Scene Segmentation Interpretation Image Characterization using Texture

Daniel Barmaimon, Roberto Marroquin Cortez

ViBOT, Escola Politècnica Superior, University of Girona

I. Introduction

Texture is commonly found in both natural scenes and in man-made objects. Sand, stones, grass, leaves and bricks could create the appearance of texture.

- Structural approach: Texture is a set of primitive *textel* in some regular or repeated relationship.
- Statistical approach: Quantitative measure of arrangement of intensities in a region.

A. Structural approach:

The repetition of a pattern across the space is usually defined as a texture. The identification of textures is not always an easy task. The texels are image regions that could be extracted through simple procedures (such as thresholding).

B. Statistical approach:

As the complexity of finding textels in real images is much bigger in real images than in artificial ones, statistics are widely used to segment textures. Some of the most used parameters will be commented in the following part.

1) Edge density an direction: The edge detection is one of the most used and well-known techniques, and is also one of the first one used for the analysis. The number of edge pixels in a given fixed size region provides some information about busyness level of the image. The directions of these edges can be useful when gradient histograms want to be implemented. The main idea of the use of this technique is to get the histogram of an area is going to be considered as texture representative, and compare it against histograms of other areas in the image. Is the distance measured is lower than a threshold, it could be considered as part of the texture area.

2) Local binary partition: Each pixel neighbors (connectivity 8) are analyzed to check if their intensity is greater than the pixel one. The result of this comparison is stored in a binary code $b_1b_2b_3b_4b_5b_6b_7b_8$where $b_i = 0$ if the intensity of the *ith* neighbor is less than than the threshold image. A histogram of these codes is used to represent texture of the image. The histograms of two images or regions are compared to measure their distances as in the previous case.

3) Co-ocurrence matrices and features: A co-ocurrance matrix is a two dimensional array *C* in which both the rows and the columns represent a set of possible image values *V*. The value of *C(i,j)* indicates how many times value *i* co-occurs with value *j* in some spatial relationship. Usually it works over gray level images, and the spatial relationship is the distance *d* that measures the distance between two pixels in a specific direction.

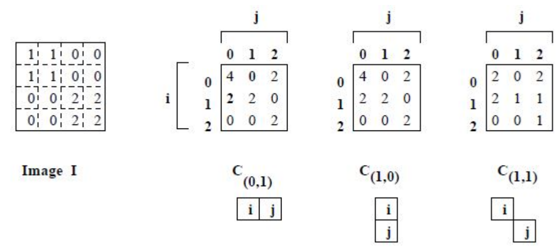

Figure 1. Co-occurrance matrices

There are two important variations of the standard gray tone co-occurrance matrix. First one is the *normalized* gray tone co-occurrence matrix N_d defined by

$$N_d(i,j) = \frac{C_d(i,j)}{\sum_i \sum_j C_d(i,j)} \tag{1}$$

that normalizes the co-occurrance values into a range between zero and one.

The second is the *symmetric* gray co-ocurrance matrix $S_d(i,j)$ defined by:

$$S_d(i,j) = C_d(i,j) + C_{-d}(i,j) \tag{2}$$

which groups pairs of symmetric adjacencies.

Co-ocurrence matrices capture properties of a texture, but they are not directly useful for comparing tow textures. From these co-occurrence matrices standard features can be used to represent the texture more compactly. The following are standard features derivable from a normalized co-ocurrence matrix.

$$Energy = \sum_i \sum_j N_d^2(i,j) \tag{3}$$

$$Contrast = \sum_i \sum_j (i-j)^2 N_d(i,j) \tag{4}$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i,j)}{1+|i-j|} \tag{5}$$

$$Entropy = \sum_i \sum_j N_d(i,j) log_2 N_d(i,j) \tag{6}$$

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j)N_d(i,j)}{\sigma_i \sigma_j} \quad (7)$$

where $\mu_i, \mu_j$are the means and $\sigma_i, \sigma_j$are the standard deviations of the row and the column.

4) Laws' Texture Energy Measures: In this case the idea is to generate several masks to detect various types of textures. The main purpose is to measure the texture energy by amount of variation within a fixed side window. A set of nine 5x5 convolution masks is used to compute texture energy, which is represented by a vector of nine numbers for each pixel of the image being analyzed. The masks are computed from the following vectors.

L5	(Level)	=	[1 4 6 4 1]
E5	(Edge)	=	[-1 -2 0 2 1]
S5	(Spot)	=	[-1 0 2 0 -1]
R5	(Ripple)	=	[1 -4 6 -4 1]

L5 returns a center weighted local average.
E5 detects the edges.
S5 gives the spots back.
R5 returns a vector that detects the ripples.

