

Improving Clustering Algorithms for Image Segmentation using Contour and Region Information

Arnau Oliver, Xavier Muñoz, Joan Batlle, Lluís Pacheco, and Jordi Freixenet
Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona, Girona, Spain
Email: {aoliver, xmunoz, jbatlle, lluispa, jordif}@eia.udg.es

Abstract—In image segmentation, clustering algorithms are very popular because they are intuitive and, some of them, easy to implement. For instance, the k-Means is one of the most used in the literature, and many authors successfully compare their new proposal with the results achieved by the k-Means. However, it is well known that clustering image segmentation has many problems. For instance, the number of regions of the image has to be known a priori, as well as different initial seed placement (initial clusters) could produce different segmentation results. Most of these algorithms could be slightly improved by considering the coordinates of the image as features in the clustering process (to take spatial region information into account). In this paper we propose a significant improvement of clustering algorithms for image segmentation. The method is qualitatively and quantitatively evaluated over a set of synthetic and real images, and compared with classical clustering approaches. Results demonstrate the validity of this new approach.

I. INTRODUCTION

Clustering methods are one of the most used algorithms in image segmentation, specially used to compare new proposals with them. In a recent review by Jain et al. [1], clustering methods are divided into hierarchical and partitional algorithms, where the main difference between them is that hierarchical methods produce a nested series of partitions (usually, these partitions are schematized using a dendrogram, representing the nested grouping of patterns and similarity levels at which groupings change), while partitional methods produce only one partition. Although hierarchical methods can be more accurate, partitional methods are used in applications involving large data sets, like the ones related with images, due to the fact the construction of a dendrogram is computationally prohibitive. Actually, hierarchical algorithms have time complexity of $O(n^3d)$ (n is the number of pixels of the image and d the number of features), while partitional algorithms usually have time complexity $O(nd)$ or $O(n^2d)$ (see the work of Jain et al. [1] for a review on these methods). However, partitional algorithms have two inherent disadvantages: 1) the algorithm

This work was partially supported by the Spanish Ministry of Education and Science under the grant TIN2005-08792-C03-01.

has to know, a priori, the number of regions which compose the image, and 2) the fact that the final segmentation could be a local solution. The latter is due to the fact that different initial seed placement could produce different segmentation results of the image. Moreover, many image segmentation clustering algorithms present another common drawback, as they disregard spatial information, which is one of the main sources of information for human segmentation [2].

As a partitional clustering algorithm usually produces clusters by optimizing a criterion function, the combinatorial search over the set of possible results would solve the inherent problems that it presents. However, as this solution is computationally prohibitive, the algorithm is run multiple times with different starting states producing a set of results [1]. This set is subsequently evaluated by measuring the distances among the different clusters of pixels in the feature space in order to choose the best final configuration. In the new proposal of Bezdek and Pal [3], three of these methods are reviewed and improved.

The main problem for image segmentation purposes of most partitional clustering algorithms is that they do not use spatial information [4], in contrast with region based methods that use themselves such information (in fact spatial information is the basis of these methods). Otherwise, clustering algorithms assume that pixels belonging to the same cluster have a similar behaviour, which is explained by uniform features like grey levels, texture or colour. Based on the complementary nature of clustering and region methods, our approach wagers for the integration of both strategies in order to solve the problems that both methods bear when are used separately. In literature there are different proposals for this integration. For instance, one of the first works integrating both strategies is the work of Pappas [5], in which spatial constraints are included by the use of a Gibbs Random Field, which looks for local intensity variations. On the other hand, the common solution consists on including the pixels coordinates as artificial features in the clustering process [1].

