

# Binary Segmentation of Multiband Images

Claudia Sánchez<sup>1</sup> and Mariano Rivera<sup>2</sup>

<sup>1</sup> Universidad Panamericana,  
Facultad de Ingeniería, Aguascalientes, Ags.,  
Mexico

<sup>2</sup> Centro de Investigación en Matemáticas,  
Departamento de Ciencias Computacionales, Guanajuato, Gto.,  
Mexico

cnsanchez@up.edu.mx, mrivera@cimat.mx

**Abstract.** We present a method for binary segmentation of multiband images based on a combination of dimensionality reduction techniques (Weighted PCA and Quadratic Programming Feature Selection), classification methods (Gaussian Mixtures Models and Random Forest) and segmentation method (Quadratic Markov Measure Field Models). In this work, four pixels descriptors are presented: Color, Discrete Cosine Transform, Gradient Fields and Adjacency Matrix. Our method combines the outcome of several classifiers using an optimization criterion. That results in a robust method for image segmentation based on color, textures and orientation. We evaluate our method capabilities with different image types for example: color images in RGB format and satellite images. Experimental results demonstrated our method performance.

**Keywords:** Image segmentation, maximum likelihood estimation, classification, multiband images, dimensionality reduction, weighted PCA, feature selection

## 1 Introduction

Image segmentation is an important issue in the image processing and computer vision areas, consists in divide an image in regions with similar features like color, texture, orientation, etc. It is the process to label each pixel in the image with the objective that pixels with the same label have similar features. More specific, image binary segmentation divide an image in two regions, it has practice applications as:

1. Separate the main object and the background,
2. Localization of tumors and other pathologies,
3. Classification of ground cover with the analysis of satellite images,
4. Face recognition.

Some algorithms and techniques of general purpose have been developed for the image segmentation, for example: [2], [5], [10], [13] and [14]. Unfortunately these methods can be used only in images with a small number of bands as RGB images.

The multiband images more common are the color images. In these images, the pixel color is obtained as the combination of three colors: red, blue and green, so these images have three bands. Another example of multiband images are the satellite images, which are the result of the information of the land cover captured by sensors of the artificial satellites. This kind of images are a tool very useful for the study of weather, land cover, etc. The application of the binary segmentation on these images could be used for land cover classification, detection of zones with particular features like agriculture areas, cities, etc. Based in [4], data of the different year seasons are used for the binary segmentation in satellite images, they use Feature Selection techniques for the reduction of the dimensionality and the Minimum Distance to the Center of the Classes or Maximum Likelihood for the classification. In [17] detect the vegetation in satellite images with a supervised learning technique for the classification, using the NDVI coefficient (Normalized Difference Vegetation Index) that is a transformation of the infrared band, near infrared band and the first main component.

Formally an image with  $n$  bands is an array of  $n$  bidimensionality arrays where each array has the information of the corresponding band, so a pixel is

$$x_{ij} = [x_{ij1}, x_{ij2}, , x_{ijk}, , x_{ijn}]. \quad (1)$$

So  $x_{ijk}$  has the value of the pixel  $(i, j)$  in the  $k$  band.

The proposed method, in contrast with the others methods mentioned in this section, is able to make a binary segmentation in different kinds of images as RGB images, infrared images, and satellite images. In addition, this method learns the feature that better difference between classes.

## 2 Brief Review of the Methods Used in this Work

### 2.1 Descriptors

A pixel descriptor is a set of data that represents the information of the color, texture and orientation of a specific pixel and its neighbors. The descriptors used in this work are:

**Color:** this descriptor has the pixel values in all the bands, as show in the equation 2.

$$D_{ij}^c = [x_{ij1}, x_{ij2}, \dots, x_{ijk}, \dots, x_{ijn}]. \quad (2)$$

**DCT (Discrete Cosine Transform)** described in [12]. The first step to calculate the DCT descriptor for a pixel  $i, j$  is get a matrix  $A^k$  with the values of the pixel  $(i, j)$  and its neighbors in the  $k$  band, as show in the equation 3.

