Université de Lille

Fuzzy Logic for Image Processing

« What men really want is not knowledge but certainty. »
Bertrand Russel

Application to image processing

- Segmentation of colour images
- Based on clustering techniques
  - Partition of a population (collection of data described by a set of features)
  - Assignment of each sample (data) to a cluster
- Some classical algorithms:
  1. HCM (Hard C-Means ; not based on fuzzy logic);
  2. FCM (Fuzzy C-Means);
  3. PCM (Possibilistic C-Means);
  4. Davé's algorithm.

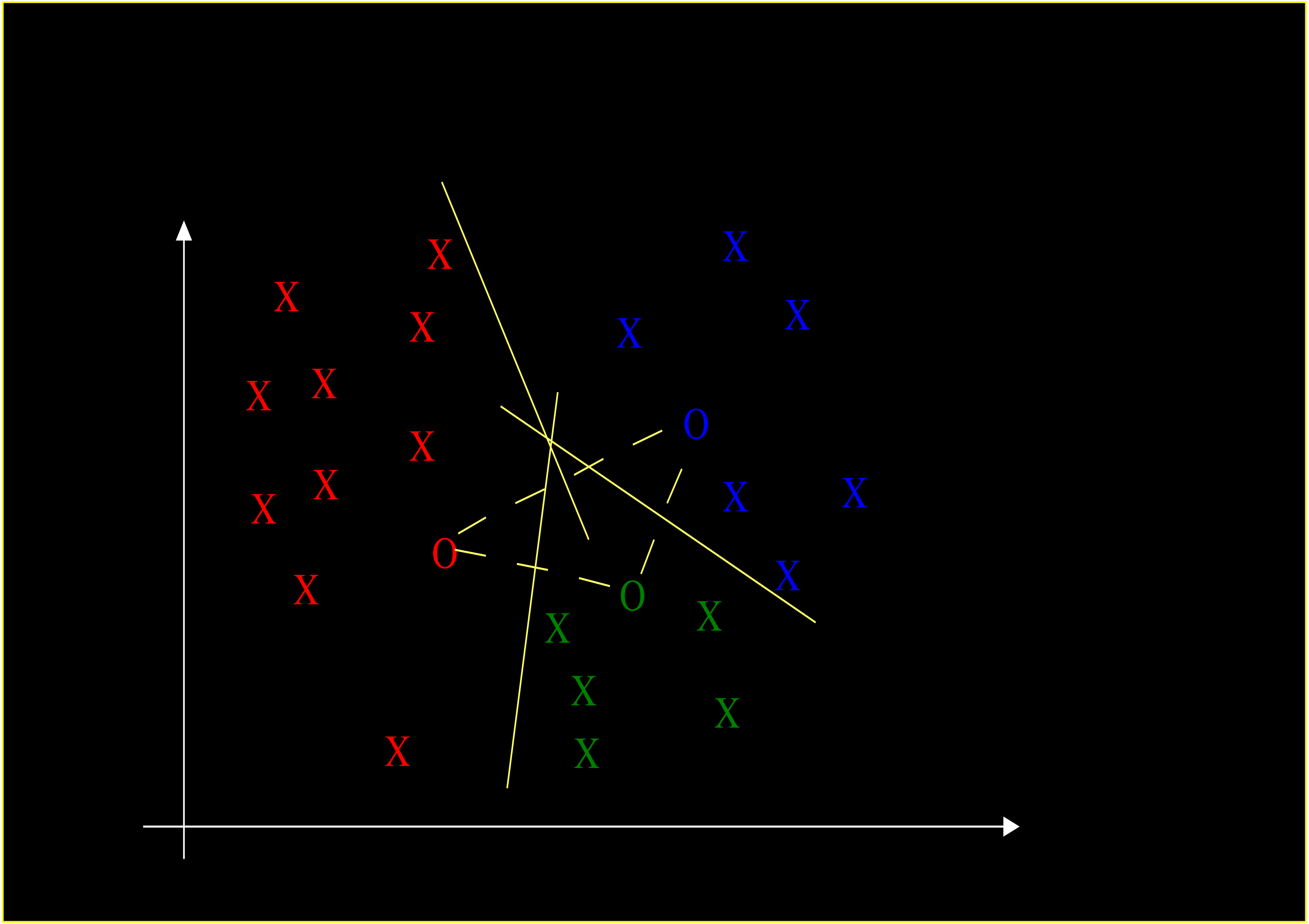
A basic approach

- C-Means algorithm = a clustering method (1967).
- Aim:
  - Partition of a population (collection of data described by a set of features)
  - Assignment of each sample (data) to a cluster
- C-Means algorithm is not a fuzzy logic-based method.

