Image Segmentation via Multiresolution Diffused Expectation-Maximisation

Giuseppe Boccignone, Vittorio Caggiano, Paolo Napoletano
Natural Computation Lab
DIIE-University of Salerno
Via Ponte Melillo 1, Fisciano (SA), Italy
Email: boccig@unisa.it

Mario Ferraro
Dipartimento di Fisica Sperimentale
University of Torino
Via P. Giuria 1, Torino, Italy
Email: ferraro@ph.unito.it

Abstract—Multiresolution Diffused Expectation Maximisation performs segmentation on vector (e.g. color) images within a multiscale framework; segmentation is carried out via the Expectation Maximisation algorithm, coupled with anisotropic diffusion on classes, in order to account for the spatial dependencies among pixels.

I. INTRODUCTION AND BACKGROUND

Segmentation, the partitioning of an image into meaningful parts, is an important research topic in image analysis and can potentially enhance the performance of many vision and multimedia applications. For instance applications supported by the MPEG-4 and MPEG-7 standard require the image content to be described at the object level so that they can be encoded differently and efficiently according to the levels of areas of interest.

A wide array of techniques, both for grey-level and color images, has been used in the past (for an in-depth survey see [1]), but so far there is no satisfactory solution to image segmentation of natural scenes. More recently an interesting class of segmentation schemes has been proposed, based on the Expectation Maximisation (EM) algorithm [2].

The EM algorithm allows one to achieve segmentation by a gradient descent search [3],[4], but a drawback of the EM algorithm, when applied to images in its standard form, is the lack of spatial constrains. To this end, several methods have tried to incorporate a prior term in order to maximise a log-posterior probability instead of log-likelihood, thus leading to quite complex EM steps (see, e.g., for a discussion [5]). With the same aim in [6] a Diffused Expectation-Maximisation (DEM) algorithm has been proposed for grey level images.

In this work we generalize the DEM algorithm in order to deal with color images within a multiresolution (pyramidal) representation, to take advantage of the fact that objects/regions to be segmented usually reside at different level of resolution.

Any image can be considered a set of $N$ unlabelled samples $F = \{f_1, f_2, \ldots, f_N\}$, on a 2-D discrete support $\Omega \subseteq Z^2$, with $N = |\Omega|$. Thus, an image segmentation /classification problem can be defined in probabilistic terms as the problem of assigning a label $k$ to each site $i$, given the observed data $F$, and where each label $k \in \{1, \ldots, K\}$ defines a particular region/model. Different models are selected with probability $P(k)$, and a sample is generated with probability distribution $p(f_i|k, \theta)$ where $\theta = \{\theta_k, k = 1, \ldots, K\}$ and $\theta_k$ is the vector of the parameters associated to label $k$. Thus $p(f_i|k, \theta)$ is the probability of $f_i$ given the parameters of all models and the fact that we have selected model (label) $k$. Each image can be conceived as drawn from a mixture density, so that, for any site (pixel),

$$ p(f_i|\theta) = \sum_{k=1}^{K} p(f_i|k, \theta) P(k), \quad (1) $$

and the likelihood of the data is $\mathcal{L} = p(f|\theta) = \prod_{i=1}^{N} p(f_i|\theta)$. For clarity’s sake, we define $p(f_i)$ and $\pi(f)$, two probability distributions; the former is the probability that a given grey level $f$ is assigned to pixel $i$, so that $\sum_i p(f_i) = 1$, whereas the latter is the probability that, given any pixel $i$, it has grey level $f$.

Image segmentation can be achieved by finding the set of labels that maximises the likelihood $\mathcal{L} = \prod_{i=1}^{N} \sum_{k=1}^{K} p(f_i|k, \theta) P(k)$, or, equivalently,

$$ \frac{1}{N} \log \mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} p(f_i|k, \theta) P(k). \quad (2) $$

By the weak law of large numbers and the ergodic theorem the term to be maximised can be written as

$$ E \left[ \log \sum_{k=1}^{K} p(f_i|k, \theta) P(k) \right] = $$

$$ = \sum_{f=0}^{L-1} \left( \pi(f) \log \sum_{k=1}^{K} p(f_i|k, \theta) P(k) \right), \quad (3) $$

