Scene Segmentation and Interpretation Image characterization: Texture analysis

Babak Rezaei, Oscar E. Ramos
MSc in Computer Vision and Robotics
Universitat de Girona, Spain 2010

Abstract. This report presents the documentation of the process needed for characterizing an image with respect to its texture. Details of the implementation and development of the algorithm are presented and the pros and cons are discussed empirically based on the results obtained.

I. INTRODUCTION

IMAGE characterization consists in obtaining properties that represent regions of the image based on different criteria like the particularities of surface and structure. Texture is one of the criterions that can be used to characterize the image. It can be understood as a repetition, either deterministic or random, of an element or pattern on a surface [1][2], leading intuitively to properties such as smoothness, coarseness or regularity [3]. For example, the pattern which characterizes the surface of wood is different from that one of grass or sand. Texture analysis is important because it constitutes a major step in texture classification, image segmentation and image shape identification [1]. Textures can be roughly classified as artificial and naturals depending on their origin [4]. Since they present very different characteristics, there is no generic texture model that can properly describe them [5]. Different approaches have been proposed and usually trial and error experiments are preferred to select the best characterization and to tune the parameters adequately.

Methods for characterizing the texture can be grouped as statistical, structural, modelization and space frequency filtering. Statistical methods analyze the spatial distribution of gray values by computing local features at each point in the image, and deriving statistics from the distributions of the local features. The most used methods in this category are energy masks, co-occurrence matrices, parametric masks and local binary patterns.

The focus of this report is on co-occurrence matrices and energy masks. The first ones were introduced by Haralick [6] and consist in measuring the probability of finding two pixels with certain values in a direction inside a window. Then, the statistics like energy, entropy, contrast, homogeneity, probability, correlation, among others, are computed. Energy masks filter small windows with predefined masks and then the mean and standard deviations are computed. The aim of this work is to produce texture descriptors using Grey Level Co-occurrence Matrix (GLCM) and the law filters (masks). As these descriptors are to be used for tasks such as image segmentation they should provide discriminating features for each one of the different textures. Texture descriptors created by GLCM are the statistics of the co-occurrence matrix of each pixel neighborhood in the image.

II. ALGORITHM

A. Co-occurrence matrices

Co-occurrence matrices M represent the probability of finding two pixels with a certain gray value and a certain oriented distance between them. An image is divided into many subimages or windows w x w among which the matrices will be computed. The window size depends on the texture itself and has to be sufficiently big as to capture the texture properties. For an image with N gray levels, the associated co-occurrence matrices (for each window) will be of size NxN ranging in each column and row from 0 to $N-1$. Each of the elements of the matrix will be called m_{ij}, where i and j describe the row and column position respectively. Then, a certain "rule" that describes the distance and orientation between two pixels, the selected one and its neighbor, in the window is given. The orientation is usually $0°$, $45°$, $90°$ and $135°$, as shown in Figure 1. These angles represent the bidirectional orientation, for instance, for $0°$ it would be equivalent to finding the neighbor that lies in the $0°$ direction as well as in the $180°$ direction. The distance is given in numbers of pixels, for instance, in figure 1 the distances from the middle pixel to the shown neighbors in all the directions has been restricted to 1.

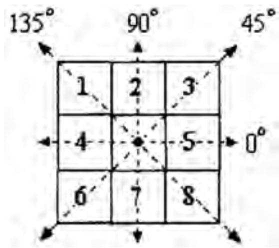

Fig. 1. Neighbors of a pixel, with different orientations.

One example of the “rule” is distance 1 and orientation *45°*. In this case, for Figure 1, considering the central pixel, the neighbors that satisfy the condition are 3 and 6. If we consider that the gray levels of the image are represented by n_i, then the above ideas can be formalized in the following way: The co-occurrence matrix *M* is composed of m_{ij} elements, where m_{ij} is an estimate of the joint probability that a pair of points satisfying the “rule” (distance and orientation) will have gray values n_i and n_j.

