information

MDPI

Article

Bit Reduced FCM with Block Fuzzy Transforms for Massive Image Segmentation

Barbara Cardone [1] and Ferdinando Di Martino [1,2,*]

[1] Department of Architecture, University of Naples Federico II, 80134 Naples, Italy; b.cardone@unina.it
[2] Interdepartmental Research Center of Research A. Calza Bini, University of Naples Federico II, 80134 Napoli, Italy
* Correspondence: fdimarti@unina.it; Tel.: +39-081-2538904; Fax: +39-081-2538909

Received: 28 May 2020; Accepted: 2 July 2020; Published: 5 July 2020

Abstract: A novel bit reduced fuzzy clustering method applied to segment high resolution massive images is proposed. The image is decomposed in blocks and compressed by using the fuzzy transform method, then adjoint pixels with same gray level are binned and the fuzzy c-means algorithm is applied on the bins to segment the image. This method has the advantage to be applied to massive images as the compressed image can be stored in memory and the runtime to segment the image are reduced. Comparison tests are performed with respect to the fuzzy c-means algorithm to segment high resolution images; the results shown that for not very high compression the results are comparable with the ones obtained applying to the fuzzy c-means algorithm on the source image and the runtimes are reduced by about an eighth with respect to the runtimes of fuzzy c-means.

Keywords: fuzzy c-means (FCM); bit-reduced fuzzy c-means (brFCM); Fuzzy Transform bit reduction FCM clustering algorithm (FTbrFCM); massive data; image segmentation

1. Introduction

The management of massive data represents a serious problem in image data clustering both in terms of memory allocation and execution times. In order to overcome this problem, it is necessary to reduce the size of the image, without producing information losses that affect the results of the clustering process. An idea to reduce the image size is to aggregate similar adjoint pixels into bins, using these bins as input data in the clustering process.

Bit-reduced fuzzy c-means (for short, brFCM) is an extension of the fuzzy c-means (for short FCM) algorithm [1,2] proposed in [3] in order to cluster large images. The image is binned removing the least significant bits in the pixels and binning adjoining identical pixel; then, a weighted FCM is applied to the binned image considering as weight the number of pixels in each bin.

In [4] brFCM is applied in very large (VL) image data clustering; the image is reduced by removing the least significant bits in the pixel values and binning adjoint pixel with identical grey level values; the weights are given by the number of pixels in each bin; the data assigned to the bin is the average of the pixel values of its pixels. The authors show that its performances are better than the ones of other FCM variations proposed in literature to cluster large and very large data.

An extension of the brFCM is proposed in [5] in hotspot detection from massive event spatial datasets. The binning strategy was to reduce the spatial scale and aggregating in a bin data points included in a determined convex region in the map; a bin is given by the centroid of the event points included in that region and its weight is given by the number of data points located inside this region.

A serious problem in applying the brFCM algorithm to massive images is the difficulty in storing in memory the entire image for binning similar adjoint pixels.

To solve this problem in this research we propose a new image bit reduction FCM clustering algorithm, called FTbrFCM, for segmenting massive image datasets in which the brFCM algorithm is applied to images compressed via F-transform.

The fuzzy transform technique [6] (for short, F-transform) is a consolidated method applied for lossy image compression [6–8]. In particular, in [7,8] is proposed a F-transform image compression method in which the image s decomposed in blocks; each block is compressed via F-transform and the compressed image is obtained merging the compressed blocks. In [9] the block F-transform image compression method is applied in image segmentation.

In FTbrFCM we apply this block F-transform image compression method to compress the image, then, the bit reduction method is executed to the compressed image, removing the least significant bits and merging adjoining identical pixels.

In this way, the whole image is not loaded in memory, but only some of its blocks; each block is compressed using the F-transform algorithm and, subsequently, the compressed image is recomposed in memory and the brFCM algorithm is applied.

In this way we obtain the following advantages:

- The clustering algorithm can be applied to the entire compressed image stored in memory;
- The runtime of the image segmentation algorithm is reduced, since it is applied on the compressed image;
- It is not necessary to remove the least significant bits in the pixels for bin the image, as the F-transform algorithm used to compress the image has already smoothed the image and the bins can be obtained spatially adjacent merging pixels with the same gray value.

The FTbrFCM algorithm is schematized in Figure 1.

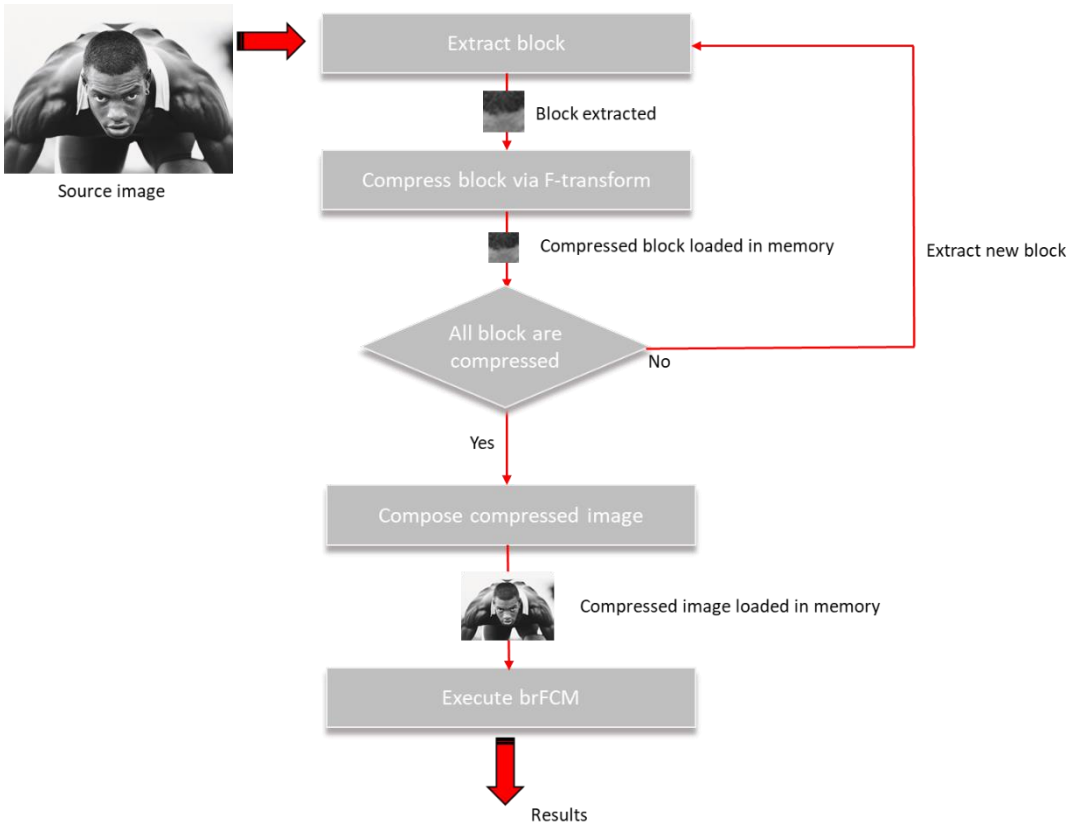


Figure 1. Schema of the image bit reduction fuzzy c-means (FCM) clustering algorithm (FTbrFCM).

After the compressed image is constructed merging adjoint compressed blocks, the brFCM algorithm is executed aggregating in a bin identical adjoint pixels and then running the weighted FCM in which the weight of a bin is given by the number of pixels aggregated in it.

The results are given by C segmented images, where C is the number of clusters; the value assigned to a pixel in the *i*th segmented image is given by the membership degree of its bin to the *i*th cluster.

We test our algorithm on high resolution color images comparing the performances with the ones measured executing FCM on the source not compressed image.

