Texture Feature Extraction for a Visual Inspection of Ceramic Tiles

I. Novak*, Ž. Hocenski**

* Faculty of Electrical Engineering/Department of Automation and Computer Engineering, Osijek, Croatia

** Faculty of Electrical Engineering/Department of Automation and Computer Engineering, Osijek, Croatia

***Abstract* – The aim of this research was to determine an acceptable method of texture feature extraction that could be used for a visual inspection of ceramic tiles quality. Traditional extraction of statistical features has been carried out as well as feature extraction based on local binary pattern operators. For the purpose of testing the proposed methods good and defective ceramic tiles have been identified. The obtained results indicate a possibility for developing a system of a visual inspection of ceramic tiles quality and point out local binary pattern operators as a powerful tool applicable in this particular case.**

I. Introduction

In the process of ceramic tiles production a visual inspection is done by human. Production lines move monotonously, and since humans tend not to work well if the job in question is uninteresting, classification often results in sets of inhomogeneous tiles. A bad classification is also influenced by limited capabilities of a human in safety manner as well as fatigue. This is hazardous and unhealthy environment for human beings. Visual quality testing done by machine vision has to be more robust and provide less costly inspection. On the other hand, it is rather difficult to implement human intelligence necessary for solving unexpected situations into a visual inspection machine system.

The influence of training data on decision-making consistency concerning proper operation is extremely large. Therefore, special attention should be paid to its setting up. The training data set must cover every possible occurrence of a ceramic tile. There are also numerous classification methods which determine classes of unknown patterns on the basis of known class affiliation of elements belonging to the training data set [1, 2, 3]. Since generally there is not the best classification method, and the effect of different methods varies by different applications, due to its simple implementation the nearest neighbor method has been selected for the purpose of this research.

Texture feature extraction is a key part of the visual inspection system [4]. The goal is to find the method that can separate those texture features on the basis of which it would be possible to detect a tile defect. Statistical methods of image analysis described in [5,6] have been lately successfully applied relative to texture analysis, thus feature extraction is carried out in accordance with these methods.

A rather recent method of texture analysis based on the local binary pattern operator (LBP) has evolved through several scientific papers [7, 8, 9] and shown a great power of extraction of spatially variant information. These scientific papers encouraged the usage of local binary patterns for this application.

Algorithms are implemented through functions written in MATLAB environment. According to feature vectors determined by some extraction method, a separation of good and defective ceramic tiles is carried out. A certain tile set previously classified by human is used as a training set by classification using the nearest neighbor method.

II. The Problem of Ceramic Tile Inspection

The class of a ceramic tile is determined by measuring dimensions, hardness, porosity, and texture imprint quality. There are various methods for determining physical characteristics of a tile, whereas imprint quality control still lies in the phase of scientific research. Various types of complex textures on ceramic tiles make this problem extremely complicated. This paper does not attempt to define the method that should classify tiles but the method aiming at separating good tiles. With respect to production quantity and major part of good tiles manufactured, the implementation of such system in ceramic tile industry implies a great deal of saving. After the separation of good tiles, the remaining defective tiles can be processed either in the existing or in some alternative way.

For some experimental data 60 ceramic tiles of a rather complex texture were selected, 30 of which were good and 30 defective ones. Fig. 1 shows an example of ceramic tile with two defects. Defective tiles had defects in the form of black dots occurring due to impurity particles during texture imprint. Monochromes images of ceramic tiles 1142x1459 pixels were obtained by a scanner with a resolution of 72 dpi.

Training images were selected by paying attention to different dimensions of defects at various locations on tiles. An experimental procedure was carried out with the training set of 10 images of good and 10 images of defective tiles. The remaining images underwent defect detection.

III. Applied Texture Features

A. First Order Statistical Features

First order statistical features provide information on the distribution of pixels on a digital image, but do not give any information on their relative positions. Thus, these are the features characterizing brightness (strength of the patterns), but not its spatial structure.

Figure 1. An example of ceramic tile with two defects

First order image histogram (*P(I)*) is defined according to [5], as a ratio between the number of pixels with value *I* and the total number of pixels. The characteristic coefficients are defined using (1), (2) and (3).

$$m_i = E\left[I^i\right] = \sum_{I=0}^{N_g-1} I^i P(I) \ , i = 1,2,... \tag{1}$$

$$\mu_i = E\left[(I - E[I])^i\right] = \sum_{I=0}^{N_g-1} (I - m_1)^i P(I) \tag{2}$$

$$H = -E[\log P(I)] = -\sum_{I=0}^{N_g-1} P(I) \log P(I) \tag{3}$$

By arranging the above mentioned coefficients into a vector we obtain a feature vector. The most efficient method so far has been obtained by combining the following coefficients, and therefore they were used in this paper: $\mu = m_1$ - mean, $\sigma_2 = \mu_2$ - variance, μ_3 - central moment related to skewness, μ_4 - central moment related to kurtosis, H – entropy.

