Surface Defect Detection with Histogram-Based Texture Features

Jukka Iivarinen*

Helsinki University of Technology, Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland

ABSTRACT

In this paper the performance of two histogram-based texture analysis techniques for surface defect detection is evaluated. These techniques are the co-occurrence matrix method and the local binary pattern method. Both methods yield a set of texture features that are computed from a small image window. The unsupervised segmentation procedure is used in the experiments. It is based on the statistical self-organizing map algorithm that is trained only with fault-free surface samples. Results of experiments with both feature sets are good and there is no clear difference in their performances. The differences are found in their computational requirements where the features of the local binary pattern method are better in several aspects.

Keywords: unsupervised segmentation, texture segmentation, defect detection, local binary pattern, co-occurrence matrix, statistical self-organizing map

1. INTRODUCTION

Texture is an important characteristic of many types of images and is hence commonly used in image segmentation, referred to as texture segmentation. Many different approaches to texture have been proposed.[1–4] This paper concentrates on feature-based histogram techniques since they are fast to compute (or can at least be approximated with fast algorithms), they are relatively simple methods, and they have been found to be good in classifying stochastic textures of which many real surfaces consist.[5–9] These properties indicate that the histogram-based techniques should be suitable for surface defect detection.

In surface defect detection one of the main requirements is that the defect detection algorithm should be simple, efficient, and realizable in a given hardware architecture. Texture-based techniques have been applied in surface defect detection since the early 1980's[10–16] although many commercial systems today are still based on more simple methods, in particular on gray level thresholding techniques. Some commercial solutions for surface inspection can be found e.g. in Refs. 17–19.

The performance of two feature-based histogram techniques are compared for surface defect detection. These techniques are the classical co-occurrence matrix method and a recently introduced method, the local binary pattern method. These methods are described in section 2. In section 3 they are applied in surface defect detection. The unsupervised segmentation procedure that is used in the experiments is based on the statistical self-organizing map algorithm.[15,20] The statistical self-organizing map is trained only with fault-free surface samples which results in many benefits, e.g. a tedious collection of defects is not needed anymore and appearance of new defects do not result in retraining of a classifier.

2. TEXTURE ANALYSIS TECHNIQUES

2.1. Co-Occurrence Matrices

The co-occurrence matrix method, known also as the spatial gray-level dependence method, has been widely used in texture analysis since the early 1970's.[21] It is based on repeated occurrences of different gray level configurations in a texture. It works well for a large variety of textures, especially for stochastic textures.[5–7]

The co-occurrence matrix is a second order statistical measure of gray level variation. It indicates the joint probability of a gray level occurrence of two pixels at a certain displacement in an image. Let **I** be a discrete image

*E-mail: Jukka.Iivarinen@hut.fi

array with G gray levels and let $\mathbf{d} = (d_x, d_y)$ be a displacement vector. Then the co-occurrence matrix $\mathbf{H}^{CO}$ is defined as a matrix, the (i, j)th element of which is the number of appearances of gray level values i and j in the image $\mathbf{I}$ in the distance and direction determined by the displacement vector $\mathbf{d}$ from each other,

$$H_{ij}^{CO} = \#\{\mathbf{x} \mid I(\mathbf{x}) = i,\ I(\mathbf{x}+\mathbf{d}) = j\}, \tag{1}$$

where $\#$ is the number of elements in the set and $\mathbf{x} = (x, y)$ runs through the image $\mathbf{I}$.

The co-occurrence matrix can be reduced to decrease calculation time and memory requirements. The most usual approach is to calculate a set of features. Haralick et al.[21] suggest 14 textural features, and Conners et al.[22] add two features more. In this study three features, energy, contrast, and mean, are used. They are defined as

$$\begin{aligned}
\text{energy} &= \frac{1}{N^2}\sum_i\sum_j (H_{ij}^{CO})^2, \\
\text{contrast} &= \frac{1}{N}\sum_i\sum_j (i-j)^2 H_{ij}^{CO}, \\
\text{mean} &= \frac{1}{2}(\mu_i + \mu_j) \;=\; \frac{1}{2N}\Big(\sum_i\sum_j iH_{ij}^{CO} + \sum_i\sum_j jH_{ij}^{CO}\Big),
\end{aligned}$$

where i and j run from 1 to G (the number of gray levels), and N is the number of entries in the co-occurrence matrix.