This document is structured as follows: in Section 2 the bi-dimensional F-transform concept and the block-wise F-transform image compression method are introduced; a detailed discussion of the bidimensional F-transform and its application in image compression is in [6–8]. The brFCM algorithm is descripted in Section 3. In Section 4 the FTbrFCM algorithm is presented in detail. In Section 5 are shown the results of comparison test applied in image. Final discussions are in Section 6.

2. F-Transform Image Compression

Let [a, b] ⊂ R be a closed interval, $n \geq 2$ and $\{x_1, x_2, \ldots, x_n\} \subset [a, b]$ be a set of points called nodes, such that $a \leq x_1 < x_2 < \ldots < x_n \leq b$. The family of fuzzy sets $A_1, \ldots, A_n$: [a, b] → [0, 1] is a fuzzy partition of [a, b] if for every $i = 1, 2, \ldots, n$ the following conditions hold:

1. $A_i(x_i) = 1$
2. $A_i(x) = 0$ if $x \notin (x_{i-1}, x_{i+1})$, where $x_0 = a$ and $x_{n+1} = b$
3. $A_i(x)$ is a continuous function on [a, b];
4. $A_i(x)$ strictly increases on $[x_{i-1}, x_i]$ and strictly decreases on $[x_i, x_{i+1}]$;
5. $\forall x \in [a, b]\ \sum_{i=1}^{n} A_i(x) = 1$. $A_1(x) + A_2(x) + \ldots + A_n(x) = 1$ (Ruspini condition).

The fuzzy sets $\{A_1, \ldots, A_n\}$ are called basic functions. They form an uniform fuzzy partition of [a, b] if $n \geq 3$ and for every $i = 1, 2, \ldots, n$ the following conditions hold:

6. $x_i = a + h \cdot i$, where $h = (b - a)/(n + 1)$ (that is, the nodes are equidistant);
7. $A_i(x_i - x) = A_i(x_i + x)$ for every x in [0, h]
8. $A_{i+1}(x) = A_i(x - h)$ for every x in $[x_i, x_{i+1}]$ and $i = 1, 2, \ldots, n - 1$.

Let *f* be a function continuous in [a, b] and let $P = \{p_1, \ldots, p_m\}$ be a discrete set of points in [a, b] in which the function *f* is known. We assume that the set *P* of these points is sufficiently dense with respect to the fixed uniform fuzzy partition, that is for each $i = 1, \ldots, n$ there exists an index *j* in $\{1, \ldots, m\}$ such that $A_i(p_j) > 0$. Then we can define the *n*-tuple $\{F_1, \ldots, F_n\}$ as the discrete F-transform of *f* with respect to $\{A_1, A_2, \ldots, A_n\}$, where each F_i is given by:

$$F_k = \frac{\sum_{i=1}^{N} f(p_i) A_k(p_i)}{\sum_{i=1}^{N} A_k(p_i)} \tag{1}$$

for $k = 1, \ldots, n$. Now we define the discrete inverse F-transform of *f* with respect to $\{A_1, A_2, \ldots, A_n\}$ to be the following function defined in the same points $p_1, \ldots, p_m$ of [a, b]:

$$f_{F,n}(p_i) = \sum_{k=1}^{n} F_k A_k(p_i) \tag{2}$$

The (2) approximate the function *f* in the interval [a, b] (cfr. [5], Theorem 18).

2.1. F-Transforms in Two Variables

The discrete direct and inverse fuzzy transforms of a function *f* continuous in [a, b] can be extended to functions in two variables. Assume that our universe of discourse is the rectangle [a, b] × [c, d] and let $m,n \geq 2$, $\{x_1, x_2, \ldots, x_m\} \subset [a, b]$ be a set of nodes in [a, b] and $\{y_1, y_2, \ldots, y_n\} \subset [c, d]$ be a set of nodes in [a, b], such that $x_1 \leq a < x_2 < \ldots < x_m \leq b$ and $y_1 \leq c < \ldots < y_n \leq d$. Furthermore, let $A_1, \ldots, A_m$: [a, b] → [0, 1] be a fuzzy partition of [a, b] and $B_1, \ldots, B_n$: [c, d] → [0, 1] be a fuzzy partition of [c, d].

Let f be a function continuous in [a, b] × [c, d] known in a discrete set of points $(p_i,q_j) \in$ [a, b] × [c, d], where $i = 1, \ldots, M$ and $j = 1, \ldots, N$, $P = \{p_1, \ldots, p_M\}$ sufficiently dense with respect to $\{A_1, \ldots, A_m\}$ and $Q = \{q_1, \ldots, q_N\}$ sufficiently dense with respect to $\{B_1, \ldots, B_n\}$.

Then, generalizing the Equation (1), we can define the discrete F-transform matrix of f $[F_{kl}]$, with respect to $\{A_1, \ldots, A_m\}$ and $\{B_1, \ldots, B_n\}$ with components $k = 1, \ldots, m$ and $l = 1, \ldots, n$:

$$F_{\mathrm{kl}} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M} f(p_i, q_j)A_k(p_i)B_l(q_j)}{\sum_{j=1}^{N}\sum_{i=1}^{M} A_k(p_i)B_l(q_j)} \tag{3}$$

By extending Equation (2) to the case of two variables, we define the discrete bi-dimensional inverse F-transform of f with respect to $\{A_1, A_2, \ldots, A_n\}$ and $\{B_1, \ldots, B_m\}$ to be the following function defined in the same points (p_i,q_j) in [a, b] × [c, d], with i in $\{1, \ldots, N\}$ and j in $\{1, \ldots, M\}$, as:

$$f_{F,m,n}(p_i\, \mathrm{q}_j) = \sum_{l=1}^{n}\sum_{k=1}^{m} F_{kl}A_kB_l \tag{4}$$

The inverse F-transform (4) approximate the bidimensional function f in [a, b] × [c, d].

2.2. F-Transforms in Two Variables for Image Compression

Let I(x,y) an image function discretized in a digital gray $M \times N$ image composed of $M \times N$ pixels with coordinates $(p_i,q_j) \in \{1, \ldots, M\} \times \{1, \ldots, N\}$. For brevity of notation, we put $p_i = i$, $q_j = j$ and [a, b] × [c, d] = [1, N] × [1, M].

The matrix I is divided in submatrices of identical size $M_C \times N_C$ called blocks, where $M_C < M$, $N_C < N$, M_C is a divisor of M and N_C is a divisor of N. The image I is then composed of $(M_C \times N_C)/(M \times N)$ blocks of equal size $M_C \times N_C$.

Each block is composed of $M_C \times N_C$ pixels with coordinates $(p_i,q_j) \in \{1, \ldots, M_C\} \times \{1, \ldots, N_C\}$. For brevity of notation, we put $p_i = i$, $q_j = j$ and [a, b] × [c, d] = [1, M_C] × [1, N_C].

Let $A_1, \ldots, A_m$: [1, M_C] → [0, 1] be a fuzzy partition of [1, M_C] with $m_C < M_C$ and let $B_1, \ldots, B_n$: [1, N_C] → [0, 1] a fuzzy partition of [1, N_C] with $n_C < N_C$.

Each block of sizes $M_C \times N_C$ is compressed in a block of sizes $m_C \times n_C$ via the discrete bi-dimensional F-transform $[F_{kl}^C]$ with components given by:

$$F_{kl}^{C} = \frac{\sum_{j=1}^{N_C}\sum_{i=1}^{M_C} I_C(i, j)A_k(i)B_l(j)}{\sum_{j=1}^{N_C}\sum_{i=1}^{M_C} A_k(i)B_l(j)} \tag{5}$$

for each $k = 1, \ldots, m_C$ and $l = 1, \ldots, n_C$. As above, naturally we do in such a way that the set{1, …, M(C) (resp., {1, …, N(C)})} is sufficiently dense to the fuzzy partition of the basic functions $\{A_1, \ldots, A_{m(C)}\}$ defined (resp., $\{B_1, \ldots, B_{n(C)}\}$) in the interval [1, M(C)] (resp., [1, N(C)]) considered.

