TEXTURE FEATURE EXTRACTION TECHNIQUES

Oana Astrid VATAMANU

University of Medicine and Pharmacy "Victor Babeş", Department of Medical Informatics

Simona APOSTOL

"Tibiscus" University of Timişoara

Mihaela IONESCU

University of Medicine and Pharmacy "Victor Babeş", Department of Medical Informatics

ABSTRACT: Texture plays an important role in numerous computer vision applications. Many methods for describing and analyzing of textured surfaces have been proposed. Variations in the appearance of texture caused by changing illumination and imaging conditions, for example, set high requirements on different analysis methods. In this thesis, methods for extracting texture features and recognizing texture categories using grey level first-order and second-order statistics, edge detectors and local binary pattern features are proposed. Unsupervised clustering methods are used for building a labeled training set for a classifier and for studying the performances of these features.
KEYWORDS: texture analysis; classification; grey level statistics; local binary pattern.

I. INTRODUCTION

Texture can be defined as a visual pattern of repeating elements that have some amount of variability in element appearance and relative positions. So the key notion is that there are several elements that repeat; they don't necessarily look exactly like each other and their placement is not necessarily identical from element to element, but there is some amount of irregularity in the appearances of each element. In computer vision, an appearance view is captured using digital imaging and stored as image pixels. Researchers have shown that texture provides information in the spatial arrangement of colors and intensities in the image. Image resolution is important in texture perception, and low-resolution images contain typically very homogenous textures. Despite the lack of a theory, we all know that real-world objects and surfaces are not flat, nor uniform, and there are numerous potential computer vision applications that could utilize texture information. Typically textures and the analysis methods related to them are divided into two main categories with different computational approaches: the stochastic and the structural methods.

II. TEXTURE ANALYSIS

A. *Texture Analysis Workflow*

The goal of texture analyzing in computer vision is to understand, model and process a texture and eventually, simulate the entire human visual learning process using computer technologies.
Any computer vision system can be divided in several components that interact to each other as described in the image below. Once the image is obtained and all necessary transformations for further processing have been done, the above texture analysis stages can be applied. First the image can be pre-processed or segmented into well defined regions. Each region will be characterized by a different texture. In the next two steps, the textures can be classified, added to a pattern or object set.

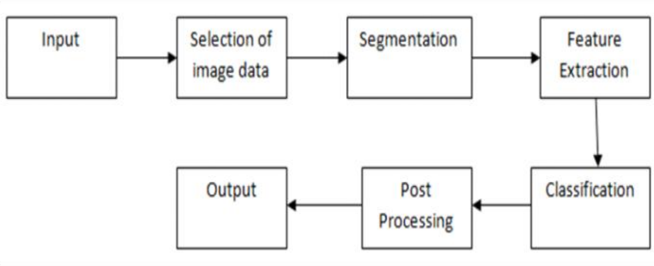

Figure 1.

B. *Texture Classification*

Texture classification is an important stage in texture analyzing and has been intensely studied and implemented in pattern recognition and computer vision related systems. An important application of the image texture is to identify distinct regions in an image using texture properties. The goal of texture classification is to categorize unknown objects from the input image into several distinct classes, or find the probability that one such object maps to a predefined category. Basically there are two important steps to follow when performing texture classification: the first one is to identify the clusters and their association with statistical classes; clusters are group of pixels that have the same spectral characteristics. This is the phase where we should separate the texture classes and build a model for the image content, for every class defined in the training data set. The training data set is built from images that have known class labels. The texture content must be defined using different image analysis methods, which eventually will lead to a set of textural features for each image in the training set. These features that can be represented in many forms: numbers, 2D histograms, empirical distributions, characterize given textual image properties, such as contrast, orientation, spatial structure and so on. The second stage in texture classification is the recognition phase. Once the knowledge is available and the texture features are at the user's disposal, one can use classical pattern classification tehniques in order to do the classification. The features that were extracted from a sample are compared with master textures from the training set using a classification algorithm, and the tested sample is assigned to a class where it fits the best.

C. Texture Segmentation

Segmentation is one of the most important steps in image analysis. The aim is to divide the image into regions that have a strong correlation with objects or image surfaces. It considers that the image is divided into disjoint regions that are homogeneous relative to some property such as brightness, color, background, reflectivity and so on. Image segmentation algorithms can also be categorized into supervised and unsupervised methods. When we refer to unsupervised classification we don't know anything about the objects, the parameters that define the image and the algorithms should be written so that they find the regions in the image automatically. Sometimes it's much easier to use supervised algorithms, where part of data (object types, number of objects) is known apriori. Typically texture segmentation algorithms try either to find the homogenous regions (region based) from the image or locate the texture in homogeneity for detecting the boundaries between regions (boundary based). Both supervised and unsupervised methods are widely used [CK05]. The first one is probably more suitable in image processing tasks; the second one is closely related to image classification.