$$A^k = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \ddots \\ \cdots & x_{(i-1)(j-1)k} & x_{(i-1)jk} & x_{(i-1)(j+1)k} & \cdots \\ \cdots & x_{i(j-1)k} & x_{ijk} & x_{i(j+1)k} & \cdots \\ \cdots & x_{(i+1)(j-1)k} & x_{(i+1)jk} & x_{(i+1)(j+1)k} & \cdots \\ \ddots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (3)$$

Then, the matrix  $B$ , that represents the Discrete Cosine Transformation of  $A$ , is calculate with the equation 4.

$$B_{pq}^k = \alpha_p \alpha_q \sum_{m=1}^M \sum_{n=1}^N A_{mn} \cos\left(\frac{\pi(2m+1)p}{2M}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right). \quad (4)$$

where the matrix  $B$  has the same size of  $A$ .  $M$  and  $N$  are the row and column size,  $p$  represents a row and  $q$  represents a column, so  $1 \leq p \leq M$ ,  $1 \leq q \leq N$ .  $\alpha_p$  and  $\alpha_q$  are calculate with the equation 5 and 6 respectively:

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & p = 1 \\ \sqrt{\frac{2}{M}} & 2 \leq p \leq M \end{cases}, \quad (5)$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & q = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq q \leq N \end{cases}. \quad (6)$$

Finally the descriptor DCT is the vectorization of all the resulting matrices of the DCT in each band, as show in the equation 7:

$$D_{ij}^{DCT} = [B_{11}^1, B_{12}^1, \dots, B_{11}^k, B_{12}^k, \dots, B_{11}^n, B_{12}^n, \dots]. \quad (7)$$

**GF (Gradient Fields)** It is based on the gradient of each pixel, it means, the magnitud and orientation of the maximum difference between neighbor pixels. The first step is calculate two matrices for each band that represent the difference between horizontal neighbor pixels and vertical neighbor pixels, as show in the equation 8:

$$DX_{ijk} = x_{ijk} - x_{(i-1)jk}, DY_{ijk} = x_{ijk} - x_{i(j-1)k}. \quad (8)$$

Then calculate the magnitud and the orientation based in the  $DX$  and  $DY$  matrices, equation 9:

$$M_{ijk} = \sqrt{(DX_{ijk})^2 + (DY_{ijk})^2}, O_{ijk} = \tan^{-1}\left(\frac{DY_{ijk}}{DX_{ijk}}\right). \quad (9)$$

After, for each band and for each pixel two sub matrices centering in the pixel  $(i, j)$  are extracted, one matrix of magnitudes and other matrix for orientations, with the objective to calculate a normalized histogram of the gradient orientation, equation 10:

$$H_{ijk} = [H_{ijk}^{0^\circ-20^\circ}, H_{ijk}^{20^\circ-40^\circ}, \dots, H_{ijk}^{340^\circ-360^\circ}]. \quad (10)$$

Finally, the GF descriptor is formed with all the values of the histograms, equation 11:

$$D_{ij}^{GF} = [H_{ij1}, H_{ij2}, \dots, H_{ijn}]. \quad (11)$$

**AD (Adjacency Matrix)**, the first step to calculate the AD descriptor consists in reduce the dimensionality of the pixels in the image to one band using PCA, described in [1]. Then the values are discretized in  $b$  bins to obtain  $I$ . After two matrices are calculate: horizontal adjacency matrix and vertical adjacency matrix, as show in the equation 12:

$$A_{rs}^H = \sum_{i=2}^M \sum_{j=1}^N \delta(I_{ij}, r) \delta(I_{(i-1)j}, s), A_{rs}^V = \sum_{i=2}^M \sum_{j=1}^N \delta(I_{ij}, r) \delta(I_{i(j-1)}, s), \quad (12)$$

where the  $A^V = \{A_{rs}^V\}_{r,s=1,\dots,b}$ ,  $A^H = \{A_{rs}^H\}_{r,s=1,\dots,b}$ ,  $M$  and  $N$  are the number of rows and number of columns in the image,  $\delta$  is the Kronecher delta where the result is one if the parameters are equals or zero if are different.