The parameter ρ: = $(m_C \cdot \times n_C)/(M_C \times N_C)$ is the compression rate of the block.

Afterwards, we decode the compressed blocks via the discrete inverse F-transform $I^F_{m(C)n(c)}$:{1, …, M(C)} × {1, …, N(C)} → [0, 1] defined as:

$$I^F_{m_C n_C}(i, j) = \sum_{l=1}^{n_C}\sum_{k=1}^{m_C} F_{kl}^{C}A_k(i)B_l(j) \tag{6}$$

which approximates I_C with arbitrary precision.

The tests conducted in [10 ÷ 14] have shown that the best performances are obtained by using a symmetric fuzzy partition of [1, M_C] whose fuzzy sets $A_1, \ldots, A_{m(C)}$:[1, M_C] → [0, 1]] are defined as

$$A_k(i) = \begin{cases} 0.5(\cos\frac{\pi}{h}(i - x_k) + 1) & \text{if } i \in [x_{k-1}, x_{k+1}] \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where k =1, 2, …, m_C, h = (M_C − 1)/(m_C + 1), x_0 = 1 and x_{m_C} + 1 = m_C, and by using a symmetric fuzzy partition of [1, N_C] whose fuzzy sets B_1, …, B_{n_C}:[1, N_C] → [0, 1] are defined as:

$$B_l(j) = \begin{cases} 0.5(\cos\frac{\pi}{s}(j - y_l) + 1) & \text{if } j \in [y_{l-1}, y_{l+1}] \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where l = 1, 2, …, n_C, s = (N_C − 1)/(n_C +1), y_0 = 1 and x_{n_C} + 1 = n_C.

The compressed image I_ϱ of I obtained using the compression ratio ρ is constructed merging the compressed blocks where each compressed block is given by the direct F-transform matrix [F_{kl}^C] calculated by (5).

Algorithm 1 shows in pseudocode the block F-transform image compression algorithm.

Algorithm 1. Block F-Transform Image Compression.

Input: *Source Image I*
Output: *Compressed image* I_ρ

```
Set Mc,Nc,nc,mc where ϱ := mc·nc / Mc Nc
Set the basic functions A1, A2, …, An as in (7) and B1, B2, …, Bn as in (8)
For each block Ic
    For k = 1 to mc
        For l = 1 to nc
            Numkl := 0      // Numerator of the Ftransform component (5)
            Denkl := 0      // Denominator of the Ftransform component (5)
            For i = 1 to Mc
                For j = 1 to Nc
                    Numkl := Ic(i,j)·Ak(i) Bl(j)
                    Denkl := Ak(i) Bl(j)
                Next j
            Next i
            Fkl^C := Numkl / Denkl // Ftransform component (5)
        Next l
    Next k
Next block
Merge the compressed block to obtain the compressed image Iϱ
Store the compressed image Iϱ
```

The compressed image I_ϱ can be decompressed dividing it in M_C × N_C blocks and decoding every block in a M_C × $N_C$block by (6); finally, the decoded blocks are merged to form the decompressed image.

3. The brFCM Algorithm

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset R^n$ be a set of data N_P data points: each data point $\mathbf{x}_j = (x_{j1}, \ldots, x_{jn})$ is a vector in the space R^n.

FCM is a partitive fuzzy clustering algorithm aimed to find as set of points in R^n the set of C fuzzy cluster centers $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_C\}$ where $\mathbf{v}_i = (v_{i1}, \ldots, v_{jn})$ (i = 1, …, C). The C × N_P partition matrix **U** with components u_{ij} i = 1, …, C; j = 1, …, N_P give the membership degree of the jth data point to the ith cluster.

FCM find **V** and **U** minimizing the following objective function:

$$J(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{C}\sum_{j=1}^{N_p} u_{ij}^{\gamma} d_{ij}^2 = \sum_{i=1}^{C}\sum_{j=1}^{N_p} u_{ij}^{\gamma} \left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2 \quad (9)$$

where $d_{ij} = \left\|x_j - v_i\right\|$ is the Euclidean distance between $\mathbf{v}_i$ and $\mathbf{x}_j$a.

The parameter $\gamma \in [1, +\infty)$ is called fuzzifier parameter; it determines the fuzziness degree of the fuzzy partition. (a constant which affects the membership values and determines the degree of fuzziness of the partition).

By applying the Lagrange multipliers, and considering the constraints:

$$\sum_{i=1}^{C} u_{ij} = 1 \qquad \forall\ j \in \{1,\ldots,N\} \quad (10)$$

$$0 < \sum_{j=1}^{N_p} u_{ij} < N_p \qquad \forall i \in \{1,\ldots,C\} \quad (11)$$

are obtained for **U** and **V** the following solutions

$$\mathbf{v}_i = \frac{\sum_{j=1}^{N_p} u_{ij}^{\gamma}\mathbf{x}_j}{\sum_{j=1}^{N_p} u_{ij}^{\gamma}} \forall i \in \{1,\ldots,C\} \quad (12)$$

and

$$u_{ij} = \frac{1}{\left(\sum_{k=1}^{c} \frac{d_{ij}^2}{d_{kj}^2}\right)^{\frac{2}{\gamma-1}}} \quad \forall i \in \{1,\ldots,C\},\ j \in \{1,\ldots,N_p\} \quad (13)$$

FCM is an iterative algorithm in which initially the membership degrees (or the cluster centers) are assigned randomly; in any cycle the cluster centers and the membership degrees are calculated via (12) and (13). The algorithm stops after t iterations if:

$$\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| < \varepsilon \quad i = 1,\ldots,C;\quad j = 1,\ldots,N \quad (14)$$

where $\varepsilon > 0$ is a parameter assigned a priori to stop the iteration process and

$$\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| = \max_{\substack{i=1,\ldots,C \\ j=1,\ldots,N}} \left\{\left|u_{ij}^{(t)} - u_{ij}^{(t-1)}\right|\right\} \quad i = 1,\ldots,C;\quad j = 1,\ldots,N. \quad (15)$$

The pseudocodes of the FCM is given in Algorithm 2.

Algorithm 2. FCM.

Input: ***INPUT DATASET D WITH N_P DATA POINTS***
Output: *Partition matrix* $\boldsymbol{U}$ *and cluster centers* $\boldsymbol{V}$
1. Set γ, ε, C
2. Initialize randomly the partition matrix **U**
3. **Repeat**
4. Calculate v_i, $i = 1, \ldots, C$ by using Equation (12)
5. Calculate u_{ij}, $i = 1, \ldots, C$ $j = 1, \ldots, N_P$ by using Equation (13)
6. **Until** $\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| > \varepsilon$

A variation of the FCM algorithm is the weighted FCM (wFCM) algorithm [10] in which a weight defines the influence of the data point to the solutions; data points with higher weights influence the determination of location of the cluster centers more than the others.

We can consider FCM a special case of wFCM where wj = 1 for each j = 1, …, N_P, in which each data point influences the determination of cluster centers in the same way.

wFCM minimize the following objective function:

$$J_w(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{C}\sum_{j=1}^{N_p} w_j u_{ij}^{\gamma} d_{ij}^2 = \sum_{i=1}^{C}\sum_{j=1}^{N_p} w_j u_{ij}^{\gamma} \left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2 \quad (16)$$

Using the Lagrange multipliers, and considering the constraints (10) and (11), are obtained for the cluster centers the solutions:

$$\mathbf{v}_i = \frac{\sum_{j=1}^{N_p} w_j u_{ij}^{\gamma} \mathbf{x}_j}{\sum_{j=1}^{N_p} w_j u_{ij}^{\gamma}} \forall i \in \{1,...,C\} \quad (17)$$

The solutions obtained for the membership degrees are given by (13).