Once the co-occurrence matrix *M* has been computed, the statistics related to it are obtained. The statistics used in this work were the energy, homogeneity, contrast and entropy, which, using the previous notation, can be expressed as follows:

$$Energy = \sum_{i,j=0}^{(N-1)^2} m_{ij}^2 \tag{1}$$

$$Homogeneity = \sum_{i,j=0}^{(N-1)^2} \frac{m_{ij}}{1+|i-j|} \tag{2}$$

$$Contrast = \sum_{i,j=0}^{(N-1)^2} m_{ij}(i-j)^2 \tag{3}$$

$$Entropy = \sum_{i,j=0}^{(N-1)^2} m_{ij}\, log_2(m_{ij}) \tag{4}$$

Note that for the entropy, the $\log_2$ was used instead of the $\log_{10}$. This won’t make difference since it only introduces a scale factor of $(log_2 10)^{-1}$ with respect to the computation using $log_{10}(m_{ij})$, since:

$$log_{10}(m_{ij}) = \frac{\log_2(m_{ij})}{\log_2(10)} \tag{5}$$

Besides that, at the end, the matrices are normalized, so the factor is completely eliminated. Summaryzing, the steps needed for computing the co-ocurrence matrices and the statistics are the following:

1. Choose a window size *w* x *w* and divide consequently the image into smaller windows or sub images.
2. For each window, compute the co-occurrence matrix M with N gray levels a direction *d* and a orientation θ.
3. Using the previous matrix M compute the statistics given by (1) to (4).

These steps have to be realized for all the existing windows in the image.

B. Energy masks

These masks were proposed by Laws [8] and they consist on a set of kernels that are convolved with each of the windows obtained from the image. After the convolution, the statistics, like the mean, standard deviation and absolute mean are computed to characterize the texture. Laws texture energy measures certain texture properties by assessing Average Gray Level, Edges, Spots, Ripples and Waves in textures. To derive these measures, the vectors shown in (6) were proposed.

$$L_3 = \begin{bmatrix}1\\2\\1\end{bmatrix} \quad E_3 = \begin{bmatrix}-1\\0\\1\end{bmatrix} \quad S_3 = \begin{bmatrix}-1\\2\\-1\end{bmatrix} \tag{6}$$

Averaging is performed by L_3, E_3 calculates first differences or edges and S_3 performs point detection since it corresponds to the second differences. Using the multiplication of these vectors the so called “Laws Masks *3*x*3*” are obtained. The order of multiplication is important and making the possible combinations we get 9 masks, which are L_3L_3, E_3L_3, S_3L_3, L_3E_3, E_3E_3, S_3E_3, L_3S_3, E_3S_3, S_3S_3. For instance, to obtain E_3L_3, the following operation is performed:

$$E_3L_3 = E_3L_3^T = \begin{bmatrix}-1\\0\\1\end{bmatrix}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = \begin{bmatrix}-1 & -2 & -1\\0 & 0 & 0\\1 & 2 & 1\end{bmatrix} \tag{7}$$

The vectors shown in (6) are also convolved to obtain the following vectors: $L_5=L_3*L_3$, $E_5=E_3*L_3$, $S_5=S_3*L_3$, $R_5=S_3*S_3$, where “*” denotes convolution. The new vectors are:

$$L_5 = \begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix} \quad E_5 = \begin{bmatrix}-1\\-2\\0\\2\\1\end{bmatrix} \quad S_5 = \begin{bmatrix}-1\\0\\2\\0\\-1\end{bmatrix} \quad R_5 = \begin{bmatrix}1\\-4\\6\\-4\\1\end{bmatrix} \tag{8}$$

These vectors can be interpreted in terms of the main operations they perform. In this case, level is given by L_5, edge by E_5, spots by S_5 and ripples by R_5. Using these new vectors, the so called “Laws Masks *5*x*5*” E_5L_5, E_5S_5, L_5S_5 and

R_5R_5 are obtained as in the case of the $3x3$ masks. These masks are then convolved with the window.