Then the matrix exponential is used to increment the response of the adjacency, equation 13:

$$EA^V = I + A^V + \frac{(A^V)^2}{2!} + \frac{(A^V)^3}{3!}, EA^H = I + A^H + \frac{(A^H)^2}{2!} + \frac{(A^H)^3}{3!}. \quad (13)$$

Finally the descriptor AD of the pixel  $(i, j)$  is the vectorization of values in the sub matrix of  $EA^V$  centering in  $(i, j)$  and the sub matrix of  $EA^H$  centering in  $(i, j)$ , equation 14:

$$D_{ij}^{AD} = [\dots, EA_{(i-1)(j-1)}^V, \dots, EA_{ij}^V, \dots, EA_{ij}^H, \dots, EA_{(i+1)(j+1)}^H, \dots]. \quad (14)$$

## 2.2 Dimensionality Reduction

**WPCA (Weighted PCA)** described in [16], is a technique based on PCA that assume each dimension contributes in a different proportion to represent the information. We can see the segmentation problem as a classification problem, so, the contribution of each variable depends on its classification capability.

Asuming  $X = \{x_{ij}\}, i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, D$  the goal of weighted PCA is project the data in a space of dimensionality  $M < D$ , the steps are:

1. Calculate a vector with the variables weight  $W = [w_1, w_2, \dots, w_D]^T$ , where  $D$  is the dimension number. The weights can be calculated with a dependency measurement as Pearson correlation coefficient between the variable and the class.
2. Normalize the vector  $W$  in order to  $w_j > 0 \forall j$  and  $\sum_{j=1}^D w_j = 1$ ,
3. Calculate the mean vector  $\bar{X} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_D]^T$  and the variance vector  $s = [s_1, s_2, \dots, s_D]^T$ ,
4. Standardize the data using the means and variances  $z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{s_j}}$ ,
5. Weighted the data using the vector  $W$ ,  $z_{ij}^* = z_{ij} \sqrt{w_j}$ ,

6. Calculate the projection vectors  $P$  as the  $M$  bigger eigenvectors of  $(Z^*)^T(Z^*)$ , where  $Z^* = z_{ij}^*, i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, D$ ,
7. Send the data to the origin  $\hat{x}_{ij} = x_{ij} - \bar{x}_j$ ,
8. Calculate the main components projecting the data  $Y = \hat{X}P$ , where  $\hat{X} = \hat{x}_{ij}$ .

Finally, the new data are  $Y = \{y_{ij}\}, i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, M$ .

**QPFS (Quadratic Programming Feature Selection)** described in [15], is a feature selection method to classification problems using quadratic programming. It reduce the redundancy between variables and maximize the dependency between the variables and the class variable. The main goal is provide a method of reasonable complex for classification problems of high dimensionality. Consists in solve the optimization problem defined in the formula 15.

$$\begin{aligned} \min_w \frac{1}{2}(1 - \alpha)w^T Qw - \alpha F^T w \\ \text{s.t.} \\ w_i \geq 0, \sum_{i=1}^D w_i = 1. \end{aligned} \quad (15)$$

$Q$  is a quadratic simetric matrix that represents the redundancy between variables. The size of  $Q$  is  $D \times D$ , where  $D$  is the variable number.  $Q = \{q_{ij}\}$ , where  $q_{ij}$  represents the dependency between the  $i$  and  $j$  variables.

$q_{ij}$  is calculate with a dependency measurement as the mutual information [6], [15] for discrete variables, equation 16:

$$q_{ij} = I(X_i, X_j) = \sum_{h_i \in X_i} \sum_{h_j \in X_j} P(h_i, h_j) \log \left( \frac{P(h_i, h_j)}{P(h_i)P(h_j)} \right), \quad (16)$$

or Pearson correlation coefficient, described in [15], for continuos variables, equation 17:

$$q_{ij} = r_{ij} = \frac{\sum_{k=1}^N (X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j)}{\sqrt{\sum_{k=1}^N (X_{ki} - \bar{X}_i)^2} \sqrt{\sum_{k=1}^N (X_{kj} - \bar{X}_j)^2}}. \quad (17)$$

$F$  is a vector that represents the dependency between the variables and class variable. The size of  $F$  is  $D \times 1$ .  $f_i$  can be calculated with the mutual information or with the Pearson correlation coefficient depending on the variable types.  $F = \{f_i\}$ , each  $f_i$  represents the dependency between the  $i$  variable and the class variable.