The pseudocode of wFCM is shown in Algorithm 3.

Algorithm 3. WFCM.

Input: *Input dataset D with N_P data points*
Output: *Partition matrix $\boldsymbol{U}$ and cluster centers $\boldsymbol{V}$*
1. Set γ, ε, c
2. Initialize randomly the partition matrix **U**
3. **Repeat**
4. *Calculate* w_j, j = 1, …, N_P by using a weight function $w(\mathbf{x}_j)$
5. *Calculate* $\mathbf{v}_i$, i = 1, …, C by using Equation (17)
6. *Calculate* u_{ij}, i = 1, …, C; j = 1, …, N by using Equation (13)
7. **Until** $\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| > \varepsilon$

A density-based wFCM was proposed in [11] for reducing the size of the input dataset. In [12,13] two weighted FCM algorithms are used for image segmentation. In [14] a wFCM algorithm is applied to solve the Source Location problem.

The brFCM algorithm is proposed in [3] to handle massive datasets. It uses a wFCM algorithm in which the weight is given by the number of data points merged in a bin.

The pseudocode of brFCM is shown in Algorithm 4.

Algorithm 4. BRFCM.

Input: *Input dataset D with N_P data points*
Output: *Partition matrix $\boldsymbol{U}$ and cluster centers $\boldsymbol{V}$*
1. Set γ, ε, c
2. Bin the dataset in N_P quantization bins
3. Assign the weight w_j, j = 1, …, N_P as the number of data points merged in a bin
4. Initialize randomly the partition matrix **U**
5. **Repeat**
6. *Calculate* $\mathbf{v}_i$, i = 1, …, C by using Equation (17)
7. *Calculate* u_{ij}, i = 1, …, C; j = 1, …, N by using Equation (13)
8. **Until** $\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| > \varepsilon$

4. F-Transform brFCM Algorithm

The FTbrFCM algorithm is made up of two phases. In the first phase, the F-transform algorithm is applied by acquiring in memory and compressing the individual blocks of the image I and, finally, merging the compressed blocks to form the compressed image Ip, stored in memory.

In the second phase the brFCM algorithm is executed on the compressed image Ip. The bins are made up of sets of adjacent pixels with the same gray value in the compressed image and the weight of each bin is given by the number of these pixels.

The binning process is accomplished by examining the pixels of the image. If the pixel value is identical to that of a neighboring pixel already binned, then the pixel is merged into that bin and the weight value associated with the bin is increased; otherwise, a new bin is created to which the pixel value is assigned and the weight of this bin is initialized to 1.

The pseudocode of FTbrFCM is shown in Algorithm 5.

Algorithm 5 FTBRFCM.

Input: *Source image I with size M × N*
Output: *Partition matrix* $\boldsymbol{U}$ *and cluster centers* $\boldsymbol{V}$

```
Set γ, ε, c
  Set Mc,Nc,nc,mc where ϱ := mc·nc/Mc Nc
  Call Block F-Transform Image Compression(I,Mc,Nc,nc,mc)
  Bin the compressed image Ip with size m × n in a dataset D with Np quantization bins
  For i = 1 to m
    For j = 1 to n
      If exists a neighboring binned pixel with same pixel value, then
        Merge the pixel in the correspondent bin
        Increase the weight associated with the bin by one unit
      Else
        Create a new bin
        Initialize to 1 the weight of the new bin
      End if
    Next j
  Next i
  Repeat
    Calculate vi, i = 1, …, C by using Equation (17)
    Calculate uij, i = 1, …, C; j = 1, …, N by using Equation (13)
  Until |U(t) − U(t−1)| > ε
```

The results can be treated as a set of C images with size m × n. In fact, we assign the membership degree of a bin to a cluster to every pixel belonging to this bin, obtaining a m × n matrix; then we normalize the values of the pixels in the number of grey levels (for example, multiplying the pixel values by 255, considering 256 grey levels, and approximating the value obtained to the nearest integer).

To measure the performance of FTbrFCM, we test the FTbrFCM algorithm in image segmentation of high-resolution images. Then a comparison with the results obtained applying FCM is performed.

To carry out this comparison, after obtained the image corresponding

(1) The m × n image corresponding to the membership degree to the c*th* cluster with c = 1, …, C, is decompressed in a M × N image by applying the block- inverse F-transform (6) and merging the decoded blocks to form a decompressed N × M image I_c.
(2) Let I^0_c be, the N × M resultant image correspondent to the c*th* cluster obtained executing FCM on the original image, the root mean square error (RMSE) index of I_c with respect to I^0_c is given by:

$$\mathrm{RMSE}(I_c, I_c^0) = \sqrt{\frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I_c(\mathrm{i}, \mathrm{j}) - I_c^0(\mathrm{i}, \mathrm{j}) \right)^2} \quad c = 1,\ldots,\mathrm{C} \tag{18}$$

(3) The final RMSE index is given by the average of the RMSE measures calculated for all the clusters:

$$\mathrm{RMSE} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{RMSE}(I_c, I_c^0) \tag{19}$$

We compress the image with different compression rates, measuring the trend of the RMSE error varying the compression rate.

In next section we show the results of our tests performed applying the FTbrFCM algorithm on massive image datasets.

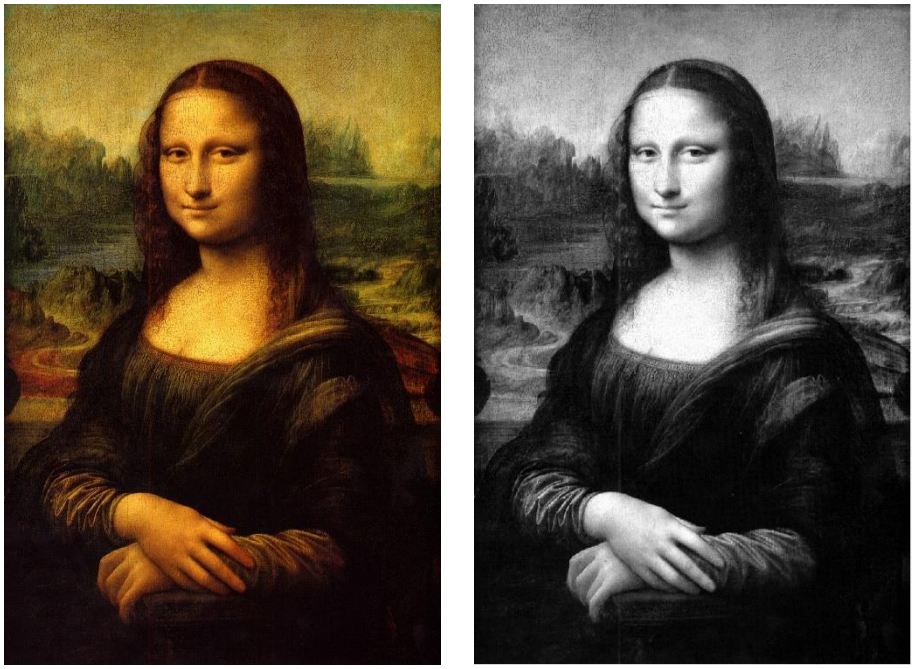
5. Test Results

The FTbrFCM is applied to a set about 200 color high-resolution images of paintings by famous painters in the Google Art & culture web page (https://artsandculture.google.com); the mean resolution of these images is of 10^8 pixels. Each image is decomposed in the three bands R, G and B, then, FTbrFCM is executed on the image in each band. The Xie-Beni validity index [15] is used to find the optimal number of clusters.

