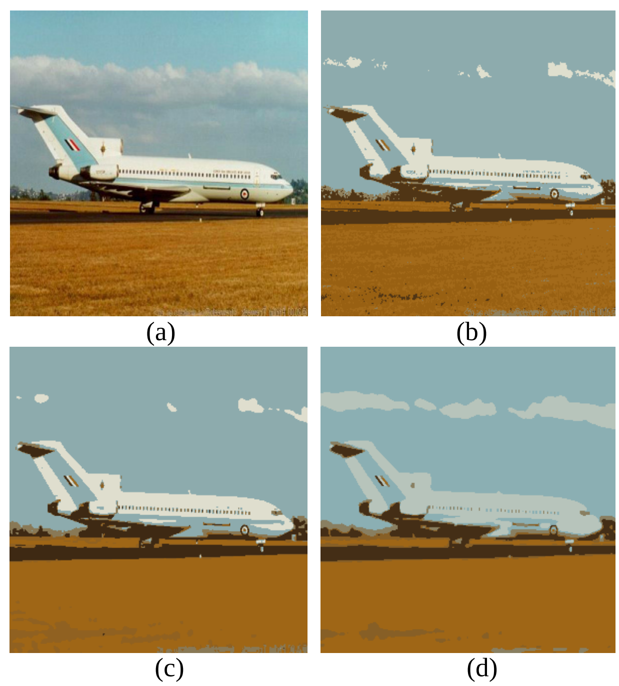


Fig. 4. Results of the color segmentation. Evaluation of the diffusion parameter. (a) Input noiseless image. (b) Color segmentation result – no diffusion. (c) Color segmentation result k=10. (d) Color segmentation result k=30.

The effect of this parameter on the color-segmented result is illustrated in Fig. 4 (the initial number of mixtures is set to 10). The initialization method for EM is based on color quantization and the level of quantization is set to 8. For all images illustrated in Fig 4(b-d) the final number of colors is 6.

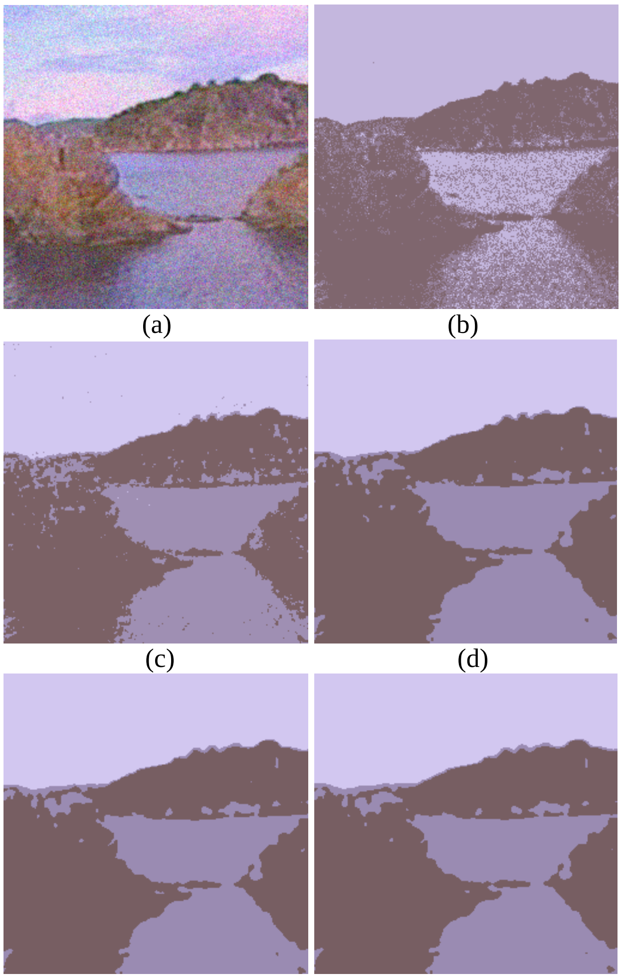
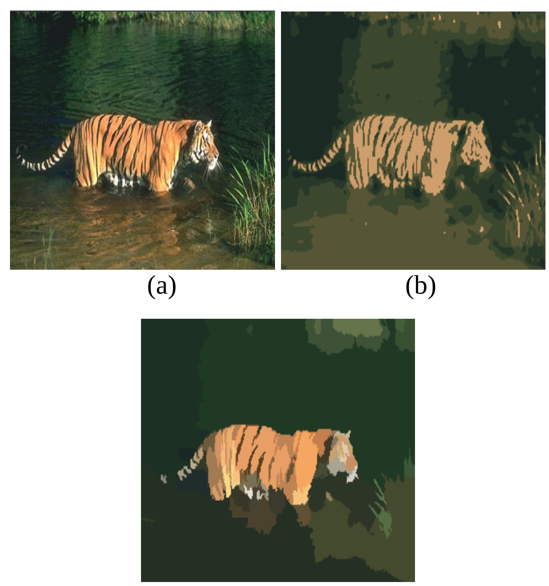


Fig. 5. Results of the color segmentation. Evaluation of the diffusion parameter. (a) Input nosy image. (b) Color segmentation result – no diffusion. (c) Color segmentation result *k*=10. (d) Color segmentation result *k*=20. (e) Color segmentation result *k*=30. (f) Color segmentation result *k*=40.