$\alpha$  controls the importance of the relevance in front of the redundancy. It must be  $0 \leq \alpha \leq 1$ . Large values of  $\alpha$  represents that the dependency is more important but the selected variables can be redundants. Small values represents reducing the redundancy is more important.

To guarantee a good solution the optimization problem have two restrictions:  $w_i \geq 0$  all the variable weighted must be bigger than zero and  $\sum_{i=1}^D w_i = 1$  the sum of all variable weights must be 1.

Finally, only the variables with the bigger weights are selected.

### 2.3 Classification

**GMM (Gaussian Mixture Model)** described in [1], allows to model complex distributions of data sets based on a linear combination of Gaussians. The parameters of a GMM are calculate with the EM algorithm. Formally the form to calculate the likelihood with a GMM is with the equation 18:

$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k), \quad (18)$$

where  $x$  is a vector,  $k$  is the Gaussian number,  $\pi_k$  the proportion of the  $k$  Gaussian,  $\mu_k$  and  $\Sigma_k$  are the mean vector and covariance matrix of the  $k$  Gaussian.  $0 \leq \pi_k \leq 1 \forall k, \sum_k \pi_k = 1$ .

**RF (Random Forest)** described in [3], is a set of random decision trees, each one created with a random subset of the training dataset. A decision tree, described in [11], is a prediction model based in a series of questions about of variable values to predict the class. In a decision tree the data are organized in rectangular regions, product of the questions, with the objective of the data in a region be as possible of the same class.

### 2.4 Segmentation

**QMMF (Quadratic Markov Measure Field Models)** described in [13], calculate the probability to set a label to the pixel, unlike hard segmentation set the label to the pixel. Once the likelihood of each pixel is calculated as a normalized vector  $\hat{v}(r)$  that shows the membership to the  $r$  pixel to each class QMMF calculates the probability  $p(r)$  as a normalized vector that shows the probability to belong to each class based on the likelihood, the neighbor probabilities, and the entropy. QMMF is based on the Quadratic Programming defined in the formula 19.

$$\min_p \frac{1}{2} \sum_r Q(p(r), \hat{v}(r)) - \frac{\mu}{2} \sum_r \|p(r)\|_2^2 + \frac{\lambda}{2} \sum_{\langle r,s \rangle} R(p), \quad (19)$$

where  $\mu$  and  $\lambda$  control the contribution to each term. The first term relates the solution  $p$  with the likelihood  $\hat{v}$ ,  $Q(p(r), \hat{v}(r)) = p(r)^T D_r p(r)$ ,  $D_r = \text{diag}(-\log(\hat{v}(r)))$ . The second term controls the solution entropy with the objective it be small. Finally, the third term produce soft spatially solution and  $r$  and  $s$  are first neighbors. The optimal solution is calculate with Gauss-Seidel Projected described in [9].

## 3 Proposed Method

The proposed method makes the binary segmentation of an image based on user clues at the beginning of the process to establish the features of each class. The



**Fig. 1.** User clues for the segmentation. a) Image b) User clues, white pixels are a sample of pixels of the class one and black pixels are another sample of pixels of the class two.

clues consist into mark a sample of pixels as class one and mark another sample of pixels as class two. The figure 1 shows an example of user clues.

Our method is divided into two phases:

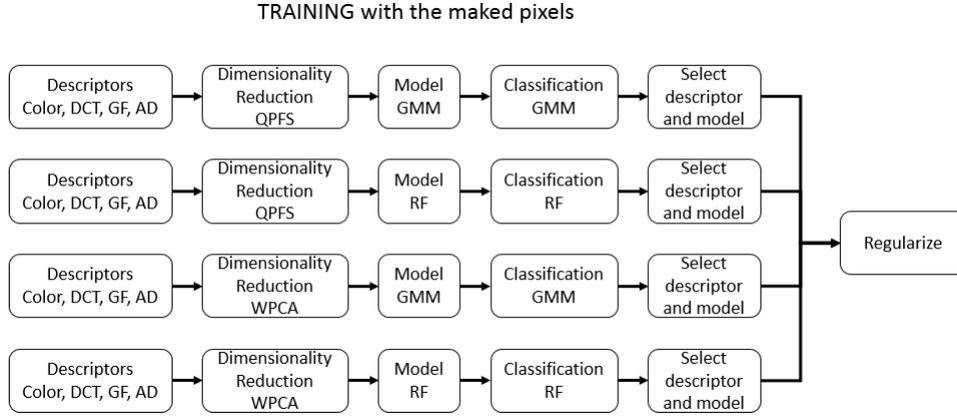
1. Training with the marked pixels,
2. Classification of the not marked pixels and Segmentation.

