

# A fast fuzzy c-means algorithm for color image segmentation

Hoel Le Capitaine and Carl Frélicot

Laboratoire Mathématiques, Image et Applications, Université de La Rochelle, FRANCE

## Abstract

Color image segmentation is a fundamental task in many computer vision problems. A common approach is to use fuzzy iterative clustering algorithms that provide a partition of the pixels into a given number of clusters. However, most of these algorithms present several drawbacks: they are time consuming, and sensitive to initialization and noise. In this paper, we propose a new fuzzy c-means algorithm aiming at correcting such drawbacks. It relies on a new efficient cluster centers initialization and color quantization allowing faster and more accurate convergence such that it is suitable to segment very large color images. Thanks to color quantization and a new spatial regularization, the proposed algorithm is also more robust. Experiments on real images show the efficiency in terms of both accuracy and computation time of the proposed algorithm as compared to recent methods of the literature.

## 1. Introduction

Image segmentation can be defined as the process of merging pixels having similar features into the same groups, or regions. The segmented image is then the union of distinct groups, where pixels of homogeneous regions are associated to the same groups. Numerous techniques have been proposed in the literature, where color, texture or edges features are used to describe each group [8]. Only gray level images were considered by early segmentation methods. As color images become the norm in a wider range of applications (*e.g.* geographical imaging, medical imaging, or video surveillance), and thanks to advancements in both color technology and computation power, the interest of color image segmentation techniques has grown. Among them, we focus on the clustering approach, especially the fuzzy c-means algorithm (*FCM*, [3]), which is used by many segmentation methods [1, 13, 5, 22, 12]. However, this algorithm requires to initialize the centers of each cluster [10], and is known to be intractable for very large data sets such as color images. In this paper, we propose a new efficient initialization algorithm especially dedicated to the problem of image segmentation. Furthermore, we introduce a novel fuzzy iterative algorithm allowing fast segmentation of images.

This paper is organized as follows. Section 2 first recalls some basic knowledge on iterative clustering

algorithms, focusing on the fuzzy approach. Then, a new initialization of cluster centers is proposed, along with some numerical examples exhibiting its efficiency in terms of both accuracy and convergence speed. Various algorithms incorporating spatial and speeding-up considerations are presented in Section 3, as well as the new *Quantized Fuzzy C-Means (QFCM)* algorithm. Numerical experiments showing the superiority of *QFCM* as compared to state-of-the-art *FCM*-based segmentation methods are given in Section 4. Finally, some conclusions and perspectives are drawn in Section 5.

## 2. Iterative clustering algorithms

### 2.1. Algorithms

Let  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be a  $n$  samples data set and assume that each sample  $\mathbf{x}_k$  is represented by a set of  $p$  features. A partition of  $X$  into  $c$  clusters is a collection of mutually disjoint subsets  $X_i$  of  $X$  such that  $X_i \cup \dots \cup X_c = X$  and  $X_i \cap X_j = \emptyset$  for any  $i \neq j$ . Partitions can be represented by  $(c \times n)$  hard partition matrices  $U$  whose general term is  $u_{ik} = 1$  if  $\mathbf{x}_k \in X_i$ , and 0 otherwise. To get a partition matrix  $U$ , one can use the so-called *Hard c-Means (HCM)* algorithm which minimizes the within-cluster distances:

$$J = \sum_{i=1}^c \sum_{\mathbf{x} \in X_i} \|\mathbf{x} - \mathbf{v}_i\|^2, \quad (1)$$

where  $\|\cdot\|$  stands for the usual Euclidean distance and  $\mathbf{v}_i$  are the cluster centers gathered into a matrix  $V$  for convenience. The objective function can be rewritten as

$$J = \sum_{k=1}^n \sum_{i=1}^c u_{ik} \|\mathbf{x}_k - \mathbf{v}_i\|^2. \quad (2)$$

In many real situations, overlapping clusters reduce the effectiveness of crisp clustering methods. Ruspini first proposed the notion of fuzzy partition [18], where samples may partially belong to several clusters through the idea of partial membership degrees. Practically,  $u_{ik} \in [0, 1]$  instead of  $\{0, 1\}$ . Some years later, Dunn [6] modified of the objective function (2) by squaring the individual membership degrees. This has been generalized by Bezdek in [3], with a fuzzifier exponent  $m > 1$ :

$$J_m = \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2, \quad (3)$$

Minimization of (3) is generally obtained by an alternating optimization procedure that successively updates the cluster centers  $V$  and the partition matrix  $U$  using:

$$\mathbf{v}_i = \frac{\sum_{k=1}^n u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^n u_{ik}^m} \quad (4)$$

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{2/(m-1)}} \quad (5)$$

The choice to first initialize a random partition matrix or the cluster centers is left to the user, both being used in the literature. The algorithm stops when the centroids stabilize, *i.e.* the matrix norm between two successive  $V$  is below a given threshold. Equivalently, the entire procedure can be shifted one half cycle, so that initialization and termination is done on  $U$ . Naturally, in terms of speed and storage, there are some advantages to initialize and terminating with  $V$ .

Application to image segmentation consists in taking  $X$  as the entire set of pixels  $\mathbf{x}_k$  of an image  $I$ , each of them being described by  $p$  features.