The effect of the diffusion parameter on the overall performance of the color segmentation algorithm is more noticeable when the EM algorithm is applied to noisy images. It can be observed that the color segmentation algorithm is not able to eliminate the image noise when no diffusion filtering is applied. The results illustrated in Fig. 5(c-f) indicate that the application of diffusion filtering virtually eliminates the influence of the image noise and local color in-homogeneities. As in the previous experiments the initialization method for EM is based on color quantization (quantization level 8) and the number of mixtures for EM algorithm is set to 10. The final number of colors in all images depicted in Fig. 5(b-f) is 3.
In order to illustrate the validity of our proposed scheme we have compared the results returned by our algorithm against those returned by a well-established mean shift color segmentation scheme developed by Comaniciu-Meer [5] at Rutgers University. In this regard, the initial evaluation was conducted on a synthetic image that has been corrupted with Gaussian noise and then on a number of natural images.

(a) (b) (c)

Fig. 6. Segmentation results. (a) Test image corrupted with Gaussian noise (mean deviation: 50 greyscale levels for each color). (b) Segmented image – our method (initial number of mixtures = 10, final number of colors = 3). (c) Segmented image – Comaniciu-Meer algorithm (final number of colors = 3).



(c)

Fig. 7. Segmentation results. (a) Natural image (b) Segmented image – our method (initial number of mixtures = 10, final number of colors = 5). (c) Segmented image – Comaniciu-Meer algorithm (final number of colors = 70).

The parameters for our algorithm are: number of mixtures for EM=10, diffusion parameter *k*=30 when applied to the noisy image depicted in Fig 6(a) and 10 for natural images, minimum difference between adjacent color regions: 20. The parameters for Comaniciu-Meer algorithm (Edison implementation [5]) are: spatial filter=8, color=50 and minimum region size=200 when applied to the noisy image illustrated in Fig. 6(a) and spatial filter=7.5, color=10 and minimum region size=25 when the algorithm was applied to the image depicted in Fig. 7(a). From these images we can notice that our algorithm outperforms the mean shift clustering algorithm when the algorithms were applied to the noisy image illustrated in Figure 6(a). When the algorithms were applied to the natural image depicted in Fig. 7(a) their performances are comparable but it is useful to notice that the results returned by the mean shift algorithm contain more colors than our algorithm. An important advantage of our color segmentation scheme over the Comaniciu-Meer algorithm is the fact that the color segmented output is relatively insensitive to changes of the input parameters whereas the segmentation result returned by the Comaniciu-Meer algorithm is very sensitive to changes in input parameters.

The third tests attempt to evaluate the efficiency of our algorithm when the color segmentation scheme is included in the development of the color-texture image segmentation algorithm outlined in Section 6. We have applied the image segmentation algorithm to natural images and medical images (detection of skin melanoma) and experimental results are depicted in Figs. 8 and 9. The input number of mixtures for EM algorithm is kept constant at 10 for all experiments.

Fig. 8. Segmentation results using the color-texture segmentation algorithm detailed in Section 6 when applied to natural image. The outlines of the segmented regions are over-imposed on the original images.

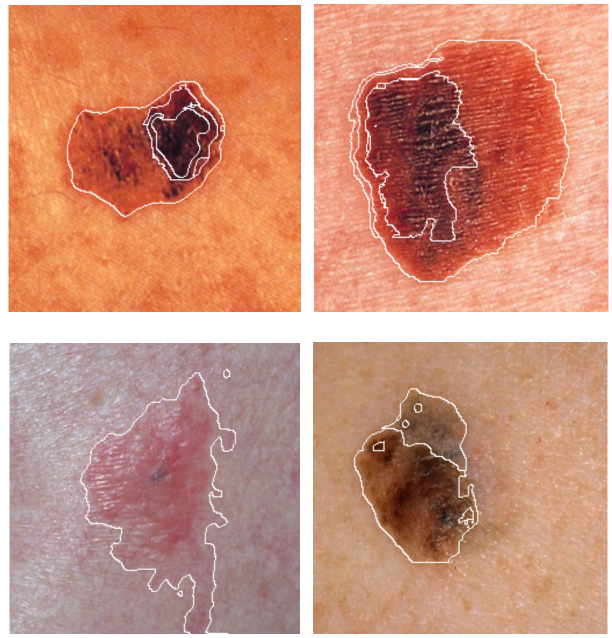


Fig. 9. Segmentation results when the color-texture algorithm has been applied to medical images. The outlines of the segmented regions are over-imposed on the original images.