$L$ being the number of grey levels. Simple manipulations lead to

$$ \frac{1}{N} \log \mathcal{L} = \sum_{f=0}^{L-1} \pi(f) \log \pi(f) + $$

$$ - \sum_{f=0}^{L-1} \left( \pi(f) \log \frac{\pi(f)}{\sum_{k=1}^{K} p(f_i|k, \theta) P(k)} \right) \quad (4) $$

0-7803-9134-9/05/$20.00 ©2005 IEEE
Hence a straightforward maximisation of \( \log \mathcal{L} \) can be obtained by minimising the second term of the last expression, namely the Kullback-Leibler (KL) distance
\[
D \left( \pi(f) \| \sum_{k=1}^{K} p(f|k, \theta)p(k) \right) \quad \text{between distributions } \pi(f) \quad \text{and } \sum_{k=1}^{K} p(f|k, \theta)p(k), \text{ while holding the first term fixed.}
\]
This is exactly what is performed by the classic EM algorithm [3] which minimises the KL distance between the manifold of the observed data and that of the true distribution. Alternatively, one could attempt a multistep approach by iteratively minimising the entropy
\[
H(f) = -\sum_{f=0}^{L-1} \pi(f) \log \pi(f), \quad \text{while holding } p(f|k, \theta), \quad P(k) \text{ fixed, and then minimising the KL distance } D, \quad \text{while keeping } H(f) \text{ fixed.}
\]
Interestingly, from the segmentation standpoint, the problem can be reformulated in a way which takes into account the spatial correlations among pixels. Define the spatial entropy as
\[
H_s = - \sum_{i=1}^{N} \pi(f_i) \log \pi(f_i) \quad \text{and probabilities } p(f_i) \quad \text{and } \pi(f) \text{ can be estimated as } p_i = p(f_i) \approx \frac{f_i}{\sum_{i=1}^{N} f_i} \quad \text{and } \quad \pi(f) \approx \frac{n_f}{N} \quad \text{where } n_f \text{ is the number of pixels with grey level } f \quad \text{and } \quad f_{tot} = \sum_{i=1}^{N} f_i = \sum_{f=0}^{L-1} n_f f.
\]
Therefore entropies \( H \) and \( H_s \) can be written as
\[
H(f) = - \sum_{f=0}^{L-1} \frac{n_f}{N} \log \frac{n_f}{N}, \quad \text{(5)}
\]
and
\[
H_s = - \sum_{f=0}^{L-1} \frac{n_f}{f_{tot}} \log \frac{f}{f_{tot}} \quad \text{(6)}
\]

As in the isotropic case, anisotropic diffusion is proved to increase the spatial entropy \( H_s \) [7]. In terms of the mixture model we are dealing with an incomplete data problem (i.e., we must simultaneously determine the labelling \( p(k|f) \) given distribution parameters \( \theta_k \) and viceversa). The DEM algorithm obtains the maximisation of \( \log \mathcal{L} \) by iteratively computing \( p(k|f, \theta), p(f|k, \theta), P(k) \) along the Expectation and Maximisation steps while diffusing on \( p(k|f, \theta) \), which in practice regularizes each \( k \) labelling field by propagating anisotropically such labels. Eventually, the segmentation is performed by using the estimated parameters \( k, \theta_k \).

II. MULTiresOLUTION DIFFUSED EXPECTATION

MAXIMISATION (MDEM)

A vector-valued (multi-valued) image is a smooth mapping from the image domain \( \Omega \subseteq \mathbb{R}^2 \) to an \( m \)-dimensional range, \( f : \Omega \rightarrow \mathbb{Z}^m \); in other terms, it is a set of single-valued images, or channels, sharing the same domain, i.e., \( f_i = (f_i)^T \), where \( i = 1, ..., m \) indexes a point in the lattice \( \Omega \). A color image can be considered a vector-valued image of three components \( m = 3 \) or color channels.