Considering that the window size is wxw and that calling the resultant matrix of the convolution Y (with elements y_{ij}), the statistics used for this case are:

$$Mean = \mu = \frac{\sum_{i,j=1}^{w\times w} y_{ij}}{w \times w} \quad (9)$$

$$\sigma^2 = \sum_{i,j=0}^{w\times w} \frac{\left(y_{ij} - \mu\right)^2}{w \times w} \quad (10)$$

$$Absilute\ mean = \frac{\sum_{i,j=1}^{w\times w} \left|y_{ij}\right|}{w \times w} \quad (11)$$

Summarizing, the steps needed for the implementation of the Laws masks are:

1. Choose a window size *w* x *w* and divide consequently the image into smaller windows or subimages.
2. Convolve each window with the Masks (3x3 or 5x5).
3. Using the resultant matrix of the convolution, compute the statistics shown in (9) to (11).

III. Design and Implementation

The texture characterization algorithms described in the previous section were completely implemented in Matlab. For the co-occurrence matrix, all the functions were implemented since the functions provided by Matlab are very general and they also use other very general functions, which take some time doing certain verifications of the number of inputs, or the types, and that is very time consuming in this case, since the co-occurrence matrix is exhaustively calculated many times. The implementation done takes approximately 20 seconds but the time depends obviously on the parameters that are used. However, it is much faster than the normal Matlab implementation. Table I gives a summary of the algorithm implemented.

TABLE I
ALGORITHM USING THE CO-OCCURRENCE MATRIX

1. Choose the size of the window w as an odd number.
2. For every window in the image
   a. Compute the co-occurrence matrix
   b. Compute the statistics
3. Normalize the images.

The normalization is performed using the function *mat2gray*, since it automatically resizes the image to the range 0 to 1. For the characterization using the Laws masks, the procedure is basically the same and the algorithm is shown in Table II.

TABLE II
ALGORITHM USING THE LAWS MASKS

1. Choose the size of the window w as an odd number.
2. For every window in the image
   a. Convolve the window with the mask
   b. Compute the statistics
3. Normalize the images.

IV. Experimental Results and Analysis

This section provides the results of the experiment carried out for finding the descriptors for images. As explained in part II, in order to extract texture descriptors using *GLCM* we should provide some parameters such as the distance and angle of the neighboring criteria and size of the window. The values of the statistics (entropy, homogeneity, etc) are dependent on these parameters and setting them properly can make a difference when finding the discriminating texture descriptor. There is no general rule for setting the parameters such that they provide an appropriate texture descriptor for all textures, but the analysis of the texture can give us some ideas to restrict the choice of the parameters to a reasonable range. In order to find the parameters, a backtracking is required. First we should see what each of the statistics obtained from the co-occurrence matrix represent. In other words, what form of co-occurrence matrix results in a high or low statistic (i.e. energy, entropy, etc.).

Energy: energy is the sum of squared elements (1) in the *GLCM* matrix. Since the elements are probabilities, the more disperse (less sparse) the matrix, the smaller the probabilities of each element. And because squaring makes the scalars smaller, it will also make the sum of all smaller. Thus, a disperse and non sparse co-occurrence matrix results in low energy and a sparse matrix results in high energy.

***Homogeneity*:** as (2) suggests, this criteria is high if the matrix is concentrated on the diagonal, which means that the patch image used has a uniform texture. On the other hand, low homogeneity means that large number of sharp edges are present in the patch. (See Figure 2)

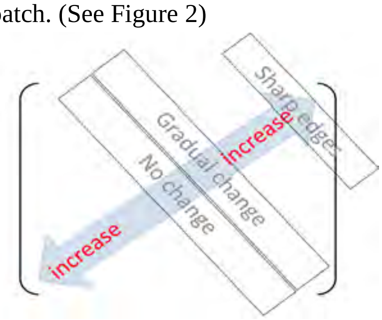

Fig 2. Illustration of the meaning of elements in the co-occurrence matrix.

Contrast**:** this value is high when there is a high number of adjacent pixels in the patch which have significant differences in terms of gray level. A high contrast patch generates a co-occurrence matrix with high values far from the diagonal and a low contrast image has high values in the diagonal.

