Image Characterisation using Texture

Tatiana Lopez Guevara , Elizabeth Vargas Vargas
University of Girona
tatiana@sirius.utp.edu.co, liz2102@gmail.com

***Abstract*—Segmentation and classification are well known unsupervised and supervised problems in artificial vision. This document review both of them based on texture feature extracted using co-ocurrence matrices. The pertinence of the features is evaluated in a set of test images and the results are presented. Depending on the characteristics of each image different features are used, in order to improve segmentation accuracy and number of correct classified images.**

***Index Terms*—Segmentation, region growing, classification**

I. Introduction

Segmentation is a well known unsupervised problem in the image processing field given it's multiple application areas, such as medicine, object recognition, etc. Different strategies has been proposed in order to solve it. However, there is not an absolute answer for the problem, given that the obtained results depends on the parameters used and on the nature of the image.

Classification on the other hand, is a supervised problem, where a set of elements should be correctly organized in classes previously defined. The problem has main applications in artificial vision, specially for object recognition.

II. Problem Definition

The segmentation problem tries to divide an image in different regions, such that objects belonging to the region do not overlap and the union is the entire image. The regions should follow a set of rules, in order to accept the segmentation as appropriate. These rules include uniformity, simple interiors without holes and spatially accurate separation.

The problem consists in determining a quantitative way to decide whether a segmentation is meaningful or not, depending on the application field. The approaches followed in the literature use different elements such as pre-processing techniques or determined parameters according to the type of image that need to be segmented. [1]

Classification deals with organizing objects in different groups, that has already been labelled according to a determined criteria. The challenge of classification is to determine the appropriate features that allows the differentiation of objects that belongs to different classes and the choose of a classifier that is able to detect those differences.

III. Implementation

A. Co-ocurrence Matrices

Co-ocurrence matrices were used in order to extract texture features from images, used to perform further segmentation and classification. Figure 1 presents the used of the MATLAB functions that implements these kind of matrices.

```
im = rgb2gray(im);
comat_im = graycomatrix(im, 'Offset',[0 2],
                        'Symmetric', true);
stats_im = graycoprops(comat_im,{'Contrast','Homogeneity',
                        'Energy', 'Correlation'});
stats_im = struct2array( stats_im );
```

Fig. 1: Co-ocurrence matrix

The function `graycomatrix` is used to create a gray co-ocurrence matrix. Figure 2 presents the values defined for the *Offset* parameter. In the code snippet, the value for the Offset is $[02]$, which means that the orientation is $0°$ and the distance is 2. The function `graycoprops` computes the statistics determined in the parameters. In this case, Contrast, Homogeneity, Energy and Correlation. In the segmentation and classification algorithms presented in this document, there are used different parameters of the matrices, such as orientation, distance and statistical measurements.

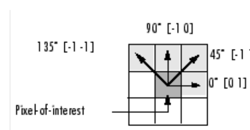

Fig. 2: Values of the offset parameter [2]

B. Segmentation

The segmentation algorithm was implemented using region growing. The implementation is the same one as in the previous exercise, with the only difference than instead three channels RGB, on each position is stored a vector of N features, additional to the color information. Additionally, each feature is normalized using:

$$norm(x_i) = \frac{(x_i - \mu_i)}{\sigma} \tag{1}$$

where x_i corresponds to the $i-th$ feature.

For each feature, the mean value of the region is stored. The agregation function is computed by using the following formula:

$$\sqrt{\sum_{i=1}^{N}(f_i - f_\mu)^2} < threshold \tag{2}$$

A new function was created to calculate the the 4 local statistics of a pixel for all the directions and angle at once. Initially, the MATLAB function *nlfilter* was used, but it can return only one statistic per one direction and angle. Since the best configuration for each image are obtained experimentally by testing with different parameters, the time for each test was considerably high..