When dealing with vector-valued, color images, clearly one should be concerned in obtaining reliable segmentation results, while keeping acceptable computational costs. Both issues can be accounted for by using a multiresolution representation of the original image, e.g., a pyramidal representation. On the one hand, propagation of information gained from coarse resolution levels makes significant objects/regions in the image more relevant respect to weak textures and noise, as it is well known in object-based coding. On the other hand, pyramids provide an efficient tool to reduce the computational load both as regards the iterations necessary to maximise the likelihood - by initializing the parameter set on the basis of the parameters estimated at the coarser level - and for what concerns the diffusion step, which, at each iteration acts upon a sub-sampled version the probability maps.

A multiresolution representation can be derived from the original color image (here a YCrCb color space has been used, [8]), by means of a Gaussian pyramid [9] obtained by performing a low-pass filtering via convolution with a gaussian \( G(\sigma) \) [9], here \( l \) is the level of resolution, followed by a sub-sampling of the smoothed scalar field (here the color channel) [10]:
\[
f^{l+1}_i(r) = S \downarrow G(\sigma) * f^l_i(r) \quad \text{(8)}
\]
where \( f^{l+1}_i(r) = f_i(r) \), namely the highest resolution level corresponds to the original input field, and \( S \downarrow \) is the downsampling operator.

We will denote by \( \mathcal{P}\{f_i\} = \mathcal{P}\{f_i(\cdot)\} \) the Gaussian pyramid build on the scalar field \( f_i(\cdot) \); in our implementation, pyramids have a depth which is automatically computed if it is fixed the size of the image at the coarsest level. Such multiresolution representation is used as follows. At a certain level \( l \) of the pyramid, different from the lowest resolution level, we obtain the maximisation of \( \log \mathcal{L} \) by iteratively computing \( p(k|f, \theta), p(f|k, \theta), P(k) \) while diffusing on \( p(k|f, \theta) \). At such level, the labelling planes...
\[ p(k^{(l)} | f^{(l)}, \theta^{(l)}) \] are initialized by up-sampling the probability maps that have been previously derived at the coarser level \( l + 1 \):

\[ p(k^{(l)} | f^{(l)}) = S \uparrow p(k^{(l+1)} | f^{(l+1)}) \]  

(9)

This way the algorithm uses the probabilistic labelling proposed at coarser levels, while reducing the iteration steps necessary to achieve convergence of the expectation-diffusion-maximisation cycle at that level. The probabilistic model is assumed to be a mixture of multivariate gaussians

\[ p(f | k, \mu_k, \Sigma_k) = \frac{\exp(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k))}{(2\pi)^d/2 |\Sigma_k|^{1/2}}, \]  

(10)

\[ \theta_k = \mu_k, \Sigma_k \] being the unknown mean vectors and covariance matrices, respectively, weighted by mixing proportions \( \alpha_k = P(k) \). Note that, we can consider the covariance matrices being diagonal because of the choice of the Y’C’rCb color space, and, furthermore we assume \( K \) fixed, in that we are not concerned here with the problem of model selection.

In summary, the MDEM algorithm works as follows. In the learning stage, after initialization at the coarsest level \( L \) of the pyramid, for levels \( L - 1 \leq l \leq 0 \) probabilities are propagated, \( h^{(l)}_{i,ik} \rightarrow h^{(l+1)}_{i,ik} \), by up-sampling, according to (9). Next, the following steps are repeated until \( |\log \mathcal{L}^{(l+1)} - \log \mathcal{L}^{(l)}| < \epsilon \), where \( t \) denotes the iteration:

1) E-step: with fixed parameters \( \alpha^{(l)}_{i,k}, \mu^{(l)}_{i,k}, \Sigma^{(l)}_{i,k} \), compute the labelling probabilities at each site \( i \) as:

\[ h^{(t)}_{i,ik} = \frac{\alpha^{(t)}_{i,k} p(f^{(t)} | k_{i}^{(t)}, \mu^{(t)}_{i,k}, \Sigma^{(t)}_{i,k})}{\sum_k \alpha^{(t)}_{i,k} p(f^{(t)} | k_{i}^{(t)}, \mu^{(t)}_{i,k}, \Sigma^{(t)}_{i,k})}. \]  

(11)

2) D-step: propagate \( h^{(t)}_{i,ik} \) by \( m \) iterations of the discrete form of anisotropic diffusion

\[ h^{(t+1)}_{i,ik} = h^{(t)}_{i,ik} + \lambda \nabla \cdot (g(\nabla h^{(t)}_{i,ik}) \nabla h^{(t)}_{i,ik}) \]  

(12)

and set \( \tilde{h}^{(t)}_{i,ik} = h^{(t+1)}_{i,ik} \)

3) M-step: with \( \tilde{h}^{(t)}_{i,ik} \) fixed, calculate the parameters that maximise log \( \mathcal{L} \):

\[ \alpha^{(t+1)}_{i,k} = \frac{1}{N} \sum_i \tilde{h}^{(t)}_{i,ik} \]  

(13)

and calculate \( \log \mathcal{L}^{(t+1)} \).