## 2.2. Centroids initialization

As defined, iterative fuzzy clustering methods do not guarantee a unique final partition because different results are obtained with different initializations of  $V$  (or  $U$ ). In particular, it has been shown that these algorithms give better results when the initials ( $U/V$ ) are sufficiently close to the final partition/centers [10]. However, most of the practitioners initialize in a random manner, which heavily affects the results. Another reason to correctly initialize the cluster centers is that it allows to speed-up the convergence, resulting in a more usable algorithm for large scale practical problems. Several methods have been proposed for the initialization of  $V$ . However, most of the methods that allow to initialize cluster centers are computationally expensive. For instance, in [11], the method requires to run the *HCM* algorithm  $p$  times on  $n$  1-dimensional samples, and then  $p$  times the same algorithm on  $n$   $p$ -dimensional samples, which is intractable for large scale data sets.

We propose a new, efficient, yet simple, manner of initializing the  $c$  cluster centers, that we call *Ordering-split*. For each  $p$ -dimensional sample  $\mathbf{x}_k$ , we define its relative mean by

$$m_k = \frac{1}{p} \sum_{j=1}^p x_{kj} \quad (6)$$

so that we obtain the  $n$ -dimensional vector  $\mathbf{m} = (m_1, \dots, m_n)$ . Note that we are working on features coming from each channel of an image so that the scale of individual features does not differ. If the features do not hold this property, a normalization is required. Let  $\sigma$  be the permutation function

$\sigma$  such that  $m_{\sigma(k)}$  is an ordered and increasing sequence. We propose to split the  $n$  relative means as follows. Assuming that the clusters are equally distributed, we uniformly split the  $n$ -dimensional vector  $\mathbf{m}$  into  $c$  groups. In other terms, we set  $c+1$  indices, say  $\ell_0, \dots, \ell_c$ , such that the  $c$  differences  $(\ell_i - \ell_{i-1})$  are roughly equals<sup>1</sup>. More formally, each index is given by

$$\ell_i = i * \lfloor n/c \rfloor \quad (7)$$

where  $\lfloor \cdot \rfloor$  is the floor function. We iteratively build  $c$  subsets  $S_i$  of  $n$  as follows

$$S_i = \{\ell_{i-1} + 1, \dots, \ell_i\} \quad (8)$$

We obtain the subset of indices in each cluster by applying the inverse function:

$$C_i = \sigma^{-1}(S_i) \quad (9)$$

Finally, each cluster center is computed using:

$$\mathbf{v}_i = \frac{1}{|C_i|} \sum_{j \in C_i} \mathbf{x}_j \quad (10)$$

where  $|C_i|$  is the cardinality of  $C_i$ . The whole *Ordering-split* centers initialization procedure is summarized in Algorithm 1. Let us show how the

---

### Algorithm 1 Initialization of cluster centers

---

**Require:**  $X$ : dataset,  $c$ : number of clusters

- 1: **procedure** ORDERING-SPLIT( $X, c$ )
  - 2:   compute  $\mathbf{m}$  by using (6) for each  $k \in \{1, \dots, n\}$
  - 3:   apply to  $\mathbf{m}$  the ordering function  $\sigma$
  - 4:   **for**  $i = 0$  to  $c$  **do** ▷ uniform splitting
  - 5:      $\ell_i \leftarrow i * \lfloor n/c \rfloor$
  - 6:   **end for**
  - 7:   **for**  $i = 1$  to  $c$  **do** ▷ build the subsets
  - 8:      $S_i \leftarrow \{\ell_{i-1} + 1, \dots, \ell_i\}$
  - 9:      $C_i \leftarrow \sigma^{-1}(S_i)$
  - 10:      $\mathbf{v}_i \leftarrow \frac{1}{|C_i|} \sum_{j \in C_i} \mathbf{x}_j$
  - 11:   **end for**
  - 12:   **return**  $V$  ▷ the matrix of centers
  - 13: **end procedure**
- 

proposed method is efficient compared to a random initialization of the cluster centers. A simple color image (see Figure 1) in the  $p = 3$  components *RGB* color space, to which we add a Gaussian white noise with various standard deviations  $s$ , is used to compose  $X$ . The mean computation times over 100 runs of the *HCM* algorithm (for simplicity) are reported in Table 1. Note that the time needed for the computation of initial cluster centers is obviously added to the clustering time. We also report the cluster centers proximity index (CCPI) defined by

$$CCPI = \frac{1}{c \times p} \sum_i \sum_j \left| \frac{v_{ij}^* - v_{ij}}{v_{ij}^*} \right|, \quad (11)$$

<sup>1</sup>Note that other hypothesis could be provided, where some information about the clusters distribution is used.

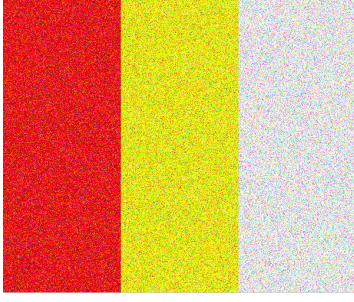


Figure 1: A  $500 \times 600$  (i.e.  $n = 300,000$ ) color image, with Gaussian white noise ( $s = 0.1$ ).