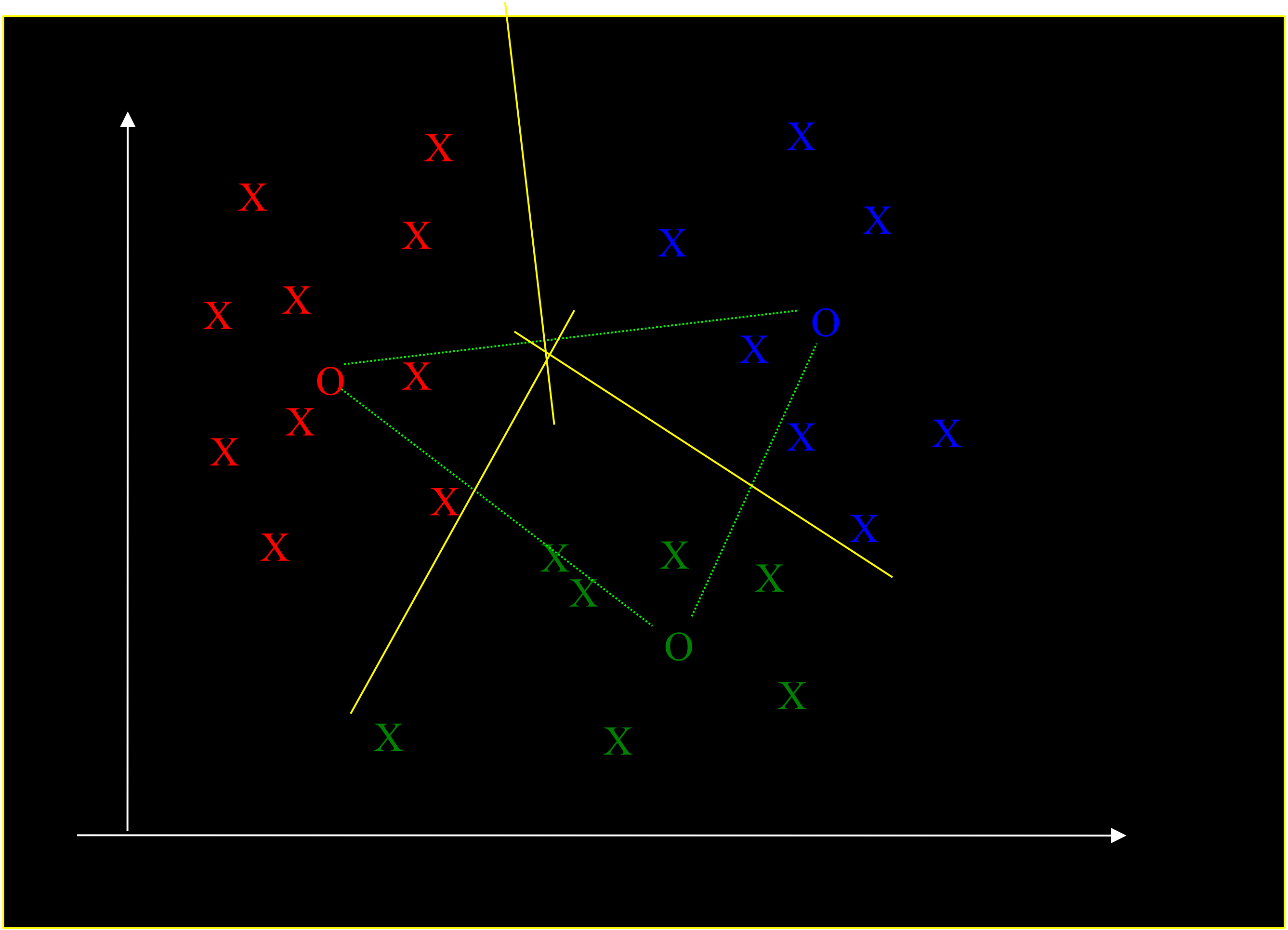
C-means algorithm

- Principle of the C-means algorithm
  - Partition of a population (collection of data described by a set of features)
  - Assignment without ambiguity ($\in$ or $\notin$) of each sample (data) to a cluster
- Algorithm:
  1. Ramdom selection of *c* samples: **centroïds.**
  2. Assignment of every sample at the closest centroïd (using a distance). Constitution of the clusters.
  3. Calculation of new centroïds: we take the mean, component by component, for all the samples of a cluster.
  4. Back to step #2 until stabilization of the borders between the clusters.

C-means: step #1



C-means: final step



Method of C-means

- Drawbacks:
  - Sensitive to the initialization
  - Problems when considering non-digital variables (required to possess a measure of distance)
    - Translation in numerical values
    - Construction of matrices of distances
  - Problem of the choice of the number of centroïds c
  - Problem of the choice of the normalization in the calculation of the distance (the same weight for every component)
    - Weighting factors, normalization, aggregation

FCM (Fuzzy C-Means)

- Generalization of the C-means algorithm
  - Fuzzy partition of the data
  - Membership functions to the clusters
- Problematic: find a fuzzy pseudo-partition and the centers of the associated clusters which better represents the structure of the data.
  - Use of a criterion allowing ensure the strong association within the cluster and a low association outside the cluster.
    - **Performance index**

FCM (Fuzzy C-Means)

- **Fuzzy pseudo-partition**
  - Set of non-empty fuzzy subsets $\{A_1, A_2, \ldots, A_c\}$

  Set of data (vector of k components): $X = \{x_1, \ldots, x_n\}$

  $$\forall x_j \in X = \{x_1, \ldots, x_n\}, \sum_{i=1}^{c} \mu_{A_i}(x_j) = 1$$

- **Fuzzy C-partition**
  - A fuzzy c-partition (c>0) of *X* is a family of c fuzzy subsets such as:

  $$P = \{A_1, A_2, \ldots, A_c\}$$

  $$\forall x_j \in X = \{x_1, \ldots, x_n\}, \sum_{i=1}^{c} \mu_{A_i}(x_j) = 1$$

$$\forall i \in \{1, 2, \ldots, c\}, \forall j \in \{1, 2, \ldots, n\} \quad \mu_{A_i}(x_j) \in [0; 1] \quad 0 < \sum_{j=1}^{n} \mu_{A_i}(x_j) < n$$

FCM (Fuzzy C-Means)

Let be $X = \{x_1, x_2, \ldots, x_n\}$ a set of data.

Each x_j can be a vector of features, i.e. $x_j = \left\{x_{j,1}, x_{j,2}, \ldots, x_{j,k}\right\}^t$.

Let $P = \{A_1, A_2, \ldots, A_c\}$ a fuzzy partition of the data set.

The centroïds (prototypes) $\nu_1, \nu_2, \ldots, \nu_c$ associated to the fuzzy partition are computed as it follows:

$$\forall i \in \{1, 2, \ldots, c\}, \qquad \nu_i = \frac{\sum_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m . x_j}{\sum_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m} = \frac{\sum_{j=1}^{n} u_{ij}^m . x_j}{\sum_{j=1}^{n} u_{ij}^m}$$

with $m \in \mathbb{R}, m > 1$, influence of the membership degrees (typically, $m = 2$).

U : matrix of the membership degrees of dimension $c \times n$

ν_i: center of the fuzzy cluster A_i

- weighted mean of the data in A_i
- The weight of data x_j is the mth power of its membership degree to A_i.

FCM (Fuzzy C-Means)

Computation of the membership degrees:

$$\forall i \in \{1, 2, \ldots, c\}, \quad u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{d^2(x_j, \nu_i)}{d^2(x_j, \nu_k)} \right)^{\frac{2}{m-1}} \right]^{-1}$$

FCM (Fuzzy C-Means)

Performance index of a fuzzy partition

Performance index of P:

$$J_{FCM}(P) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m \|x_j - \nu_i\|^2 = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m .d_{ij}^2$$

$\|\|$: norm on $\mathbb{R}^k$

Lower is $J(P)$, better is P.

- The index of performance is an objective function. Its aim is to optimize the data partition in *c* clusters.
- The algorithm is iterative. Several iterations are made until obtaining a stable partition of the data (minimization of $J_{FCM}(P)$).

FCM (Fuzzy C-Means)

Algorithme du FCM :

1. Choisir le nombre de classes : c // Information à priori, algorithme supervisé.

2. Initialise la matrice de partition U, ainsi que les centres c_k (initialisation aléatoire) ;

3. Faire évoluer la matrice de partition et les centres suivant les deux équations :

(1) $u_{ik} = 1 \,/\, \left(\sum_{j=1,c} (d_{ik} / d_{ij})^{(2/(m-1))} \right)$, // mise à jour des degrés d'appartenances,

où : $d_{ij} = ||x_i - c_j||$,

(2) $c_k = (\Sigma_i (u_{ik})^m . x_i) \,/\, (\Sigma_i (u_{ik})^m)$, // mise à jours des centres.