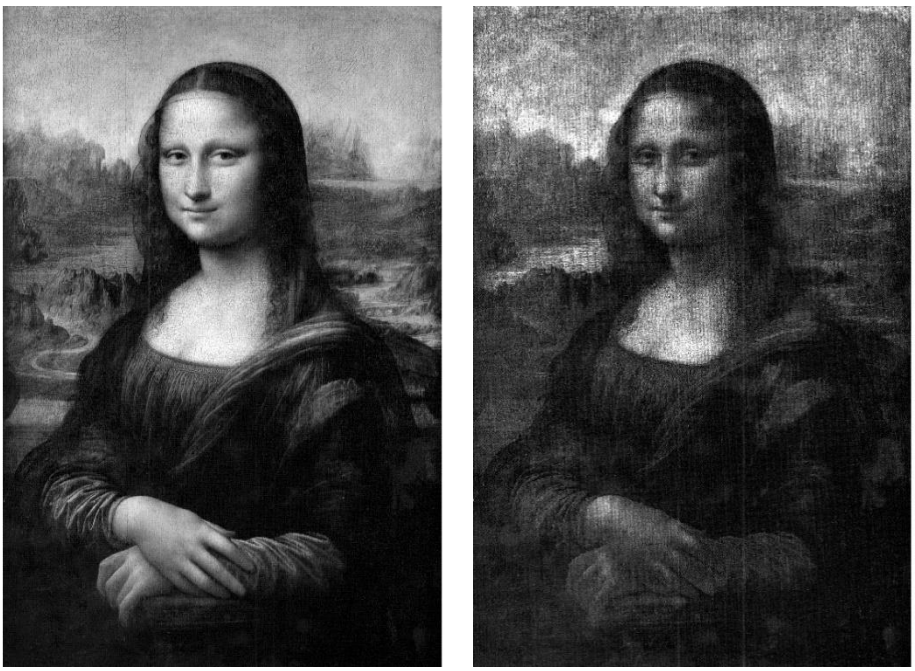
We apply FTbrFCM on the image on each band by using various compression rates. We execute FTbrFCM on an Intel core I5 3.2 GHz processor.

For brevity we present the detailed results obtained for the color images Mona Lisa, which represents the homonymous oil painting by Leonardo da Vinci preserved in the Louvre, and Sunflowers, which represents the Van Gogh's the painting on canvas, Vase with 15 Sunflowers, preserved at Van Gogh Museum

In Figure 2a–d is shown the high-resolution image Mona Lisa decomposed in the three band R, G and B.



(**a**) (**b**)



(**c**) (**d**)

Figure 2. (**a**) Mona Lisa; (**b**) Mona Lisa R Band; (**c**) Mona Lisa G Band; and (**d**) Mona Lisa B Band.

The images in the three bands are been compressed and segmented setting C = 3 clusters.

Figures 3–5 show, respectively, the segmented images obtained in the bands R, G and B, compressing the original image with a compression rate $\rho = 0.25$, obtained compressing each block 4 × 4 in a block 2 × 2.

Figure 3. Mona Lisa R Band—compression rate 0.25—segmented images.

Figure 4. Mona Lisa G Band—compression rate 0.25—segmented images.

Figure 5. Mona Lisa B Band—compression rate 0.25—segmented images.

A segmented image can be subsequently processed to be classified. As an example, the binary image in Figure 6 show the results of the classification of the second segmented image in the G band and the pixel values frequency histogram used to classify the pixels.

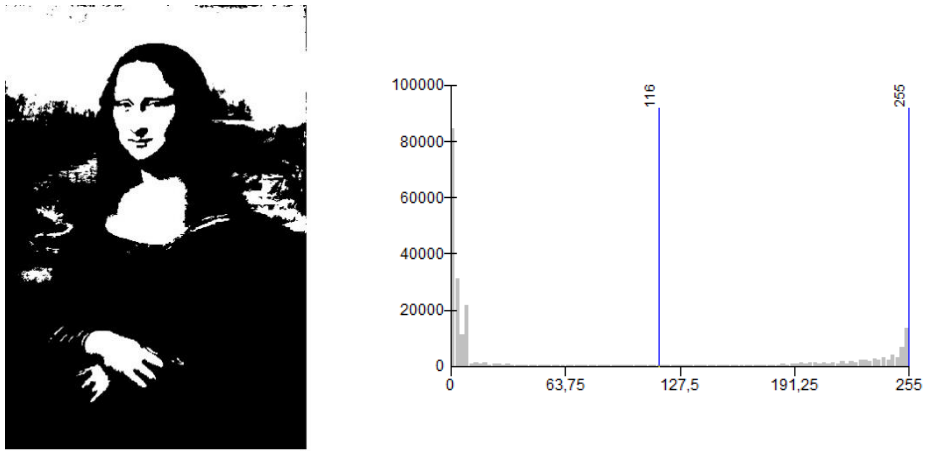


Figure 6. Classified frequency image and its pixel values frequency histogram.

We calculate the RMSE index of the segmented image obtained in a band with respect to the correspondent segmented image obtained applying FCM on the source image in that band. To

perform these measures, we decompress the segmented images obtained by executing the FTbrFCM algorithm using the bidimensional inverse F-transform.

In Table 1 are shown the RMSE measures obtained in each band changing the compression rate.

Table 1. Root mean square error (RMSE) measures for the image Mona Lisa.

Compression Rate	Band	RMSE
0.563	R	1.58
	G	1.52
	B	1.65
0.250	R	1.69
	G	1.60
	B	1.87
0.111	R	1.94
	G	1.88
	B	2.16
0.063	R	2.18
	G	2.01
	B	2.41
0.028	R	2.39
	G	2.26
	B	2.64
0.016	R	3.47
	G	3.36
	B	3.69
0.004	R	4.51
	G	4.38
	B	4.72
0.001	R	5.95
	G	5.71
	B	6.46

The trend of RMSE varying the compression rate is shown in Figure 7.

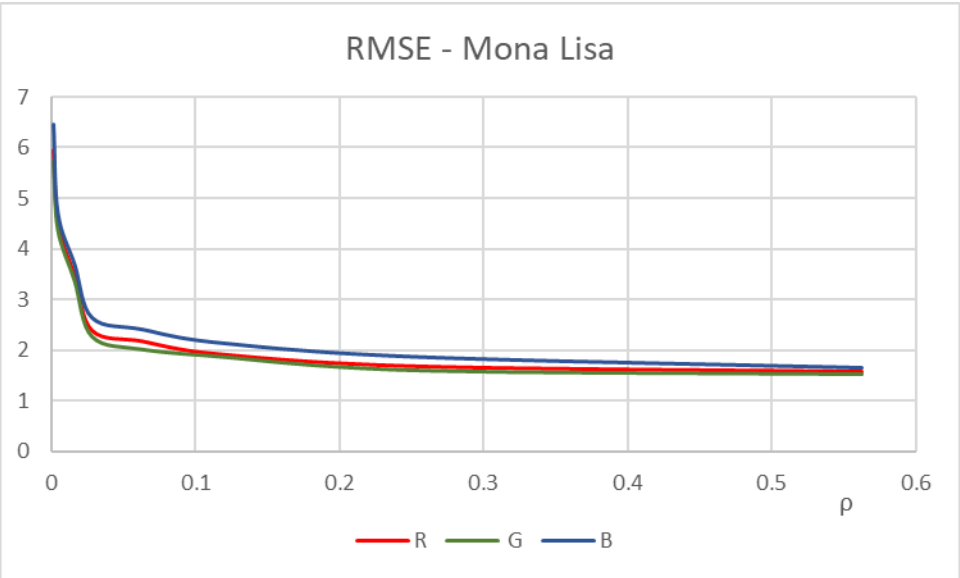


Figure 7. Mona Lisa—RMSE trend in the three band.