### 3.1 Training with the Marked Pixels

The first phase consists in recognize the features of each class based on the user clues, this is: found the descriptors that identify the features that separate the classes, calculate the parameters to an optimal dimensional reducing and calculate the parameters to the classification models.

In this phase we only use the information of the marked pixels, the user clues, to learn the models for the classification. The steps, shown in the figure 2, are the following:

1. Create the descriptors: Color, DCT, GF and AD for the marked pixels.
2. Reduce the descriptors dimensionality using: WPCA and QPFS. So we have eight reduce descriptors into two groups:
  - (a) Group 1: Color reduced with QPFS, DCT reduced with QPFS, GF reduced with QPFS and AD reduced with QPFS.
  - (b) Group 2: Color reduced with WPCA, DCT reduced with WPCA, GF reduced with WPCA and AD reduced with WPCA.
3. Create the models, as we have two dimensionality reduction techniques and two classification methods, we group the models as the next form:
  - (a) Group 1: GMM of the descriptors reduced with QPFS,
  - (b) Group 2: RF of the descriptors reduced with QPFS,
  - (c) Group 3: GMM of the descriptors reduced with WPCA,
  - (d) Group 4: RF of the descriptors reduced with WPCA.
4. Classification of the marked pixels with the models created in the previous step. We calculate 16 different likelihoods for each marked pixel because we have four groups and four descriptors in each group.



**Fig. 2.** Proposed method, first phase.

5. Select the descriptor and model for each group. For each group:
  - (a) Calculate the accuracy for each descriptor with the equation 20.

$$Accuracy = \frac{\sum_{\langle i \rangle} \varphi(V_i^1, V_i^2, C_i)}{Nmp}, \quad (20)$$

where

$$\varphi(V_i^1, V_i^2, C_i) = \begin{cases} 1 & \text{if } C_i = 1, V_i^1 \geq V_i^2 \\ 1 & \text{if } C_i = 2, V_i^2 \geq V_i^1 \\ 0 & \text{otherwise} \end{cases}, \quad (21)$$

where  $Nmp$  is the number of marked pixels and  $\langle i \rangle$  represents only the marked pixels.

- (b) Select the descriptor with the bigger accuracy.
- (c) Select the model, RF or GMM, that corresponds to the selected descriptor.
- (d) Normalize the likelihoods of the descriptor selected, with the goal of  $V_i^1 + V_i^2 = 1$ , with the equation 22:

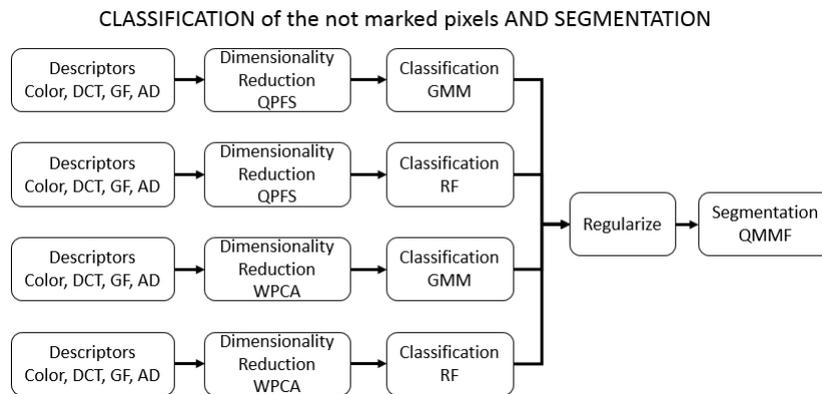
$$V_i^1 = \frac{V_i^1 + \varepsilon}{V_i^1 + V_i^2 + \varepsilon}. \quad (22)$$