The bottleneck of the co-occurrence matrix method lies in its computation requirements. For example, in a 9×9 image window with a displacement $\mathbf{d} = (1, 0)$ there are 72 pixel pairs to be processed and thus $N = 72$ entries in the co-occurrence matrix. If the co-occurrence matrix is formed for adjacent window positions in an image (a sliding window) then processing can be speeded up by updating the co-occurrence matrix so that pixel pairs that are moved out of the window are subtracted from the matrix and pixel pairs that are moved into the window are added to the matrix. Thus if the window is 9×9, there are at most 18 pixel pairs (compared to 72) to be processed. From the three features above, energy needs the co-occurrence matrix computation while contrast and mean can be computed without actually forming the co-occurrence matrix.

2.2. Local Binary Patterns

A local binary pattern (LBP) method is a two-level version of the texture unit method.[23] It has been used successfully in metal surface inspection[13] and it performed well in a recent comparison in Ref. 9. The texture unit elements $TU_k(\mathbf{x}),\ k = 1, 2, ..., 8$, are simple functions of gray level values I around a given pixel position $\mathbf{x}$. The texture unit element $TU_k(\mathbf{x})$ is given as

$$TU_k(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad I(\mathbf{x}+\mathbf{d}_k) < I(\mathbf{x}), \\ 1 & \text{if} \quad I(\mathbf{x}+\mathbf{d}_k) \geq I(\mathbf{x}) \end{cases} \tag{2}$$

where $\mathbf{d}_k,\ k = 1, 2, ..., 8$, are displacement vectors. They define a 3×3 neighborhood (from left to right and from top to bottom) of the center pixel $\mathbf{x}$.

The texture unit number TN is a unique number that is assigned to each different binary pattern in a 3x3 window,

$$TN(\mathbf{x}) = \sum_{k=1}^{8} TU_k(\mathbf{x}) 2^{k-1}. \tag{3}$$

For each 3x3 image window, there are $2^8 = 256$ possible texture unit numbers.

In addition to the texture unit number TN, a simple contrast measure was also proposed in Ref. 9. It is computed after the texture unit element calculation and it is a difference of the mean gray level values of the pixels with $TU_k(\mathbf{x}) = 1$ and the pixels with $TU_k(\mathbf{x}) = 0$. Both features are computationally simple.

3. CASE STUDY: SURFACE DEFECT DETECTION

Defect detection of a textured surface is used as a case study. Defect detection is based on the following idea: *an unknown sample is classified as a defect if it differs enough from the estimated distribution of fault-free samples.* This means that only samples taken from a fault-free surface are needed in training the classifier. No defect samples are needed or even wanted in training although the system is robust enough to allow some defect samples in a training set. This kind of training has some important advantages: a tedious collection of different kind of defects is not needed anymore and appearance of new defects do not result in retraining of a classifier.

The statistical self-organizing map is used to estimate the distribution of fault-free samples.[15,20] It does not require the distribution to have any specific form (for example, the Gaussian form) but the distribution can be of any shape.

3.1. Segmentation Procedure

Segmentation is done as follows. An image is scanned by a gliding window. A feature vector is extracted from the gliding window for a center image pixel. Each image pixel is then classified as a defect or as a fault-free pixel. Three different steps are thus recognized (Figure 1):

1. Extract a feature vector $\mathbf{f}$ for an image pixel.
2. Compare $\mathbf{f}$ with the statistical SOM and find the best-matching unit c. Show the minimum error e_c as the output.
3. If $e_c > 0$, the pixel is classified as a defect. Otherwise the pixel is fault-free.
4. Repeat steps 1-3 for each image pixel.

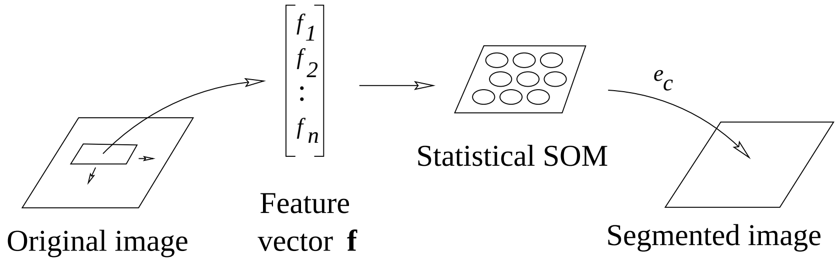


Figure 1. The segmentation procedure used in the experiments.