Table 1: Comparison of centers initialization: total time needed for initialization + clustering, proximity and accuracy.

	time (sec.)	CCPI	Accuracy
$s = 0.05$			
Random	1.286	0.729	98.33%
Ordering-split	0.578	0.088	98.33%
$s = 0.1$			
Random	1.870	0.714	92.18%
Ordering-split	0.787	0.118	92.27%
$s = 0.5$			
Random	2.776	0.787	53.13%
Ordering-split	1.670	0.181	65.90%

which measures the degree of closeness between the cluster centers  $V$  obtained by the initialization algorithm and the desired cluster center  $V^*$ . Clearly, the less the CCPI, the better the result. We finally give the accuracy performance defined by the ratio of correctly labeled pixels over the total number of pixels. According to Table 1, we see that the Ordering split method allows to obtain a faster algorithm (time), with an output that is more close to the reality (CCPI), and producing a higher accuracy than the method which consists in randomly initializing cluster centers.

### 3. Clustering for image segmentation

#### 3.1. Spatial FCM algorithms

Whilst the conventional FCM algorithm works well on noise-free images, it is very sensitive to local irregularities, which occur very often in real images. This sensitivity is due to the absence of consideration of the spatial context of each pixel. In [1], Ahmed *et al.* modify the original objective function by adding a penalty term that allows the memberships of each pixel  $\mathbf{x}_k$  to be influenced by its neighborhood. The new objective function is defined as

$$J_S = J_m + \frac{\alpha}{N_R} \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m \sum_{r \in N_k} \|\mathbf{x}_r - \mathbf{v}_i\|^2 \quad (12)$$

where  $N_R$  is the cardinality of  $N_k$ , which stands for the set of neighbors in a window around the pixel

$\mathbf{x}_k$ . The balance parameter  $\alpha$  allows to control the effect of the neighboring terms. However, computing the neighboring terms in each iteration is computationally expensive. Moreover, tuning  $\alpha$  is not easy, because a slight variation of  $\alpha$  produces very different segmentations. This algorithm is denoted *FCM\_S* in the sequel.

In [13], the authors propose another objective function where the relationship between neighboring pixels is taken into account. The usual Euclidean distance between pixels and centers by a weighted mean distance of the pixel and its neighbors to each center is taken. However, here again, in each iteration, all the pixels of the image are considered, leading to a large computation time.

In [5], the authors propose to reduce the computation time of the solutions derived from (12) by computing in advance the mean value of the pixel within the specified window:

$$J_{S1} = J_m + \alpha \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m \|\bar{\mathbf{x}}_k - \mathbf{v}_i\|^2 \quad (13)$$

where  $\bar{\mathbf{x}}_k$  is the mean of neighboring pixels in the window around  $\mathbf{x}_k$ . Additionally, they propose another objective function  $J_{S2}$  where  $\bar{\mathbf{x}}_k$  is the median value of the neighboring pixels. Then, they introduce the use of kernel-induced distances instead of the usual Euclidean one. The corresponding algorithms are respectively denoted as *KFCM\_S1* and *KFCM\_S2* in the sequel. More recently, Yang and Tsai [22] propose to adapt the balance parameter  $\alpha$  of (13) to each cluster, namely  $\alpha_i$ :

$$J_G = J_m + \sum_{k=1}^n \sum_{i=1}^c \alpha_i u_{ik}^m \|\bar{\mathbf{x}}_k - \mathbf{v}_i\|^2 \quad (14)$$

Moreover, since they use kernel-induced distance, they also propose to automatically set the parameters of the Gaussian kernel. The corresponding algorithm is claimed to be a generalized version of *KFCM*. Here again, the authors allow to consider mean-based or median-based spatial filtering. The derived algorithms are respectively denoted as *GKFCM\_S1* and *GKFCM\_S2* in the sequel.

Finally, in [12], the authors propose another modification of the objective function somewhat similar to (12) as follows

$$J_{FL} = J_m + \sum_{k=1}^n \sum_{i=1}^c G_{ik} \quad (15)$$

where the penalty term  $G_{ik}$  is defined by

$$G_{ik} = \sum_{j \in N_k, k \neq j} \frac{1}{d_s(k, j) + 1} (1 - u_{ij})^m \|\mathbf{x}_j - \mathbf{v}_i\|^2, \quad (16)$$

where  $d_s(k, j)$  is the spatial Euclidean distance between pixels  $\mathbf{x}_j$  and  $\mathbf{x}_k$ . The obtained corresponding updating functions are called *FLICM*. Naturally, *FLICM* suffers of the same high computation time as all the previous algorithms.

