

Improved Textured Images Segmentation Using an Energy Functional

Antoni Grau and Jordi Saludes*

Dept. of Automatic Control and Computer Engineering

* Dept. of Applied Mathematics II

Polytechnic University of Catalonia UPC, Barcelona, Catalonia

Abstract. In this paper we present a new classification and image segmentation system based on the addition of a variational method to a classic clustering algorithm. This system constitutes an improvement respect traditional segmentation methods. Often due to the nature of the texture features obtained from an image, the segmentation results are not quite precise. If this happens, using the energy functional and its minimization can improve the segmentation. This functional takes into account the information in the feature space and the information in the 2D image domain. The extracted characteristics from the image are texture features that have been defined in order to obtain an admissible trade-off between their discriminant capacity and their effectiveness to be implemented in a vision board to operate at real time. We show some results to appreciate this improvement in the segmentation using the energy functional.

1 Introduction

Texture perception is an important part of the human vision. The objects can often be discriminated by their texture features despite colors and similar shapes. Despite its ubiquity in scene analysis a precise definition of texture does not exist, but texture is the term used to characterize the surface of a given phenomenon in an image and it is undoubtedly one of the principal features used in image processing and pattern recognition, but in a wide sense, there is no generally accepted definition of texture. In computer vision, all treatments of texture have taken one of the following two approaches [1]. The statistical approach attempts a global characterization of texture. Statistical properties of the spatial distribution of gray levels are used as texture descriptors. The structural approach conceives of texture as an arrangement of a set of spatial subpatterns according to certain placement rules. The first step in objects recognition is at present the segmentation, because normally, each region corresponds to an individual object or each contour corresponds to the border between different objects [2]. For a correct texture discrimination it is necessary to find the differences between them.

In this work we present an image segmentation system which uses texture features and works at high speed, that is, video rate. Due to this reason, the computed texture features are not very sophisticated. We search an admissible trade-off between their effectiveness in terms of quality and their low computational cost. If the features do not have an excessive quality, it will be a hard task for the classifier to segment the parts of the image with certain uniformity: it exists a deficiency in the feature space

that the classifier can not solve. It is just in this moment, after classification, when the concept of the energy functional appears: why not take advantage of the information in the feature space together with the information in the 2D image domain?. With this sum of information, the misclassified objects will be reclustered and, at the end of this process, we will find a set of clusters that can be considered correct in terms of visual perception of the image textures. When the final clusters have been obtained from some training images, the system changes of stage and is already prepared to work and classify new objects in an on-line process at high speed. The cluster assignment for every new object entering in the system is achieved in real time and the clusters information can be updated with these news objects. Obviously, a bottle neck in the system is the feature extraction system and we are preparing a specific architecture to implement these texture features. In [3], these features are widely explained and detailed, but we will define them in Section 3. In Section 4 we will see how to classify these texture features in order to create perceptually different texture classes.

2 System overview

For image segmentation using the texture information we propose a system that extracts texture features from the image and groups all the texture elements having something in common. There will be three different and separate steps to achieve the segmentation of the original image. In figure 1 the segmentation system is shown.



Fig. 1. Diagram of the image segmentation system using texture features.

In low-level processing the thinned edges of the image are obtained and the image is prepared for the feature extraction module. In this second module a feature space is generated. Each new point (represented as a vector) in this new space is a texel (texture element) which has texture parameters as components. In the classification step, the texels are grouped in order to obtain texture classes with similar behavior. This classification will be improved with the use of a energy functional. Then, the segmentation output is represented as the borders between texture groups.

3 Feature extraction and feature space

Being perceptible is the principal characteristic of the texture parameters and this perceptible nature of the parameters allow to compute their value with masks created in a perceptible manner too. These masks are local boolean expressions which are applied over each pixel in the image. Due to the masks, it will appear some constraints when the system will try to discriminate the textures, but in the other hand, there is a predisposition for a hardware implementation of the masks to find the texture parameters. The definitions of the parameters are as follows:

1. **Straightness.** This parameter indicates the straight line density over a region, and it has been derived from the linear regression model.
2. **Blurriness.** The blurriness is a visual effect where a progressive and slow gray level increasing or decreasing along an image area is noticed. We only consider blur pixels if the change in their intensity is inside a fixed interval, otherwise we consider sudden changes.
3. **Abruptness.** This parameter indicates sudden changes in the directionality of the texture.
4. **Granularity** has a high perceptual significance in textures. This value will indicate how many elements in the image are isolated or non-concatenated.
5. **Discontinuity.** This parameter measures the density of cut edges in the image. When the edges are continuous the value will be high.