Now the question is, what forms a sparse co-occurrence matrix? As an example, Figure 3.a shows a patch from the body of the cheetah. As it can be seen, there are gradual increases in intensity levels in east, west north and south direction. Let's assume a *8* by *8* co-occurrence matrix. Due to the quantization of gray levels into 8 bins, some of the gray levels will be put into the same category. This suggests that some elements in the diagonal of the co-occurrence matrix will have high values. Thus, increasing the distance to 2 or 3 (i.e. 2/3 pixels to left) will increase the value of the elements close to the diagonal. This results in a high low pattern in a neighborhood, similar to the patch of the cheetah's body shown in figure 3.a. Defining such co-occurrence matrix (window size = 30, distance= 3, angle=90) will result in high energy for patches, similar to sky regions, as the co-occurrence matrix will have high value only in very few places in the diagonal. Over all such setting will result in a proper texture descriptor for the image below.

(a)

(b)

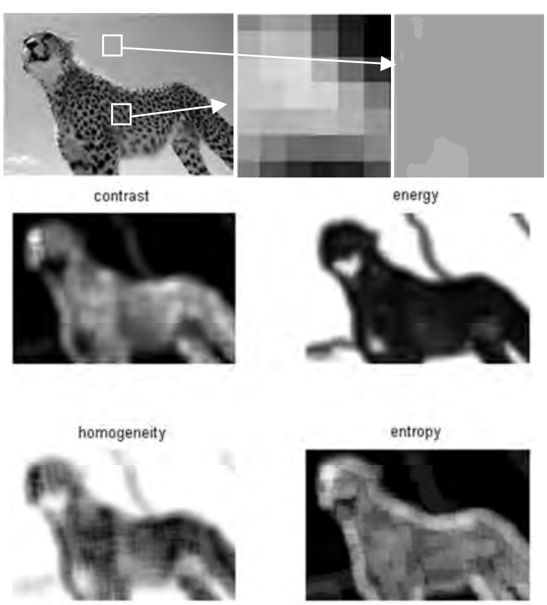

Fig 3. Sample image and textures within the image. Results of texture characterization using GLCM.

As it can be seen in the above picture, the textures have been represented in a separable way. However, there are several distortions (lines) which are created as artifacts of this process. Creation of these lines is due to the selection of distance, direction and size of the co-occurrence matrix. In other words, in the sky texture (in the place where the artifact appears) the gray levels of the pixels happen to fall into different bins of gray levels thus creating values with distance to the diagonal of the co-occurrence matrix. This leads to low energy, homogeneity and high contrast and entropy. To handle this, we will reduce the number of gray levels so that the number of gray levels in the sky will seem more uniform. Care should be taken as over reduction of gray levels may lead to losing reparability of other texture regions. In this case if we reduce the number of gray levels to 4 we still have enough gray levels for accommodating the patterns of the other textures (i.e. body of cheetah). Using these new settings we obtain following results.

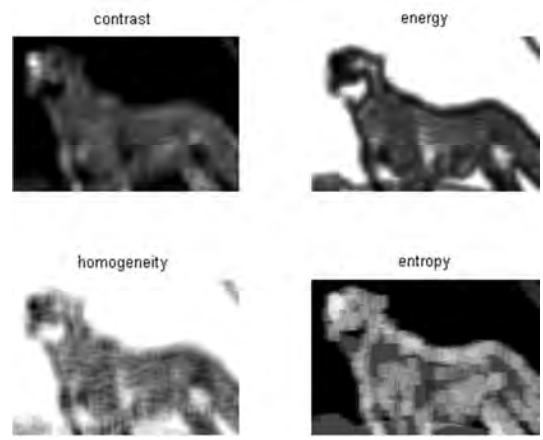

Fig 4. Removing the artifacts by reduction of gray level to 4 and increasing distance to 5 east.

Below are the test results for other images:

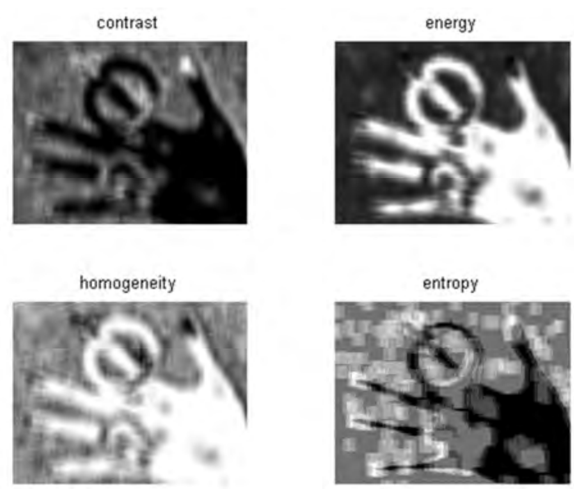

Fig 5. Result of texture characterization with 4 gray levels window size of 15 and distance 5 pixel east. L=4 W= 15 D= [0 5]

As it can be observed, most of the images have artifacts, thus, textures in the image are not separable. This is due to improper parameter setting. Below is another example of improper parameter settings.