### 3.2. Speeded up clustering algorithms

The main drawback of such iterative clustering algorithms is their running time. In [7], the authors propose a fast and accurate clustering method of images. The time reduction is operated by aggregating similar examples and using the weighted prototype in the clustering, giving the *brFCM* algorithm. In order to speed up the segmentation, Szilagyi *et al.* [19] used the idea of Eschrich *et al.* [7] to propose the *EnFCM* algorithm, which consists in applying the *brFCM* algorithm to a smoothed image as follows. They first construct a linearly-weighted sum image with local neighbors of each pixel as follows:

$$\mathbf{x}'_i = \frac{1}{1 + \alpha} \left( \mathbf{x}_i + \frac{\alpha}{N_R} \sum_{j \in N_i} \mathbf{x}_j \right) \quad (17)$$

where  $N_i$  is the set of neighbors of the pixel  $\mathbf{x}_i$ , and the parameter  $\alpha$  controls the influence of the neighbors. Instead of considering each pixel of the image, the objective function uses the number of gray levels in the image, which drastically reduces the computation time, since the number of gray levels of an image is generally much lower than the number of pixels. In [4], Cai *et al.* propose an improvement of the *EnFCM* algorithm by adding a local similarity measure  $S_{ij}$ . The new image to be clustered is then defined as

$$\mathbf{x}'_i = \frac{\sum_{j \in N_i} S_{ij} \mathbf{x}_j}{\sum_{j \in N_i} S_{ij}} \quad (18)$$

where  $S_{ij}$  is a factor incorporating the spatial and gray level relationships in the neighborhood of  $i$ . They propose various definitions of the local similarity measure. The first one is given by

$$S_{ij} = \begin{cases} \exp \left( -\frac{d_s(i,j)}{\lambda_s} - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\lambda_g \sigma_i^2} \right) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad (19)$$

This similarity measure leads to the *FGFCM* algorithm. They also propose two other local similarity measures, that we denote *FGFCM\_S1* and *FGFCM\_S2*. The corresponding similarity measures are respectively defined by  $S_{ij} = 1$  for all  $i$  and  $j$ , so that  $\mathbf{x}_i$  is equal to the mean of the neighbors, and  $S_{ij} = \text{median}(I(j))$ , so that  $\mathbf{x}_i$  is the median value of the neighbors. *EnFCM*, *FGFCM* and their variants provide quite good segmentation. However, they heavily depend on the internal parameters  $\lambda_s$  and  $\lambda_g$ , or  $\alpha$ . Moreover, to the best of our knowledge, they are restricted to gray level images, and there are no propositions dedicated to color images. We propose to extend these approaches to color images by color quantization.

### 3.3. Fast quantized fuzzy c-means

Nowadays, processing an image without pre-processing and regularization is pointless. In [20],

the authors proposed the bilateral filtering procedure, which is an anisotropic approach based on both spatial and photometric considerations. Formally, the filtered image  $I'$  is obtained by

$$\mathbf{x}'_i = \frac{\sum_{j \in N_i} w(i,j) \mathbf{x}_j}{\sum_{j \in N_i} w(i,j)} \quad (20)$$

where  $w(i,j)$  are the weights applied to every pixel  $\mathbf{x}_j$  in  $N_i$ . The weights are decomposed by a conjunction into two weights corresponding to the spatial and the color weights,  $w(i,j) = w_s(i,j) \times w_c(i,j)$ . For instance, in [20],  $w_s$  and  $w_c$  are respectively defined by

$$\begin{aligned} w_s(i,j) &= \exp \left( -\frac{d_s^2(i,j)}{2\sigma_s^2} \right) \\ w_c(i,j) &= \exp \left( -\frac{d_c^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma_c^2} \right) \end{aligned} \quad (21)$$

In fact, the proposition of [4] consists in adding a bilateral filter process before the clustering algorithm. It can be shown that (20) reduces to (18) if we use  $d_s = L_\infty$  and  $d_c = L_2$ , where  $L_p$  stands for the  $p$ -norm. Extending the gray level algorithm [19] to color spaces is not immediate. If we use the same resolution, and take the RGB space, it would lead to compute  $U$  and  $V$   $2^8 * 2^8 * 2^8 = 2^{24}$  times. In other terms, this would be equivalent to run the usual fuzzy c-means algorithm on a  $(4096 \times 4096)$  color image. Since this is computationally intractable in practice, we propose to use a color space quantization into  $q_i$  bins,  $i$  corresponding to the channel index. Most of the color spaces use three channels, so that we have to define  $q_1$ ,  $q_2$  and  $q_3$ . Obviously, this can be extended to any multi spectral images, or images where several color spaces are used to describe each pixel (e.g. RGB, HSV and CIE  $L * a * b^*$ ). In the sequel we consider that each color component is divided into the same number of bins:  $q_1 = q_2 = q_3 = q$ .

Various studies have shown that many color spaces proposed for computer graphic applications are not well adapted to image processing. As pointed out in [2], a convenient representation should yield distances and provide independence between chromatic and achromatic components. For this reason and comparison purpose, we use the CIE  $L * a * b^*$  color space. There is another advantage of quantizing the  $L * a * b^*$  information rather than RGB information. If  $L * a * b^*$  is approximately uniformly distributed, then a uniform quantization yields a constant distance between any two quantization levels, resulting in small variation of perceptual color difference. This is not the case with RGB data where this variation can be very large.