This results show that for not high compressions (ρ greater than 0.016), the RMSE index is less than 3, i.e., the average difference of the pixel values between the segmented image obtained using the FTbrFCM algorithm and the corresponding one obtained using the FCM algorithm is less than 3, therefore the loss of information due to the compression of the source image can be considered acceptable. In fact, if the mean square error obtained is less than 3 the average absolute difference between the membership degree of a pixel to the cluster obtained with the two algorithms is less than $3/255 \approx 1.2 \times 10^{-2}$ and the loss of information can be neglected. For strong compressions ($\rho < 0.1$), however, the RMSE value rises rapidly and the average differences between the membership degree values assigned to each pixel become significant.

In Figure 8a–d are shown the image Sunflowers and its decomposition in the three bands R, G and B.

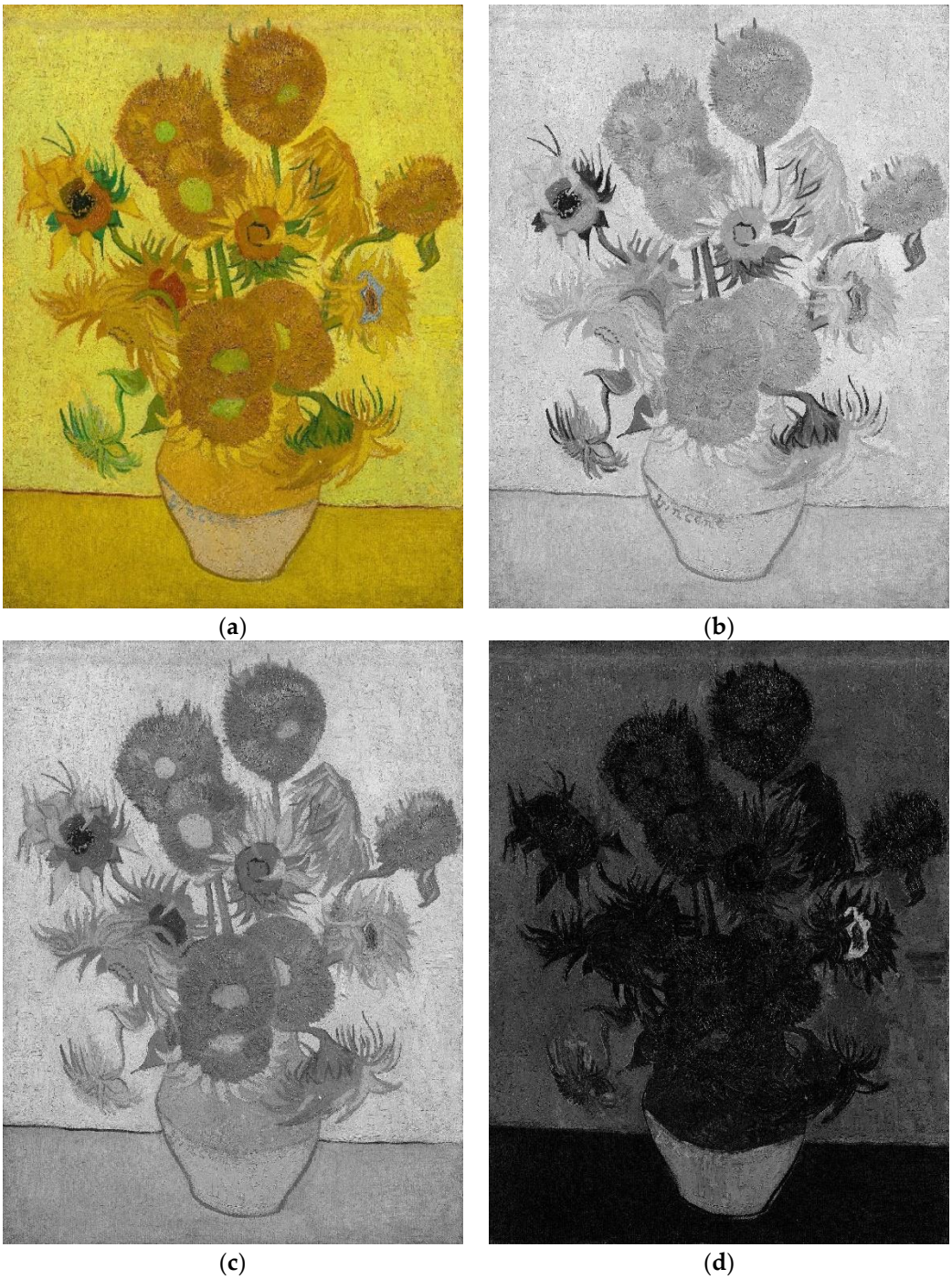


Figure 8. (**a**) Sunflowers; (**b**) Sunflowers R Band; (**c**) Sunflowers G Band; (**d**) Sunflowers B Band.

This image in each band is compressed using various compression rates and segmented setting the number of clusters (C) to 3. In Figures 9–11 shown the segmented images obtained using a compression rate = 0.25, respectively, in the R, G and B bands.

Figure 9. Sunflowers R Band—compression rate 0.25—segmented images.

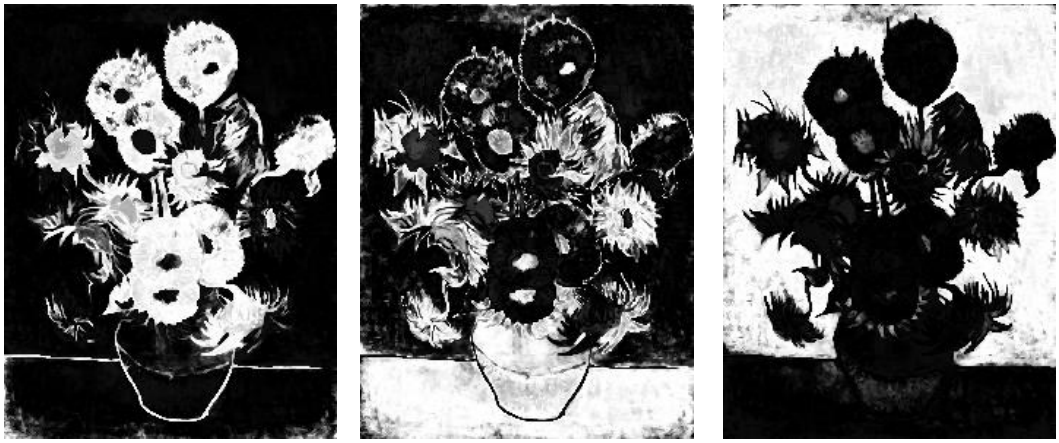


Figure 10. Sunflowers G Band—compression rate 0.25—segmented images.

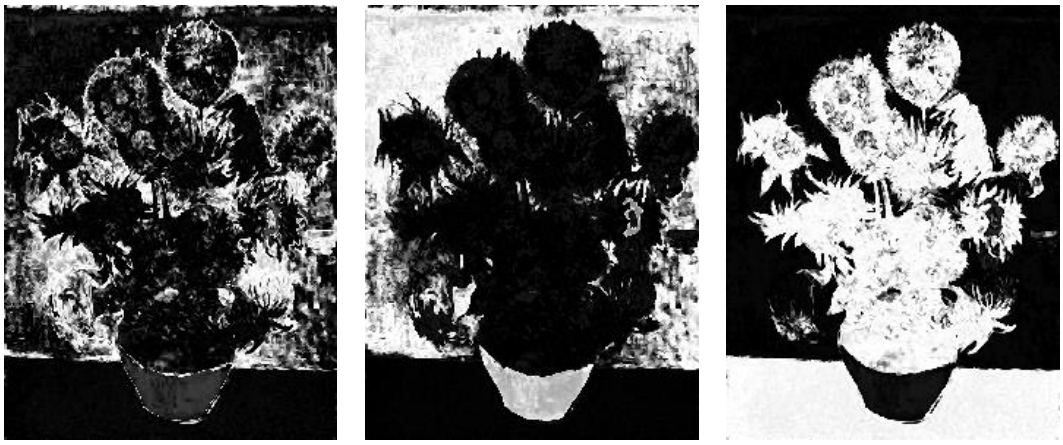


Figure 11. Sunflowers B Band—compression rate 0.25—segmented images.