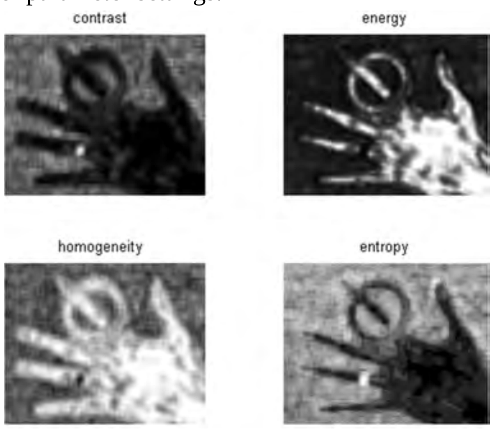

Fig 6. Another example of imporper parameter setting. L=10,W=11,D=[0 1]

Results can be improved by reducing the number of gray level to discount changes of intensity on the hand and using smaller window to reduce the blurring effect on the fingers.

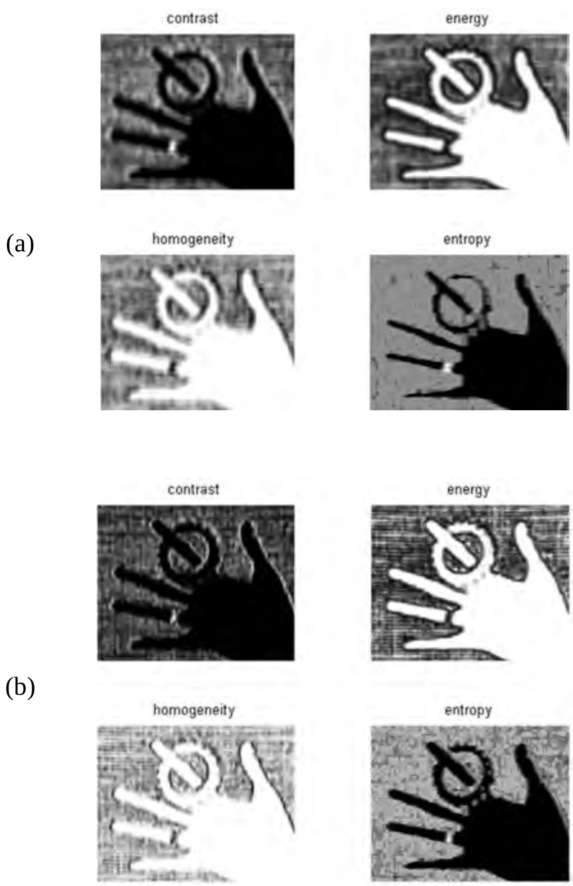

Fig 7. Improved characterization.(a) bigger window, blurred back ground. (b) smaller window, better boundaries .

It should be noted that there is a tradeoff between obtaining the original boundaries of some textures (e.g. hand) and obtaining uniform textures of another (e.g. background).

If the following test case the textures are very similar. There exist 2 main textures which are the hand and the background. The texture of the hand, the ball and the table are all uniform. As the texture of the background is low in variance but changing rapidly (i.e. patches are small) we chose a small window and a low grey level. Figure 8 shows the results of this experiment.