B. Second Order Statistical Features

Second order statistical features consider pixels in pairs and give information on their relative positions. Hence, they characterize a texture by providing information on brightness and spatial structure. A pair of pixels is determined by two values, relative distance and relative orientation. Relative distance *d* is measured by the number of pixels (*d*=1 for neighboring pixels, etc.). Relative orientation Φ is quantized in four directions: horizontally, diagonally, vertically, and anti-diagonally (0°, 45°, 90°, and 135°). For every combination of *d* and Φ a two-dimensional histogram is defined, according to [5], as a ratio between the number pixels pairs at distance *d* in direction Φ with values (I_1, I_2) and the total number of possible pairs of pixels.

$$0° : P\left(I(m,n) = I_1, I(m \pm d, n) = I_2\right) \tag{4}$$

$$45° : P\left(I(m,n) = I_1, I(m \pm d, n \mp d) = I_2\right) \tag{5}$$

$$90° : P\left(I(m,n) = I_1, I(m, n \mp d) = I_2\right) \tag{6}$$

$$135° : P\left(I(m,n) = I_1, I(m \pm d, n \pm d) = I_2\right) \tag{7}$$

For each of those histograms we define a field known as a co-occurrence matrix. A co-occurrence matrix for pair (d,Φ) is defined as an N_g x N_g matrix, where N_g is the image depth. E.g. if $N_g = 4$, then the co-occurrence matrix is equal to:

$$A(d,\phi) = \frac{1}{R}\begin{bmatrix} \eta(0,0) & \eta(0,1) & \eta(0,2) & \eta(0,3) \\ \eta(1,0) & \eta(1,1) & \eta(1,2) & \eta(1,3) \\ \eta(2,0) & \eta(2,1) & \eta(2,2) & \eta(2,3) \\ \eta(3,0) & \eta(3,1) & \eta(3,2) & \eta(3,3) \end{bmatrix} \tag{8}$$

where $\eta(I_1,I_2)$ is the number of pixel pairs at relative position (d,Φ) with values (I_1,I_2). R is the total number of possible pixel pairs. The definition of the co-occurrence matrix implies its symmetry, which enables further reduction and gives a possibility of separating certain coefficients that efficiently encode image texture. According to [5, 6], the following seven coefficients are used in this paper: energy, entropy, homogeneity, inertia, correlation, cluster shade, cluster prominence.

By connecting these elements into a vector we obtain a feature vector. There is a great correlation of pixel pairs within small relative distances, whereas by increasing the distance, that correlation vanishes. The consequence is an efficient texture representation by co-occurrence matrices for small relative distances. Co-occurrence matrices are sensitive to changes of brightness, which makes them inadequate for comparing objects recorded in various light conditions.

C. Features of the original LBP operator

The original LBP (local binary pattern) operator was introduced for the first time according to [7]. It operates in the neighborhood of eight pixels, using the value of the central pixel as a threshold. An LBP code of a particular neighborhood is calculated by assigning corresponding weights to every pixel and adding products as shown in Fig. 2. Boundary pixels do not have neighborhoods, therefore a central pixel of first existing neighborhood is pixel addressed by second row and second column. Every existing image neighborhood is encoded by a respective 8-bit code. Thus, there can appear the total of 2^8=256 different LBP codes. The image can be given by an LBP histogram of 256 elements, which is also a feature vector at the same time.

Correlation between pixels is great within a small environment defined by a neighborhood, so that the spatial texture structure can be efficiently represented by this method. The histogram of LBP operators contains only information on local patterns, i.e. spatial image structure, thereby neglecting brightness (strength of patterns) completely.