Therefore we define the new objective function to be minimized as

$$J_Q = \sum_{i=1}^c \sum_{q_1=1}^q \sum_{q_2=1}^q \sum_{q_3=1}^q h_{q_1, q_2, q_3} u_{i, q_1, q_2, q_3}^m \|\mathbf{x}_{q_1, q_2, q_3} - \mathbf{v}_i\|^2 \quad (22)$$

where  $\|\cdot\|$  is a convenient norm in the quantized space, e.g. the Euclidean one. For convenience, we rewrite the objective function as

$$J_Q = \sum_{i=1}^c \sum_{l=1}^{q^3} h_l u_{il}^m \|\mathbf{x}_l - \mathbf{v}_i\|^2. \quad (23)$$

Since the gradient of  $J_Q$  with respect to  $u_{il}$  and  $\mathbf{v}_i$  vanishes when reaching the local optimum, and knowing that  $\sum_i u_{il}$  sums up to one for all  $l$ , it is easy to show that the optimal updating equations of  $U$  and  $V$  are given by

$$u_{il} = \frac{\|\mathbf{x}_l - \mathbf{v}_i\|^{-2/(m-1)}}{\sum_{j=1}^c \|\mathbf{x}_l - \mathbf{v}_j\|^{-2/(m-1)}} \quad (24)$$

and

$$\mathbf{v}_i = \frac{\sum_{l=1}^{q^3} h_l u_{il}^m \mathbf{x}_l}{\sum_{l=1}^{q^3} h_l u_{il}^m} \quad (25)$$

Although the introduction of a bilateral filtering process before clustering improves the effectiveness of segmentation, it still lack enough robustness and neighborhood importance should be taken into account in the clustering algorithm. To this aim, we propose, instead of considering the entire image, to regularize the partition matrix, based on the neighborhood of each pixel. The advantage of this proposition is to get rid of the selection of the crucial balance parameter  $\alpha$  in the methods of [1, 7, 5, 22]. This parameter ensures a balance between robustness to noise and effectiveness of preserving details. Hence it is hard to set and have considerable impact on the performances. However, due to quantization, the elements of the partition matrix do not have spatial relationships. To overcome this problem, we introduce a mapping of the quantized partition matrix of size  $(c \times q^3)$  to the usual partition matrix of the pixels, of size  $c \times (m \times n)$ , where  $m$  and  $n$  are the width and the height of the image. The basic idea of this mapping is, from a given pixel of the image, to obtain its corresponding bin, say  $l_i$ , in the quantized space. Then, this pixel inherits from the membership degrees of  $l_i$  obtained by  $u_{il}$ . In order to avoid ambiguities, an element of the usual partition matrix is denoted  $u_{ik}$ , while an element of the quantized partition matrix is denoted  $u_{il}$ .

Each element of the regularized partition is obtained by

$$u_{ik} = \frac{r_{ik}}{\sum_{j=1}^c r_{jk}} \quad (26)$$

where  $r$  is a spatial function. We define two general form for  $r$ :

- $r_{ik} = \sum_{j \in \mathcal{N}(\mathbf{x}_k)} u_{ij}$ ,
- $r_{ik} = \text{median}_{j \in \mathcal{N}(\mathbf{x}_k)} u_{ij}$ ,

which respectively correspond to a mean and a median filter applied to the membership degrees of the neighborhood of the pixel. However, in order to keep the algorithm fast, this regularization

cannot be done in each updating step, so that  $U$  is smoothed when the local optimum has been reached. The resulting algorithm, that we call QFCM, is given in Algorithm 2.

---

#### Algorithm 2 Quantized Fuzzy C-Means (QFCM)

---

- 1: **procedure** SEGMENTATION(Image  $I$ , No. of clusters  $c$ , No. of bins  $q$ )
  - 2:   Pre-process the image  $I$  using (20).
  - 3:   initialize cluster centers  $V$  using the Ordering-split procedure (Algorithm 1).
  - 4:   **repeat**
  - 5:     Update partition matrix  $U$  using (24).
  - 6:     Update prototypes matrix  $V$  using (25).
  - 7:   **until**  $\|V - V_{old}\| < \epsilon \triangleright \|\cdot\|$  is a matrix norm.
  - 8:   Regularize the partition  $U$  using (26).
  - 9:   **return**  $(U, V)$     $\triangleright$  Partition and centers.
  - 10: **end procedure**
- 

## 4. Results and comparison

### 4.1. Performance measures

The first measures of evaluation of segmentation were subjective, and the ever growing interest in this topic led to numerous metrics allowing proper evaluation.

In order to objectively measure the quality of the segmentations produced, 4 evaluation measures are considered in this paper. The first one is the Probabilistic Rand Index (PRI, [21]). This index compares results obtained from the tested algorithm to a set of manually segmented images. Since there is not a single correct output, considering multiple results allows to enhance the comparison and to take into account the variability of human perception. The PRI is based on a soft nonuniform weighting of pixel pairs as a function of the variability in the ground-truth. The ground-truth set is defined as  $\{G_1, G_2, \dots, G_L\}$  where  $L$  is the number of manually segmented images. Let  $S$  be the segmentation provided by the tested algorithm,  $l_i^{G_k}$  the label of pixel  $\mathbf{x}_i$  in the  $k$ -th manually segmented image and  $l_i^S$  the label of pixel  $\mathbf{x}_i$  in the tested segmentation. Then, PRI is defined by

$$PR(S, G_k) = \frac{2}{N(N-1)} \sum_{i,j,i < j} (p_{ij}^{c_{ij}} (1 - p_{ij})^{1-c_{ij}}), \quad (27)$$

where  $N$  is the number of pixels,  $c_{ij}$  is a boolean function denoting if  $l_i^S$  is equal to  $l_j^{G_k}$ , and  $p_{ij}$  is the expected value of a Bernoulli distribution for the pixel pair. The PRI metric is in the range  $[0, 1]$ , where high values indicate a large similarity between the segmented images and the ground-truth.