4. Test d'arrêt : $|J^{(t+1)} - J^{(t)}| <$ seuil .

FCM (Fuzzy C-Means)

Some comments

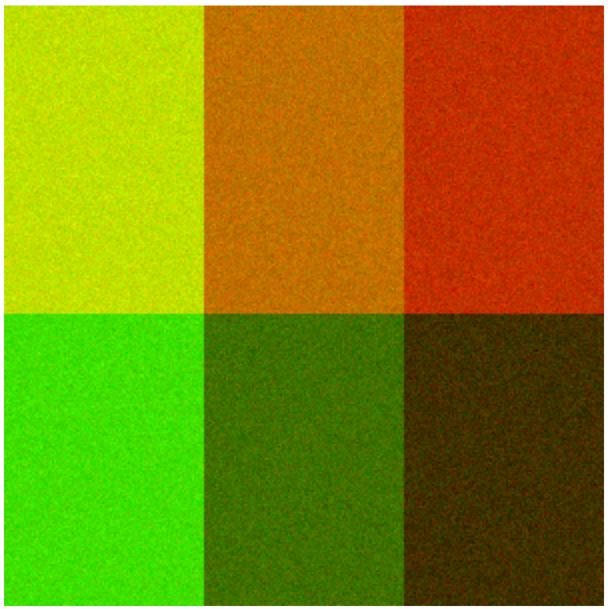
- FCM algorithm minimizes a weighted sum of the squared distances between vectors to group together and the centers of the clusters.

- The membership degree of any element (vector) to a given cluster has to be all the more raised that the vector is a typical element of the cluster.

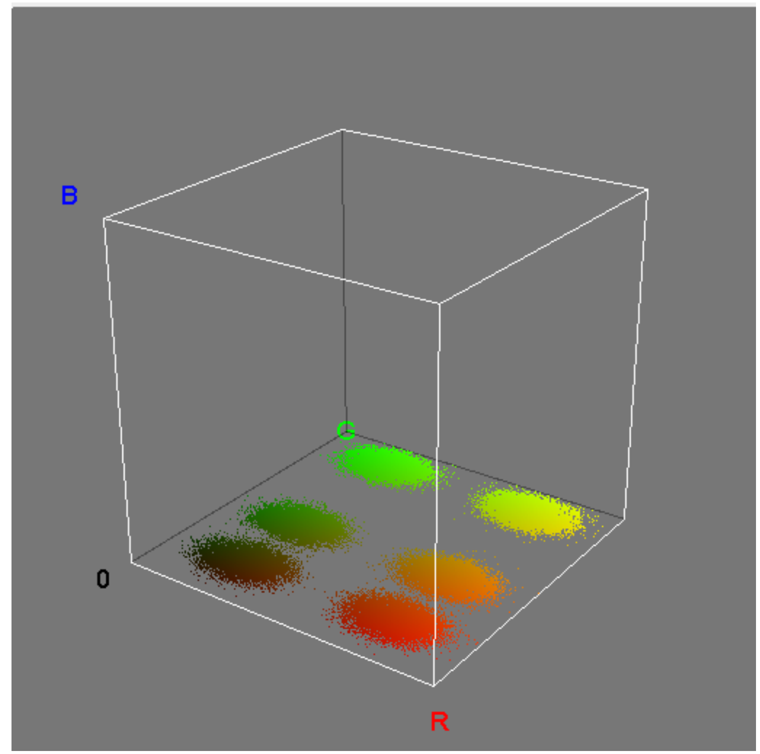
- Gustafson and Keller have proposed a modified version of FCM for non-spherical distributions of data.

FCM (Fuzzy C-Means)

Example



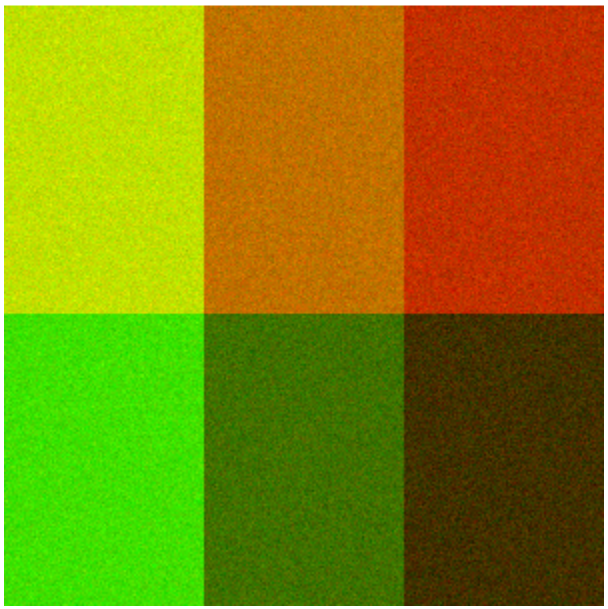
Image



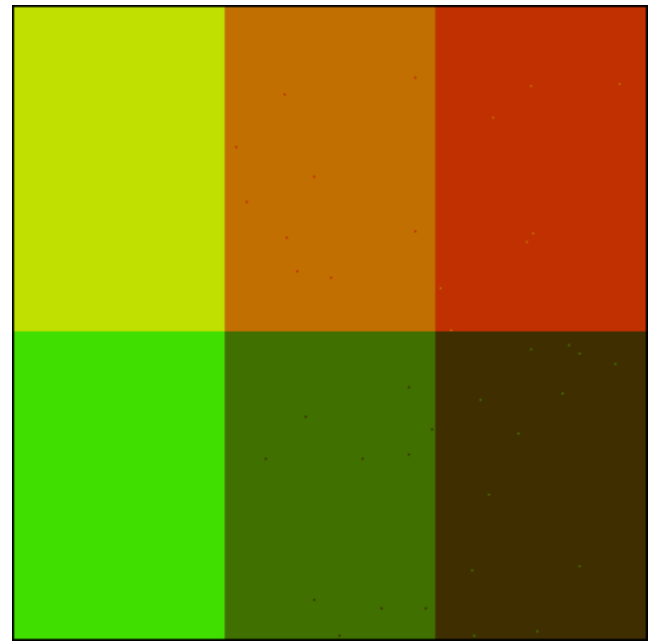
RGB color space

FCM (Fuzzy C-Means)