The 2D convolution masks are obtained by computing outer products of pairs of vectors. The combination of all pair of maps lead to nine resultant energy maps; that combines into a single image with a vector of nine texture attributes at each pixel. These texture attributes can be used to cluster an image into regions of uniform texture.

5) Autocorrelation and power sptectrum: The autocorrelation function of an image can be used to detect repetitive patterns of texture elements and also describes the fineness of the texture. The autocorrelation function of an (*N+1*)x(*N+1*) image for displacement *d = (dr, dc)* is given by

$$\rho(dr, dc) = \frac{\sum_{r=0}^{N} \sum_{c=0}^{N} I[r,c]I[r+dr, c+dc]}{\sum_{r=0}^{N} \sum_{c=0}^{N} I^2[r,c]} \quad (8)$$

If the texture is not very fine, then the autocorrelation drops off slowly, otherwise, it will drop off very quickly. The autocorrelation function is related to the power spectrum of the Fourier transform. If *I(r,c)* is the image function and *F(u,v)* is its Fourier transform, the quantity $|F(u,v)|^2$is defined as the power spectrum wher | ů | is the modulus of a complex number.. The frequency domain can be divided into regions bounded by circular rings (for frequency content) and edges (for orientation content) and the total energy in each region is computed to produce a set of texture features.

II. Classification

The purpose of this lab is the study of different classifiers for detecting images. For achieving this co-occurrance matrices will be used. The choice of the parameters for the classifiers and the influence of this choice over the result images will be explain during the report.

There are two main tasks to be done. First one implies the selection of the parameters needed for the classifier and testing the results over several textures. It was needed several images of each texture, some of the to train and other to test the classifiers. For the second task we use the best classifier found with a region growing algorithm to segment the image.

A. Global classification

For obtaining the co-occurrance matrices is necessary to define some features at the beginning. These features are the distance and the orientation between two gray levels to be measured, and number of gray levels to consider. Another parameter related with the orientation of the gray level changes is the consideration of symmetry in the direction or not (considering opposite directions as equal or different directions).

Once the co-occurrance matrices are obtained the choice of parameters like contrast, energy density, homogeneity, entropy or correlation could be used as parameters for the classifier.

During the implementation of the algorithm several parameters were used as tries to find the best classifier. It is necessary to point that not all textures have similar characteristics, so the parameters needed to classify them will be different too. That leads to the conclusion that there is no a unique and perfect classifier for all the images, and it should be adjusted depending on the application. The directions for the orientation considered for all the experiments were 0, 45, 90, 135°. The results obtained for the set of experiments are collected in the table below. The measure of the accuracy is given in percentage.

D=1				**D**=1 with RGB			
Gr=16		**Gr**=32		**Gr**=16		**Gr**=32	
S	**NS**	**S**	**NS**	**S**	**NS**	**S**	**NS**
75.0	75.0	85.0	81.3	82.5	78.8	90.0	75.0

D=2				**D**=3			
Gr=16		**Gr**=32		**Gr**=16		**Gr**=32	
S	**NS**	**S**	**NS**	**S**	**NS**	**S**	**NS**
72.5	80.0	76.3	76.3	75.0	73.8	73.8	75.0

where:

D : Distance in gray levels between two pixels.

Gr : Number of gray levels.

S / NS : Symmetric or non-symmetric.

RGB : Consideration of the RGB channels as extra features.

All the classifiers were implemented with 12 parameter, but the RGB, that had 15.

The best results were obtained with a 90% of accuracy (measured with Weka library for Matlab) classifier. In this case the number of gray level were 32, the three channels RGB for colors were considered and the symmetric directions were considered as well.

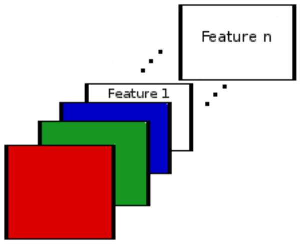

Figure 2. Layers to use in a classifier

B. Local classification

For a local classification it will be necessary to use a window for analyzing each pixel characteristics according to its neighborhood. Combining one of the classifiers studied in the previous case, and the region growing algorithm, it is possible to detect regions with different textures. The smaller is the window the more time it will take to run the algorithm. The size of this window also affect to the detection of the texture, and there is no a perfect size for all the textures.
A new image is obtained for each feature required by the classifier, and that means that each pixel will have a value for each feature. All layers should be normalized, being sure that each layer will have same weight while the average process. For each pixel the average of all the channels will be obtained, and if it is below a threshold, this pixel will be considered as a part of the region, or in our case, of the texture.

III. APPLICATION AND SEGMENTATION RESULTS

The original images used for analyzing the goodness of the classifier and the different result over different images are the ones shown in Fig 3.

Figure 3. Original images for testing

In the Fig 4. it can be observe the comparison between two homogeneity channels that were obtained with different angles and with symmetry. The angles were for *a)* 90° and *b)* 135°. It can be notice that there is very few difference between the features (redundant information), therefore the calculation of one of them could be avoided.