The second one is the Variation of Information (VOI, [15]). The VOI metric measures the sum of information loss and gain between two clusterings belonging to the lattice of possible partitions. It is

defined by

$$VOI(S, G_k) = H(S) + H(G_k) - 2I(S, G_k), \quad (28)$$

where  $H$  is the entropy  $-\sum_{i=1}^c \frac{n_i}{n} \log \frac{n_i}{n}$ ,  $n_i$  being the number of points belonging to the  $i$ -th cluster. The term  $I$  is the mutual information between two clustering, and it is defined by

$$I(S, G_k) = \sum_{i=1}^c \sum_{j=1}^c \frac{n_{i,j}}{n} \log \frac{n_{i,j}}{n} \frac{n_i}{n} \frac{n_j}{n} \quad (29)$$

where  $n_{i,j}$  is the number of points in the intersection of cluster  $i$  of  $S$  and  $j$  of  $G_k$ . The VOI measure is a distance, therefore the smaller it is, the closer the segmentation obtained and the ground-truth are.

The Global Consistency Error (GCE [14]) evaluates to what extent a segmentation can be viewed as the refinement of the other. A measure of error at each pixel  $\mathbf{x}_i$  is defined by

$$E(S, G_k, \mathbf{x}_i) = \frac{|R(S, \mathbf{x}_i) \setminus R(G_k, \mathbf{x}_i)|}{|R(S, \mathbf{x}_i)|}, \quad (30)$$

where  $|\cdot|$  is the cardinality,  $\setminus$  is the set difference, and  $R(S, \mathbf{x}_i)$  is the set of pixels corresponding to the region in segmentation  $S$  that contains the pixel  $\mathbf{x}_i$ . The GCE measure, which forces all local refinements to be in the same direction, is then defined by

$$GCE(S, G_k) = \frac{1}{n} \min \left( \sum_{i=1}^n E(S, G_k, \mathbf{x}_i), \sum_{i=1}^n E(G_k, S, \mathbf{x}_i) \right). \quad (31)$$

The closer  $GCE$  is to zero, the better the segmentation  $S$  with respect to the ground-truth  $G_k$ .

The Boundary Displacement Error (BDE [9]) evaluates the average displacement error of boundary pixels between two segmented images by computing the distance between the pixel and the closest pixel in the other segmentation. Given an arbitrary point  $\mathbf{x}_i$  of  $S$ , the BDE index uses the minimal Euclidean distance from  $\mathbf{x}_i$  to all points of  $G_k$ . A distance distribution signature  $D_S^{G_k}$  is then obtained by summing the distances over all points of  $S$ . The BDE index is finally obtained by

$$BDE(S, G_k) = \frac{1}{2} (D_S^{G_k} + D_{G_k}^S) \quad (32)$$

Here again, a small value of  $BDE$  indicates that  $S$  is a good segmentation considering the ground-truth segmentation  $G_k$ .

## 4.2. Experiments

In all experiments, the maximum number of iterations is set to 100, the termination criterion  $\varepsilon$  is set to 0.001, and a  $(3 \times 3)$  window is taken. As suggested by the authors in [1], the value  $\alpha$  of FCM\_S is set to 0.85. In [5], the authors suggest to use a value  $\alpha = 3.8$  for KFCM\_S1 and KFCM\_S2. For this experiment, the number of bins of QFCM is

Table 2: Average performance measures on the eight images.

	PRI	VOI	GCE	BDE
<i>FCM_S</i>	0.775	2.534	0.256	4.976
<i>KFCM_S1</i>	0.791	2.257	0.214	4.565
<i>KFCM_S2</i>	0.791	2.364	0.226	4.722
<i>GKFCM_S1</i>	0.762	3.245	0.282	5.347
<i>GKFCM_S2</i>	0.777	3.156	0.280	5.394
<i>FLICM</i>	0.799	2.412	0.227	4.462
<i>QFCM_S1</i>	0.818	1.871	<b>0.158</b>	4.154
<i>QFCM_S2</i>	<b>0.848</b>	<b>1.812</b>	0.167	<b>3.793</b>

Table 3: Average performance measures of *QKFCM\_S2* for different values of  $q$  ( $N$  fixed) and  $N$  ( $q$  fixed).

$q = 5$	PRI	VOI	GCE	BDE
$N = 3$	0.820	2.007	0.171	3.995
$N = 5$	0.848	1.812	0.167	3.793
$N = 7$	0.802	2.106	0.197	3.982
$N = 3$	PRI	VOI	GCE	BDE
$q = 5$	0.820	2.007	0.171	3.995
$q = 7$	0.812	2.142	0.202	4.213
$q = 9$	0.831	2.045	0.162	3.997