D. Texture Shape

Shape from texture is the problem of estimating a 3D surface shape by analyzing texture property of a 2D image. Weak homogeneity or isotropy of a texture is likely to provide a shape cue [Cle02]. For instance, texture gradient is usually resulted from perspective projection when the surface is viewed from a slant, which infers the parameters of surface shape or the underlying perspective transformation. Therefore, via a proper measure of texture gradient, a depth map and the object shape could be recovered.

E. Texture Synthesis

In computer graphics, texture synthesis is a common technique to create large textures from usually small texture samples, for the use of texture mapping in surface or scene rendering applications. A synthetic texture should differ from the samples, but should have perceptually identical texture characteristics [DeB97]. The main advantage of texture synthesis in this case is that it can naturally handle boundary condition and avoid verbatim repetitions. In computer vision, texture synthesis is of interest also because it provides an empirical way to test texture analysis. Because a synthesis algorithm is usually based on texture analysis, the result justifies effectiveness of the underlying models. Compared to texture classification and segmentation, texture synthesis poses a bigger challenge on texture analysis because it requires a more detailed texture description and also reproducing textures is generally more difficult than discriminating them.

III. STATE OF THE ART

Texture perception is very different in many ways, thus there are several methods to represent texture from a variety of images

A. Model Based Methods

This approach takes into consideration the image model that describes the texture using a parametrically view. These methods are usually used for specific textures analysis tasks like using fractals to adjust the textural properties of images. In practice, the computational complexity arising in the estimation of stochastic model parameters is the primary problem. The fractal model has been shown to be useful for modeling some natural textures. It can be used also for texture analysis and discrimination however; it lacks orientation selectivity and is not suitable for describing local image structures.

This model has proved good results especially in synthesize of images. A very popular approach for modeling images is the Markov random fields approach. They are able to capture the local (spatial) contextual information in an image. These models assume that the intensity at each pixel in the image depends on the intensities of only the neighboring pixels. MRF models have been applied to various image processing applications [CJ83].

B. Structural Methods

This texture analysis method can be categorized as a heading of geometrical (structural) methods, which main property is that of being composed of texture elements (textons). This class of texture analysis usually relies on processing the geometric properties of these texture elements. To describe the texture, we need to define the primitives and their placement rules. The choice of a primitive (from a set of primitives) and the probability of the chosen primitive to be placed at a particular location can be a function of location or the primitives near the location. The advantage of the structural approach is that it provides a good symbolic description of the image; however, this feature is more useful for synthesis than analysis task. The method was successfully applied in medicine, especially for detection of changes in bone micro-structure.
Zucker [Zuc76] has proposed a method in which he regards the observable textures (real textures) as distorted versions of ideal textures. The placement rule is defined for the ideal texture by a graph that is isomorphic to a regular or semi regular tessellation. These graphs are then transformed to generate the observable texture. Which of the regular tessellations is used as the placement rule is inferred from the observable texture. This is done by computing a two-dimensional histogram of the relative positions of the detected texture tokens.

C. Signal Processing Methods

Signal processing methods represent an image in a space whose coordinate system has an interpretation that is closely related to the characteristics of a texture (such as frequency or size). Both spatial and frequency domain approaches can be used for filtering images and capturing relevant information. Methods based on the Fourier transform perform poorly in practice, due to its lack of spatial localization. Gabor filters provide means for better spatial localization; however, their usefulness is limited in practice because there is usually no single filter resolution at which one can localize a spatial structure in natural textures. Compared with the Gabor transform, the wavelet transforms feature several advantages:

- varying the spatial resolution allows it to represent textures at the most suitable scale there is a wide range of choices for the wavelet function, so one is able to choose
- Wavelets best suited for texture analysis in a specific application.

They make the wavelet transform attractive for texture segmentation. The problem with wavelet transform is that it is not translation-invariant.

D. Statistical Methods

The main quality of these methods is the spatial distribution of grey values. These are the earliest methods of texture analyzing in computer vision. Texture is described by a collection of statistics and selected features (mean, variance). These features can also be classified in first-order statistics – apply operators directly on grey pixel values, second-order statistics – calculate the illumination difference for pixels fixed at a distance d one from each other and apply the previous operators. These methods, based on statistics obtained from pairs of pixels have provided higher discrimination rates than the power spectrum and structural methods.
Probably the most important second-order statistical features for texture analyzing are co-occurrence matrices. In paper [Har79] the authors provide a co-occurrence matrix method named (GLCM) which eventually has become one of the most important and widely used statistical derivation approach in texture analyzing.

IV. FEATURE EXTRACTION TECHNIQUES

Several feature extraction operators are discussed in this section. There are several aspects that need to be taken in consideration when choosing such a technique: is it invariant to illumination, rotation, scaling, how much is affected by the input noise and so on.

