Automatic Defects Classification and Feature Extraction Optimization

Bernd Kuhlenkötter, Carsten Krewet, and Xiang Zhang

Summary. This paper introduces an automatic classification system that can identify defects on product surfaces in manufacturing, especially in processes like grinding and polishing. The identification process is based on grayscale images taken by a vision system. Some technologies that extract features from digital images are discussed. The support vector machine (SVM) is used in this paper as a multiclass classifier. It is shown that the overall classification rate can be close to the level that a skilled operator can obtain. The issues concerning the optimization of feature extraction are also covered in this paper.

Key words: Defect classification, Feature extraction, Support vector machine, Optimization.

1 Motivation

Nowadays the manufacturing process is tending to high automation level, as much as possible relieving workers from the laborious tasks and unpleasant working environment [1, 2]. Nevertheless, many inspection tasks are still done manually due to the difficulty of automatic execution. One example is that the flaw inspection and identification on the surface of fittings, e.g., water tap heads, have long been done by human operators in sanitary industries. It is very beneficial to automate this process. First of all, the efficiency of this process will be dramatically increased. Second, the job is monotonous and tedious, leading to less concentration of operators over the time, which causes classification errors. Third, operators have their own standards of inspecting and classifying the defects. It is possible that one defect, which is identified by one operator to class A, is classified by another operator into class B. It is also possible that one operator might make different judgments at different times.

The work in this paper is aimed, but not limited, to automatically classify defects on water tap heads after grinding and polishing processes. From manufacturing practice, possible defects are defined into 15 categories in advance. Figure 1 shows samples of seven kinds of defects. From practical experience, an

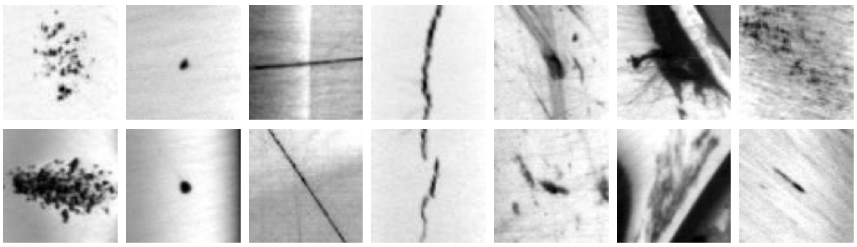

Fig. 1. Defect samples (from left to right: casting peel, pore, lined mark, crack, burned residues, grease residues, polishing shade)

operator reaches a classification rate in the range from 60% to 90% depending on their experience and on their concentration level. The wrong inspections come from the lack of concentration and subjective errors. In comparison, an automatic inspection and classification system can evaluate the defects using a constant criterion and overcome the varied standards among different operators.

2 Automatic Classification System

The vision system consists of a carrier, a camera system, a lighting system, other accessories and the software. The system hardware is responsible to provide a constant lighting environment and obtain the digital images of surfaces under this constant circumstance. The software provides the solution to examine the images from the camera system, locating and classifying the defects on workpiece surfaces.

Two steps are included in the software implementation, the feature extraction and the classifier design. The feature extraction is the most important part in the system. It defines the rules to describe and express the defects inside an image in a form that the classifier can understand and utilize to distinguish one class from others. Generally, feature extraction digitizes the defect images in a way that enlarges the distinctions among categories and discards the similarities at the same time. After that, the features are applied as the training data to the classifier. Support vector machine (SVM) [3] is an effective artificial method to solve both regression and classification problem, especially when the input dimension is very high. It has been successfully applied in many research and industrial classification tasks [4, 5]. Therefore it is also used as the classifier in this project described in this paper. The one-against-one scheme is used to combine a group of two-class classifier into a multiclass classifier. In most cases, the one-against-one scheme yields a better result than the one-against-all scheme [6].

3 Feature Extraction Technologies

The feature extraction is the most crucial step to the final accuracy of the classification. However, no single feature extraction method is consistently

superior to other methods [7] because the result of a method highly depends on the task to be solved. Therefore, several feature extraction technologies are implemented and tested, including shape features, statistical features, the local energy of some filtering channels and grayscale information.

3.1 Shape Features

Shape features indicate some values that represent the size related to the object contour. Some of the shape features are illustrated in Fig. 2, in which C is the contour of the detected defect, C_{con} is the convex hull of C and f_{max} is the maximal Feret diameter. The Feret diameter is defined as the projection length of the convex envelope of an object in a given direction. Besides f_{max}, seven shape features are used in this paper, area S, length l_a, breadth l_b, elongation e, compactness c, roughness r and area ratio s_r. The features are either shown in Fig. 2 or can be computed by following formulas

Length	$l_a + l_b = \frac{P}{2}$	Breadth	$l_a * l_b = S$
Elongation	$e = \frac{l_a}{l_b}$	Compactness	$c = \frac{P^2}{4\pi S}$
Roughness	$r = \frac{P}{P_{con}}$	Area ratio	$s_r = \frac{S}{S_{con}}$

where P is the perimeter of the contour C, P_{con} and S_{con} are the perimeter and area of its convex hull C_{con}, respectively.