Following the ideas presented by Freixenet et al. [6], in this paper we propose an unsupervised segmentation method which

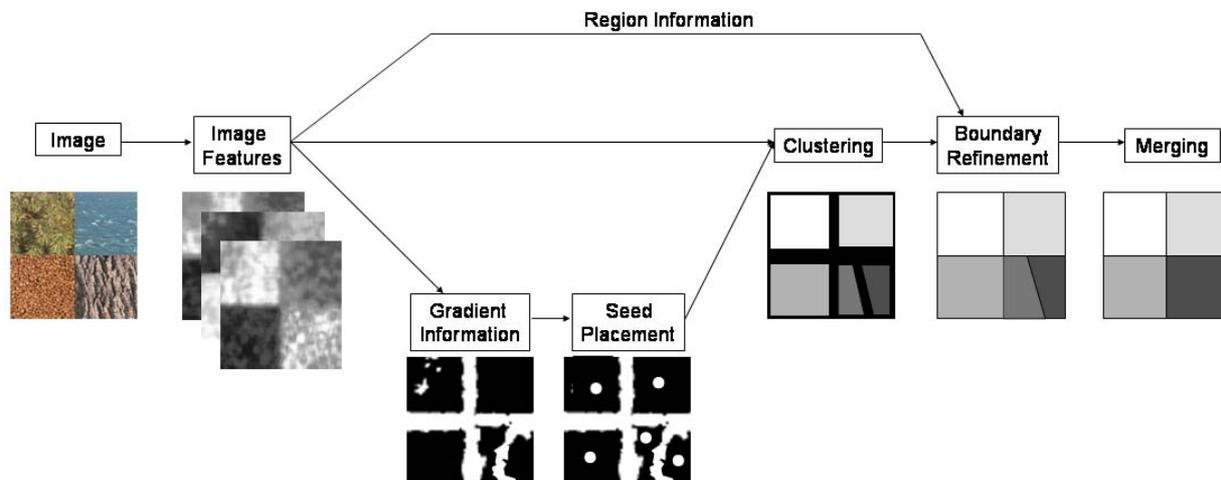


Fig. 1. Scheme of our fully unsupervised segmentation algorithm.

takes advantages of the integration of different strategies. The method begins by detecting the main contours of the image in order to place a number of seeds in the appropriated positions. The parameter that controls the number of seeds is always set up with the aim that it will be larger than the real number of regions. Following, “non boundary pixels” are classified using typical clustering algorithms obtaining the core regions. Pixels placed near boundaries are classified according to region and boundary information. Finally, the method ends with a merging step in order to refine the initial over-segmented result. The remainder of this paper is structured as follows: Section II describes the proposed segmentation technique, detailing the method used to detect the starting seeds, as well as the strategy to classify boundary-pixels. Experimental results proving the validity of our proposal are shown in Section III. Finally, conclusions are given in Section IV.

II. THE SEGMENTATION METHOD

Our fully unsupervised approach, roughly depicted in Figure 1, begins with an automatic seeds placement using contour information. This information is used to determine an initial number of clusters in the image, larger than the final number of regions. Next, a clustering algorithm classifies the pixels into regions. In this stage, we introduce two main restrictions: 1) do not classify the pixels with a high gradient; that is, pixels that belong to a boundary region and 2) do not classify pixels far away from the center of all clusters (noise pixels). As result, when the clustering finishes, the core of the regions are segmented, while there is a number of pixels not yet classified. The segmentation of these pixels is subsequently performed by using the information of the core of the regions, as well as boundary information. The last step consists on merging the regions previously obtained. This over-segmentation scheme is

used in order to ensure the placement of seeds into the main regions of the image.

A. Initialization

The placement of the initial seed points can be stated as a central issue on the obtained segmentation results of a clustering algorithm. Despite their importance, traditional algorithms chooses them randomly. In order to make a more reasonable decision, edge information can be used to decide which is the most correct position to place the seeds. To obtain a sample of each region large enough to model its homogeneity behaviour, initial seeds have to be placed completely inside the regions. Boundary information allows us to extract these positions in the core of the regions by looking for places far away from the contours.