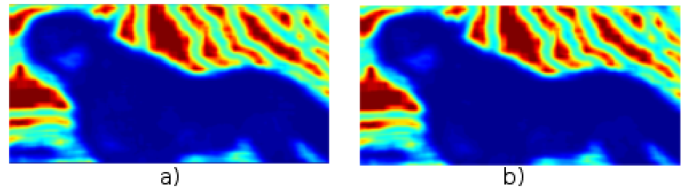

Figure 4. Homogeneity

For the analysis of the contrast some differences are easy to find when the orientation changes. The image *a)* , shown in the Fig 5. , is with 90° an the image *b)* 0°. It can be observe that the background could be easily found and differently from the Fig 4. the change of direction gives useful information.

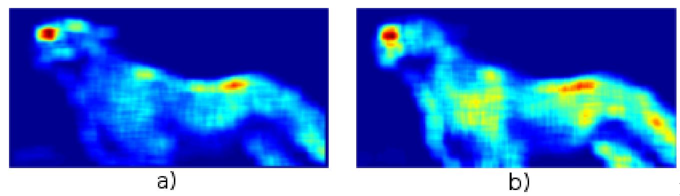

Figure 5. Contrast

In the Fig 6. is shown the energy for 90° and 0° respectively, the results are quite different and is easy to subtract the feline from the background.

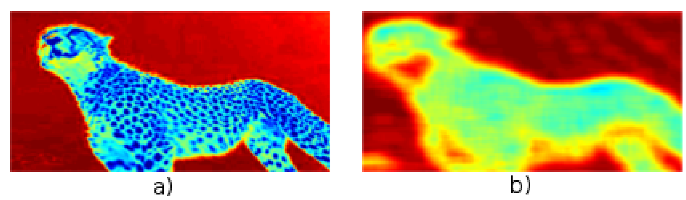

Figure 6. Energy

The best results for the images where found by using the classifier with greater accuracy (15 features) with the following parameters:

- For *feline* : mask size = 5 , threshold = 0.2; Post-processing-> median filter with size = 9. The resulting image is shown in Fig 7. it can be notice that the texture of the feline is mainly one region. It is important to remark that similar regions has similar color after segmentation because of the colormap used in Matlab, in this case 'lines'.
- For *hand* : mask size = 5 , threshold = 0.126; Post-processing-> median filter with size = 9. In Fig 8. it can be observe that the background, the hand, ring and the 'wheel' are considered as different regions. Two of the fingers were consider as a different region due to the region growing algorithm.
- For *mosaic* : mask size = 5 , threshold = 0.08; Post-processing-> median filter with size = 15. The resulting image is presented in Fig 9. As it can observe there

is over-segmentation in one of the textures, this means that the parameters chosen did not fit for that particular texture.

- For *path* : mask size = 21 , threshold = 0.13; Post-processing-> median filter with size = 7. Shown in Fig 10. In this case the three main regions where detected as textures but a lot of small regions appears even using a median filter. In fact, the use of this filter could join two separated and similar regions, if the separation between them is small enough.
- For *ceiling* : mask size = 5 , threshold = 0.08; Post-processing-> median filter with size = 7. Shown in Fig 11. As it is reflected in the image the texture of the ceiling is well segmented, but not the separation between the different blocks. It could be possible because of the algorithm itself (parameters not optimized for this texture) or because of the illumination.