6. Regularize, this is, calculate the weight of each group to the classification.
  - (a) Calculate the efficiency to the classification for each group.  $a^g$ ,  $g = 1, 2, 3, 4$ , represents the classification efficiency of the group  $g$ , and it is calculate as the accuracy of the normalize likelihood.
  - (b) Regularize the accuracy, this is,  $\hat{a}^1 + \hat{a}^2 + \hat{a}^3 + \hat{a}^4 = 1$ , whit the equation 23:

$$\begin{aligned} \hat{a}^1 &= \frac{a^1}{a^1 + a^2 + a^3 + a^4} & \hat{a}^2 &= \frac{a^2}{a^1 + a^2 + a^3 + a^4} \\ \hat{a}^3 &= \frac{a^3}{a^1 + a^2 + a^3 + a^4} & \hat{a}^4 &= \frac{a^4}{a^1 + a^2 + a^3 + a^4} \end{aligned}. \quad (23)$$

### 3.2 Classification of the not Marked Pixels and Segmentation

The second phase consists in the classification of the no marked pixels and the segmentation of the results, it means: calculate the descriptor of the no marked pixels, reduce the descriptors dimensionality and calculate the likelihood of the not marked pixels with the models created in the first phase, combination of the four classifiers results and segmentation. The steps of this phase, show in the figure 3, are the following:



**Fig. 3.** Proposed method, second phase.

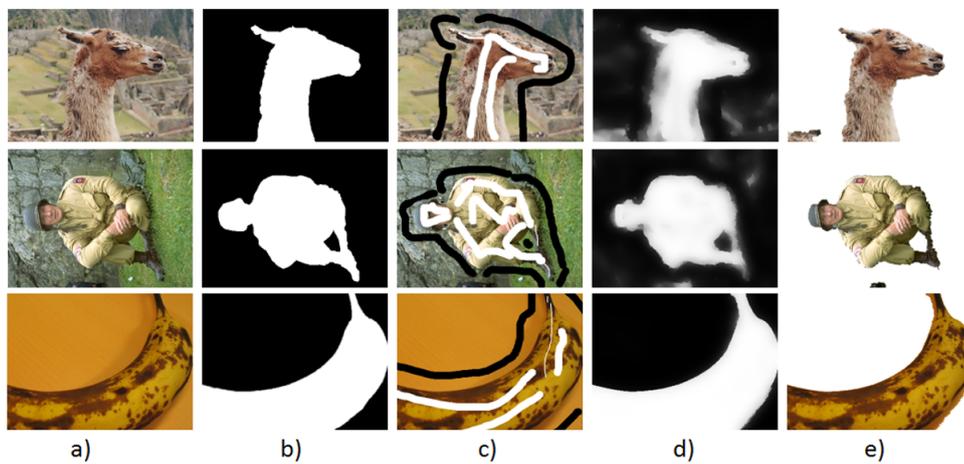
1. Create the selected descriptors for the no marked pixels.
2. Reduce the descriptors dimensionality using: WPCA and QPFS.
3. Classify the no marked pixels using the reduce descriptors and the models created and selected in the first phase. We must to have four likelihoods, one for each group:
  - (a) Group 1: QPFS and GMM,
  - (b) Group 2: QPFS and RF,
  - (c) Group 3: WPCA and GMM,
  - (d) Group 4: WPCA and RF.
4. Calculate the general likelihood using the accuracy of each group.
  - (a) Normalize the likelihood of each group, as mentioned in the equation 22.
  - (b) Combine the classifiers results 24:

$$\hat{V}_i^1 = \sum_{g=1}^4 \hat{a}^g V_{gi}^1. \quad (24)$$

5. Segment the likelihood using QMMF.

**Table 1.** Percentage of error in the classification of the Statlog dataset.

Class	QPFS	QPFS	WPCA	WPCA	Combination
	and GMM	and RF	and GMM	and RF	
1	2.97	4.93	1.97	1.94	<b>1.18</b>
2	3.75	2.36	<b>1.12</b>	1.46	1.43
3	7.04	5.56	5.28	4.55	<b>3.82</b>
4	14.39	8.81	9.29	7.94	<b>6.53</b>
5	7.32	5.19	7.52	4.72	<b>3.43</b>
7	11.62	8.86	8.17	7.02	<b>5.87</b>
Mean	7.85	5.95	5.56	4.61	<b>3.71</b>