The error e_c determines on how many feature planes the feature vector $\mathbf{f}$ differs from the best-matching unit c. It takes integer values between zero and n (where n is the dimension of the feature vector $\mathbf{f}$). For more information on the used segmentation procedure, see Refs. 15,20.

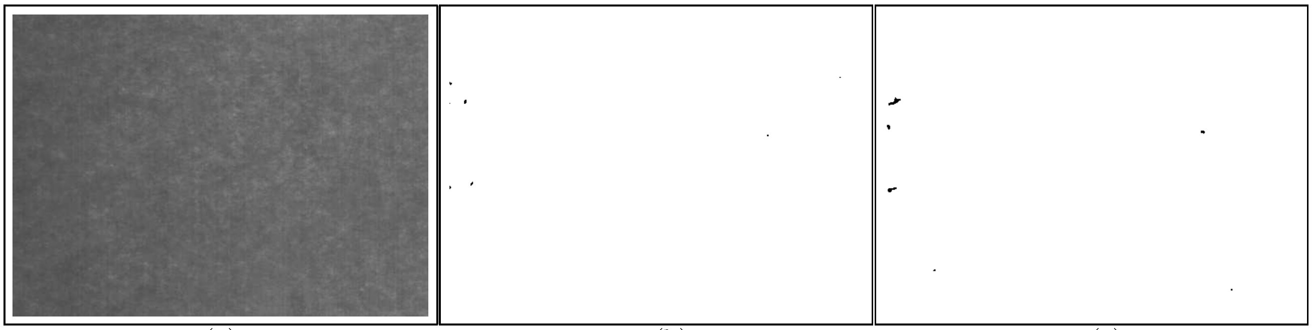
3.2. Feature Extraction

Two different feature sets are used in the experiments. The feature set 1 has the co-occurrence matrix features, energy, contrast, and mean. These are calculated in a 9x9 image window using the displacement $\mathbf{d} = (1, 0)$. The feature set 2 has the local binary pattern features, the texture unit number TN, the simple contrast measure, and the mean value. These are extracted from a small 3x3 image window. The surface images used in the experiments were quantized to 32 gray levels (the co-occurrence matrix features) or to 256 gray levels (the local binary pattern features). It is not efficient to use the whole 256 gray levels with the co-occurrence matrix features since the size of the co-occurrence matrix comes too large for any efficient computations. In the case of the local binary pattern features, the number of gray levels does not affect the computational properties and thus all available gray levels can be efficiently used.

The mean value brings the gray-level dependency into both feature sets. It is an essential feature here since many dark defects have their texture similar to a good fault-free surface; only the average gray-level value is different. Thus these defects cannot be found only with gray-level invariant texture measures. On the other hand, there are other defects that cannot be found with pure gray-level thresholding. So in our case there is a need for both the gray-level invariant measures and the gray-level dependent measures.

3.3. Experimental Setup and Results

The size of the training set was 100000 feature vectors extracted from fault-free samples taken randomly from the image in Figure 2(a). The size of the statistical SOM was 25 units.



(a) (b) (c)

Figure 2. (a) Surface image used for training and (b) its segmentation with the feature set 1 and (c) with the feature set 2. Defects are marked with black color.

In Figure 2 is the surface image that was used in training and its segmentation with both feature sets. As expected, only few pixels are classified as a defect. Three defect images (that were not used in training) with their segmentations are shown in Figure 3. The morphological closing operation has been applied to all segmentations to close small gaps between found defects. The first image has some dark spots, the second image has a large hole with some dark and light areas around it, and the third image has a few holes with some weak areas around them. All defects are found nicely with both feature sets. However, there are also small differences. The feature set 1 seems to produce more noise than the feature set 2. It also enlargens found defects which is due to the use of bigger image window. The dark spot in the uppermost image is found better with the feature set 2 whereas the weak areas in the undermost image are found better with the feature set 1. So, based on these images (and about hundred other images) we may conclude that there is no critical difference in performances of these feature sets. The differences are in computational requirements where the feature set 2 is better in several aspects: smaller image windows are used (using 3x3 windows with the feature set 1 results in a dramatic reduction in performance), less memory is required, and the features are faster to compute.