C. Classification

A set of 20 different classes of images were classified using the Random Forest classifier provided by the data mining tool WEKA. For each class, a set of six images were provided, with labels 1, 5, 9, 13, 17 and 21. Images with labels 1 and 5 were used as a training for the classifier and the rest were used for testing. The implementation consisted in determining a set of features from each image, in order to improve the accuracy of the classification. The proposed features were the following:

- **Gray channel mean:** The image was transformed into a gray representation, (in case we were dealing with a color image) and the mean value was computed.
- **Texture features (CHECorr):** Using a global co-ocurrence matrix, in orientation 0°, distance 2 and the statistics Contrast, Homogeneity, Enery and Correlation, a set of texture features of the image were computed, using the matlab functions `graycomatrix` and `graycoprops`.
- **Texture features (CH):** It is a simplied version of the previous set of features, where only Contrast and Homogeneity are considered.
- **RGB mean:** The images that were using for classification were analysed, in order to determine by visual inspection which kind of features are the most representative for each image. The conclusion was that the color is the characteristic that is easier to differentiate between different sets. Table I presents the summarize of the color features from each set.

Set	Color	Set	Color
t1	yellow	t11	light green
t2	white	t12	reddish brown
t3	green	t13	light yellow
t4	beige	t14	light yellow
t5	blue	t15	beige
t6	purple	t16	red
t7	blue	t17	white
t8	brown	t18	green
t9	white	t19	white
t10	light blue	t20	brown

TABLE I: Color features

- **RGB mean & Texture features:** This feature is the combination of the two type of features previously computed.
- **CIELab mean:** The RGB image were transform into the *CIELab* color space, given that euclidean distance in that color space presents in general better results as comparison measurement than in other colors spaces.
- **CIELab mean & Texture features:** This feature is the combination of two type of features previously computed: textures using co-ocurrence matrices and the mean value of each channel in the CIELab color space.

IV. Experimental Results

A. Co-ocurrence Matrices Performance

The system described in Table II was used to run the performance tests.

Hardware Overview:
Model Name: MacBook Pro
Model Identifier: MacBookPro10,1
Processor Name: Intel Core i7
Processor Speed: 2.3 GHz
Number of Processors: 1
Total Number of Cores: 4
L2 Cache (per Core): 256 KB
L3 Cache: 6 MB
Memory: 8 GB

TABLE II: Hardware Overview

Table III presents the time table with the execution time of the algorithm that extracts statistics, according to the image used and the parameters used to compute them.

Image	Size	# Ang.	Dist.	# Statistics	Time
Feli	382 x 265	0	2	3	160.374
Feli	382 x 265	2	3	3	280.554
Hand	288 x 228	1	3	3	141.983
Hand	288 x 228	4	1	3	151.944
PingPong	344 x 228	4	1	3	193.45

TABLE III: Texture Feature Extraction Execution Time

B. Segmentation Results

The segmentation results obtained for different images are presented individually in this section, given that for each image in particular, the parameters were determined by analysing the characteristics of the image. In each case, a comparison between the segmented image using colors and textures is presented, in order to evaluate the pertinence or not of the features.

Feli Texture Features:
Gray Levels: 8
Window Size: 13
Distance: [1 2]
Orientation: 45°, 135°

TABLE IV: Feli texture features parameters

The textures features from *Feli* were extracted using the parameters of the Table IV. The parameters were determining by observing the different images generated from different features, in order to identify which one presents the most notable differences between regions. Figures 3 and 4 presents the features contrast, homogeneity and energy for the orientations 45° and 135° respectively. As can be seen in the images, the difference between the background and foreground is remarkable, making these kind of features useful for performing a meaningful segmentation.

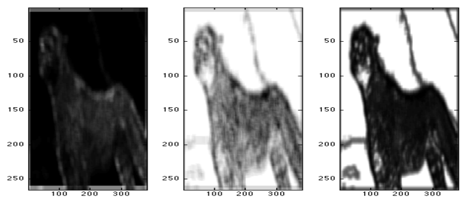

Fig. 3: Statistics using 45°

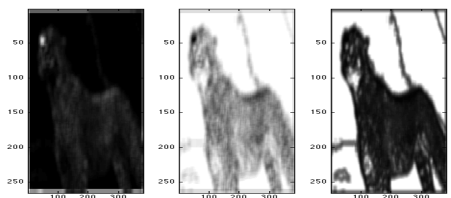

Fig. 4: Statistics using 135°

Figure 5 presents a patch of the texture of the animal that was analysed, in order to determine the parameters (distance, angle) used on the co-ocurrence matrix calculation. In the figure can be seen the orientation of the texture, which corresponds with the selected angles.