To measure the RMSE index of each segmented image with respect to the correspondent one obtained executing FCM on the source image, we decompress the segmented image via the bidimensional inverse F-transform.

In Table 2 are shown the RMSE measures obtained in each band changing the compression rate.

Table 2. RMSE measures for the image Sunflowers.

Compression Rate	Band	RMSE
0.563	R	1.78
	G	1.80
	B	1.85
0.250	R	2.05
	G	2.06
	B	2.32
0.111	R	2.56
	G	2.58
	B	2.63
0.063	R	2.77
	G	2.76
	B	2.86
0.028	R	3.16
	G	3.14
	B	3.25
0.016	R	4.25
	G	4.24
	B	4.50
0.004	R	6.19
	G	6.21
	B	6.46
0.001	R	7.41
	G	7.44
	B	8.12

The trend of the RMSE index in the three bands is shown in Figure 12.

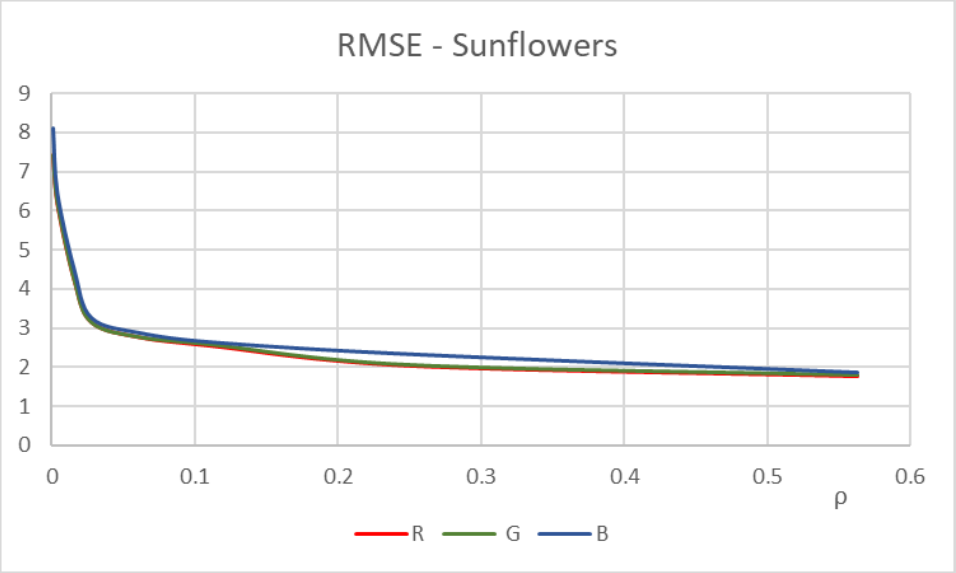


Figure 12. Sunflowers—RMSE trend in the three band.

The results in Figure 13 show that for compression rates greater or equal to 0.063 the value of the RMSE index is less than 3 in each band and the segmented images obtained executing FtbrFCM are comparable with the ones obtained executing FCM on the source images; conversely, for compression rates less than 0.063 the RMSE index rises rapidly and the loss of information due to compression becomes significant.

In Figure 12 is shown the mean trend of the RMSE measured for all the images in the dataset in the three bands varying the compression rate.

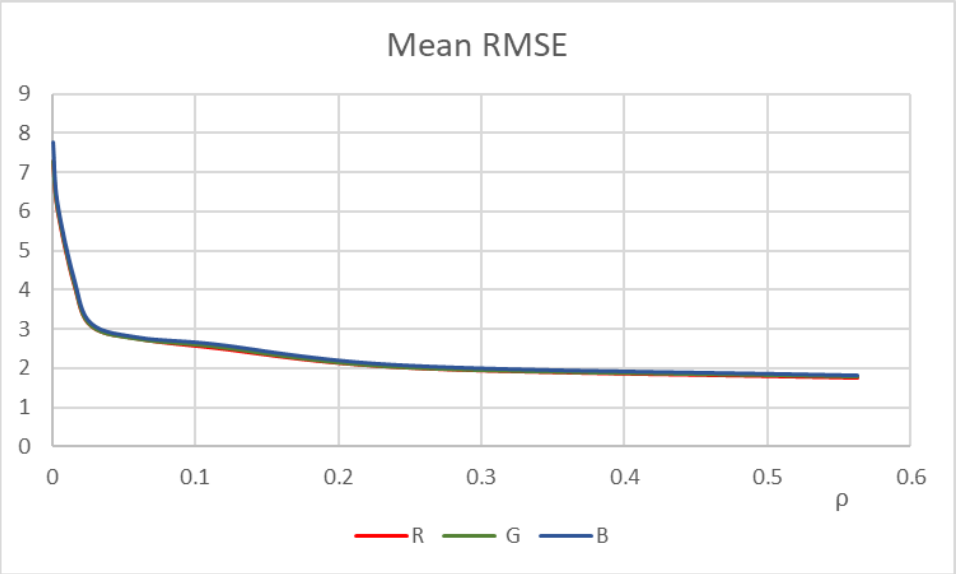


Figure 13. Mean RMSE trend in the three band.

For a compression rate ρ less than 0.11 the mean RMSE is greater than 3 and increases exponentially for higher compression. For compression rates greater than 0.11 the mean RMSE is less than 3 and the results obtained running FTbrFCM are comparable with the ones obtained running FCM on the source images.

To compare the performances of FTbrFCM also with other image segmentation methods in literature we measure also the RMSE of FTbrFCM with respect to two fast FCM image segmentation variations called fuzzy generalized fuzzy c-means (for short FGFCM) [16] and improved intuitionistic fuzzy c-means (for short IIFCM) [17]; both these two FCM-based algorithms incorporate local spatial information considering spatial relations between near pixels and are more robust to noise than FCM improving its performances.

Figure 14 shows the trend of the mean RMSE calculated in any band varying the compression rate with respect to FCM, FGFCM and IIFCM.

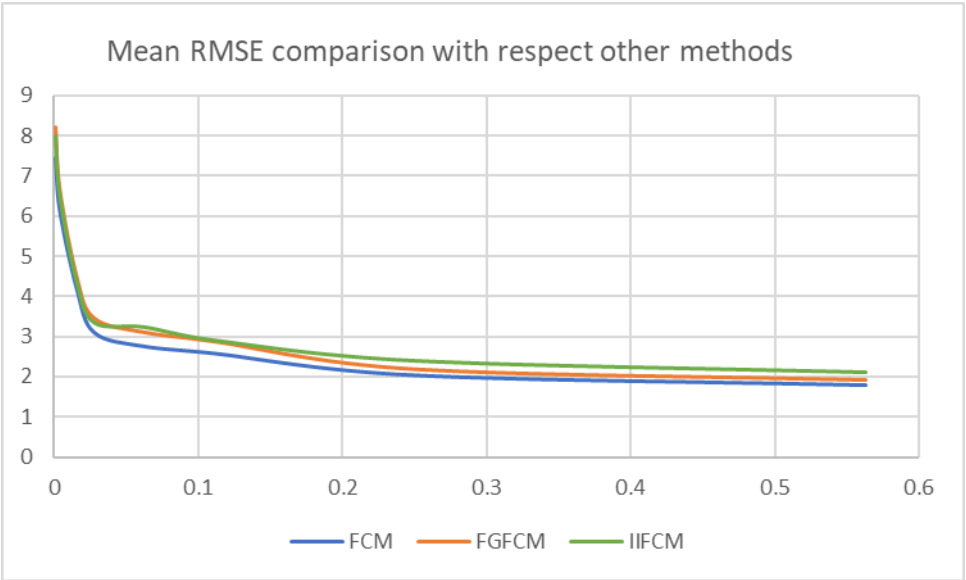


Figure 14. Mean RMSE trend with respect to FCM, fuzzy generalized fuzzy c-means (FGFCM) [16] and improved intuitionistic fuzzy c-means (IIFCM).

