Image segmentation by multidimensional clustering and boundary refinement with oriented filters

P. Schroeter and J. Bigün

Signal Processing Laboratory, Swiss Federal Institute of Technology
EPFL-Ecublens, CH-1015 Lausanne, Switzerland

Résumé

Dans cet article, on presente un nouvel algorithme de segmentation multi-dimensionelle pour images 2-D. On propose un processus d'estimation de frontière basé sur l'orientation locale de celles-ci. Ce processus, imbriqué dans une structure pyramidal à plusieurs niveaux de résolution, permet l'utilisation d'algorithmes de classifications sans contraintes de connectivité spatiale. La présence de bruit dans l'espace des attributs provoque un recouvrement de classes et peut être réduit par l'utilisation de pyramides. Après classification, la connectivité spatiale est restaurée par une relaxation déterministe et la segmentation finale est obtenue par le processus de rafinage.

Abstract

In this paper we present a new multi-dimensional segmentation algorithm for 2-D images. We propose an orientation-adaptive boundary estimation process, embedded in a multiresolution pyramidal structure, that allows the use of different clustering procedures without spatial connectivity constraints. The presence of noise in the feature space causes a class-overlap which can be reduced in a multiresolution pyramid. After classification, the spatial connectivity is restored by a deterministic relaxation labeling and the final segmentation is obtained by the orientation-adaptive boundary refinement process.

1 Introduction

In many cases, especially when dealing with textures, the segmentation of images is performed using a set of local features. Progress in texture analysis has led to models which allow to describe increasingly complex structures, thereby leading to high dimensional feature spaces. Therefore, we need to develop segmentation algorithms capable of handling such complicated data structures.

In this paper we address unsupervised image segmentation, where "unsupervised" means that no training of the classifier takes place. In that case, both the prototypes (most representative feature-vector of a class) and the segmentation have to be determined. These two processes are highly inter-dependent because well defined prototypes are needed to find accurate boundaries, and accurate boundaries are needed to compute the prototypes (principle of uncertainty [1]). Here, we propose an orientation-adaptive boundary estimation process, embedded in a multiresolution structure, that allows the use of different clustering procedures without spatial connectivity constraints. The class separation at the coarsest resolution level is significantly increased so as to make possible the detection of the different classes and of their associated prototypes. We also reduce the problem of an exact estimation of the class number by overestimating it and reassigning small and scattered classes to their neighborhing classes by means of a spatial restoration process. This process also removes isolated labels produced by unconstrained clustering. The orientation-adaptive boundary refinement process further eliminates spatially misclassified pixels. More Details can be found in [2].

2 Segmentation algorithm

The algorithm is embedded in a multiresolution framework using quadtrees [3, 4]. Multiple resolutions are very useful in reducing the uncertainty because at lower resolutions the class-prototypes are better defined while higher resolutions are needed to obtain accurate borders. The algorithm is explained in the next four sections.

2.1 Pyramid building

The noise reduction can be done by means of quadtrees [4]. Lower resolution levels are obtained by projecting the average of non-overlapping squares of size 2x2 to the next coarsest level. Let $I_p(i,j,l)$ be the value of the p-th feature at location (i,j) and level l of the pyramid. The value of a father node is simply the mean of its four children value

$$I_p(i,j,l) = \frac{1}{4}\sum_{m=0}^{1}\sum_{n=0}^{1} I_p(2i+m, 2j+n, l-1) \quad (1)$$

The different levels are computed using (1) in a bottom-up manner starting from level $l = 1$ up to a predefined level ($l = l_{max}$). The choice of the number of levels is important. If l_{max} is too small the uncertainty is not enough reduced, whereas if l is too high, small regions will disappear. Pyramids also reduce the computational cost by progressively reducing the number of feature vectors on which a clustering algorithm will be applied.

2.2 Clustering

At the coarsest level of the pyramid, a clustering algorithm is used to find the different classes and their prototypes. We assume that objects with similar properties belong to the same cluster. The problem is to find a partition of the feature space into c homogeneous subsets. Different clustering criteria will lead to different results depending on the shape of the clusters. However, our method does not critically depend on the choice of a clustering technique. Experiments (section 3) show that similar segmentation results are obtained with different algorithms.

We mainly used the fuzzy c-means clustering algorithm proposed by Bezdek [5] which gives good clustering results while being simple. It minimizes the family of objective functions given by

$$J_m(U, \mathbf{v}) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m (d_{ik})^2 \quad (2)$$

where $U = [u_{ik}]$ is a fuzzy c-partition of the feature space $X = \{\mathbf{x_1}, ..., \mathbf{x_n}\}$, $u_{ik} \in [0,1]$ is the degree of belongingness of vector k to class i, $\mathbf{v} = \{\mathbf{v_1}, ..., \mathbf{v_c}\}$ is the set of cluster centers, $(d_{ik})^2 = \|\mathbf{x_k} - \mathbf{v_i}\|^2$ and $\|\cdot\|$ is any inner product induced norm on $\mathbb{R}^p$. m is the weighting exponent controlling the amount of fuzziness, $m \in [1, \infty[$, n is the number of feature-vectors and c is the number of classes. For comparison we will use the general agglomerative algorithm proposed by Lance and Williams (see [6] for a description) with three clustering creteria: 1) sum of squares, 2) centroid and 3) complete link.