The main idea is to calculate how much the contours of the image affect to each pixel of the image. Obviously, contours near one pixel would affect it more than contours far away from it. Therefore, the following equation is defined in order to calculate this effect:

$$G_e(p) = \sum_{i \in I} \frac{g(i)}{1 + d(i, p)} \quad (1)$$

where I is the whole image, $g(i)$ is the value of the gradient of one pixel (we use Sobel filters for calculating it) and $d(i, p)$ is the Euclidean distance between pixels i and p . The main drawback of this approach is the high computational cost, making it inviable for larger images. However, we can compute this information locally, considering only contours near each pixel. Therefore, Equation 1 can be re-written as

$$G_e(p) = \sum_{i \in E(p)} \frac{g(i)}{1 + d(i, p)} \quad (2)$$



Fig. 2. Seed placement in the image of Lenna.

where $E(p)$ is the set of pixels near the pixel p . This manner, pixels with low G_e would be pixels far away of all contours (pixels situated in the core of the regions) while pixels with high G_e will be those near boundaries. Thus, that pixel with lowest value of G_e of the region will be selected as a seed. We define a region as the set of pixels connected among them which appear when binarizing G_e at different thresholds, with the restriction that the area of the set of pixels must be greater than the original image divided by 100. Figure 2 shows the seed placement in the image of Lenna obtained using the described strategy.

When segmenting real images sometimes is necessary to work with texture features. The problem of texture edge detection is considered as a classical edge detection scheme in the multidimensional set of k texture features which are used to represent the region characteristics. Thus, boundaries of homogeneously textured regions are defined to be located where sudden changes in local texture characteristics occur. Nevertheless, as is well known, texture is an inherently non-local image property. All common texture descriptors, therefore, have a significant spatial support which renders classical edge detection schemes inadequate for the detection of texture boundaries. Hence, the result of this simple method is inaccurate and thick contours are obtained. However, this information is enough to perform the seed placement according to the algorithm proposed above, in which seeds are placed in zones free of contours or, in other words, the “core” of the regions.

Nevertheless, before these seeds are used as initial central points of a clustering algorithm, a seed merging must be considered. If the image has a cluster which is split into two (or several) spatially not connected regions, the seed placement algorithm will produce two different seeds. In order to merge them, all seeds are compared using the Euclidean distance on texture features space. If two seeds are separated by a short distance they are merged.

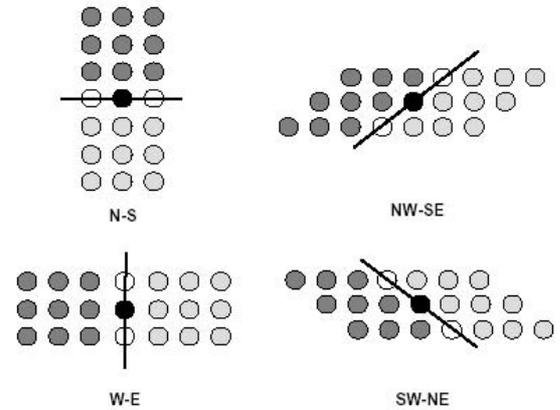


Fig. 3. Four partitions are considered in the classification of boundary pixels.

B. Clustering with boundary restriction

The clustering process is performed by using a classical algorithm, such as k-Means, but including one important restriction: pixels with a high gradient are not classified. As is above noted, gradient of pixels is obtained over the set of texture features, and not over the original image as is usual, in order to avoid false contours due to microtexture. The reason of this restriction is that, in images rich on texture, the clustering will not be effective in boundary pixels. This is due to the fact that texture features are extracted using the local neighbouring information, and in these positions the texture value will be a mixture of two or more regions. Hence, in order to avoid a bad classification of these pixels, the algorithm simply does not classify them.

When initial clustering finishes, the algorithm has to classify the unsegmented pixels. Nevertheless, it is well known that the extraction of boundary information for textured images is a very tough task. Moreover, a texture model of the adjacent textures is required to enable precise localization. Current segmentation of non-boundary pixels allows to obtain this required knowledge, and the core of regions are used to model each region by a multivariate Gaussian distribution considering the set of texture features.

Following the work of Paragios and Deriche [7], we shall consider that a pixel constitutes a boundary between two adjacent regions, A and B , when the textural properties at both sides of the pixel are different and fit with the models of both regions. Therefore, the probability of a pixel to be boundary between regions A and B is maximum when the pixel is exactly the edge between textures A and B because textures at both sides fit better with both models. As is shown in Figure 3, four possible neighbourhood partitions (vertical, horizontal and two diagonals) are considered. The algorithm looks for pixels with maximum probability to be boundary in order to determine the exact place of the edge.

