International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:1, 2012

Variance Based Component Analysis for Texture Segmentation

Zeinab Ghasemi, S. Amirhassan Monadjemi, and Abbas Vafaei

Abstract—This paper presents a comparative analysis of a new unsupervised PCA-based technique for steel plates texture segmentation towards defect detection. The proposed scheme called Variance Based Component Analysis or VBCA employs PCA for feature extraction, applies a feature reduction algorithm based on variance of eigenpictures and classifies the pixels as defective and normal. While the classic PCA uses a clusterer like Kmeans for pixel clustering, VBCA employs thresholding and some post processing operations to label pixels as defective and normal. The experimental results show that proposed algorithm called VBCA is 12.46% more accurate and 78.85% faster than the classic PCA.

Keywords—Principal Component Analysis; Variance Based Component Analysis; Defect Detection; Texture Segmentation.

I. Introduction

IMAGE Segmentation is a difficult yet very important task in many image analysis or computer vision applications. Differences in mean grey level or in colour in small neighborhoods alone are not always sufficient for image segmentation. Rather, one has to rely on differences in the spatial arrangement of grey levels of neighboring pixels–that is, on texture differences. The problem of segmenting an image based on textural cues is referred as texture segmentation problem [3].

Texture segmentation is a very fundamental area of study in computer vision and image processing. It is a key problem in many applications such as object recognition, defect detection, quality inspection, remote sensing, and so on [4]. The particular approach of this paper is defect detection in steel surfaces or as it is usually called visual inspection. Non-destructive visual inspection for texture abnormalities has got applications on a variety of surfaces, e.g. wood, steel, ceramics, etc. It is highly demanded by industry in order to replace the subjective and repetitive process of manual inspection. Fig. 1 shows some defect samples in different types of material. Many techniques have been developed on texture analysis and segmentation towards visual inspection. These techniques are divided into four categories, statistical approaches, structural approaches, filter based approaches, and model based approaches [6].

The surface defects are loosely separated into two types. One is local textural irregularities which is the main concern for most visual surface inspection applications. The other is global deviation of colour and/or texture, where local pattern or texture does not exhibit abnormalities. We refer this type of defects as shade or tonality problem. In this paper we deal with the first type of defects in steel surfaces and focus on a particular approach in defect detection, which is referred to as Principal Component Analysis that is categorized under filter based approaches of texture analysis and segmentation.

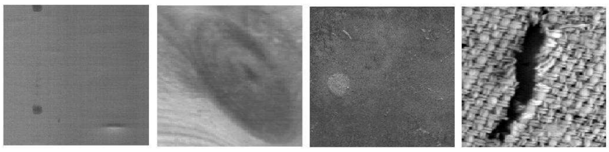


Fig. 1. Defects on different types of surfaces; from left: Steel, Wood, Ceramic Tile, and Textile.

Principal Component Analysis (PCA) which is also called eigenpictures decomposition [2] is a standard and popular approach used in pattern recognition and signal processing studies, since it is a simple and non-parametric method [5] for data extraction and reduction, and has recorded a great performance in fields such as face recognition and texture analysis [7]. As the pattern often contains redundant information, mapping it to a feature vector can decrease this redundancy and yet preserve most of the major information content of the pattern that are called principal components. So, PCA reduces problem dimensionality by seeking principal features called principal components, and eliminating redundant information [1]. It will be the basic algorithm behind our proposed method used for detecting defective and normal areas in steel surfaces. The rest of the paper is organized as follows. Section II describes Principal Component Analysis, its formulas and its particular usage in texture analysis of an image. Section III elaborates the proposed method, Variance Based Component Analysis (VBCA), and before concluding in Section V, Section IV performs the percentile and visual comparison of classic PCA and VBCA methods.

II. Background Information

A. Principal Component Analysis (PCA)

Principal Component Analysis [2] finds a rather small set of the most important characteristics of data (here: image) called "eigenpictures," which may be thought as the principal components of the original images. We aim to find the principal components of image, or eigenvectors of the covariance matrix of an image. These eigenvectors are ordered, based on different values of variance among the texture image–that is

Z. Ghasemi is M.Sc. Student of the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran, e-mail: (zeinab.ghasemi@eng.ui.ac.ir)
S. A. Monadjemi is Assis. Professor at the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran, e-mail: (monadjemi@eng.ui.ac.ir)
A. Vafaei is Assis. Professor at the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran, e-mail: (abbas_vafaei@eng.ui.ac.ir)

presented by eigenvalues. Therefore the eigenfaces are ordered based on the amounts of their corresponding eigenvalues. Each individual image can be represented exactly in terms of linear combination of its eigenpictures and can be approximated using only the "best" eigenpictures, i.e., those that have the largest eigenvalues.