2.3 Restoration of the spatial connectivity

At the coarsest resolution level l_{max}, we can define a label image $I_\gamma(i,j,l_{max})$, result of the classification. Some pixels are spatially misclassified because no spatial connectivity constraints were present in the clustering algorithm (see section 3). Let $N_8(i,j)$ be the neighborhood of a pixel at location (i,j) (see figure 1). A pixel (i,j) is considered as spatially misclassified if $I_\gamma(i,j,l_{max})$ is different from all the labels in $N_8(i,j)$. In that case, it is reassigned to the most represented class in $N_8(i,j)$.

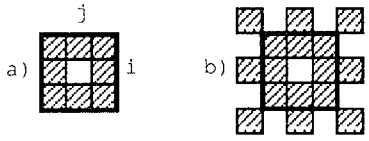

Figure 1: a) $N_8(i,j)$ and b) extended neighborhood $N_e(i,j)$

The next step is to reassign small and scattered classes to their spatial neighborhood. A class in the feature space is distributed in one or more subregions in $I_\gamma(i,j,l_{max})$ and is considered as "significant" only if its largest subregion contains a sufficient number of connected pixels (9 in our experiments).

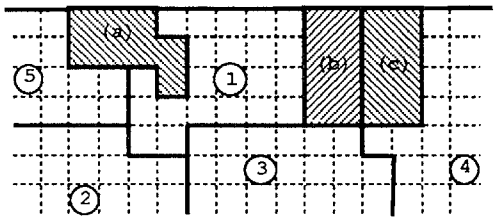

Figure 2: Example of 3 classes that have to be reassigned (hatched regions).

Consider, for instance, the hatched regions in Figure 2. These classes can be reassigned using a simple deterministic multi-pass relaxation. First, we reassign the isolated classes like C_a, represented by the n_a pixels of region (a). A set of candidate classes, among the neighboring classes of C_a, is determined for each pixel $(i,j) \in C_a$. A new candidate class is obtained for each label $I_\gamma \in N_e(i,j)$ (see figure 1) different from $I_\gamma(i,j,l_{max})$. Each pixel $(i,j) \in C_a$ is then reassigned to its closest candidate class.

Finally, the non-isolated classes are reassigned. However, the reassignment order leads to different results and multi-passes are needed to avoid this problem. This process can be illustrated by considering, for instance, the classes (b) and (c) in figure 2. In the first pass, both classes are temporarily reassigned as if they were isolated classes. The same initial state of $I_\gamma(i,j,l_{max})$ is used for both classes. The maximum number of pixels $n_{i,max}$ reassigned to the same candidate class is determined for i =(b),(c). Then a ratio r_i between $n_{i,max}$ and the respective population n_i of the classes is computed, i.e. $r_i = n_{i,max}/n_i$, i =(b),(c). A value of $r_i = 1$ means that the class i tends to be reassigned to a single candidate class. The reassignment of the region with the largest ratio r_i is validated. Then, we check again for "insignificant classes" and repeat the preceeding operations until all such classes are reassigned.

2.4 Orientation-adaptive boundary refinement

The last step of the algorithm is a boundary refinement procedure that gradually improves the spatial resolution of the label image $I_\gamma(i,j,l_{max})$.

First, at the coarsest level l_{max}, the boundary pixels are determined. Each pixel (i,j) is considered as a boundary pixel if at least one label in $N_8(i,j)$ is different from $I_\gamma(i,j,l_{max})$. The children of the boundary nodes define a boundary region at the next higher resolution. The non-boundary nodes at the children level are given the same labels and properties as their parents.

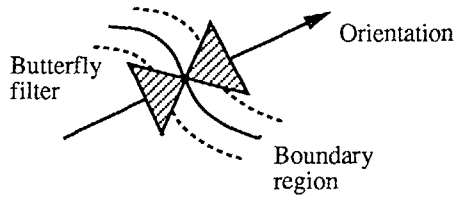

Figure 3: Illustration of the oriented butterfly filters.

As stated by the spatial principle of uncertainty [1], accurate prototypes can be obtained only at the expense of the spatial resolution. Therefore, at coarser resolutions, the uncertainty on both sides of the boundaries is high. We propose to reduce it by means of orientation-adaptive filters. The principle is illustrated in figure 3. The dominant local orientations of the boundaries can be simply obtained by the linear symmetry algorithm [7] which defines the orientation in the least squares sense. For each dominant local orientation, a butterfly-like filter is defined. Butterfly-like shape reduces the influence of feature-vectors along the boundary. The coefficients of the filters are function of a dissimilarity measure $d = |\mu_1 - \mu_2|/\sqrt{\sigma_1^2 + \sigma_2^2}$, where μ and σ^2 are the mean and the variance of the two classes on both sides of the boundary. A stronger smoothing is performed for small values of d. For each boundary pixel, the filter corresponding to the local orientation is applied. The two halves of the filter (applied separately) produce two responses. The distances between these two vectors and the two prototypes, associated to the classes defining the boundary, are computed. Then, each boundary vector is reassigned to the closest class. All these operations are applied in a top-down manner, level by level, until the bottom of the pyramid is reached.