C. Merging the regions

When the classification of all the pixels in the image has finished, the result is the image segmented in as many regions as the initialization stage produced. As we have above noted, the algorithm produces more regions than the human vision would appreciate. Therefore, the optimum number of regions has to be calculated. In this sense, we first construct a dendrogram with its leafs corresponding to each region. As regions are modelled using a gaussian approach, the links between the different leafs (clusters) are calculated using the Bhattacharyya distance. We use this averaging distance because, as Jain et al. [1] states, the single linkage tends to produce “long” and “stringy” clusters, while complete linkage tends to produce “small” and “compact” clusters. An average distance produces intermediate clusters. Bhattacharyya distance between two clusters is defined as:

$$d(C_1, C_2) = \frac{1}{8}(m_1 - m_2)^T(S_1 + S_2)^{-1}(m_1 - m_2) + \frac{1}{2} \ln\left(\frac{\frac{1}{2}|S_1 + S_2|}{\sqrt{|S_1||S_2|}}\right) \quad (3)$$

In this equation, C_i represents each class, m_i the mean of the i class, and S_i its covariance. The first term of Equation 3 represents the separation due to the averages of each class, while the second is the separation due to the covariances.

When the dendrogram has been constructed, the problem of finding the optimum number of classes is equivalent to the question of where to cut the dendrogram. A review of techniques for cutting the dendrogram is detailed in the book of Milligan and Cooper [8]. However, with these methods, the values have to be recomputed if a merge is done. We decide the level to cut by computing the average of the link distances that the cut will break. As larger the distance, more separated are the clusters among them. In this way, the problem of finding the optimum number of classes is equivalent to minimize the following energy function:

$$E(\text{cut}) = \frac{1}{D_{av}} \quad (4)$$

In this equation, D_{av} is the average of the distances of the links which the cut will broke.

If an estimation of the possible number of regions is available, this information can be taken into account by including a new term on the decision criterion. Specifically, Equation 4 can be generalized as:

$$E(\text{cut}) = \frac{1}{D_{av}} * (1 + \alpha(N_k - N_c)) \quad (5)$$

In this equation, D_{av} is the average of the distances of the links to cut, N_k is the previously estimated number of clusters given by a supervisor or extracted from the previous experimentation, and N_c is the number of clusters that the cut will produce. Note that the parameter α permits to modify the importance of the last term.

TABLE I

PIXEL CLASSIFICATION RATES FOR SYNTHETIC IMAGES FOR THE THREE CLUSTERING ALGORITHMS. FIRST ROW ARE THE RATES FOR THE ORIGINAL ALGORITHMS; SECOND ROW ARE THE IMPROVED VERSIONS USING PIXEL COORDINATES AS ARTIFICIAL FEATURES; THIRD ROW ARE THE RESULTS OBTAINED BY USING OUR APPROACH.

	KM		FCM		GM	
	Avg	Std	Avg	Std	Avg	Std
Original	60.2	12.5	61.1	14.7	66.5	12.5
Coord.	62.3	11.2	63.2	13.6	68.7	10.5
Proposal	70.1	10.8	71.3	9.8	75.5	9.6

III. EXPERIMENTAL RESULTS

Traditional clustering algorithms have been considered to compare the performance of classical versus the new proposal. Moreover, we have also compared our algorithms with those clustering versions that use pixel coordinates as artificial features. Note that we do not make direct comparison to known and more sophisticated algorithms, like Mean Shifts [9] or Normalized Cuts [10], because the aim of this paper is to provide significant improvements in classical clustering algorithms.

A. Clustering algorithms

In this paper we use three typical clustering partitioned algorithms, which are the k-Means, the Fuzzy C-Means and the Gaussian of Mixtures algorithm. We briefly review the implemented methods.

1) *k-Means Algorithm (KM)*: The popular k-Means algorithm [11] is an error minimization algorithm where the function to minimize is the sum of squared error:

$$e^2(K) = \sum_{k=1}^K \sum_{i \in C_k} (x_i - c_k)^2 \quad (6)$$

where c_k is the centroid of cluster C_k , and K the number of clusters (known a priori). Two factors have made the k-Means popular: it has linear time complexity and its easy implementation [1].