D. Features of a general LBP operator

A general LBP operator is defined according to [8]. Let texture T of an image local neighborhood be defined as an integrated distribution of grey scale values on P+1 (P>0) pixels.

$$T = t(g_c, g_0, \ldots, g_{P-1}) \tag{9}$$

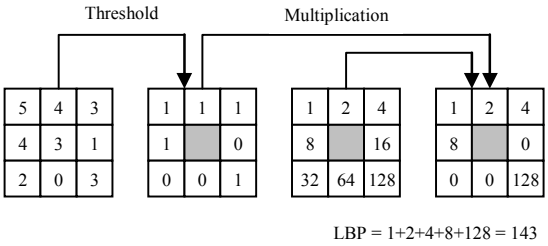

Figure 2. Calculation of LBP code of a particular neighborhood

g_c is the value of the central pixel of a local neighborhood. $g_P(P=0,\ldots P\text{-}1)$ are values of P pixels equidistant along the circumference with $R(R>0)$. The circumference forms a set of circular symmetric neighbors. On the domain of the digital image coordinates of individual neighbors g_p are given by (10).

$$x_c + R\cos\frac{2\pi p}{P},\ y_c - R\sin\frac{2\pi p}{P}, \tag{10}$$

Where (x_c,y_c) are coordinates of the central pixel. Fig. 3 illustrates three circular symmetric neighborhoods for different values of P and R. Values not hitting the exact pixel center are determined by a bilinear interpolation.

If the central pixel value is subtracted from the values of individual neighbors, local texture can be represented as an integrated distribution of the central pixel value and differences using (11).

$$T = t(g_c, g_0 - g_c, \ldots, g_{P-1} - g_c) \tag{11}$$

If we assume that differences are independent of g_c, the distribution can be factorized in the following way:

$$T \approx t(g_c)t(g_0 - g_c, \ldots, g_{P-1} - g_c) \tag{12}$$

Equation (12) has a sign „≈“ since the assumption concerning independence of differences is false, but with an insignificant loss of information we have independence of grey scale shifts. $T(g_c)$ describes a general image brightness and most of the information on original integrated distribution texture remains in the distribution of differences:

$$T \approx t(g_0 - g_c, \ldots, g_{P-1} - g_c) \tag{13}$$

P-dimensional difference distribution records occurrences of various texture patterns in the neighborhood of every pixel. In order to have independence on any monotone grey scale transformation, only the difference signs are taken into consideration.

$$T \approx t(s(g_0 - g_c), \ldots, s(g_{P-1} - g_c)), \tag{14}$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}.$$

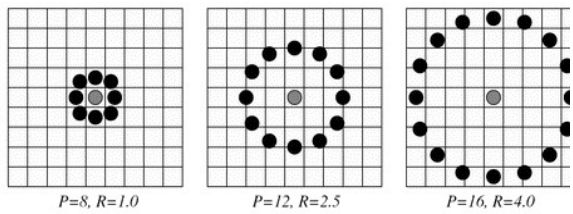

Figure 3. An example of circular symmetric neighborhoods

Binomial weight 2^P is assigned to every sign $s(g_p-g_c)$, transforming neighborhood differences into a unique LBP code using (15), which characterizes local texture around (x_c,y_c).

$$LBP_{P,R}(x_c, y_c) = \sum_{P=0}^{P-1} s(g_P - g_c)2^P \tag{15}$$

An LBP code is a P-bit binary number that can assume 2^P various values. A texture feature vector is obtained as a 2^P element histogram of LBP codes. A balance between P and R values has to be established. LBP code could contain a lot of redundant texture information or could not contain texture information at all if the balance is not established.

Rotation invariance is based upon circular indexing of neighborhoods, since every image pixel is considered to be a rotation center. Rotation of every LBP code into a referential position results in the same LBP code for every occurrence of its rotation versions. That transformation is defined using (16).

$$LBP_{P,R}^{ri} = \min\{\mathrm{ROR}(LBP_{P,R}, i) \mid i = 0,1...P-1\} \tag{16}$$

Where *ri* comes from „*rotation invariant*“. Function *ROR(x,i)* shifts a P-bit binary number x i times to the right.

Fig. 4 shows an example of 8-bit rotation invariant LPB codes. Binary value 0 is represented by black and binary value 1 is represented by white. The first row comprises referential positions of LBP codes, whereas other rows contain LBP codes resulting in the same referential LBP code. We can notice that the first two LBP codes are invariant as such, the third one can specifically occur only in two rotation versions, whereas the other three codes might appear in seven various rotation versions, two of which are represented.