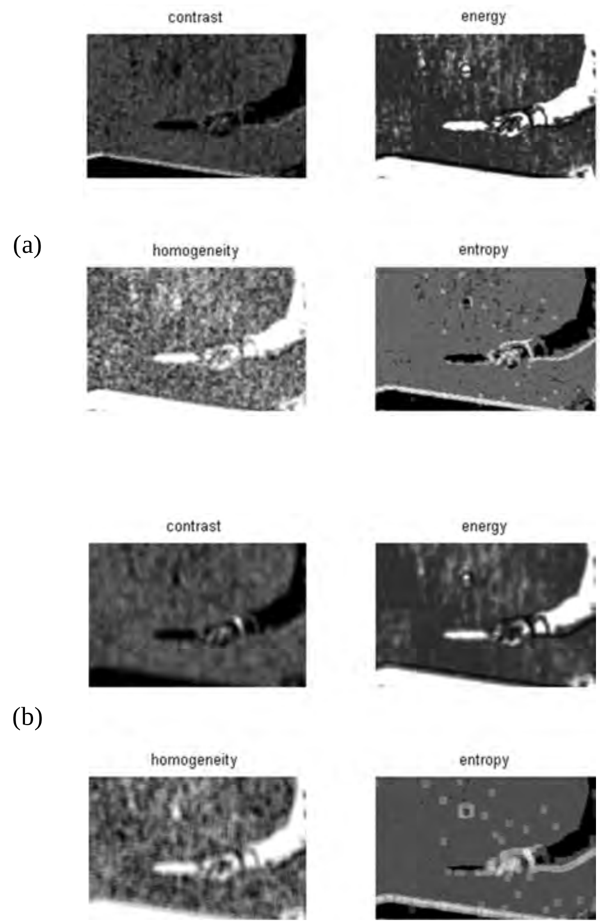

Fig 8. Result of 3rd experiment using GLCM. (a) using smaller window (b) using bigger window.

The forth experiment is done using the most difficult case in which the image is composed of four different textures. Textures in this example are similar in term if co-occurrence of grey levels. Except the top right texture which has finer texture, the other textures produce similar results in most experiments done using different parameters. However, setting parameters as Gray level=4, Window size=19 and distance= [5 3] we could generate result such that at least 3 out of 4 are separable. The results obtained are shown in Figure 9.

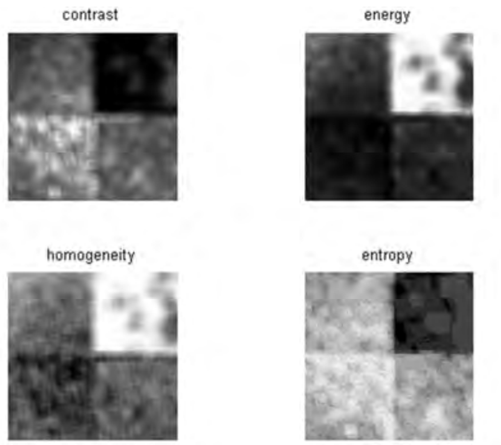

Fig 9. Result obtained for characterization of texture.

As stated before, the method may not be appropriate for some textures. We try to compare the results of GLCM with the results of the Laws masks in the following section. It should be noted that the Laws mask implemented have the capabilities to find ripples, spots, edges etc in a texture. This masks work well when there is clear difference in density of such patterns in the texture residing in the image.

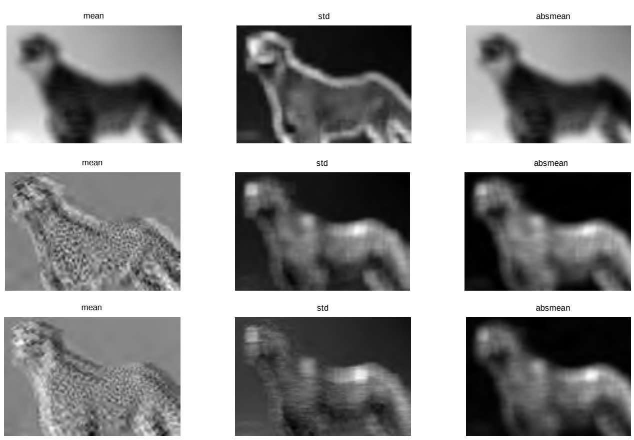

Fig 10. Results of texture characterization using Law masks. (Top row) L3L3. (Mid row)E3E3. (Bottom row) E3S3.

The best result show in Figure 10 is obtained with L3L3 which is a Gaussian filter. Due to the lack of negative values, the results of mean and absolute mean of a neighborhood are the same. The size of the window for this experiment is set to 25 which is almost the size of a Texel in the body of the animal.