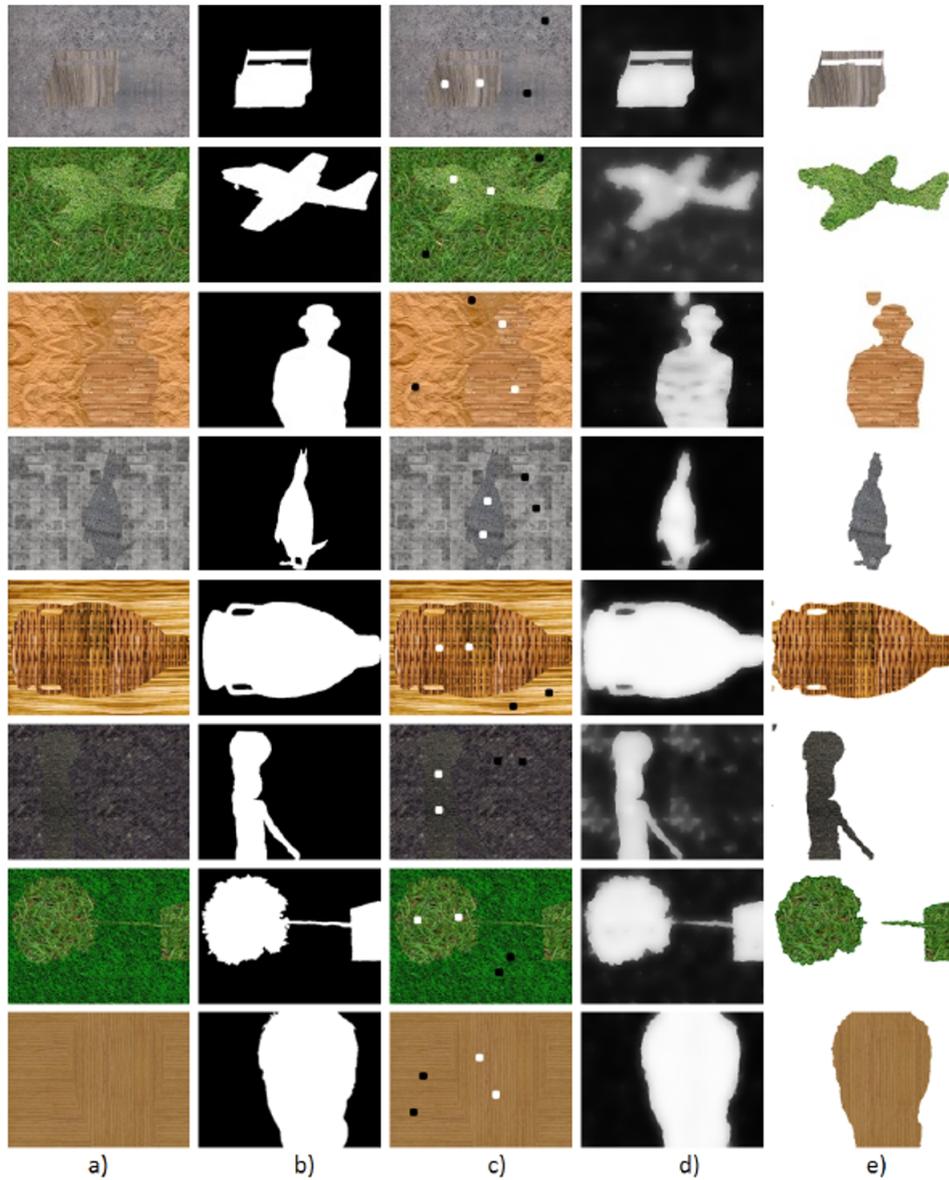


**Fig. 4.** Results of binary segmentation of images with real textures a) Image b) Real segmentation c) User clues d) Proposed method segmentation e) Main object, calculated with the proposed method