8. Conclusion

In this paper we detailed the implementation of a new color segmentation method based on the EM algorithm. Since the EM algorithm is sensitive to the starting condition in this paper we detailed the implementation of two histogram-based techniques that are able to extract the dominant colors contained in the image to be analysed. The experimental data indicates that the devised algorithm offers reproducible and accurate color segmentation results and its performance is comparable with the performance offered by other established color segmentation techniques. The developed algorithm has been applied to a large number of images including synthetic, natural and medical images. The developed color segmentation scheme has been included in the development of an adaptive color-texture segmentation algorithm and we are particularly interested to apply this algorithm to practical implementations including the early detection of skin cancer and product inspection.

Acknowledgments

We would like to express our gratitude to Science Foundation Ireland (SFI) for supporting this research.

References

[1] Y. Deng & B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, *IEEE Trans. Pattern Analysis Machine Intell,, 23*(8), 2001, 800-810.

[2] J. Mukherjee, MRF Clustering for segmentation of color images, *Pattern Recognition Letters*, *23*(8), 2002, 917-929.

[3] H.D. Cheng, X. Jiang, Y. Sun, & J. Wang, Colour image segmentation: advances and prospects, *Pattern Recognition*, *34*(12), 2001, 2259-2281.

[4] L. Shafarenko, M. Petrou, & J. Kittler, Histogram-based segmentation in a perceptually uniform color space, *IEEE Trans. on Image Processing*, *7*(9), 1998, 1354-1358.

[5] D. Comaniciu & P. Meer, Mean Shift. A robust approach toward feature space analysis, *IEEE Trans. Pattern Analysis Machine Intell*, *24*(5), 2002, 603-619.

[6] A. Diplaros, T. Gevers, & N. Vlassis, Skin detection using the EM algorithm with spatial constraints, *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Hague, Netherlands, 2004, 3071-3075.

[7] L. Hermes, T. Zöller, & J. M. Buhmann, Parametric distributional clustering for image segmentation, *7th European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, 2002, 577-591.

[8] S. Lakshmanan & H. Derin, Simultaneous parameter estimation and segmentation of Gibbs Random Fields using simulated annealing, *IEEE Trans. Pattern Analysis Machine Intell.*, *11*(8), 1989, 799-813.

[9] J.A. Bilmes, *A Gentle Tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixed and Hidden Markov Models* (U.C. Berkely, California, April 1998, TR-97-021).

[10] P. McKenzie & M. Alder, Initializing the EM algorithm for use in Gaussian mixture modeling, *In E.S. Gelsema and L.N. Kanal, Editors, Pattern Recognition in Practice IV, Elsevier*, 1994, 91-105.

[11] A.P. Dempster, N.M. Laird, & D.B. Rubin, Maximum likelihood estimation from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society (B)*, *39*(1), 1977, 1-38.

[12] J.M. Pena, J.A. Lozano, & P. Larranaga, An empirical comparison of four initialization methods for the K-Means algorithm, *Pattern Recognition Letters*, vol. *20*(10), 1999, 1027-1040.

[13] C. Biernacki, G. Celeux, & G. Govaert, Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models, *Computational Statistics and Data Analysis*, *41*(3), 2003, 561-575.

[14] S. Khan & A. Ahmad, Cluster centre initialisation algorithm for K-Means clustering, *Pattern Recognition Letters*, *25*(11), 2004, 1293-1302.

[15] M. Meila & D. Heckerman, *An experimental comparison of several clustering methods* (*Microsoft Research Technical Report*, 1998, MSR-TR-98-06).

[16] M. Sonka, V. Hlavac, & R. Boyle, *Image processing, analysis and machine vision* (2nd edition, PWS Boston, 1998).

[17] K. Tang, J. Astola, & Y. Neuovo, Nonlinear multivariate image filtering techniques, *IEEE Trans. Image Processing*, *4*(6), 1995, 788-797.

[18] P. Perona & J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis Machine Intell*, *12*(7), 1990, 629-639.

[19] J. Weickert, Coherence-enhancing diffusion of colour images, *Image and Vision Computing*, *17*(3), 1999, 201-212.

[20] M. Black, G. Sapiro, D. Marimont, & D. Heeger, Robust anisotropic diffusion, *IEEE Trans. Image Proc.*, *7*(3), 1998, 421-432.

[21] F.S. Hill, *Computer graphics* (Macmillan Publishing 1990).

[22] X. Wu, Efficient statistical computations for optimal color quantization, *Graphics Gems 2*, Academic Press, 1991.

[23] J. Puzicha, M. Held, J. Ketterer, J.M. Buhmann, & D. Fellner, On spatial quantization of color images. *IEEE Trans. Image Processing*, *9*(4), 2000, 666-682.

[24] P. Nammalwar, O. Ghita, & P.F. Whelan, Integration of feature distributions for colour texture segmentation, *17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 1, 2004, 716 – 719.

[25] T. Ojala, M. Pietikainen, & T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Analysis Machine Intell.*, *24*(7), 2002, 971-987.

[26] V. Takala, T. Ahonen, & M. Pietikäinen, Block-based methods for image retrieval using local binary patterns, *14th Scandinavian Conference on Image Analysis*, Joensuu, Finland, 2005, 892-900.