Below are the results of the experiments using another test image. In this case, the textures have different intensity values. That is, the back ground is darker than the hand, thus, the Gaussian filter can be used to smooth the image so that both foreground and background are more uniform. A window size of 15 has been used and it provides a good tradeoff between the uniformity of the result and preservation of original boundaries.

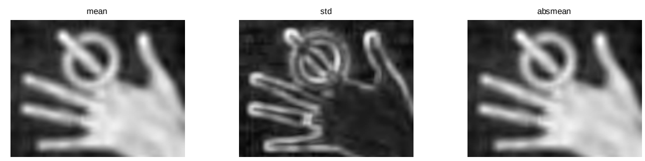

Fig 11. Texture characterization using only L3L3.

In the following experiment L3L3 provides the best results due to the uniformity of the textures in the image. We have chosen smaller window size due to the small size of the Texel of the background. Best results are as below

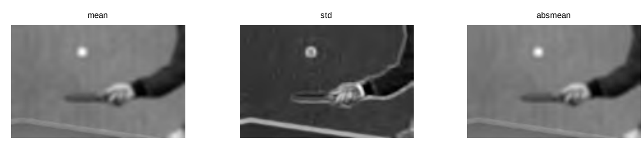

Fig 12. Texture characterization using only L3L3.

To utilize the difference of gray level intensity between textures in the following example we have used the L_3L_3 Filter with window size of 17 which was set with respect to the Texel size. Figure 13 illustrates the results of this experiment with comparison to S3L3 filter with the same window size.

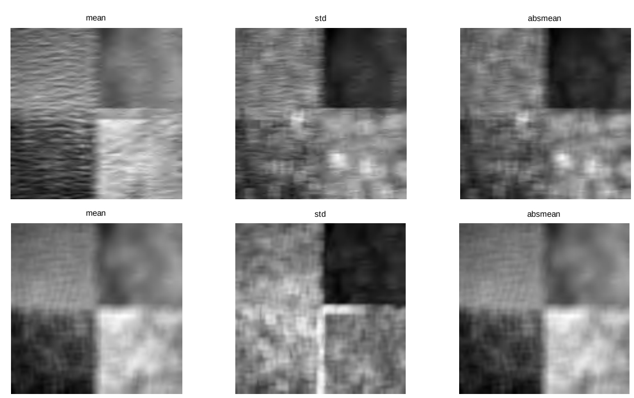

Fig 13. (Top row) Results obtain using S3L3 filter . (Bottom row) Results obtained from L3L3.

It can be observed that the results of the L3L3 are more separable than that of S3L3. The best results in this experiment are obtained using the absolute mean statistics.

As the Characterization done in this work is a preparation step for scene segmentation we have used the region growing segmentation algorithm developed previously to test the performance of the texture descriptors developed in this work practically. Figure 14 shows some of the results obtained.