1) Calculating EigenPictures: Let an image Im(x,y) be a two-dimensional $n \times n$ array and suppose that we have an image set with n different images $x_1, x_2, ..., x_n$. After altering every 2D image matrix to a vector of dimension $n \times n$ and placing each vector in a row of matrix, we will have a matrix of train images with n rows where each row represents one image. Covariance matrix C will be calculated by (1), and the eigenvectors and eigenvalues can be derived from matrix C using (2).

$$C = \frac{1}{n}\sum_{i=0}^{n}(x_i - \overline{x})(x_i - \overline{x})^T \tag{1}$$

$$\lambda_k = \frac{1}{n}\sum_{i=0}^{n} u_k^T (x_i - \overline{x})(x_i - \overline{x})^T \tag{2}$$

Where $\overline{x}$ denotes the average of all n samples, C is the covariance matrix and x_i is the i_{th} sample, and the vectors u_k and the scalars λ_k are the eigenvectors and eigenvalues, respectively. As mentioned before, the eigenvectors are ordered according to amount of eigenvalues. After choosing *m* ($m \leq n$) eigenvectors with largest eigenvalues as the principal components of the image set:

$$F = \mathit{Im} \times X \tag{3}$$

X is the eigenvector matrix with *m* rows of eigenvectors and *F* is the matrix of 'eigenpictures' of the image set.

B. Defect Detection in Texture

As mentioned before, when we work with textures, an area around a pixel is as important as the pixel itself. It means that we consider an area around a pixel as an observation of it. Consider texture image as a matrix of dimensions $n \times n$. The neighborhood size, *d*, can be variant as 3, 5, 7, etc. Every $d \times d$ matrix will be extracted as an observation of data and reshaped as a $d \times d$ vector, and it will be a row of neighborhood matrix, *Ngb*, of dimensions $k \times (d \times d)$. The Covariance matrix C is then computed and the eigenvectors and eigenvalues are obtained:

$$C(Ngb) = \frac{1}{n}\sum_{i=0}^{n}(Ngb - \overline{Ngb})(Ngb - \overline{Ngb})^T \tag{4}$$

$$(C(Ngb) - \lambda I) \times u = 0 \tag{5}$$

I is the unit matrix and λ and *u* are eigenvalues and eigenvectors of neighborhood matrix *Ngb* respectively. The matrix of eigenpictures B is obtained by 2D spatial domain convolution of image by the members of eigenfilter bank:

$$B = \mathit{Im} \otimes V_i \qquad i = 1, \ldots, n \tag{6}$$

Based on the value of *d*, it will be $d \times d$ eigenpictures. We can reduce these numbers of eigenpictures by choosing the most important eigenpictures and reconstruct image for determining defective areas in texture.

III. The Proposed Method: Variance Based Component Analysis (VBCA)

Our approach in texture segmentation involves the following steps:

1) Rearranging 2D texture image to achieve the neighborhood matrix.
2) Calculating Covariance C, of neighborhood matrix.
3) Obtaining Eigenvectors and EigenValues from C.
4) Applying 2D spatial convolution of eigenvectors to the image and retrieving eigenpictures.
5) Eigenpicture Selection.
6) Image Reconstruction.
7) Thresholding.
8) Post Processing.
9) Evaluation of Results.

The first four steps of the proposed method are done by statements and equations presented in previous Section (Section II-B). Calculating covariance matrix C by eq. 4, achieving eigenvector and eigenvalue matrixes by eq. 5, and 2D spatial convolution of eigenvectors to the image by eq. 6.

A. Eigenpicture Selection

All of eigenpictures are not required for segmentation process. As a matter of fact, to achieve a high level of accuracy, it is necessary to eliminate some of the eigenpictures. After obtaining eigenpictures, we use a technique to choose the most important ones. We calculate the variance of eigenpictures and choose the eigenfaces with variance values exceeding a predefined threshold, θ. After examining different values, 0.05 was chosen as the best value for parameter θ. For majority of images, the number of eigenpictures chosen by this threshold would be 2. It means that we will use two eigenpictures with variance values larger than the threshold for image reconstruction process.

B. Image Reconstruction and Thresholding

Simple addition is been used for combination of eigenpictures and obtaining reconstructed image. Then a threshold criteria is used to determine the boundaries of defective and normal areas. Different values of threshold criteria, μ, have been examined to determine the most suitable value for every single image ($0 \leq \mu \leq 0.2$).

C. Post Processing

We applied three different kinds of post processing operations to improve the result. Removing isolated pixels to clean normal areas, filling holes in final binary image to fill the defective areas and using the majority operation to smooth both defective and normal areas. The size of majority, α, is been examined for every single image and the best of it is been used for image ($1 \leq \alpha \leq 21$).