set to  $q = 5$  and the size of the window is  $N = 5$ . In order to provide fair comparison, all the algorithms are initialized with the cluster centers matrix obtained with the Ordering-Split method described in Section 2. The pre-processing step (20) is also applied before running all methods. All the algorithms are implemented in C++ language, and run on a Intel Core2 Duo, 2.4 GHz, with 3Gb of memory. In the sequel, eight color images publicly available [14] are considered. The eight images are plotted in Figure 2. For each of these images, several ground-truth segmentations that have been designed by humans are available. The images segmented by the *QFCM\_S2* algorithm are plotted in Figure 3, where pixels are defined by the mean color of each region  $\mathcal{R}_l$ ,  $l = 1 \dots r$ . In the second experiment, in order to study the influence of each parameter of *QKFCM\_S2*, one of the two parameters  $q$  or  $N$  stays constant, while the other varies. The results on the eight images are reported in Table 3. As can be seen, results obtained with the proposed algorithm are better than those of literature's algorithms, regardless the two input parameters,  $N$  and  $q$ . According to Table 2, we can draw the following remarks. Clearly, *QFCM\_S1* and *QFCM\_S2* are the two best methods on the considered images. An interesting point is that *QFCM\_S1* performs better than *QFCM\_S2* for the GCE measure, and worst for the other three. This is due to the definition of the GCE index, which does not penalize over-segmented images.



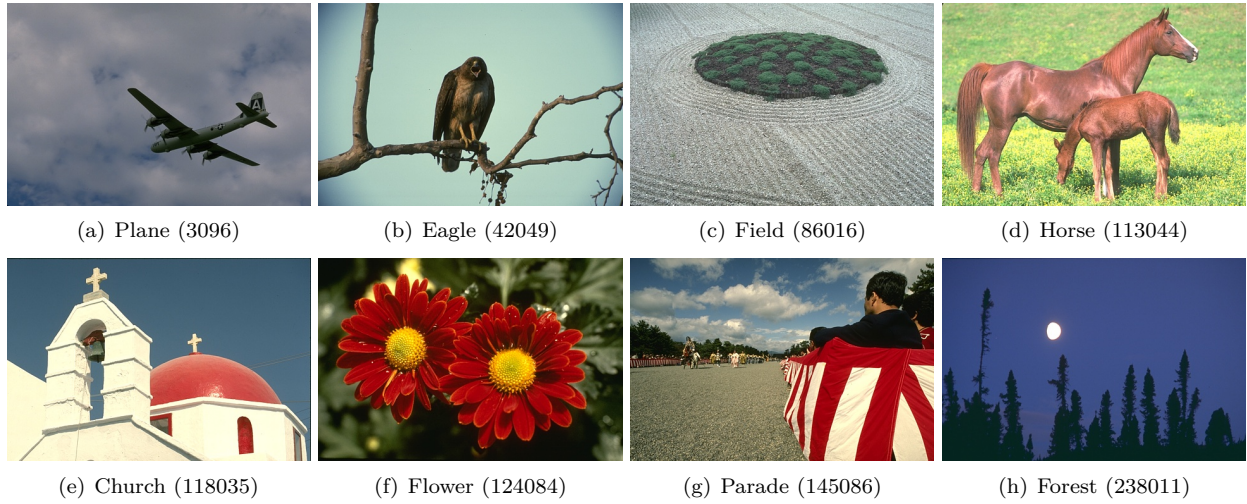


Figure 2: The eight color images to be segmented.

Table 4: Computation times (in seconds)

	Plane	Eagle	Field	Horse	Church	Flower	Parade	Forest
<i>FCM_S</i>	702.94	116.09	191.1	96.62	790.07	172.44	151.41	85.55
<i>KFCM_S1</i>	321.53	60.31	85.44	45.34	362.66	77.53	72.98	39.48
<i>KFCM_S2</i>	411.78	57.70	84.50	39.98	340.19	75.56	70.95	39.92
<i>GKFCM_S1</i>	438.08	446.29	121.60	62.70	799.65	144.56	208.94	454.57
<i>GKFCM_S2</i>	357.39	457.50	128.54	73.44	820.52	147.15	215.23	453.85
<i>FLICM</i>	507.38	516.55	462.66	268.68	898.91	920.02	326.10	211.31
<i>QFCM_S1</i>	0.57	0.56	0.55	0.58	0.81	0.58	0.56	0.56
<i>QFCM_S2</i>	0.71	0.71	0.69	0.70	1.09	0.75	0.71	0.70