It is also noticed that certain local patterns represent fundamental texture features and that, according to [9], they carry more than 90% of information. These patterns are uniform since they share one common feature, i.e. in their circular binary codes occur at most two transitions one-zero and zero-one. For a formal definition of uniformity a local uniformity measure U of neighborhood G_P is used:

$$U(G_P) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{17}$$

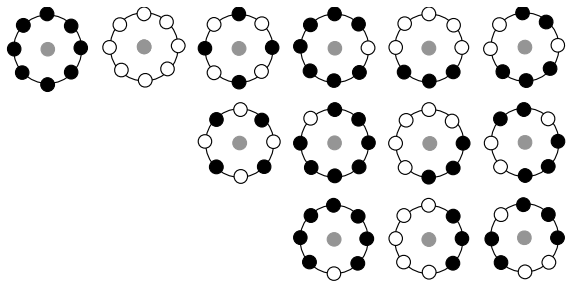

Figure 4. An example of 8-bit rotation invariant LBP codes

Patterns in which value U assumes 2 or less than 2 are called uniform. Uniform LBP patterns can be seen in Fig. 5. The first (00000000_2) and the last (11111111_2) LBP patterns have value U=0. The remaining LBP patterns have value U=2, since in their circular binary code there occur exactly two transitions, one from 0 to 1, and the other from 1 to 0.

The total number of uniform patterns is P+1. The feature vector is obtained as a histogram of P+1 elements of uniform patterns and one element of all other patterns summed together. By applying uniformity and rotation invariance the number of possible LBP codes is significantly reduced. E.g. by applying 8-bit rotation invariant and uniform LBP codes we obtain a histogram of only 10 elements, which represents a significant decrease with respect to a 256-element histogram of the original LBP operator.

IV. Classification method

The problem of separating good ceramic tiles can be considered as a classification of tiles into two groups: good and defective class, whereby defective tiles are all those tiles with any possible sort of damage. There is not a general classification method which could be said to be better than any other, but efficiency of various methods varies depending on the application. For the purpose of a simple implementation, classification in this paper is done according to the nearest neighbor method. The training set of vectors with defined class affiliation is used for determining the class of an unknown vector. On the basis of calculating the unknown vector distance from every individual training vector it is assigned the class of the closest training vector. „Proximity“ of vectors is measured by Euclidean distance. If feature vectors consist of b coefficients and if we denote a pattern feature vector by S, and one of the training feature vectors by T, Euclidean distance D can be calculated using (18).

$$D(S,T) = \sqrt{\sum_{i=1}^{b} (S_i - T_i)^2} \tag{18}$$

Hence, equation (18) should be first applied to all training vectors, and then an unknown vector is assigned the class of the training vector with minimal value D.

V. Experimental Results

As it was expected, the influence of the image spatial structure information is much greater than the influence of the image brightness information, so that the feature extraction methods based upon local binary patterns were much more efficient. For every texture feature extraction method a classification was made with a training set of 10 good and 10 defective tiles. Percentage error ranges from

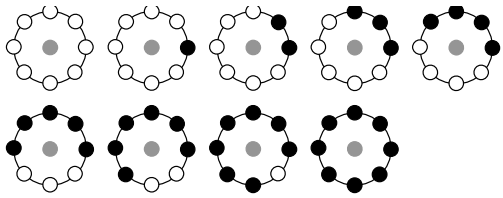

Figure 5. 8-bit uniform LBP codes

TABLE I.
CLASSIFICATION ERRORS IN PERCENTAGES

	false defect error	undetected defect error	total error (%)
STAT_r1	25	20	**22.5**
STAT_r2d1	0	5	**2.5**
STAT_r2d2	0	10	**5**
LBP	10	5	**7.5**
LBP_riu81	15	0	**7.5**
LBP_riu162	10	10	**10**
LBP_riu243	0	0	**0**

22.5% (first order statistical coefficients) to outstanding 0% (LBP_riu243). Percentage errors and classification efficiency are given in Table 1 and Fig. 6, respectively.

Most of the errors are made by the first order statistical coefficient method, since it neglects an image pixel spatial structure. Results of the second order statistical coefficients for distances d_1=1 and d_2=2 are almost identical and characterize homogeneity very well, since they did not make any false defect error in any case. Results obtained by the original LBP operator and LBP_riu81 operator method are identical. These two operators are defined similarly, but the original LBP operator uses 256 elements in the feature vector, whereas LBP_riu81 uses only 10, which is, if we reject the inefficient first order statistical coefficient method, a method representing an image by the smallest feature vector. Despite a more complex algorithm and a greater feature vector, LBP_riu162 operator was a bit worse than LBP_riu81**,** which indicates imbalanced values of number of neighboring points (*P*) and radius of neighborhood (*R*). In both classifications the LBP_riu243 method was dominant, with no errors at all. The balance of *P* and *R* values is obviously well established. That method has the most complex algorithm, starting computing demands on the system. However, feature vectors are still small enough; only 26 elements, which makes the classification process faster.