4. CONCLUSIONS

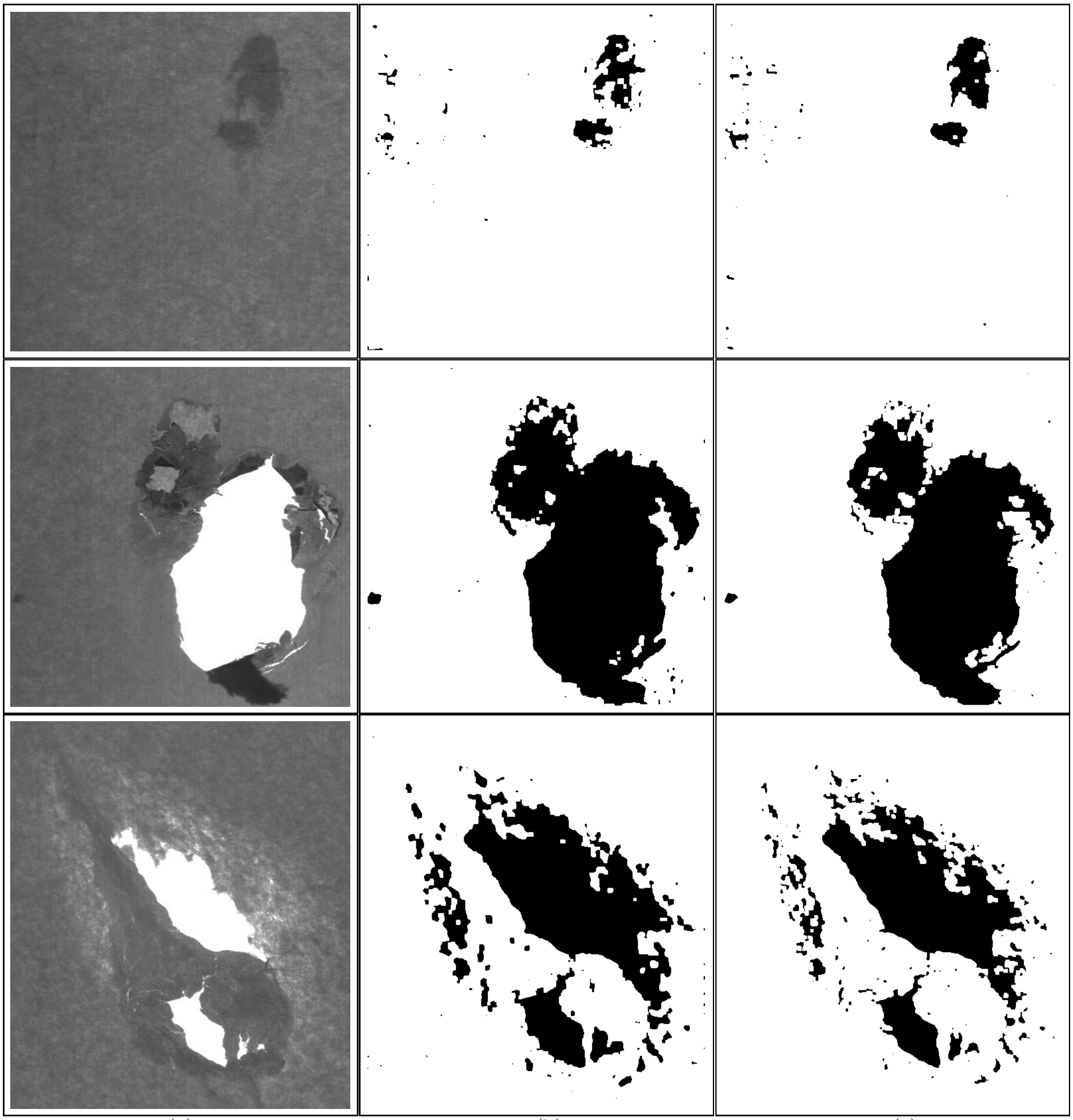
In this paper the evaluation of two histogram-based texture analysis techniques (the co-occurrence matrix method and the local binary pattern method) for surface defect detection was carried out. Both techniques gave similar results, but the difference between them in computational complexity is noticeable. The local binary pattern method is a simple and fast one, and it works well with small 3x3 image windows. These properties make it a promising approach when considering real-time implementations.