A. Grey Level Difference Operators

The first order statistical methods provide no information about the repeating nature of the texture. The Grey Level Co-occurrence Matrix method is a way of extracting second order statistical texture feature and contains information about the position of pixels having similar grey values. The co-occurrence matrix is a complex but relatively compact descriptor of the contents of the image. Let's consider the following matrix Pij(x,y). Each element (x,y) in this matrix tells us how many pixels with intensity x have a pixel at intensity y that is i columns to the right and j rows below. So basically, (i,j) tells us how the two pixels are positioned one relative to the other and the (x,y) pair tells us how different their intensities are. All we have to do is to decide which of these points are we interested in and we can use a set of these as a descriptor very rich in information for the image patch. Actually this is one of the first descriptors that were proposed for texture analyzing.
Several features can be extracted and further analyzed:

a) The *energy* or homogeneity sums up all the codes of the matrix and this gives us details on the dispersion of appearance. See figure 1 for description. The homogeneity is also known as Angular Second Moment (ASM).

$$ASM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) * P(i,j)$$

b) If we have a regular texture that keeps getting repeated in exactly the same way over and over again, it means that one of the elements in the co-occurrence matrix will have a huge size and most of the other elements will be close to 0. In this case the entropy is going to be relatively low, because low entropy suggests a spiked aspect. The *entropy* formula is displayed below

$$Entropy = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) * \log\left(P(i,j)\right)$$

c) The *contrast* is also defined as the local variation of the intensity

$$Contrast = \sum_{n=0}^{G-1} n2 \left\{\sum_{i=0}^{G-1}\sum_{j=1}^{G-1} P(i,j)\right\},$$

d) The *variance* operator puts very high weights on the elements that differ from the average value in

$$Variance = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} (i-\mu)^2 * P(i,j)$$

the matrix

B. Edge Detection Operators

The edges of items in an image hold much of the information in the image. The edges tell you where items are, their size, shape, and something about their texture. An edge is where the gray level of the image moves from an area of low values to high values or vice versa. The edge itself is at the center of this transition. Edge detectors can be used as texture operators. This is because a textured area has many edges compared with a smooth area. Applying an edge detector to a texture produces many strong, bright edges while edge detecting a smooth area yields nothing. Smoothing the edge detector result gives a bright area that can be separated from the dark area that lacked edges.

The range operator is an edge detector that does work well on some textures. It takes the pixels in an n x n area, sorts them by value, and replaces the center pixel with the range (the largest pixel value minus the smallest). Other edge operators that have proved good results in texture analysis are the variance, sigma and skewness.

C. Local Binary Pattern

The Local Binary Pattern (LBP) operator is a simple yet a powerful gray-scale invariant texture primitive, derived from a general definition of texture in a local neighborhood. The LBP method can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its invariance against monotonic gray level changes caused, for example, by illumination variations. Another equally important is its computational simplicity, which makes it possible to analyze images in challenging real-time settings. LBP is also very flexible: it can be easily adapted to different types of problems and used together with other image descriptors. Consider the following patch from a texture:

(0,0)	(1,0)	(2,0)
(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)

This is a piece of the image having pixel (1, 1) centered. What we want to do is to calculate the Local Binary Pattern for this pixel. Suppose we have the following grey levels for the above configuration:

90	95	105
101	100	105
90	101	200

From the above we get a binary result: 00111101 = 61, which is the LBP value for pixel (1,1) . Then we perform this step for all other pixels in the image and store the count of LBP for each pel in the texture. Eventually we get 36 unique LBPs or features for an image with 256 grey levels and the resulting histogram will be the model for the texture.

There are several extensions that were added to the initial Local Binary Pattern operator. One is the use of circular neighborhood and bilinear interpolating values at non-integer pixel coordinates which allow any radius and number of pixels in the neighborhood [OPM02]. The gray scale variance of the local neighborhood can be used as the complementary contrast measure.

V. EXPERIMENTS AND RESULTS

Several experiments were performed using the previous texture feature extraction techniques.

The first experiment involves an image that contains two different patterns. The purpose of this experiment is to test the edge detecting operators. Following we have the result of applying the range detector on a grey level image. On the right side we have the output bitmap. The result was normalized, the output histogram was equalized, for a better view on the process. It can be seen that there are clearly two separate objects having different grey levels.

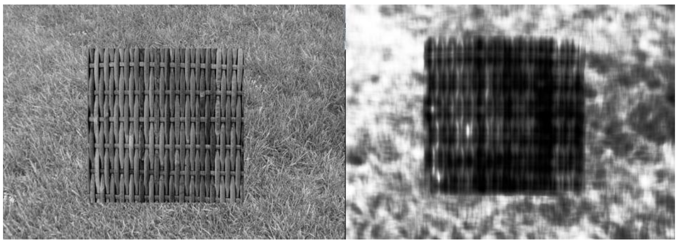

Figure 2.