Each texel determines a point in the 5-dimension texture space. The images used in this work contain 32-by-32 texels each and this will be the number of points in the feature space. The parameters can be found in a parallel manner because it does not exist any data dependence among them. Then, for an input image, where the minimum element is the pixel, each parameter is calculated in parallel with its masks to produce an output image where the minimum element is the texel (see figure 2). The size of the masks used to calculate the parameters is 4-by-4 elements. A bigger size will generate a large amount of masks for each parameters and the cost in time will be excessive for our intentions. A smaller size of the masks is contradictory with the own definition of the parameters.

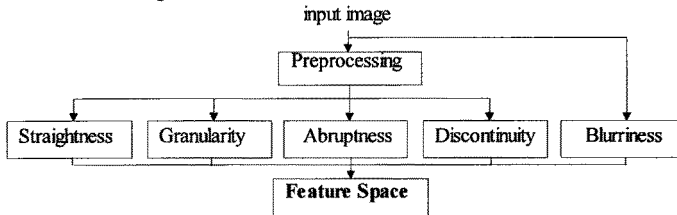


Fig. 2. Feature extraction and generation of the feature space.

The masks are calculated through the image following the raster scan. Therefore, every pixel in the image will be analyzed. These masks are template matching masks.

4 Classification and segmentation

This is the third and last step in the image segmentation system. There are two major techniques in grouping: discriminant analysis and cluster analysis. All varieties of discriminant analysis require prior knowledge of the classes, usually in the form of a sample from each class. In our case, the data do not include information on class membership and moreover we do not know how many classes we are looking for. For this reason, the technique used in this work is cluster analysis and, among the available methods, the algorithm uses a hierarchical technique with an agglomerative method explained below.

A cluster is defined as a collection of points which are close between them [5]. The various clustering methods differ in how the distance between two clusters is computed. There will be two steps in cluster processing: the learning phase and the working phase. In the learning phase, after cluster analysis, the supervisor chooses the number of clusters cutting at specified level of the hierarchical tree. Once the clusters are defined, in the working phase for each new object in the system the centroid distance is computed and the object is assigned to the closest cluster. The learning phase takes a lot of time to compute all the clusters but, luckily, it is an off-line task. The working phase is critical in computing cost, but the system have to compute only several distance measures (as distances as clusters).

In the figure 3 the recognition strategy can be seen. The different phases in recognition are separated. First, from the learning pattern samples the clusters are generated and, in the same phase, by the energy functional a decision rule is derived. The energy functional performs a reclustering of misclassified samples and, in this submodule, the correct clusters are created. In the working phase, the new pattern samples are classified using the clusters generated by the energy functional module. This is the on-line phase.

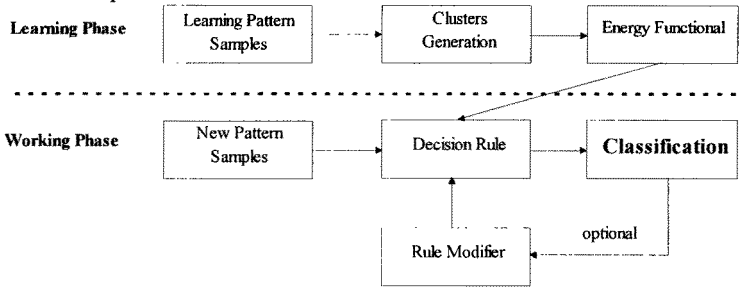


Fig. 3. Diagram of the recognition strategy. Learning phase and working phase.