ACKNOWLEDGMENTS

The author wishes to thank the ABB Industry Oy and the Technology Development Centre of Finland for financial support (TEKES's grant 4172/95).

REFERENCES

1. R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE* **67**, pp. 786–804, May 1979.
2. H. Wechsler, "Texture analysis – a survey," *Signal Processing* **2**, pp. 271–282, July 1980.



(a) (b) (c)

Figure 3. (a) Surface images and (b) their segmentation with the feature set 1 and (c) with the feature set 2. Defects are marked with black color.

3. L. Van Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis anno 1983," *Computer Vision, Graphics and Image Processing* **29**, pp. 336–357, Mar. 1983.
4. M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, eds., pp. 235–276, World Scientific, Singapore, 1993.
5. J. S. Weszka, C. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-6**, pp. 269–285, Apr. 1976.

6. R. Conners and C. Harlow, "A theoretical comparison of texture algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**, pp. 204–222, May 1980.
7. J. M. H. du Buf, M. Kardan, and M. Spann, "Texture feature performance for image segmentation," *Pattern Recognition* **23**(3/4), pp. 291–309, 1990.
8. P. P. Ohanian and R. C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognition* **25**(8), pp. 819–833, 1991.
9. T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition* **29**, pp. 51–59, Jan. 1996.
10. M. Unser and F. Ade, "Feature extraction and decision procedure for automated inspection of textured materials," *Pattern Recognition Letters* **2**, pp. 185–191, Mar. 1984.
11. F. S. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, pp. 803–808, Aug. 1991.
12. C. Neubauer, "Segmentation of defects in textile fabric," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, vol. I, pp. 688–691, (The Hague, The Netherlands), Aug. 30–Sept. 3 1992.
13. M. Pietikäinen, T. Ojala, J. Nisula, and J. Heikkinen, "Experiments with two industrial problems using texture classification based on feature distributions," in *Intelligent Robots and Computer Vision XIII: 3D Vision, Product Inspection and Active Vision*, Proc. SPIE 2354, pp. 197–204, 1994.
14. D. Brzakovic and N. Vujovic, "Designing a defect classification system: A case study," *Pattern Recognition* **29**(8), pp. 1401–1419, 1996.
15. J. Iivarinen, J. Rauhamaa, and A. Visa, "Unsupervised segmentation of surface defects," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. IV, pp. 356–360, (Wien, Austria), Aug. 25–30 1996.
16. A. Meier and U. Priber, "On-line surface inspection with fuzzy-asic modules," in *5th European Congress on Intelligent Techniques and Soft Computing*, vol. 3, pp. 2474–2479, (Aachen, Germany), Sept. 8–11 1997.
17. J. C. Badger, "An automatic web inspection system with advanced defect detection and classification capabilities," in *Proceedings of 1992 Nonwovens Conference*, pp. 69–80, (Marco Island, Florida), May 10–14 1992.
18. J. E. Graf, S. T. Enright, and S. I. Shapiro, "Automated web inspection ensures highest quality nonwovens," *Tappi Journal* **78**(9), pp. 135–138, 1995.
19. L. Mäkelin, J. Moisio, and J. Järvinen, "From hole detection to image processing and information technology," in *Proceedings of 50th Appita Annual General Conference*, vol. 1, pp. 817–822, 1996.
20. J. Iivarinen and A. Visa, "Unsupervised image segmentation with the self-organizing map and statistical methods," in *Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, D. P. Casasent, ed., Proc. SPIE 3522, pp. 516–526, 1998.
21. R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**, pp. 610–621, Nov. 1973.
22. R. Conners and C. Harlow, "Toward a structural textural analyser based on statistical methods," *Computer Graphics and Image Processing* **12**, pp. 224–256, 1980.
23. D.-C. He and L. Wang, "Texture features based on texture spectrum," *Pattern Recognition* **24**(5), pp. 391–399, 1991.