2) *Fuzzy C-Means Algorithm (FCM)*: One restriction of the k-Means algorithm is that it associates each pattern of the image into one, and only one, cluster. With the use of fuzzy theory, each pattern can be associated with every cluster using a membership function. See the book of Bezdek [12] for the source of these algorithms.

3) *Gaussian of Mixtures Algorithm (GM)*: Another way to allow each pattern to belong to different clusters is by using the Gaussian of Mixtures algorithm [13]. In this probabilistic model, each pattern is characterized by a set of Gaussian mixtures:

$$p(x_i; K) = \sum_{k=1}^K \pi_k g_k(x_i) \quad (7)$$

where g_i is a Gaussian distribution and π_i a prior distribution ($\sum_k \pi_k = 1$). The model parameters and the cluster membership function are determined by maximizing the log-likelihood

TABLE II

TABLE SHOWING THE RELATION WITH THE NUMBER OF REGIONS THAT OUR ALGORITHM FINDS (N_c) AND THE REAL NUMBER OF REGIONS (N_{gt}).

Real Images	KM	FCM	GM
$N_c < N_{gt}$	81%	83%	82%
$N_c = N_{gt}$	8%	7%	9%
$N_c > N_{gt}$	11%	10%	10%

TABLE III

PIXEL CLASSIFICATION RATES FOR REAL IMAGES FOR THE THREE CLUSTERING ALGORITHMS. FIRST ROW ARE THE RATES FOR THE ORIGINAL ALGORITHMS; SECOND ROW ARE THE IMPROVED VERSIONS USING PIXEL COORDINATES AS ARTIFICIAL FEATURES; THIRD ROW ARE THE RESULTS OBTAINED BY USING OUR APPROACH.

	KM		FCM		GM	
	Avg	Std	Avg	Std	Avg	Std
Original	69.6	17.5	62.8	19.7	75.6	13.3
Coord.	73.6	13.3	71.8	12.5	79.8	11.2
Proposal	81.2	11.2	82.2	8.2	87.3	8.6

function:

$$l(K) = \sum_{i \in I} \log(p(x_i; K)) \quad (8)$$

where I is the whole image. This step is efficiently done by using the Expectation Maximization algorithm [14].

B. Experimental Data

The described segmentation method can be performed over any set of real and synthetic images. In order to work with real images, texture features have to be used. The result of comparing the relative merits of the different types of texture features have been nonconclusive and a clear winner has not emerged in all cases. For the experimental trials showed in this paper we used the co-occurrence matrices proposed by Haralick et al. [15]. Two of the most typical features, contrast and entropy, are computed for distances one and five, and for orientations 0° , 45° , 90° , and 135° , constituting a 16-dimensional feature vector.

The evaluation of image segmentation has been performed by measuring the percentage of not-correctly segmented pixels considering the segmentation as a multi-class classification problem. Therefore, we used for this evaluation two image databases including ground-truth. Concretely, we used 50 synthetic mosaic images extracted from the Outex Database [16] and 20 real images extracted from the Berkeley Segmentation Dataset [17].

C. Discussion

As we can see in Table I an error mean of 39.8%, 38.9%, and 33.5% was obtained for the originals KM, FCM and GM algorithm when applied over the set of synthetic images. Meanwhile, an error mean of 37.7%, 36.8%, and 31.3% was obtained using pixel coordinates as artificial features and an error of 29.9%, 28.7%, and 24.5% was obtained when our proposal was considered (we have used in this experiment Equation 5 with $N_k = 5$). We want to emphasize the