Example



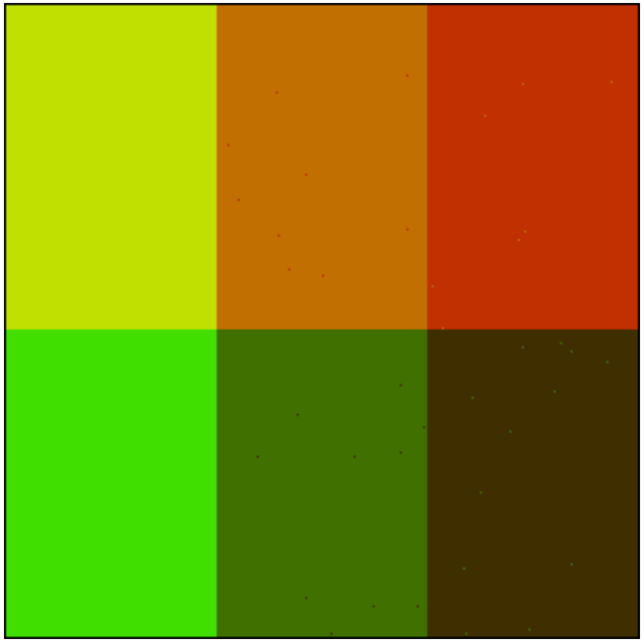
Image



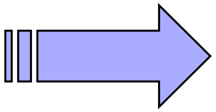
Segmented image

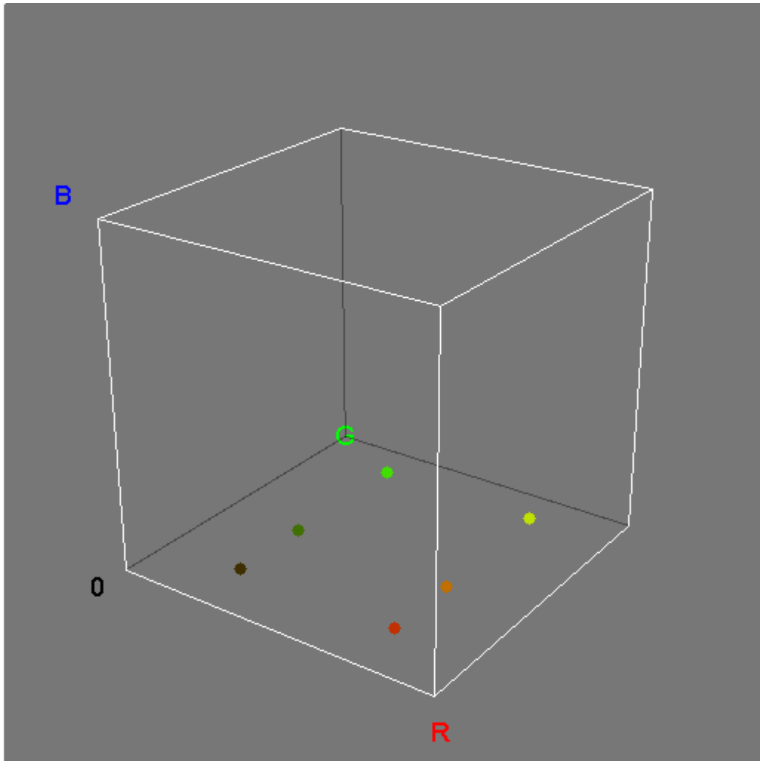
FCM (Fuzzy C-Means)

Example



Segmented image

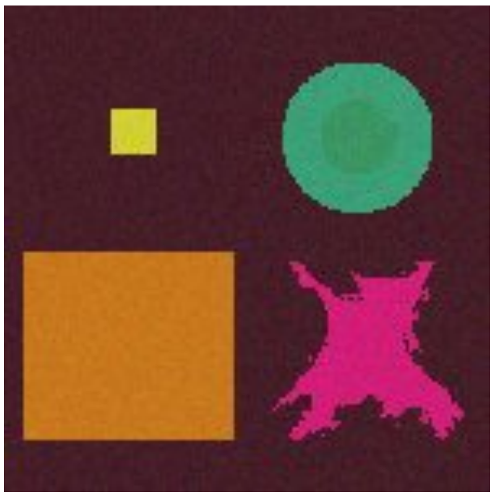




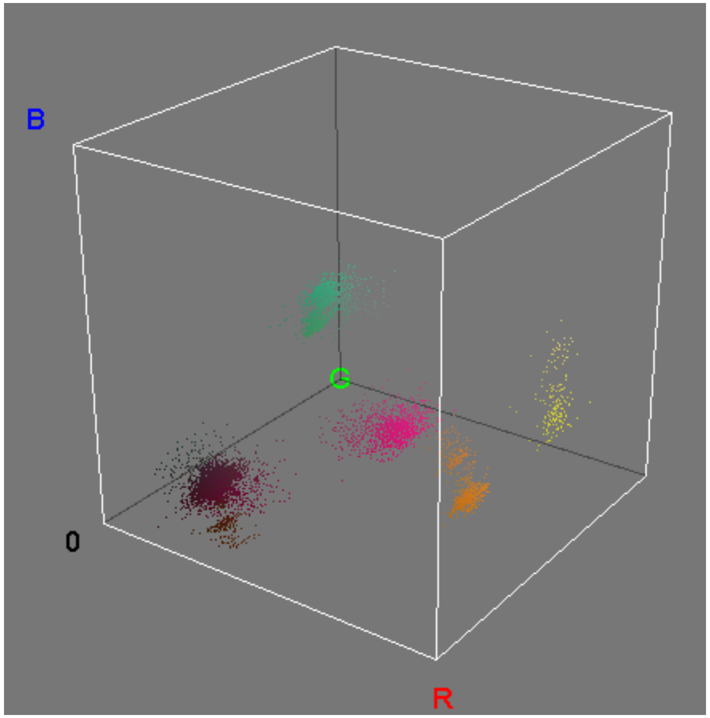
RGB color space

FCM (Fuzzy C-Means)

Example



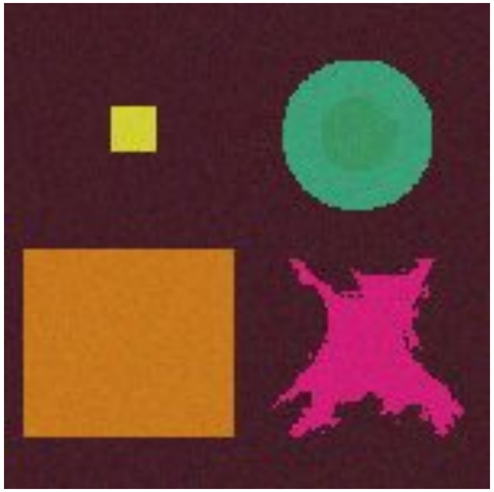
Image



RGB color space

FCM (Fuzzy C-Means)