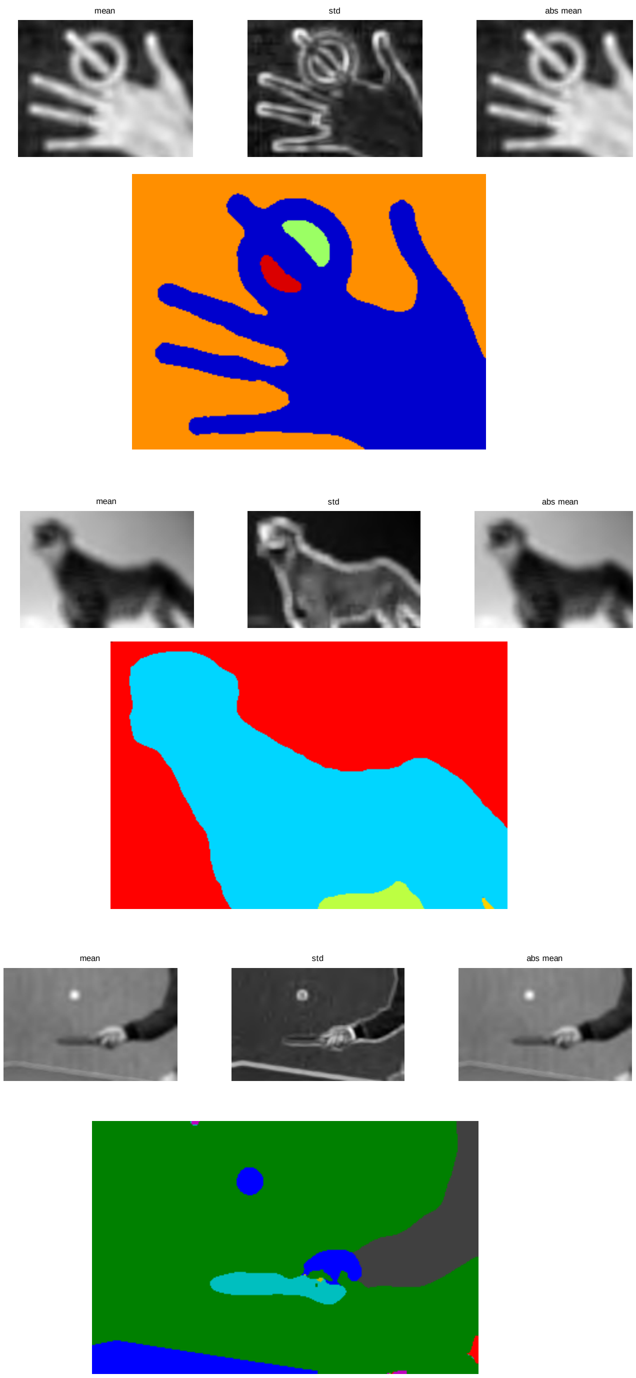

Fig 14. Segmentation using the Texture Discriptors.

V. Further Improvements

Texture characterization is a complex task. For improving some of the poor results obtained in this work we suggest using the edge direction. For example, entropy of edge direction histogram can be used as an approximation to measure directivity. Further, co-occurrence of edge pairs with the same edge direction at constant distances can be used as measure of linearity for characterizing the texture[9]. These improvements may contribute in cases such as the last test case in which one of the significant discriminant features of the textures in the image was the direction of edges. Other co-occurrence based statistics can also be used to characterize textures, for example, auto correlation, which is based on finding the linear spatial relationships between different primitives.[10]

VI. Conclusions

To sum up the experiment we can say that texture characterization using the statistics of the co-occurrence matrix and the Laws Masks is an empirical way to describe properties of texture, since there are many degrees of freedom when choosing the parameters. However, some insight into how the texture itself is can help when deciding which parameters approximately have to be chosen, but experiments have to be carried out to obtain acceptable results. The distance that repetitions of the texture structure present determines the window size and helps to decide which direction and orientation to consider.

VII. References

[1] G. Srinivasan, G. Shobha, “Statistical Texture Analysis”, in *Proceedings of world academy of science, engineering and technology*, Vol.36, 2008 ISSN 2070-3740.

[2] S.W. Zuckera and D. Terzopoulosa, “Finding structure in Co-occurrence matrices for texture analysis” in *Computer Graphics and Image Processing*, Vol.12, Issue 3, 1980.

[3] R. C. Gonzales and R.E. Woods, *Digital Image Processing*, Pearson/Prentice Hall, Third Edition, 2008.

[4] W.K. Pratt, *Digital Image Processing, PIKS Scientific inside*, John Wiley & Sons, Fourth Edition, 2007.

[5] R. Zwiggelaara, et al. “Texture Segmentation in Mammograms”, in *Medical Image Understanding and Analysis*, 2003.

[6] R.M. Haralick, et al., “Textural features for image classification”, in *Proc. IEEE 67*, 1979.

[7] N. Pican et al. “Texture analysis for seabed classification: co-ocurrence matrices”, in *Proceedings from Oceans*, 1998.

[8] K. Laws, “Rapid texture identification”, in *Image Processing for Missile Guidance*, Vol. 238, 1980.

[9] G. N. Srinivasan, and G. Shobha G.” Statistical Texture Analysis”, in *Proceedings of world academy of science, engineering and technology*, volume 36 december 2008 ISSN 2070-3740.

[10] M. Singh and S. Singh, “Spatial Texture Analysis: A comparative Study”, *PANN* Research, Department of Computer Science,University of Exeter, Exeter, UK.