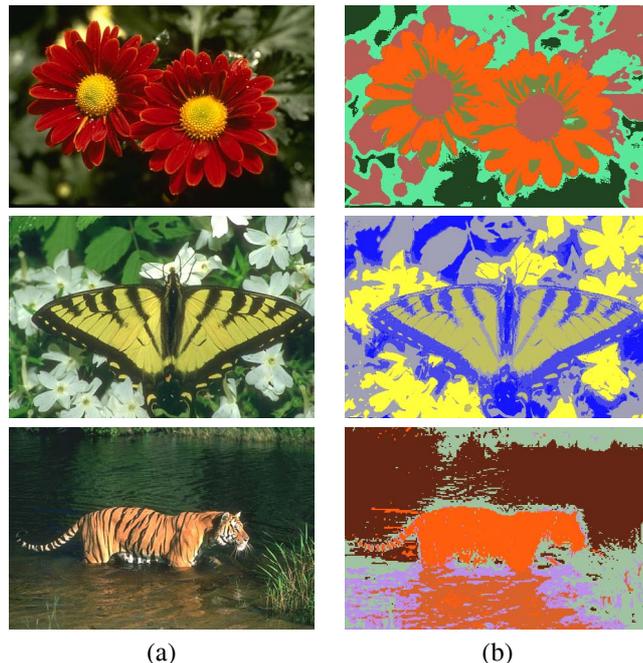


Fig. 4. Real image segmentation results. The images are extracted from the Berkeley Segmentation Dataset [17]. Column (a) shows the original images and column (b) the segmentation results.

improvement of the rate of well-segmented pixels, while not the absolute rate of pixel classification. Note that the Outex Database is designed as a framework for empirical evaluation of texture classification and segmentation algorithms [16]. In this sense, the results of the algorithms we have used, could be improved using another kind of texture features, like features extracted from Gabor filters, wavelets, etc.

The second set of images we tested was the real ones. In this case, we used Equation 4 because we do not know a priori the number of regions of the images. In Table II we analyse the number of regions that the algorithm found, comparing it with the number that ground-truth found. We can see that our algorithm tends to under-segment the images, i.e., our algorithm tends to merge regions.

Table III shows the rates of pixels well-classified we obtained when dealing with real images. We can see that the error mean using our approach is again reduced compared with the other cases. Moreover, Figure 4 shows three examples of real images segmented using our proposed method.

Finally, we want to remark that our approach is not slower than the remaining ones, although our algorithm calculates the best placement for the seeds, and does a final merge step. This is due to the fact that the clustering step of our approach is faster than the clusterings alone, because in our case, pixels which can modify the clusters (those near boundaries or far away of all clusters) are excluded of the iterations and classified later. In our experiments, our approach is 1.2 times slower than the k-Means algorithm, while the k-Means with pixels coordinates as artificial features is 1.5 times slower than the original k-Means. These rates are increased with the FCM

or the GM, but still satisfying the relation above mentioned.

IV. CONCLUSIONS

A new strategy for clustering segmentation which integrates region and boundary information has been described. The algorithm uses boundary information in order to initialize, in an unsupervised way, the number and the location of the seeds which are used as initial central points of the clustering algorithm. The clustering is subsequently applied only for pixels with low gradient, in order to not miss-classify boundary pixels. Subsequently, precise edges are determined by comparing neighbourhood of pixels with features of adjacent regions. Finally a merging step is required to merge adjacent regions. The method has been compared with classical clustering algorithms, and has been tested on a set of synthetic and real images. The experimental results demonstrate the effectiveness of the proposed algorithm in estimating regions with high accuracy, and the improvement of the classical clustering approaches.

REFERENCES

- [1] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM: Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [2] D. Marr, *Vision*. New York: Freeman and Company, 1982.
- [3] J. Bezdek and N. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 3, pp. 301–315, 1998.
- [4] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.
- [5] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 901–914, 1992.
- [6] J. Freixenet, X. Muñoz, J. Martí, and X. Lladó, "Colour texture segmentation by region-boundary cooperation," in *European Conference on Computer Vision*, vol. II, Prague, Czech Republic, 2004, pp. 250–261.
- [7] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002.
- [8] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 5, pp. 159–179, 1985.
- [9] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [11] J. MacQueen, "Some methods of classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
- [12] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [13] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.
- [14] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via em algorithm," *Journal of the Royal Statistical Society: Series B*, pp. 1–38, 1977.
- [15] R. Haralick, K. Shanmugan, and I. Dinstein, "Texture features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, November 1973.
- [16] T. Ojala, T. Menp, M. Pietikinen, J. Viertola, J. Kyllnen, and S. Huovinen, "Outex - new framework for empirical evaluation of texture analysis algorithms," in *IAPR International Conference on Pattern Recognition*, vol. 1, 2002, pp. 701–706.
- [17] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.