3 Experimental results

In our experiments, we used two different "patch" images of size 256x256. The first image, p1, is composed of 7 natural textures taken from aerial images where each texture was normalized with respect to the mean and variance. The second image, p2, is composed of textures that can be found in Brodatz album.

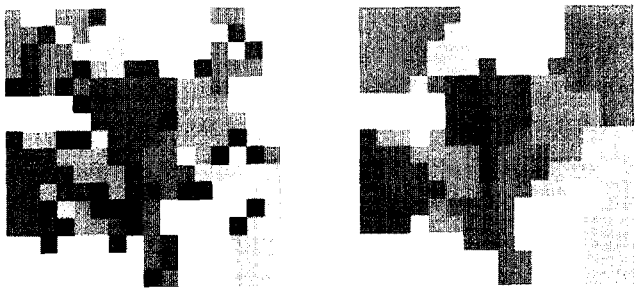

Figure 4: a) Results of the fuzzy c-means clustering algorithm applied at the coarsest resolution level $l = 5$. b) Results of the spatial restoration

The complex moments of the Gabor power spectrum were computed on these two images [8] resulgin in 6 features. The noise in this space is mainly due to modeling errors and is progressively reduced in the pyramid. Experiments showed that a number of levels of 5 is the optimum compromise between the noise reduction and the size of the smallest detectable region. We used the fuzzy c-means algorithm with $c = 10$ and $m = 1.6$. The result of the clustering operation applied on "p2" is shown in Figure 4 a) where each class is represented by a different gray level. For convenience, a pixel

at the coarsest level is represented by a square of size 16x16. The result of the spatial restoration is shown in Figure 4 b). Isolated pixels are removed and insignificant classes are reassigned, reducing the number of classes from 10 to 7. The boundary refinement procedure is then performed and the final segmentation can be seen in Figure 5. All classes are found and the borders correspond well to those that can be seen by a human observer.

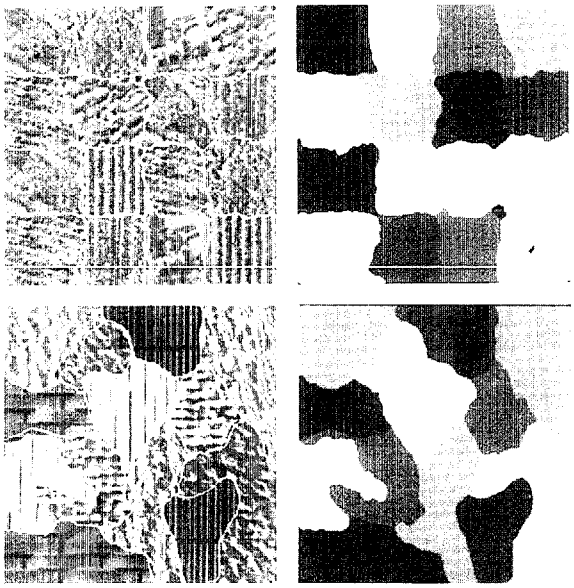

Figure 5: Segmentation results of "p1" and "p2" superimposed on the original image, and the corresponding class assignments.

Similar segmentation results can be obtained with a large variety of clustering algorithms as long as the clustering criterion does not excessively missmatch the shape of the clusters in the feature space. The boundaries are very similar when using the fuzzy c-means or the general agglomerative algorithm of Lance and Williams. However, the detected number of classes changes. With the fuzzy c-means and centroid algorithms all 7 classes are found, while one class (resp. two) is missed by the sum of squares criterion (resp. complete link). We also tested our algorithm on 1-D features, being the gray levels of the original images. Very good segmentation results were obtained in that case too.

4 Conclusions

Detecting image boundaries often requires multidimensional segmentation algorithms. In this paper we have presented a new algorithm that gives good segmentation results on various images such as texured or grey level images. It accepts different kinds of features and is robust with respect to noise. A spatial restoration process combined with an orientation-adaptive boundary refinement allow the use of different clustering algorithm without spatial connectivity constraints.

References

[1] R. Wilson and G.H. Granlund. The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:758–767, 1984.

[2] P. Schroeter and J. Bigün. Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement. *International Journal of Computer Vision*, 1993. Submitted.

[3] R. Wilson and M. Spann. *Image Segmentation and Uncertainty*. Research Studies Press Ltd., Letchworth, England, 1988.

[4] W.I. Grosky and R. Jain. Optimal quadtrees for image segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:77–83, 1983.

[5] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithm*. Plenum Press - New York and London, 1981.

[6] A.D. Gordon. *Classification*. Chapman and Hall, New York and London, 1981.

[7] J. Bigün, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:775–790, 1991.

[8] J. Bigün and J.M.H. Du Buf. N-folded symmetries by complex moments in gabor space and their application to unsupervised texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993. Accepted.