Even if the average RMSE obtained with respect to FGFCM and IIFCM is greater than the average RMSE obtained with respect to FCM as the compression rate changes, the average RMSE obtained with respect to FGFCM and IIFCM remains below the threshold 3 for compression rates not lower than 0.1. These results show that for not substantial compression ($\rho \geq 0.1$) the quality of the segmented images obtained executing FTbrFCM is also comparable with the ones obtained executing. FGFCM and IIFCM.

In Figure 15 is shown the mean runtime in seconds, varying the compression rate. The run time measured where $\rho = 1$ is the one obtained executing FCM on the source image.

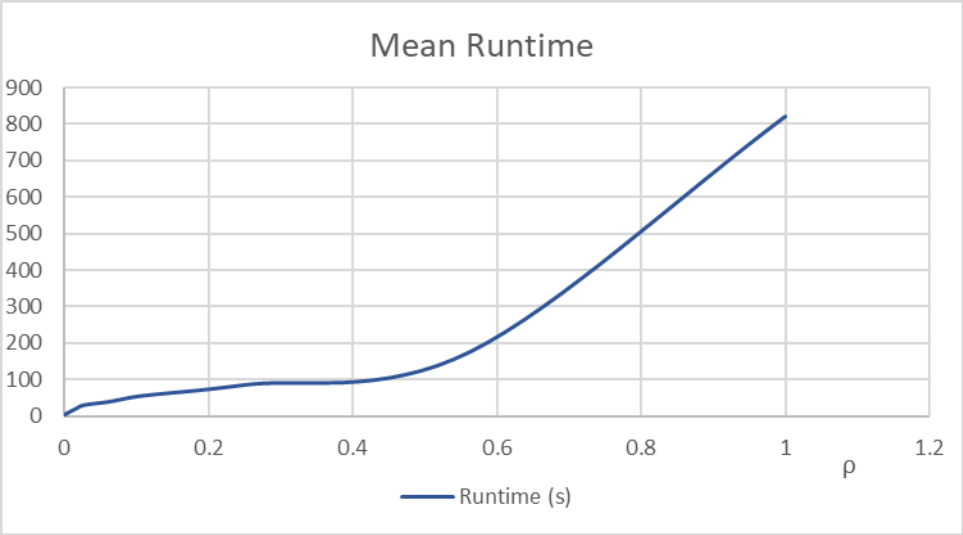


Figure 15. Mean runtime trend.

Figure 15 show that for $\rho \leq 0.25$ the runtimes are less than 1/8 of the runtime of FCM applied on the source image. Since in our tests for all the images and in all bands using compression rates greater than $\rho = 0.11$ the RMSE is less than 3, we deduce that using compression rates $\rho = 0.11$ and $\rho = 0.25$ all the segmented images are comparable with the ones obtained by executing FCM, and the runtimes are on average reduced to 1/8 with respect to the ones obtained executing FCM.

6. Conclusions

In order to handle massive data in image segmentation, we propose a bit reduced FCM algorithm applied on images compressed by bi-dimensional F-transforms. To perform the compression of the image, the block F-transform compression method is used in which each block of the image is acquired sequentially and compressed; when all the blocks have been compressed the compressed image is reconstructed and is binned by merging all adjacent pixels with the same gray value in a bin. FTbrFCM is tested on a set of high definition color images; the results show that for not excessively high compressions ($\rho \geq 0.11$) the results are comparable with the ones obtained applying FCM and other FCM-based more robust image segmentation algorithms on the source images; in addition, for $\rho \leq 0.25$ the runtimes are not exceeding one eighth of the runtimes measured using FCM on the source images.

In the future, we intend to further increase the performance of this method by exploring the use of variations of the FCM algorithm that are more robust than the presence of noise in the data and the initialization of clusters, to be applied to binned datasets for the segmentation of the compressed image. In addition, we intend to carry out further tests considering different types of massive image data such as high-resolution multiband satellite data and high-resolution diagnostic images used in many medical fields, varying the size of the image.

Author Contributions: Conceptualization, B.C. and F.D.M.; methodology, B.C. and F.D.M.; software, B.C. and F.D.M.; validation, B.C. and F.D.M.; formal analysis, B.C. and F.D.M.; investigation, B.C. and F.D.M.; resources, B.C. and F.D.M.; data curation, B.C. and F.D.M.; writing—original draft preparation, B.C. and F.D.M.; writing—

review and editing, B.C. and F.D.M.; visualization, B.C. and F.D.M.; supervision, B.C. and F.D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest

References

1. Dunn, C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. Cybern.* **1973**, *3*, 32–57.
2. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press: New York, NY, USA, 1981; p. 272.
3. Eschrich, S.; Ke, L.; Hall, L.; Goldgof, D. Fast accurate fuzzy clustering through data reduction. *IEEE Trans. Fuzzy Syst*. **2003**, *11*, 262–269.
4. Havens, T.C.; Bezdek, J.C.; Leckie, C.R.; Hall, L.O.; Palaniswami, M. Fuzzy C-means algorithms for very large data. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 1130–1146.
5. Di Martino, F.; Sessa, S. Extended Fuzzy C-Means Hotspot Detection Method for Large and Very Large Event Datasets. *Inf. Sci.* **2018**, *441*, 198–215.
6. Perfilieva, I. Fuzzy transforms. *Fuzzy Sets Syst.* **2006**, *157*, 993−1023.
7. Di Martino, F.; Sessa, S. Compression and decompression of images with discrete fuzzy transforms. *Inf. Sci.* **2007**, *177*, 2349−2362.
8. Di Martino, F.; Loia, V.; Sessa, S. An image coding/decoding method based on direct and inverse fuzzy trans-forms. *Int. J. Approx. Reason.* **2008**, *48*, 110−131.
9. Di Martino, F.; Loia, V.; Sessa, S. A segmentation method for images compressed by fuzzy transforms. *Fuzzy Sets Syst.* **2010**, *161*, 56–74.
10. Kaufman, L.; Rousseeuw, P. *Finding Groups in Data: An Introduction to Cluster Analysis*; Wiley-Blackwell: New York, NY, USA, 2005; p. 342.
11. Hathaway, R.; Hu, Y. Density-Weighted Fuzzy c -Means Clustering. *IEEE Trans. Fuzzy Syst.* **2008**, *17*, 243–252.
12. Ji, Z.; Xia, Y.; Chen, Q.; Sun, Q.-S.; Xia, D.; Feng, D.D. Fuzzy c-means clustering with weighted image patch for image segmentation. *Appl. Soft Comput.* **2012**, *12*, 1659–1667.
13. Gong, M.; Liang, Y.; Shi, J.; Ma, W.; Ma, J. Fuzzy C-Means Clustering with Local Information and Kernel Metric for Image Segmentation. *IEEE Trans. Image Process.* **2012**, *22*, 573–584.
14. Nadalin, E.Z.; Silva, R.C.; Attux, R.; Romano, J.M.T. Analysis of the Weighted Fuzzy C-means in the Problem of Source Location. In Proceedings of the ESANN 2014, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 23–25 April 2014; pp. 219–224.
15. Xie, X.L.; Beni, G. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell*. **1991**, *13*, 841–847.
16. Cai, W.; Chen, S.; Zhang, D. Fast and robust fuzzy c-means clustering algorithm incorporating local information for image segmentation. *Pattern Recognit.* **2007**, *40*, 825–838.
17. Verma, H.: Agrawal, R.K.; Sharan, A. An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation, *Appl. Soft Comput.* **2016**, *46*, 543–557.