4.1 Clustering Algorithm

In [6], Ward proposed a method of forming clusters that is based on the loss of information resulting from the grouping of objects into clusters as measured by the total sum of squared deviations of every observation from the mean of the cluster to which it belongs. The assignment rule rests on the increase in the error sum of squares induced from combining every possible pair of clusters. This value, which will be denoted by E.S.S., is used as an objective function. The algorithm also employs a hierarchical grouping procedure. The E.S.S. is computed as

$$E.S.S = \sum_{j=1}^k \left(\sum_{i=1}^{C_j} x_{ij}^2 - \frac{1}{C_j} \left(\sum_{i=1}^{C_j} x_{ij} \right)^2 \right) \quad (4.1)$$

where x_{ij} denotes the point value in the feature space (n-dimensional) for the object i th in the j th cluster, k is the total number of clusters at each stage, and C_j is the number of objects in the j th cluster. The E.S.S. is zero at the first stage, since each object constitutes a cluster. For example, we consider the objects shown in figure 4a). After considering all possible clusters of size 2 and combine those objects that yield the smallest E.S.S., we fuse objects E and F. The next stage entails a computation of E.S.S. induced from (1) adding each of the remaining objects to the first cluster and (2) forming all possible pairs of the unclustered objects. Finally, the resulting dendrogram is shown in the figure 4b). After the examination of the dendrogram, if we choose N3 as the level to cut the tree, the number of total clusters will be 3, as it can be seen in the figure 4c). For simplicity in the figures, we consider a 2-dimensional feature space.

4.2. Energy Functional

The segmentation problem is the problem of subdividing an image into regions so that in each region, the image properties are relatively uniform. When clustering algorithms have been applied to group points in a n texture feature space, it is not sure that we have arrived at the end of the segmentation process with success and correct results. Therefore, in order to improve the segmentation we will study the problem with a variational approach [4]. This approach is motivated in part by occasional failures of traditional methods, which are based exclusively on clustering objects formed by features extracted from images without taking into account the image domain. The general idea is to define an energy functional over a set of possible segmentations in terms of penalty measures that correspond to various desired properties of a good segmentation. In particular, the functional we have studied is the following. Consider C a segmentation, that is, a partition $C = \{C_1, \dots, C_M\}$ of the set of texel indexes.

$$J(C) = \sum_{k=1}^M \sum_{j \in C_k} \|\vec{x}_j - \bar{x}_k\|^2 + \lambda |F| \quad (4.2)$$

where \bar{x}_k is the center of mass of the texels in C_k , \vec{x}_j is the j th object of the k th cluster, M is the number of clusters, λ is a weight, F is the union of the borders between texture regions in the image and $|F|$ is the length of these borders. The problem is to find a segmentation C that minimize the functional $J(C)$. The functional works in two different spaces: R is the image domain and H is the n-dimensional domain of texture features. While the first term of the functional tends to get the maximum number of clusters in the H space, the second term tends to get to minimize the borders between clusters, trying to avoid segmenting the image into too many regions. Thus, the formulation is designed to find a simple segmentation such that in every region the deviation of the objects to the center is minimum. Then, the functional penalties are 1) the minimum number of clusters and 2) the maximum

length of their borders. Resuming, this method combines the classification in the H space together with the actual placement of the texels over the image domain R .

The following algorithm finds the pair that minimizes the functional. These are the steps in the algorithm:

1. Choose randomly a texel t_{xy} in the image domain.
2. Seek for any neighbor of texel t_{xy} over the R space that belongs to a different cluster in the H space. If the whole neighbors texels belong to the same clusters of texel t_{xy} go to step 1.
3. Join temporally texel t_{xy} to the clusters of the chosen neighbor.
4. Compute the functional $J(C)$.
5. If the functional value has decreased, then join definitively texel t_{xy} to the cluster of its neighbor. Update \bar{x}_k and go to step 1.
6. If after n iterations the functional value has not decreased we are over a local minima and the algorithm finishes.

Theoretically and practically there are not algorithms that find global minima to this problem due to the nonlinearity and also to the existence of many local minima.

In figures 4c) and 4d) we can see an example of migration of objects to different clusters after minimization of the functional. An object can be moved from a cluster to a different one decreasing the number of borders between regions even if the deviation of this cluster increases.

4.3 Working Phase

In this second phase of grouping, it is required that the clusters be clearly defined. Thus, for each new object entering in the system it will be assigned to a former cluster created in the learning phase with the use of the energy functional. Graphically, this effect is shown in the figure 4e). A new object \mathbf{X} can be placed in any point in the feature space and it is assigned to the closest cluster. The creation of new clusters with new objects is not allowed in this working phase. As assignment rule the minimum distance to the centroids of clusters is chosen.

$$\text{Assignment rule} = \{ \mathbf{X} \in i \mid \min\{d(\mathbf{X}, \bar{x}_i)\} \quad (4.3)$$

where i is the cluster to which the object is assigned and \bar{x}_i is the center of mass of the cluster i . For each new object, we compute the distances between the object and the centroids of clusters. The object is assigned to the closest cluster. This process has a low computational cost and it allows its implementability in an on-line system.