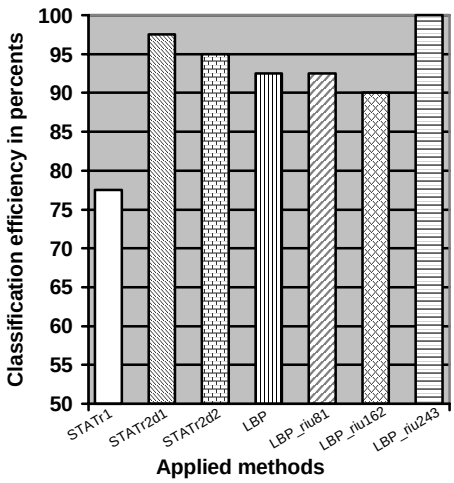

Figure 6. Classification efficiency

VI. CONCLUSION

Results indicate that it is possible to build a visual inspection system of ceramic tiles using one of the proposed texture feature extraction methods. An efficient quality inspection can be reached by analyzing ceramic tiles texture features from an image obtained by a black-and-white camera. Such camera is easily mounted on the assembly line enabling thereby computer-made decisions about possible defects. Such simple implementation of a visual inspection system of ceramic tiles is of great practical importance.

First order statistical coefficients were less efficient. Since they do not have any pixel spatial structure information, but just the information on their values, we can conclude that for defect detection it is important to use features characterizing spatial relations between pixels. All other methods include spatial structure information which in line with that show good results.

The most efficient method of texture feature extraction is the method of a general rotation invariant and uniform LBP operator defined for 24 points distant form the central pixel for radius 3 (LBP_riu243). This method represents the first choice for a visual quality inspection system.

An interesting case is a rather high level of efficiency of an LBP operator defined for 8 points distant from the central pixel for radius 1 (LBP_riu81). Namely, that histogram of LBP codes characterizes the whole tile texture, demanding small computer power, with only 10 elements. By a proper selection of the training set, this method could give excellent results even on very weak computer systems.

REFERENCES

[1] Ž.Hocenski, S.Rimac-Drlje, T.Keser, "*Automatic Inspection of Defects in Plain and Texture Surfaces* ", 5th IEEE Int.Conf. Intell. Engn.Sys.2001, INES 2001,16.-18.9.2001, On Viking Line m/s Mariella on route Helsinki, Finland - Stockholm, Sweden - Helsinki, Finland, 2001., pp.221-227

[2] Ž.Hocenski, A.Baumgartner, "*Image Comparison Method for Visual Quality Control Based on Matrix Decomposition* ", IEEE Int.Symposium on Industrial Electronics, ISIE '2000, Universidad de las Américas-Puebla, México, 4-8 Dec.2000, Vol.2, pp.508-585

[3] Ž.Hocenski, E.K.Nyarko, *Surface Quality Control of Ceramic Tiles Using Neural Networks Approach,* in Proc. 2002 IEEE International Symposium on Industrial Electronics, IEEE-ISIE'2002, July 8/11, 2002, L'Aquila, Italy, CD ROM

[4] R.Haralick, K.Shanmugam, L.Dinstein, "Textural features for image classification," *IEEE Transactions on System Man and Cybernetics*, Vol. 3, 1973, pp.610-621

[5] S. Theodoridis, K. Koutroumbas, "Pattern Recognition", Academic Press, New York, 1998.

[6] A. Bradley, P. Jackway, B. Lovell, "Classification in scale space: Applications to texture analysis", *Proc. 14th International Conference on Information Processing in Medical Imaging*, Ile de Berder, France, Vol. 1, 1995, pp.375-376

[7] T. Ojala, M. Pietikäinen, D. Harwood, "A comparative study of texture measures with classification based on feature distributions", *Pattern Recognition*, Vol. 29, 1996, pp. 51-59

[8] T. Ojala, M. Pietikäinen, T. Mäenpää, "Multiresolution gray scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, 2002, pp.971-987

[9] T. Mäenpää, T. Ojala, M. Pietikäinen, M. Soriano, "Robust texture classification by subsets of local binary patterns", *Proc. 15th International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 3, 2000, pp.947-950