Example

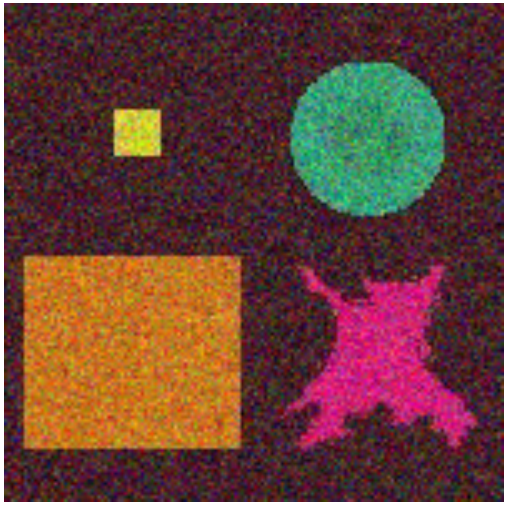


Image

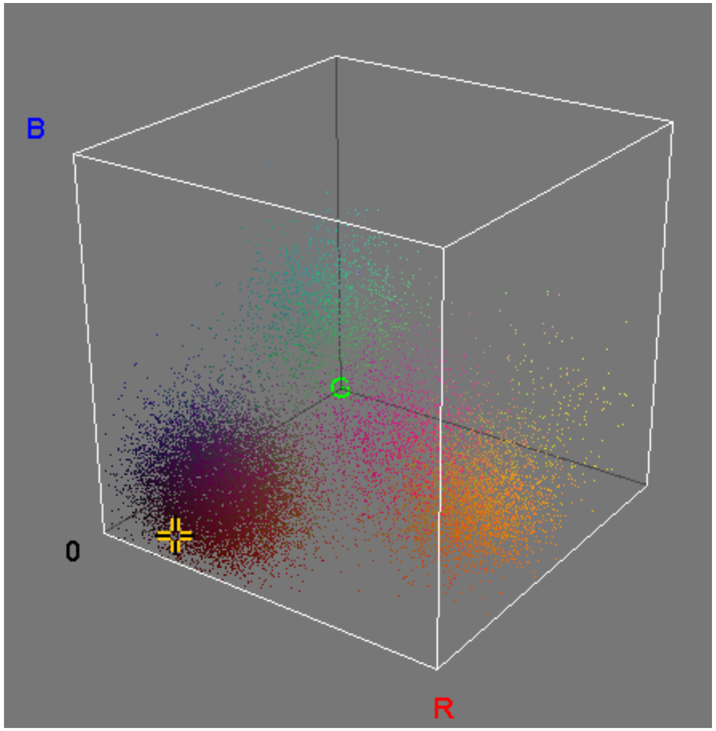
Segmented image

FCM (Fuzzy C-Means)

Example



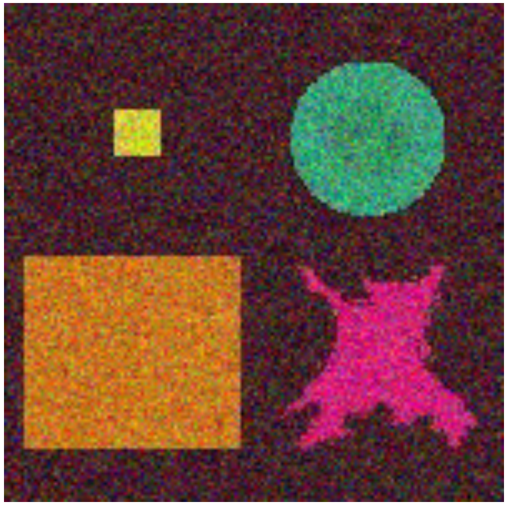
Noisy Image



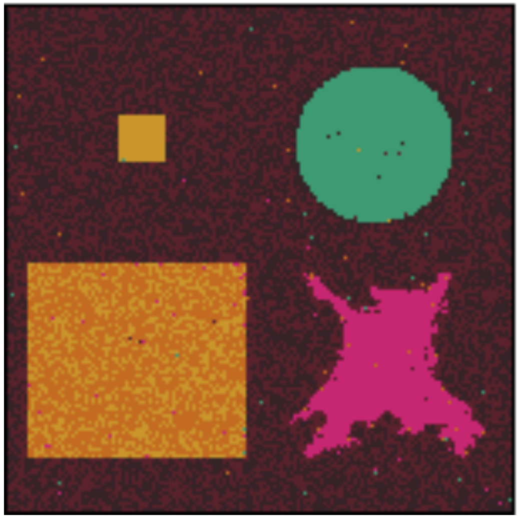
RGB color space

FCM (Fuzzy C-Means)

Example



Noisy Image



Segmented image

HCM (Hard C-Means)

Some steps backward

Let be $X = \{x_1, x_2, \ldots, x_n\}$ a set of data.

Each x_j can be a vector of features, i.e. $x_j = \left\{x_{j,1}, x_{j,2}, \ldots, x_{j,k}\right\}^t$.

Let $P = \{A_1, A_2, \ldots, A_c\}$ a partition of the data set.

The centroïds (prototypes) $\nu_1, \nu_2, \ldots, \nu_c$ associated to the fuzzy partition are computed as it follows:

$$\forall i \in \{1, 2, \ldots, c\}, \qquad \nu_i = \frac{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m .x_j}{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m} = \frac{\sum\limits_{j=1}^{n} u_{ij}^m .x_j}{\sum\limits_{j=1}^{n} u_{ij}^m}$$

HCM (Hard C-Means)

Some steps backward

$$\forall i \in \{1, 2, \ldots, c\}, \quad \nu_i = \frac{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m .x_j}{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m} = \frac{\sum\limits_{j=1}^{n} u_{ij}^m .x_j}{\sum\limits_{j=1}^{n} u_{ij}^m}$$

Computation of the membership degrees:

$$\forall i \in \{1, 2, \ldots, c\}, \forall j \in \{1, 2, \ldots, n\} \quad u_{ij} = \begin{cases} 1 \text{ iif } d^2(x_j, \nu_i) < d^2(x_j, \nu_k) & \forall k \neq i \\ 0 \text{ otherwise} \end{cases}$$

Hard assignment: $x_j \in A_i$ or $x_j \notin A_i$

HCM (Hard C-Means)

Performance index

Performance index of P:

$$J_{HCM}(P) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m \|x_j - \nu_i\|^2 = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m .d_{ij}^2$$

$\|\|$: norm on $\mathbb{R}^k$

$m = 1$

Lower is $J(P)$, better is P.

- The index of performance is an objective function. Its aim is to optimize the data partition in *c* clusters.
- The algorithm is iterative. Several iterations are made until obtaining a stable partition of the data (minimization of $J_{FCM}(P)$).

PCM (Possibilistic C-Means)

Introduction

PCM (Possibilistic C-Means) is a variant of the FCM algorithm [Krishnapuram & Keller].

Aim: to be more robust in presence of noise.

Comments:

- The PCM algorithm aims to overcome the relative behaviour of the membership degrees provided in FCM: a vector is « shared » between the different clusters.

- Krishnapuram and Keller replace the notion of membership by the notion of typicality.

- The result of a clustering should describe the absolute relationship between a vector and each of the c clusters independently of the relationship between the vector and the (c-1) other clusters .

PCM (Possibilistic C-Means)

Details

- The membership degrees given by PCM are not relative degrees, they are absolute values reflecting the strength with which each vector belongs to all the clusters.

- The elimination of the interferences between the different prototypes needs to define a new objective function (performance index) for the optimization of the partition.

- <u>Remark</u>: only one of the membership degrees of a vector to be classified has to be not equal to zero.

PCM (Possibilistic C-Means)

Let be $X = \{x_1, x_2, \ldots, x_n\}$ a set of data.

Each x_j can be a vector of features, i.e. $x_j = \left\{x_{j,1}, x_{j,2}, \ldots, x_{j,k}\right\}^t$.

Let $P = \{A_1, A_2, \ldots, A_c\}$ a fuzzy partition of the data set.

The centroïds (prototypes) $\nu_1, \nu_2, \ldots, \nu_c$ associated to the fuzzy partition are computed as it follows:

$$\forall i \in \{1, 2, \ldots, c\}, \qquad \nu_i = \frac{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m .x_j}{\sum\limits_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m} = \frac{\sum\limits_{j=1}^{n} u_{ij}^m .x_j}{\sum\limits_{j=1}^{n} u_{ij}^m}$$

with $m \in \mathbb{R}, m > 1$, influence of the membership degrees (typically, $m = 2$).