Fig. 5: Parameters Estimation

Figure 6 presents the results obtained when segmenting the image using color features (left) and color combined with texture features (right). The threshold value used for color segmentation was 40, while for the combined feature vector was 58. By visual inspection can be verified that the image on the left side is oversegmented, given that the animal appears in different regions. On the other hand, the right image presents a clear different between background and foreground and the silhouette of the animal can be easily recognized.

Table V illustrates the parameters used for the image Hand. In this case, only one orientation were used: 135°. Figure 7 presents the statistics images generated with that orientation.

Hand Texture Features:
Gray Levels: 8
Window Size: 15
Distance: 1
Orientation: 135°

TABLE V: Hand texture features parameters

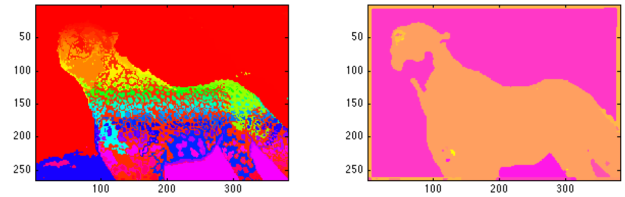

Fig. 6: Segmentation results using color and texture features

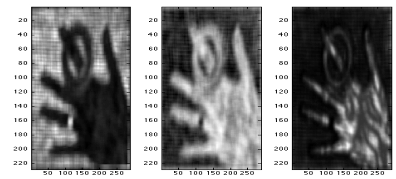

Fig. 7: Statistics using 135°

Figure 8 illustrates the parameters obtained when segmenting the image using color features and color with texture features. The threshold value used for the color is 65, while for color and texture were 2.8015. In this case, however, the segmentation using color presents better results, given that the statistics generated are noise, so the background in the segmented image is not homogeneous.

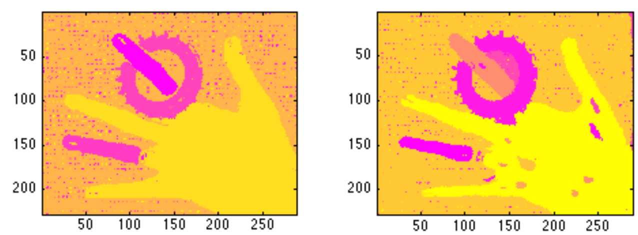

Fig. 8: Segmentation results using color and texture features

For the image Ping Pong, two different set of parameters were used. At first, the values defined on Table VI were used. Figure 9 presents the statistics obtained for the orientation 135°. As can be seen on the picture, the background is noise, which leads to poor segmentation results as can be seen in Figure 10, where the threshold value for 65 for the color segmentation and 2.259 for texture segmentation were used.

Ping Pong Texture Features:
Gray Levels: 8
Window Size: 3
Distance: 1
Orientation: 135°

TABLE VI: Ping Pong texture features parameters I

A new set of parameters were defined and are illustrated by Table VII. In this case, all the orientations were used. The results of the statistics are presented on Figure 11. The result is in this case smoother as with the previous parameters, which leads to better segmentation results, as can be observed in Figure 12. The threshold used for the features is 4.17.

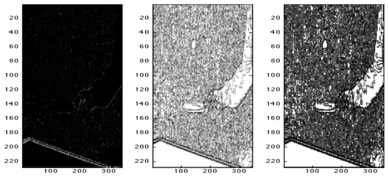

Fig. 9: Statistics using 135°

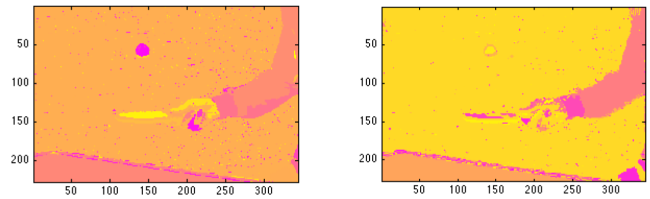

Fig. 10: Segmentation results using color and texture features

Ping Pong Texture Features:
Gray Levels: 6
Window Size: 30
Distance: 3
Orientation: 0°, 45°, 90°, 135°