In any on-line process the incoming images can vary gradually with time. Therefore, it exists the possibility of updating the centers of mass of the clusters concurrently with the classification of new objects. Each new object in the system is assigned to a

cluster by the rule (4.3), and the center of mass of this cluster is computed again. To update the center of mass of a cluster with a new object can be done at real time.

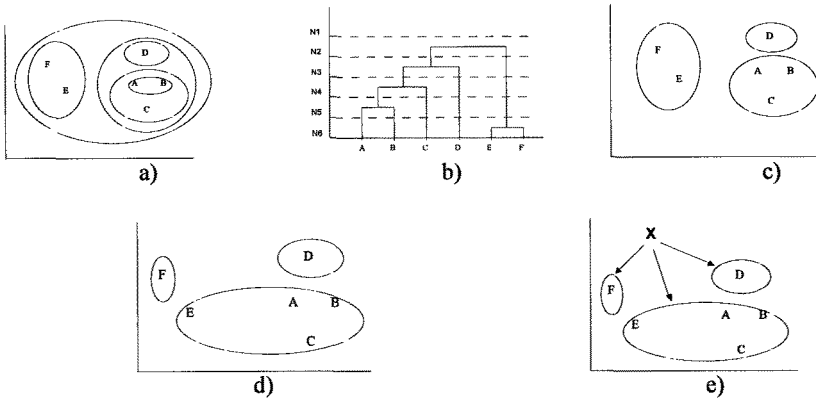


Fig. 4. a) Objects in feature space; b) dendrogram; c) clustering process; d) final clustering after energy functional minimization; e) cluster assignment in the working phase.

5 Results and concluding remarks

To show the performance of our system we have chosen as input images some aerial images obtained from a plane.

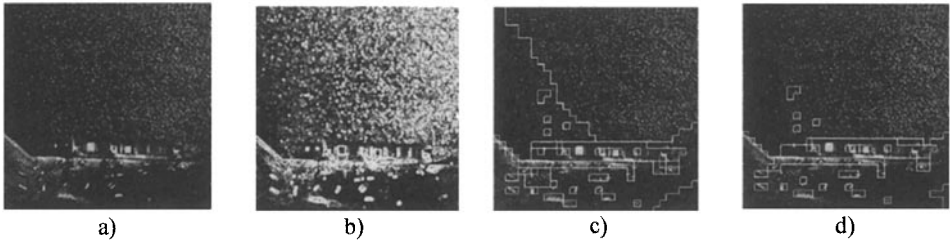


Fig. 5. Learning phase. a) Input; b) Gradient image; c) Clustering; d) Output segmented image.

In the learning and in the working phase, images pass through three different processes: 1) preprocessing: in this stage the gradients of the original image are extracted, then they are binarized and thinned; after this process the image enters in 2) the texture features extractor where five texture parameters are computed generating 32×32 texels per image. The combination of these five parameters forms the feature space with 1024 points with five components each. The next stage is 3) the clustering process: over this 5-dimensional space the points are grouped to form clusters with similar objects. The use of the energy functional will take part only in the learning phase after the classification. All this process can be seen in figure 5 where appears a harbor, the sea, a ship and other objects: from the input image a) the gradients are extracted in b). In order to evaluate the effectiveness of the use of the energy

functional versus the traditional clustering methods is interesting to look at the next table (table 1), made up from data in figure 5.c) and 5.d).

		Actual Cluster					Total	Actual Cluster					Total
		1	2	3	4	1		2	3	4			
Assigned Cluster	1	489 71%	10	6	5	510	1	675 98%	0	0	4	679	
	2	194	176 76%	4	4	378	2	9	217 94%	7	4	237	
	3	6	20	31 72%	0	57	3	5	7	37 86%	0	49	
	4	0	24	2	53 85%	79	4	0	6	0	53 85%	59	
	Total	689	230	43	62	1024	Total	689	230	43	62	1024	

Table 1. Clustering results before (left table) and after (right table) energy functional use.