U : matrix of the membership degrees of dimension $c \times n$

ν_i: center of the fuzzy cluster A_i

- weighted mean of the data in A_i
- The weight of data x_j is the mth power of its membership degree to A_i.

PCM (Possibilistic C-Means)

Formulas

$$P = \{A_1, A_2, \ldots, A_c\}$$

$$\forall x_j \in X = \{x_1, \ldots, x_n\}, \sum_{i=1}^{c} \mu_{A_i}(x_j) = \sum_{i=1}^{c} u_{ij} = 1$$

$$\forall i \in \{1, 2, \ldots, c\}, \forall j \in \{1, 2, \ldots, n\} \quad \mu_{A_i}(x_j) \in [0; 1]$$

$$\forall i \in \{1, 2, \ldots, c\}, \forall j \in \{1, 2, \ldots, n\}$$

$$\begin{cases} 0 < \sum_{j=1}^{n} u_{ij} < n \\ \max_i u_{ij} > 0 \end{cases}$$

PCM (Possibilistic C-Means)

Performance index

Performance index of P:

$$J_{PCM}(P) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m \|x_j - \nu_i\|^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \left[1 - u_{ij}\right]^m$$

$$J_{PCM}(P) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m .d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} \left[1 - u_{ij}\right]^m$$

A penality term which avoids the trivial solution $u_{ij} = 0 \quad \forall i$ and $\forall j$

η_i : squared distance between the center of the cluster A_i and the set of vector having a membership degree to this cluster equal to 0.5

The membership degree of a vector to a specific cluster only depends on the distance to the cluster (degree of typicality). It allows to detect absurd data (outliers).

PCM (Possibilistic C-Means)

Performance index

Performance index of P:

$$J_{PCM}(P) = \sum_{i=1}^{c}\sum_{j=1}^{n}\left[u_{ij}\right]^m .d_{ij}^2 + \sum_{i=1}^{c}\eta_i\sum_{j=1}^{n}\left[1-u_{ij}\right]^m$$

In practice: $\eta_i = \dfrac{\sum\limits_{j=1}^{n} u_{ij}^m . d_{ij}^2}{\sum\limits_{j=1}^{n} u_{ij}^m}$

Also: $\eta_i = \dfrac{\sum\limits_{x_j\in(\Pi_i)_\alpha} d_{ij}^2}{|(\Pi_i)_\alpha|}$ with $(\Pi_i)_\alpha$ an α–cut of Π_i

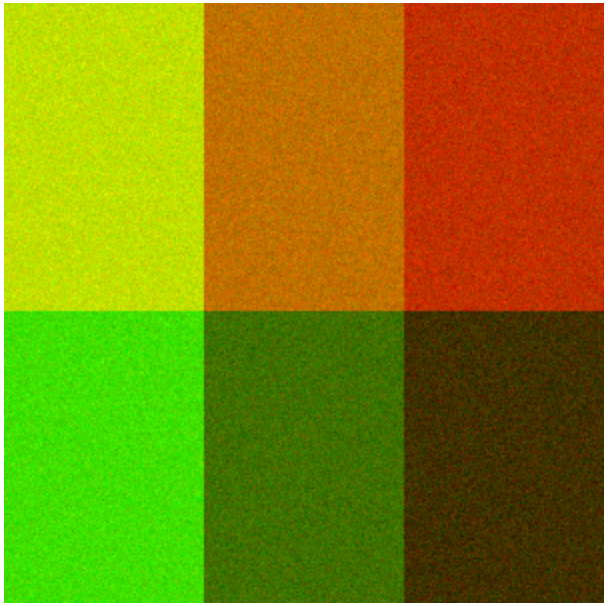
PCM (Possibilistic C-Means)

Computation of the membership degrees:

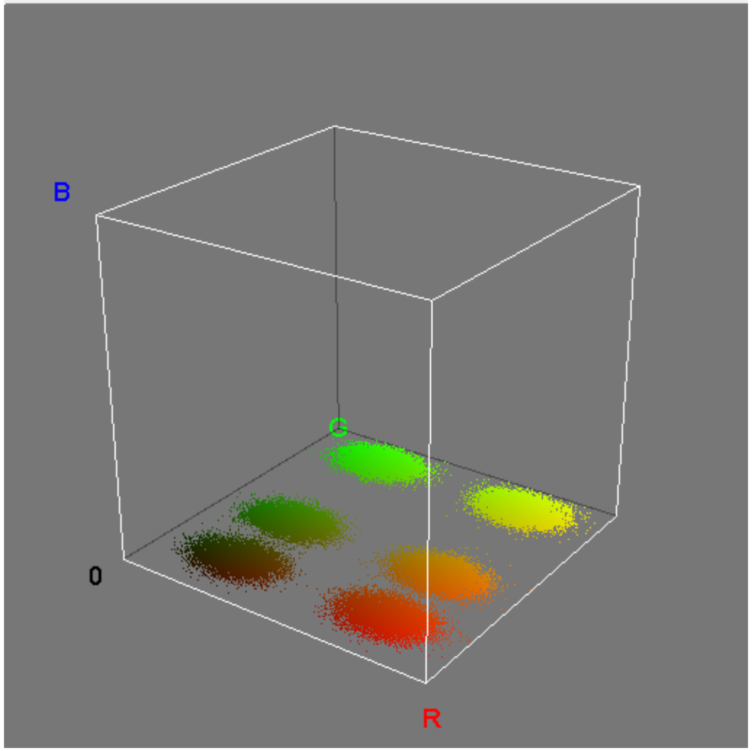
$$\forall i \in \{1, 2, \ldots, c\}, \forall j \in \{1, 2, \ldots, n\} \quad u_{ij} = \frac{1}{1 + \left(\frac{d^2(x_j, \nu_i)}{\eta_i}\right)^{\frac{1}{m-1}}} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}}$$

PCM (Possibilistic C-Means)