D. Evaluation of Results

Three criterions are used in evaluating the results, Sensitivity or True Positive Rate (Sns), which is the percentage of true classification of positive samples (defective pixels), and Specificity or True Negative Rate (Spc), which is the percentage of true classification of negative samples (normal pixels), and Classification Accuracy (CA) that is $(Sns + Spc)/2$. We also use visual evaluation that is showing determined defective and normal areas on image and compare the results with original image.

IV. Experimental Results

We implement the proposed method called VBCA on a dataset consisting of 15 different steel surface images with dimensions 512×512. These images are chosen to show different types of defect on steel surfaces. In Fig. 2 some prototype images from dataset are shown.

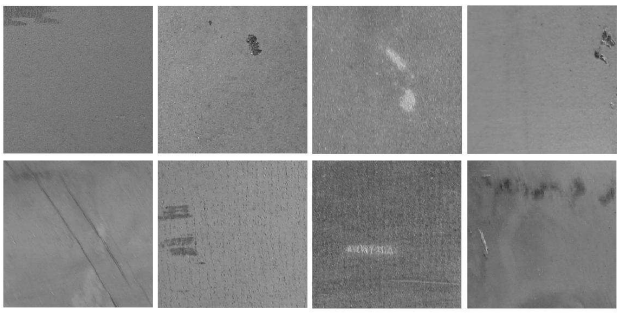


Fig. 2. Eight Different Samples of Steel DataSet

The achieved results will be compared to a basic algorithm called classic PCA. In classic PCA, pca technique will be used for feature extraction and a standard unsupervised clusterer, Kmeans, is adopted for clustering and segmentation of pixels. In Tables I and II, the results of the proposed method called VBCA are compared with classic PCA results regarding both accuracy and time complexity.

TABLE I
Accuracy of Methods

DataSet	**VBCA**			**Classic PCA**		
Image No.	**Sns**	**Spc**	**CA**	**Sns**	**Spc**	**CA**
Image 1	91.94	92.32	92.13	82.13	77.77	79.95
Image 2	86.30	96.28	91.29	63.60	94.70	79.15
Image 3	92.24	86.79	89.52	93.09	82.43	87.76
Image 4	100	99.73	99.86	79.43	65.41	72.42
Image 5	87.88	94.48	91.18	92.93	97.95	95.44
Image 6	98.10	92.32	95.21	76.01	69.84	72.92
Image 7	92.63	99.09	95.86	81.82	88.80	85.31
Image 8	93.71	97.77	95.74	73.91	86.45	80.18
Image 9	96.40	97.73	97.07	87.64	67.51	77.58
Image 10	99.53	99.44	99.49	86.79	69.31	78.05
Image 11	98.78	97.76	98.27	95.98	82.84	89.41
Image 12	99.58	99.71	99.65	86.21	78.87	82.54
Image 13	99.23	99.02	99.13	99.32	98.95	99.14
Image 14	99.52	98.97	99.24	85.64	95.87	90.76
Image 15	95.87	95.91	95.89	60.18	89.36	74.77
Mean	94.72	96.25	95.49	84.06	83.08	83.03

As it can be seen in Table I, the results show that VBCA is 12.46% more accurate than classic PCA. In most of images, Sns and Spc and therefore CA in VBCA are more than classic PCA, but in image 5, classic PCA has better results, and in image 13 the results are rather the same. We also see in Table II that as time consumption of VBCA is 6.9 seconds while it is 32.7 seconds for classic PCA, VBCA is 78.8% faster than classic PCA . Though it was expected due to less number of eigenpictures in VBCA. Visual test for three different images are also shown in Figure 3 where the advantages of VBCA are visible as well. In image 5 it can be seen that classic PCA has a slightly better performance, in image 7 and 11 VBCA has separated defective and normal areas precisely, while classic PCA has misclassified some part of normal areas as defective and in image 13 VBCA has a slightly better performance. So, as it is visible, VBCA outperforms classic PCA at most of the samples regarding visual tests.

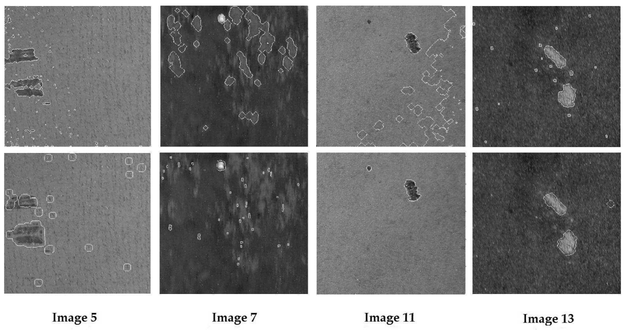


Fig. 3. Visual Comparison of classic PCA and VBCA results. First row is classic PCA results and second row is VBCA results.