TABLE VII: Ping Pong texture features parameters II

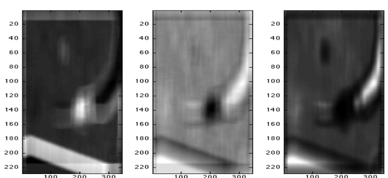

Fig. 11: Statistics using 135°

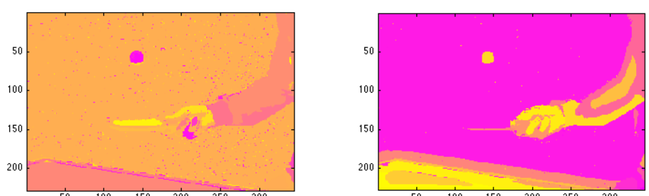

Fig. 12: Segmentation results using color and texture features

C. Classification Accuracy

Table VIII presents the percentage of images that were correctly classified for each of the features extraction approaches previously presented.

The gray channel, which is the naive approach, turns in a percentage of images that are correctly classified of 66.25%. When using normalized texture features however, the percentage decreased to 47.50%. This can be explained given the orientation used, given that was calculated only in one direction, and it is maybe not the most representative direction for all the set of images. In a further test, only the first two statistics were used, obtaining a result of correctly classified images of 63.75%. Additional tests showed that using only contrast did not lead to a good result (42.5%) and neither homogeneity (11.25%). So it means, that is the combination of both statistics, without normalization, that leads to a good result. A better result was obtained when the mean of the RGB values was used, 93.75%. It can be explained because the most prominent feature of the set of images is color, given that each image has only one type of object. The best result is obtained when the RGB mean is combined with texture features. In that case, the obtained result is 100% of images correctly classified. In this part, the experiment of changing the order of the feature vector was made, but the result was not the same one: 87.5%. At last, the color space was changed, in order to use CIELab and the mean of each channel was used as a feature. The result of applying the classification algorithm using those features is 95%. Adding the texture features to that vector does not change the result, as can be seen on the table. So we can say that the texture features only improves the accuracy on the RGB color space, but in $CIELab$ it has no positive or negative effect.

Strategy	**Correct Classified (%)**
Gray channel mean	66.25
Texture features (CHECorr)	47.50
Texture features (CH)	63.75
RGB mean	93.75
RGB mean & Texture features	100
CIELab mean	95.00
CIELab mean & Texture features	95.00

TABLE VIII: Classification accuracy

On the appendix, the most relevant confusion matrices are presented with the diagonal highlighted (representing the images that are correctly classified). Table XII presents the matrix where all the images has been correctly classified. It can be seen that in that matrix, there are values only in the diagonal. The feature that leads to this classification is the **RGB mean & Texture features**

V. Organization of Coursework

In order to organize the tasks to develop as well as the version control of the code, the tool **github** was used. The repository of this project can be find at: *https://github.com/ZePoLiTaT/SegmentationUsingTexture*.

The distribution of the work per person is presented in Table IX.

Activity	Time (hours)
Laboratory	4
Implementation	6
Test	8
Report	6
TOTAL	24

TABLE IX: Time distribution

VI. Conclusions and Future Work

- The use of texture features to perform segmentation can leads in some cases to a better segmentation results than when only colour features are used. One example of this situation is the Feli image, because when only colour features are used, the image gets oversegmented and the use of textures leads to a meaningful segmentation, where the background and foreground are easily recognized.
- In some cases, the results obtained using colour features leads to better results than using textures. This is the case of images where the difference between colors is very strong, while the textures can hardly be recognized.
- The features used in order to perform a proper classification relies mostly on the nature of the image. It could be appreciated in the classification exercise, when the use of color features increased the percentage of images correctly classified in around 40%.
- The order in which the features are used in the vector had an impact on the Random Forest Tree classifier, given that for different order of the features inside the vector, the obtained result was different. A future work in this area is the use of another classifier.
- The challenge of the segmentation and classification problems relies on the selection of features that presents meaningful differences when compared using diverse metrics. The most reliable method so far, consists in performing visual inspection on the images, in order to select the features that will be computed.

References

[1] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.

[2] Mathworks. Documentation center. http://www.mathworks.es/es/help/images/ref/graycomatrix.html. Accessed: 2014-03-20.

Appendix

t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

TABLE X: Gray channel mean

t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

TABLE XI: RGB mean

t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

TABLE XII: RGB mean & Texture features