When the texture features have been extracted and hierarchically classified, we cut the tree at a certain level and four clusters have been generated. The cluster #1 corresponds to a image area where it appears the sea. In the cluster #2 are grouped the objects corresponding to the land (in the harbor) and also these elements that will form clusters with a few objects to be considered relevant. In the cluster #3 are grouped those objects with a small area such as trucks, containers in the harbor and also those small components in the ship, while in the cluster #4 are grouped those objects corresponding to the ship.

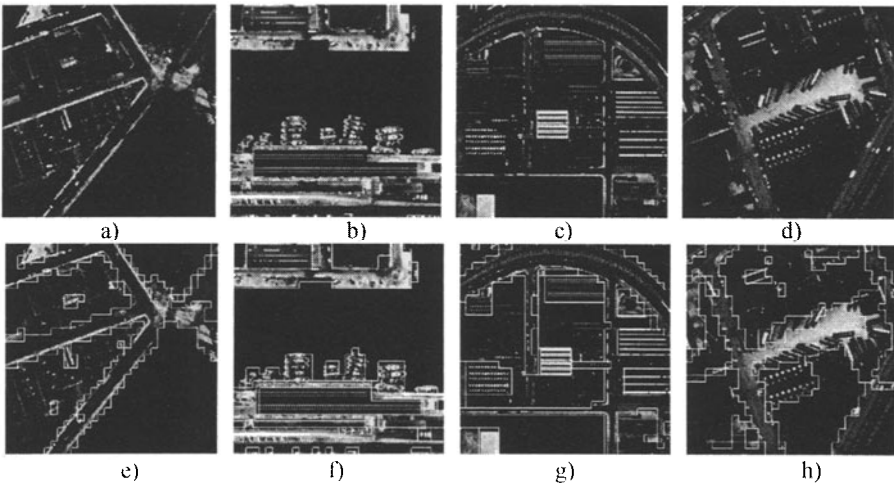


Fig. 6. Results in the working phase. a), b), c) and d) Input images; e), f), g) and h) Output segmented image.

With this information we made two tables: the first table is made after classification but before the use of the energy functional; the second table is made after use it. The

table shows the number of objects that appear in the clusters. The columns indicate the objects appearing in the actual clusters while the rows indicate the number of objects appearing in each cluster. The optimal solution is when all the objects appear in the diagonal of the tables, that is, when the assigned objects in the clusters correspond with the actual clusters and the classification is perfect. This is not the case and we indicate the % of correct classification in each cluster. As it can be seen in the tables, the number of objects classified correctly augment in the right table and, specially, in the cluster #1 (corresponding to the sea); practically, the objects that correspond actually to the sea but considered as objects in the land (194 objects in the left table) are moved to their correct clusters after the reclassification with the energy functional. With the cluster #2 there is a similar effect, some objects corresponding actually to land are grouped as ship elements or even as small elements, then many of these elements are moved to its correct cluster. After an evaluation of the system, we pass to the working phase using the data obtained in the learning phase. This can be seen in the figure 6 where some input images are segmented with their texture information. As it is appreciated in the output images, the segmentation is not perfect due to different aspects: first, the input to the image segmentation system is not the gray level image but their edges. That produces an information loss and a degradation in the image. In second place, the parameters computation is not completely accurate because there is a time limitation if we want fast speed, near real time (video rate in this case). Actually, there is a trade-off between the quality of segmentation and the computation time. Our system prioritizes the low processing time in front of the high quality in the segmentation not always required in many applications.

6 References

- [1] Lipkin, B.C. and Rosenfeld, A., *Picture Processing and Psychopictorics*, Academic Press, New York, 1970.
- [2] Tomita, F. and Tsuji, S., *Computer Analysis of Visual Textures*, Kluwer Academic Publishers, 1990.
- [3] Casals, A., Amat, J. and Grau, A., "Texture Parametrization Method for Image Segmentation", *Proc. 2nd European Conf. on Computer Vision ECCV'92*, pp. 160-164, 1992.
- [4] Mumford, D. and Shah, J., "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems", *Comm. Pure Appl. Math.*, Vol. 42, pp. 577-685, 1989.
- [5] Young, T.Y. and Calvert, T.W., *Classification, Estimation and Pattern Recognition*, Elsevier Publishing Company, Inc., 1974.
- [6] Ward, J., "Hierarchical Grouping to Optimize an Objective Function", *Journal of the American Statistical Association*, Vol. 58, pp. 236-244, 1963.