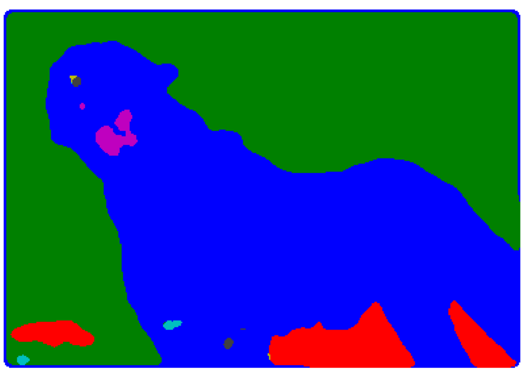

Figure 7. Best result for *felino*

Figure 8. Best result for *hand*

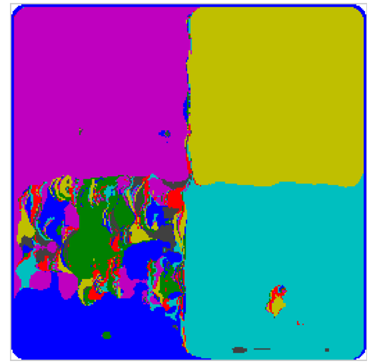

Figure 9. Best result for *Mosaic*

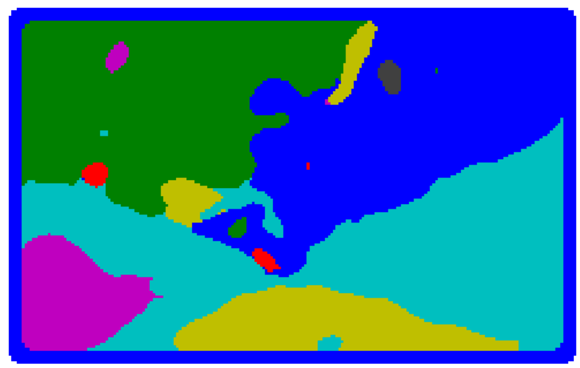

Figure 10. Best result for *Path*

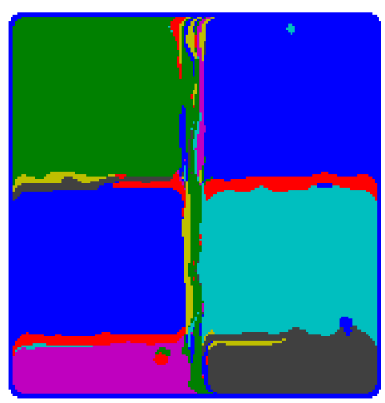

Figure 11. Best result for *Ceiling*

The election of a greater window size leads to a faster processing time but as it can be checked in the Fig 12-*b).* it over-segment the image. Also small details are lost during the process as shown in Fig 13-*b).* where the ring disappears and some regions are merged. For the Fig 12. it toke 19 minutes for image a) and 17 for image b) .

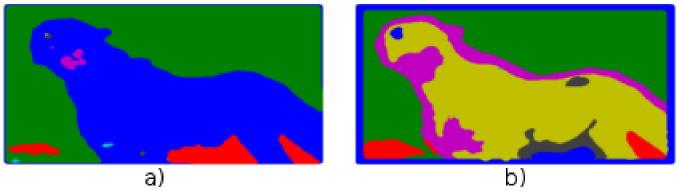

Figure 12. (a) Window size = 5, (b) Window size = 21

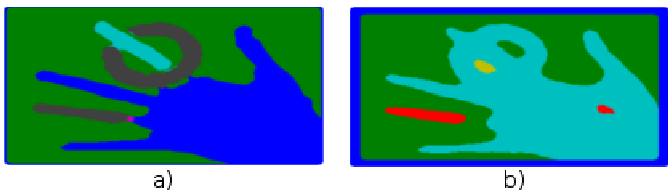

Figure 13. (a) Window size = 5, (b) Window size = 21

The figures 14 and 15 shows the advantages of using the texture features in comparison with only using the color features. It can be notice that with texture features the texture was considered as only one region and for color as multiple regions.

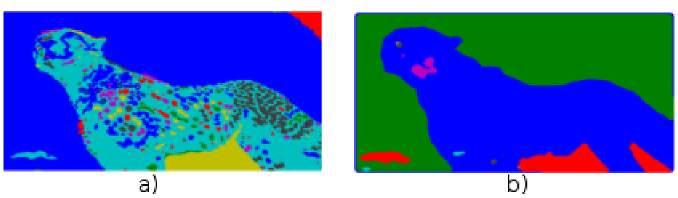

Figure 14. Comparison with only RGB and RGB (a) + features for *felino* (b)

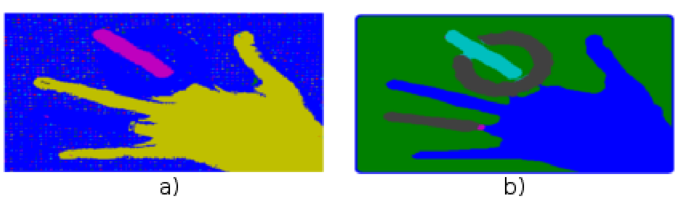

Figure 15. Comparison of RGB and RGB (a) + features for *hand* (b)

IV. Conclusions

Normally are used statistics that measure a collective feature such as the entropy, energy or homogeneity that represents pretty well the textures. After the experiments performed over the set of images the conclusions are the following:

- There is no an unique classifier that works properly with all kind of textures. For each application the number of features and its parameters should be optimized.
- The use of a big quantity of features doesn't imply that result will be better than with fewer features.
- The use of a bigger amount of gray levels is always returning a better result, but it will be much more computational expensive.
- The use of bigger distance doesn't imply a different in efficiency of matching.
- Adding the color channels as features increase up to 90 % the efficiency of the classifier performance.
- The use of a median filter as post-processing helps to homogenize the image, reducing the number of little spots that could appear. On the other hand, if the mask is too big it could be possible that the separation between two areas with the same texture disappears (as in the Fig 10).

References

[1] Shapiro and Stockman, Computer Vision, Prentice-Hall, 2001