Example



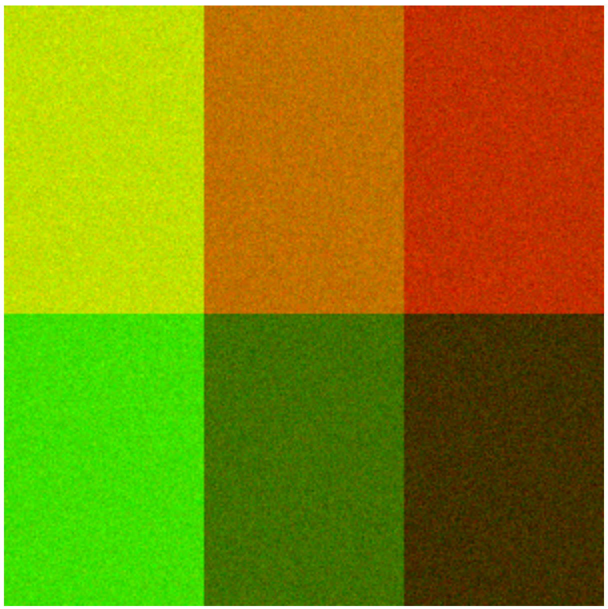
Image



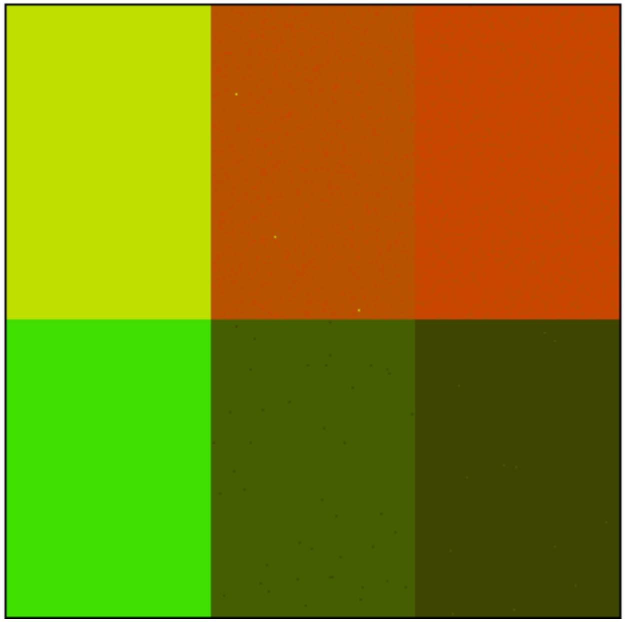
RGB color space

PCM (Possibilistic C-Means)

Example



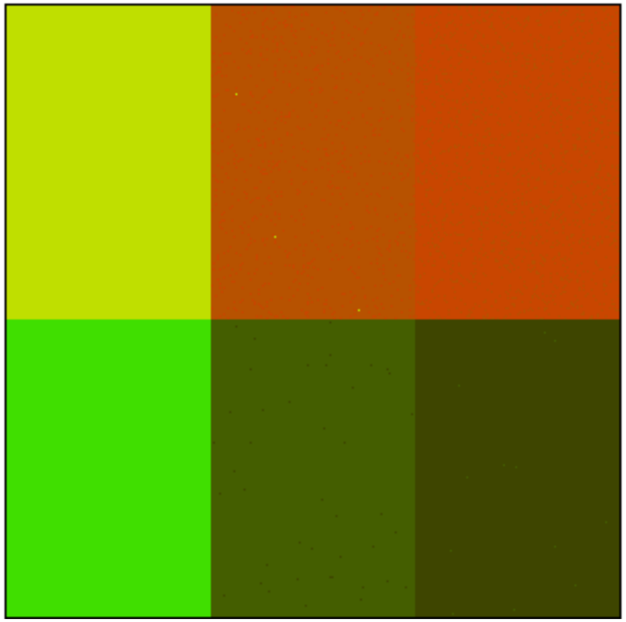
Noisy Image



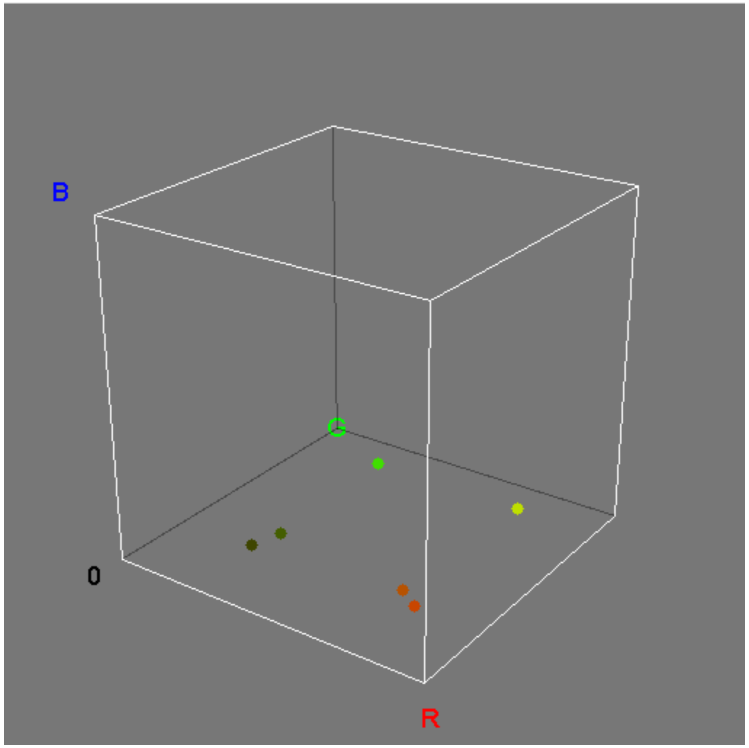
Segmented image

PCM (Possibilistic C-Means)

Example



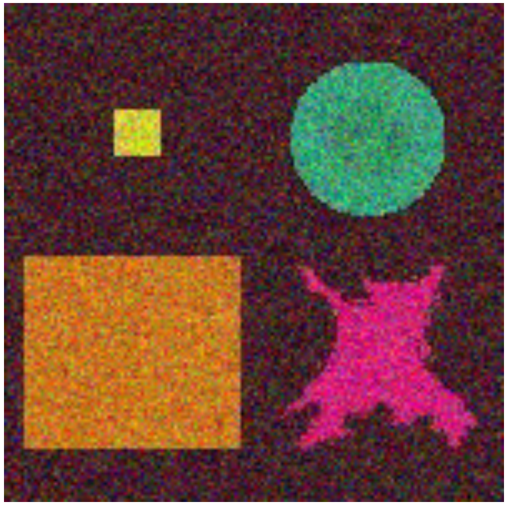
Segmented Image



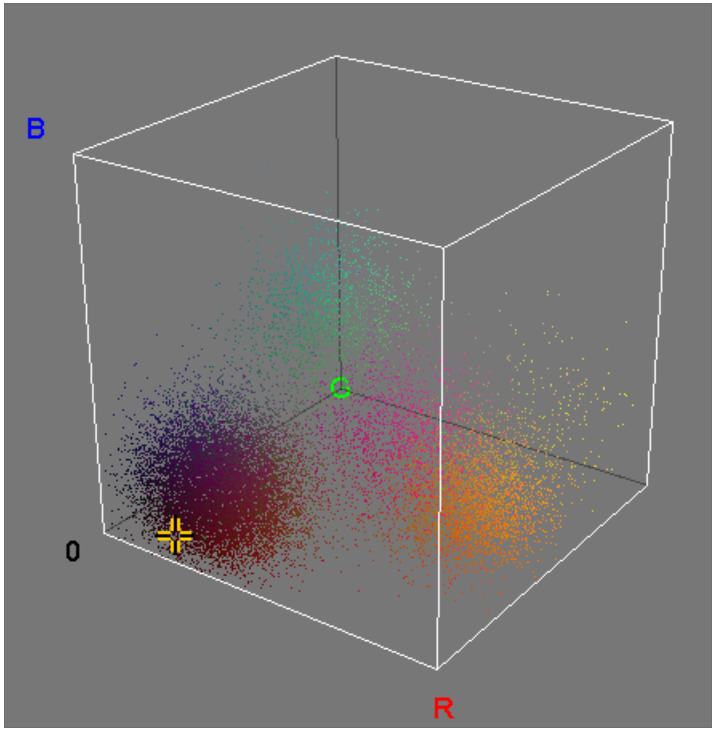
RGB color space

PCM (Possibilistic C-Means)