TABLE II
Time Consumption of Methods

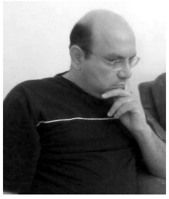
DataSet	**Time Consumed (s)**	
Image No.	**VBCA**	**Classic PCA**
Image 1	6.35	34.07
Image 2	7.88	34.73
Image 3	6.36	38.39
Image 4	6.35	50.11
Image 5	7.80	58.52
Image 6	6.97	34.41
Image 7	6.19	37.78
Image 8	6.94	35.66
Image 9	6.29	38.92
Image 10	6.63	39.39
Image 11	9.92	20.11
Image 12	5.90	13.21
Image 13	6.55	8.69
Image 14	6.46	24.55
Image 15	7.66	21.93
Mean	6.92	32.70

V. Conclusion

In this paper we presented a PCA-based algorithm called Variance Based Component Analysis or VBCA for segmentation of steel plate surfaces regarding defect detection. The results of the proposed algorithm has been compared with a basic algorithm called classic PCA regarding their accuracy and time consumption. According to the results, it seems that our proposed method, VBCA, with sensitivity of 94.72% and specificity of 96.25% has performed better than classic PCA (PCA+Kmeans) with sensitivity of 84.06% and specificity of 83.08% in detection of both defective and normal areas. The mean of time consumed by VBCA is 6.92 seconds, while mean time needed for classic PCA is 32.7 seconds. It should be mentioned that it is likely for our method to have less time complexity, becuase of its lower dimension. Therefore, based on rather high level of accuracy, it seems that VBCA is a powerful algorithm in the field of defect detection. With some modification, such as using different criteria for choosing the best eigenpictures, instead of overall variance, we may be able to achieve better results in this particular application.

REFERENCES

[1] A. Monadjemi. Towards efficient texture classification and abnormality detection. PhD Thesis, Department of Computer Science, University of Bristol, October 2004.
[2] M. Turk, A. Pentland. Eigenfaces for Recognition. Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.
[3] A. K. Jain, F. Farrokhnia. Unsupervised Texture Segmentation Using Gabor Filters. Journal of Pattern Recognition, Vol. 24, No. 12, pp. 1167-1186, 1991.
[4] Y. Chen, R. Wang. Texture Segmentation Using Independent Component Analysis of Gabor Features. The 18th International Conference on Pattern Recognition (ICPR'06), pp. 147-150, 2006.
[5] J. Shlens. A Tutorial on Principal Component Analysis. http://www.snl.salk.edu/∼ shlens/, 2009.
[6] X. Xie. A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques. Electronic Letters on Computer Vision and Image Analysis, Vol. 7, No. 2, pp. 1-22, 2008.
[7] A. Eleyan, H. Demirel. PCA and LDA based Neural Networks for Human Face Recognition. Face Recognition, Book edited by: Kresimir Delac and Mislav Grgic, June 2007.



S. Amirhassan Monadjemi is born in Isfahan, Isfahan, Iran, 1968. He got his B.Sc. degree at computer hardware engineering from Isfahan University of Tech., in 1991, his M.Sc. degree at computer engineering, machine intelligence and robotics from Shiraz University, Shiraz, in 1994 and his PhD in computer science, image processing and pattern recognition from University of Bristol, Bristol, UK. in 2004. He is now working as assistant professor at the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran. His research interests are image processing, computer vision and pattern recognition, computer aided learning and physical detection and elimination of viruses.

Abbas Vafaei is born in Isfahan, Isfahan, Iran, 1947. He received his B.Sc. degree in electrical engineering from Tehran Polytechnic, Tehran, Iran, in 1968, the M.Sc. degree from Toulouse Polytechnic, France in 1972, and the PhD degree in 1976 from Paul Sabatier University, Toulouse, France both in electronics. He also got a M.Sc. degree from South Bank University, London, in integrated circuit design in 1988. He jointed National Iranian Radio and Television in 1976. Since 1991 he is working as assistant professor at the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran and teaches courses in computer hardware and signal processing. His research interests contain industrial automation, display system, image processing and electromagnetion.

Zeinab Ghasemi is born in Ahwaz, Khuzestan, Iran, 1986. She is a M.Sc. Student of artificial intelligence at the Department of Computer Engineering from University of Isfahan, Isfahan, Iran. She got her B.Sc. degree at computer hardware engineering from University of Isfahan, Isfahan, Iran, in 2008. Her research interests contain image processing, pattern recognition, defect detection and data mining.