## 4 Experiments and Results

### 4.1 Statlog Dataset

The Statlog dataset [7] has information of satellite images of the Landsat satellite. This dataset consists in multivariate data of pixels in 3x3 neighborhood and the classification of the central pixel. The objective is predict the classification. The pixel class is coded with a number that represents: 1 red soil, 2 cotton crop, 3 grey soil, 4 damp grey soil, 5 soil with vegetation stubble, 6 mixture class and 7 very damp grey soil. Each neighborhood is represented by 36 variables more the class, the data set is composite by 6,435 data. The objective of this experiment is verify the classifiers combination is better than each classifier by separated, the results are show in table 1.



**Fig. 5.** Results of binary segmentation of images with different textures. a) Image b) Real segmentation c) User clues d) Proposed method segmentation e) Main object calculated with the proposed method

#### 4.2 Images with Real Textures

The images with real textures was obtained of the Microsoft Research Cambridge dataset [8]. The objective to this experiment is measure the efficiency of the

proposed method with real images in RGB format. The dataset consists in 50 images as the images shown in the figure 4. The mean percentage of error obtained with the proposed method is 3.6%.

### 4.3 Images with Different Textures

The images with different textures was created based on the Microsoft Research Cambridge dataset [8] replacing the main object and the background with texture images. The objective to this experiment is measure the efficiency of the proposed method to segmentation by color, texture and orientation. The results are show in the figure 5.

## 5 Conclusions

The use of four descriptors: Color, DCT, GF and AD creates a robust segmentation by color, texture and orientation. Descriptor selection improvements the results because only the features that maximize the difference between classes are used, so this makes easy the models creation because we have a moderate dimensionality instead of use the information of all descriptors. With the selection of the better descriptors we can say that the method is adapted to each image.

The lineal combination of four classifiers: (1) WPCA and GMM, (2) WPCA and RF, (3) QPFS and GMM, and (4) QPFS and GMM produces better results than the use of only one classifier.

The proposed method can be implemented in parallel form so can be applied efficiently in satellite images.

## References

1. Bishop, C.: Pattern Recognition and Machine Learning. Springer (2006)
2. Boykov, Y., Jolly, M.: Interactive graph cut for optimal boundary and region segmentation. IICIP (2001)
3. Breiman, L.: Random forest. Machine Learning pp. 5–32 (2001)
4. Caetano, M.: Image classification. ESA Advanced training course on Land Remote Sensing (2009)
5. Grady, L.: Random walks for image segmentation. ECCV 1 (2004)
6. L., H.P.F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans pp. 1226–1238 (2005)
7. Machine learning repository. statlog dataset (2012), "[http://archive.ics.uci.edu/ml/datasets/Statlog+\(Landsat+Satellite\)](http://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite))"
8. Microsoft research cambridge. dataset of test image (2012), "<http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>"
9. Nocedal, J., W., S.J.: Numerical Optimization. Springer (2006)

10. Ocegueda, O., Rivera, M., Marroquín, J.: Entropy-controlled quadratic markov measure field models for efficient image segmentation. *IEEE Trans. Image Processing* (2007)
11. Olshen, C.R.A., Stone, J., Breiman, L.: *Classification and regression trees*. CRC Press (1998)
12. Pennebaker, W., Mitchell, J.: *Jpeg: Still image data compression standard*. Van Nostrand Reinhold (1993)
13. Rivera, M., Dalmau, O.: Variational viewpoint of the quadratic markov measure field models: Theory and algorithms (2010)
14. Rivera, M., Dalmau, O., Mio, W., Ramirez, A.: Spatial sampling for image segmentation. *The computer Journal* (2010)
15. Rodríguez, L.I.: *Selección de variables mediante programación cuadrática*. Universidad Autónoma de Madrid (2009)
16. Thomaz, C.: A simple and efficient supervised method for spatially weighted pca in face image analysis. Imperial College London, Technical Report (2010)
17. Viana, H., Aranha, J., Rodrigues, R.: Vegetation classification and quantification by satellite image processing. a case study in north portugal. *International Conference and Exhibition on Bioenergy* (2008)