The initialization of the MDEM algorithm at the coarsest level is performed, by running step 2 but with parameters initialized as reported in the next Section.

Segmentation is eventually achieved for each site \( i \in \Omega \), via estimated parameters by assigning to \( i \), the label \( k \) for which

\[ \max_k \{ p(f^{(0)} | k^{(0)}, \mu^{(0)}_{i,k}, \Sigma^{(0)}_{i,k}) \} \]  

holds.

### III. SIMULATION

We have experimented the method on different kinds of natural and sports images. The test images used in this paper are two, the first image is shown in Fig. 1(a) and the second is shown in Fig. 2(a).

At first, we applied the MDEM algorithm to the image of a landscape, Fig. 1(a), the depth of the pyramid is computed considering the size of the coarsest image be 20×20 pixel, then non uniform initial estimates were chosen for \( \alpha^{(0)}_{L,k}, \mu^{(0)}_{L,k}, \Sigma^{(0)}_{L,k} \) parameters at the coarsest level. Parameter \( \{\mu^{(0)}_{L,k}\} \) ranged from the minimum to the maximum value of \( f_i \), \( \{\sigma^{(0)}_{L,k}\} \) took values in the range from 1 to \( \max \{f_i\} \) and finally \( \{\alpha^{(0)}_{L,k}\} \) parameters.

The conductance function \( g \) used in the D-step of the DEM algorithm can have a quite general form, but must be such that that label boundaries are preserved, numerical stability guaranteed; the functions \( h_{ik}^{(t)} \) are renormalized so that their sum is one after each iteration [11]. In our experiments we set \( g(\nabla h_{ik}) = |\nabla h_{ik}|^{-9/5}, \lambda = 0.1 \). The number of iterations in (12) is automatically set as \( m = 3 \times (1 + L - l) \), \( L \) and \( l \) being the maximum depth and the current level of the pyramid. The number of classes was \( K = 6 \) convergence of the algorithm was assumed when \( \epsilon = 0.1 \). For visualisation purposes, pixels belonging to the same class were represented with the same color, as shown in Fig. 1(b).
In order to compare MDEM method to the EM and DEM methods, the algorithm has been tested on a sports image Fig. 2(a). We have used the same initialization mode of the first experiment to compute the parameters and the depth; the same set of numbers for the D-step and for the convergence parameter. More important, by comparing the results obtained by standard EM (Fig. 2(b)), DEM method (Fig. 2(c)), and MDEM method (Fig. 2(d)), it is apparent the higher perceptual significance and the reliability of the latter as regards region classification.

The number of iterations in the case of EM and DEM algorithms, was equal to the total iterations of MDEM (sum of iterations carried out at each level). Always referring to Fig. 2, in Fig. 3 log $L$ is plotted versus the number of iterations; it is apparent that likelihood is much higher in the MDEM case which already at the coarser level yields a value of log $L$ larger than those of the other methods. Note that the large jumps denote the change from a coarse scale to a finer one. Note that MDEM can be thought of as a multiscale competition/cooperation scheme on the $k$ label probability planes: the E and M steps implement a competition among the different labels at site $i$, while the D-step can be considered as a cooperation step among sites on the same plane.

IV. CONCLUSION

MDEM is a novel scheme that results in a simple but effective segmentation algorithm for color images that: 1) retains the appealing characteristics of a feature clustering based approach; 2) takes into account spatial constraints while avoiding complex schemes such as MRFs; 3) it operates within a multiresolution framework, in order to reliably define regions of interest and efficiently perform required computations. Simulations show that it performs quite well compare with more standard methods, and it is flexible enough to be used in a wide range of application.

REFERENCES