The next figure displays the results of the skewness operator. This operator is strongly related to the symmetry in the image. Two experiments are presented here. The first one uses the same input image used in experiment 1. As you can notice in the right image, the operator wasn't a correct choice for this texture because nothing can be extracted from the output bitmap.

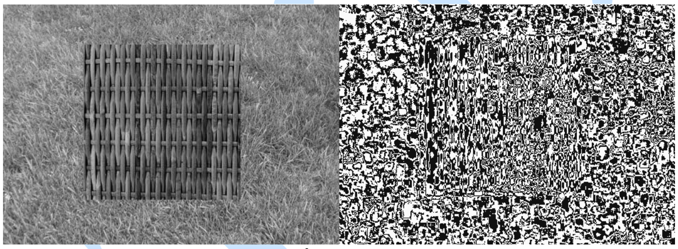

Figure 3.

On the other hand, we have a symmetrical input pattern – chessboard pattern. This pattern had a perfectly symmetrical histogram, so skewness returned zero for every pixel, case when the skewness operator proves excellent results.

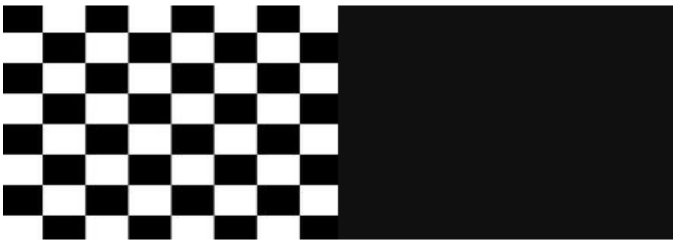

Figure 4.

Another set of tests was done to evaluate the local binary pattern operator's performances. A training set and a unsupervised clustering technique was used to create a model. A test set was compared to this model to check the accuracy of the operator. The model contained local binary pattern values for the training set. Below you can notice an example of a uniform LBP histogram.

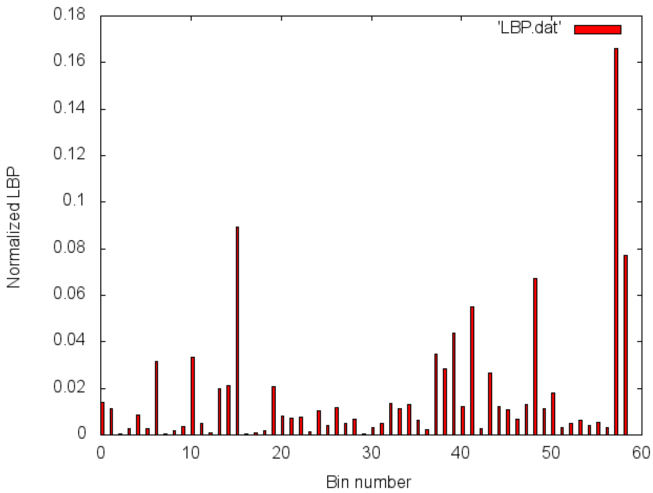

Figure 5.

VI. CONCLUSION

Image analysis applications are seldom solved by using out of-box methods, but often tuning and modification of the algorithms are required. Utilization of texture is typically also not straightforward and one must consider, for example what kind of features are being extracted and how these features are further used.

In this paper, methods were proposed for early stage data analysis to study the performance of texture features. Another goal was to recognize different textures, considering some predefined classes.

ACKNOWLEDGEMENTS

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD) 2007-2013, financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/107/1.5/S/82839.

REFERENCES

[Cle02] **M. Clerc** - *Weak homogeneity for shape from texture.* In Texture 2002 Workshop in conjunction with ICCV 2002, pages 31-34, 2002.

[CJ83] **G. C. Cross, A. K. Jain** - *Markov Random Field Texture Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5, pp. 25-39, 1983.

[CK05] **J. L. Chen, A. Kundu** - *Unsupervised texture segmentation using multichannel decompositions and hidden Markov models*. IEEE Transactions on Image Processing 4(5): 603–619, 2005

[DeB97] **J. S. DeBonet** - *Multiresolution sampling procedure for analysis and synthesis of texture images*. Computer Graphics, 31(Annual Conference Series):361-368, 1997

[Har79] R. M. Haralick - *Statistical and Structural Approaches to Texture*, Proceedings of the IEEE, 67, pp. 786-804, 1979

[OPM02] **T. Ojala, M. Pietikainen, T. Maenpaa** - *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on PatternAnalysis and Machine Intelligence 24(7): 971–987 , 2002

[Zuc76] **S. W. Zucker** - *Toward a model of Texture*, Computer Graphics and Image Processing, 5,pp. 190-202, 1976