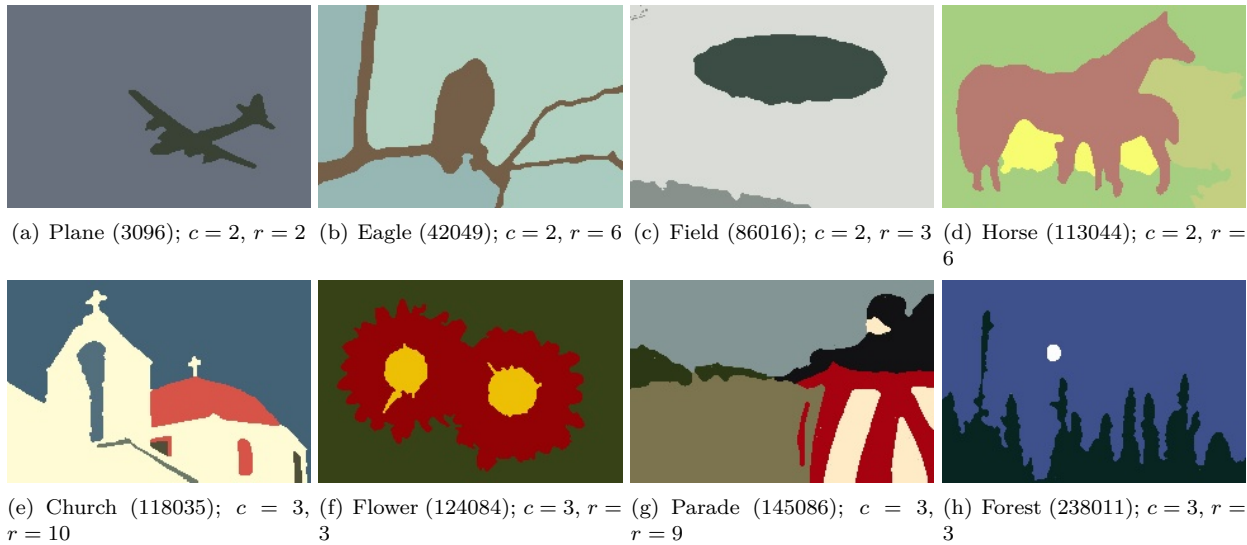


Figure 3: Results of the *QFCM\_S2* algorithm applied to the eight color images.

## 5. Conclusion

This paper presents a fast and efficient method for color image segmentation. We proposed several new methods and extensions in this paper. First, a new initialization method of cluster centers, allowing faster convergence of the iterative algorithm is presented. Second, we generalize the enhanced fuzzy c-means algorithm [4] to color images (in fact any  $N$ -d image) by using a quantization of the color space. Thus, we keep the ability of the algorithm to be computed quickly. Moreover, using a quantization of the color space allows to obtain a more robust clustering: noisy pixels are allocated to bins shared by noise-free pixels. According to an extensive comparison with state of the art segmentation methods, our approach gives satisfactory results. Moreover, the computation time has been drastically reduced, enabling to process very large images in a reasonable time. In the future, we plan to study a non uniform quantization of the color values in the CIE  $L^*a^*b^*$  space, so that regions having high density in the color space will benefit of a finer resolution, while areas with low density will have coarse resolution. Combining and fusing segmentations of different color spaces, as in [16], is also under study. An interesting work is also reported in [17], where the authors determine the number of bins according to the goodness-of-fit between the observed and expected frequencies given the  $q$  bins.

## References

- [1] M. N. Ahmed, S. M. Yamany, N. Mohamed, and A. A. Farag and T. Moriarty. A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *IEEE Transactions on Medical Imaging*, 21(3):193–199, 2002.
- [2] J. Angulo and J. Serra. Color segmentation by ordered mergings. In *IEEE International Conference on Image Processing*, 2003.
- [3] J. C. Bezdek. *Pattern Recognition with fuzzy objective function algorithm*. Plenum Press, 1981.
- [4] Weiling Cai, Songcan Chen, and Daoqiang Zhang. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition*, 40:825–838, 2007.
- [5] S. Chen and D. Zhang. Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man and Cybernetics, Part B.*, 34(4):1907–1916, 2004.
- [6] J. C. Dunn. A fuzzy relative isodata process and its use in detecting well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1974.
- [7] S. Eschrich, J. KE, L. O. Hall, and D. B. Goldgof. Fast accurate fuzzy clustering through data reduction. *IEEE Transactions on Fuzzy Systems*, 11(2):262–270, 2003.
- [8] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.
- [9] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cuff. Yet another survey on image segmentation: Region and boundary information integration. In *European Conference on Computer Vision*, pages 408–422, 2002.
- [10] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999.
- [11] S. K. Khan and A. Ahmad. Cluster center initialization algorithm for k-means clustering. *Pattern Recognition Letters*, 25:1293–1302, 2004.
- [12] S. Krinidis and V. Chatzis. A robust fuzzy local information c-means clustering algorithm. *IEEE Transactions on Image Processing*, 19:1328–1337, 2010.
- [13] A. Liew, S. H. Leung, and W. H. Lau. Segmentation of color lip images by spatial fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 11(4):542–549, 2003.
- [14] 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 *Proc. 8th Int. Conf. Computer Vision*, volume 2, pages 416–423, 2001.
- [15] M. Meila. Comparing clusterings: an axiomatic view. In *International Conference on Machine Learning*, pages 577–584, 2005.
- [16] M. Mignotte. Segmentation by fusion of histogram-based k-means clusters in different color spaces. *IEEE Transactions on Image Processing*, 17(5):780–787, 2008.
- [17] N. R. Pal and J. C. Bezdek. Complexity reduction for large image processing. *IEEE Transactions on Systems, Man, And Cybernetics - Part B*, 32(5):598–611, 2002.
- [18] E. H. Ruspini. A new approach to clustering. *Information and control*, 15(1):22–32, 1969.
- [19] L. Szilagyi, Z. Benyo, S. Szilagyi, and H. Adam. Mr brain image segmentation using an enhanced fuzzy c-means algorithm. In *Proc. of 25th Annual International Conference of IEEE EMBS*, 2003.
- [20] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, pages 839–, 1998.
- [21] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944, 2007.
- [22] M-S. Yang and H-S. Tsai. A gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction. *Pattern Recognition Letters*, 29(12):1713–1725, 2008.