Example



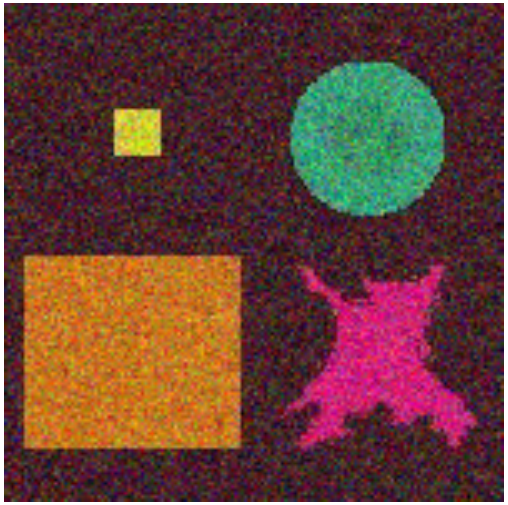
Noisy Image



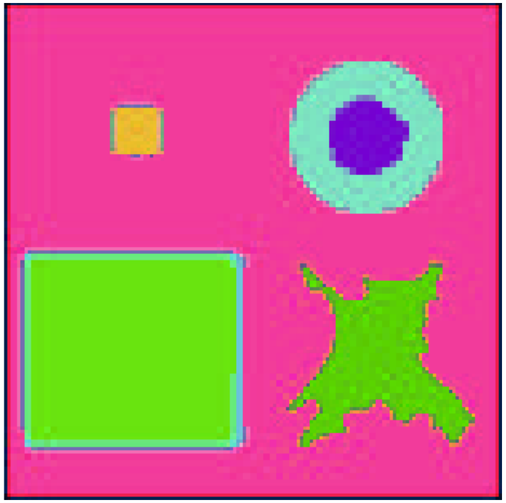
RGB color space

PCM (Possibilistic C-Means)

Example



Noisy Image



Segmented image

Davé's algorithm

Let be $X = \{x_1, x_2, \ldots, x_n\}$ a set of data.

Each x_j can be a vector of features, i.e. $x_j = \left\{x_{j,1}, x_{j,2}, \ldots, x_{j,k}\right\}^t$.

Let $P = \{A_1, A_2, \ldots, A_c\}$ a fuzzy partition of the data set.

The centroïds (prototypes) $\nu_1, \nu_2, \ldots, \nu_c$ associated to the fuzzy partition are computed as it follows:

$$\forall i \in \{1, 2, \ldots, c\}, \qquad \nu_i = \frac{\sum_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m .x_j}{\sum_{j=1}^{n} \left[\mu_{A_i}(x_j)\right]^m} = \frac{\sum_{j=1}^{n} u_{ij}^m .x_j}{\sum_{j=1}^{n} u_{ij}^m}$$

with $m \in \mathbb{R}, m > 1$, influence of the membership degrees (typically, $m = 2$).

U : matrix of the membership degrees of dimension $c \times n$

ν_i: center of the fuzzy cluster A_i

- weighted mean of the data in A_i
- The weight of data x_j is the mth power of its membership degree to A_i.

Davé's algorithm

Introduction of a "noisy" cluster (rejection option):

$$\forall j \in \{1, 2, \ldots, n\} \quad u_{\star j} = 1 - \sum_{i=1}^{c} u_{ij}$$

The cluster of noise (rejection) allows to collect outliers (absurd data) which seem to be different compared with « normal » data.

Davé's algorithm

Performance index

Performance index of P :

$$J_{Dav}(P) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[u_{ij}\right]^m \|x_j - \nu_i\|^2 + \sum_{j=1}^{n} \delta^2 \left(1 - \sum_{i=1}^{c} u_{ij}\right)^m$$

δ : a fixed distance of the cluster of noise to all the vectors.

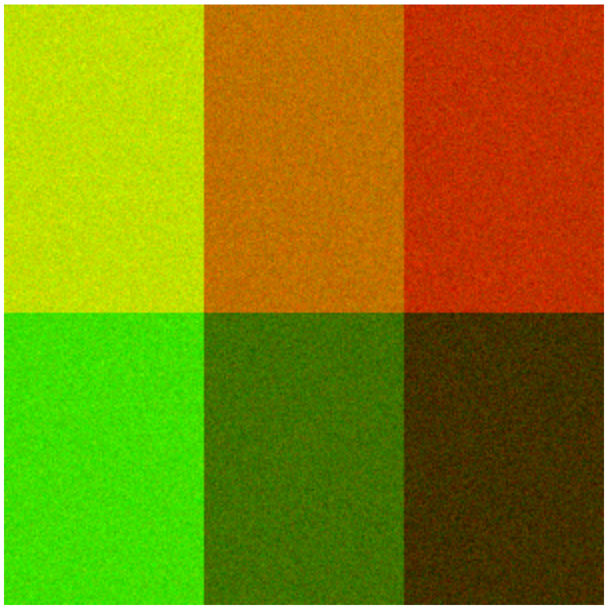
δ allows to control the ratio of outliers (absurd data).

$$\delta^2 = \lambda . \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} \left[d_{ij}\right]^2}{n.c}$$

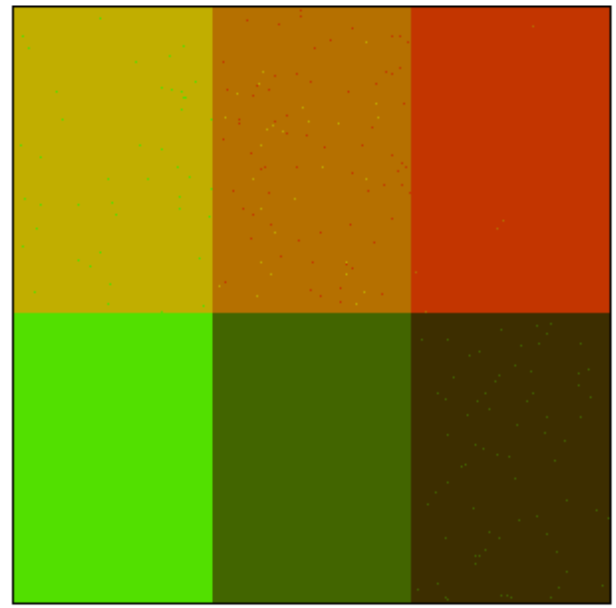
δ^2 has to be updated at each iteration.

Davé’s algorithm

Example



Noisy Image



Segmented image

This is the end of this part!