The length l_a and breadth l_b are the logical length and breadth that can be calculated by the area and the perimeter. Elongation is the quotient of the length divided by the breadth, thus always greater than 1. Compactness is the square of the ratio of the perimeter of the original contour and the perimeter of a circle that has an equal area as the original contour. Ideally the compactness is 1 when the contour is a circle, otherwise it is greater than 1. Roughness and area ratio are two measures to indicate the convexity of the contour. A convex contour has the value 1 for the both measures. These eight shape features are not all independent.

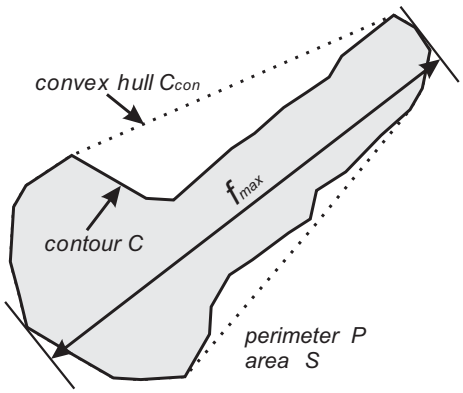

Fig. 2. Shape features

3.2 Filter Bank

Another technology to extract features from the texture image is the filter bank. The filter bank is also called multichannel spatial filtering method. The idea is to apply a sequence of filters on the image and take the local energy of the filtered images as features. The inspiration for this method comes from neurological studies. These research works suggest that the preprocessing stages in the human vision system involve a set of parallel and quasi-independent mechanisms or channels which resemble band-pass filters. Each filter in the filter bank contains intensity variations over a narrow range of frequency and orientation, specifying the regularity, coarseness and directionality of the original image [8]. One filtering transaction is computed by applying a convolution kernel to the original image and the local energy is calculated from the filtered image in a specified window. The general processing flow is shown in Fig. 3. The kernel or unit impulse response of the filter ℓ is given by a square matrix f_ℓ. The filtered image $y_\ell(i,j)$ is obtained by centrally convoluting the original image $x(i,j)$ with the filter f_ℓ, which can be written as

$$y_\ell(i,j) = x(i,j) * f_\ell(i,j) \tag{1}$$

Then the ℓ^{th} feature is specified by the local variance of the filtered image $y_\ell(i,j)$ in a $W \times W$ window and can be expressed as

$$FEA_\ell = \frac{1}{W^2} \sum_{m,n=0}^{W} \left\{ y_\ell(\frac{W}{2} - m, \frac{W}{2} - n) - u_\ell(i,j) \right\}^2 \tag{2}$$

where $u_\ell(i,j)$ is the mean value of the filtered image $y_\ell(i,j)$ in the $W \times W$ window and W is the window size which is specified by users. The different filter banks differ from each other mainly in the formulating of the filters f_ℓ.

Two kinds of filter banks are used: Laws filters [9, 10] and Gabor filters [11, 12]. Refer to our previous paper [13] for formulation and parameter configuration of these filters.

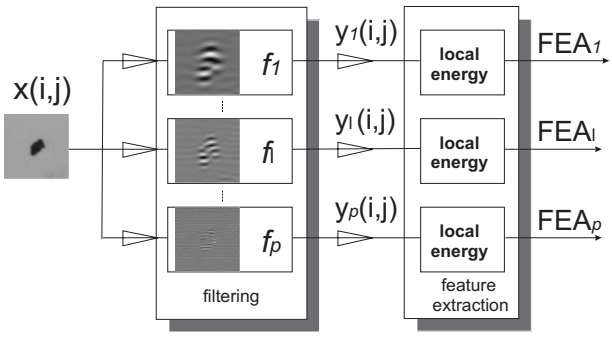

Fig. 3. Processing flow of filter bank

3.3 Statistical Features

The gray level co-occurrence matrix [14,15] is a well-known statistical tool for extracting second-order texture information from images. The co-occurrence matrix P_d is a $N_g \times N_g$ square matrix defined on a given displacement vector $\overrightarrow{d} = \{dx, dy\}$ where N_g is the grayscale level of the image. The entry (i, j) of the matrix P_d is the number of occurrences of the pair of gray level i and j which is a distance $\overrightarrow{d}$ apart. An example is given in Fig. 4 to demonstrate how to compute the co-occurrence matrix of a grayscale image. The left side of Fig. 4 shows an image of three grayscale levels, in which numbers denotes the pixel grayscales. The right side is the corresponding co-occurrence matrix P_d, which is a 3×3 square matrix, with respect to the displacement vector $\overrightarrow{d} = (1, 1)$. After that the co-occurrence matrix is calculated and a large range of features can be computed from this co-occurrence matrix. Five of them are used in this paper.

$$\begin{aligned}
\text{Energy } f_1 &\quad \textstyle\sum_i \sum_j P^2(i,j) \\
\text{Entropy } f_2 &\quad \textstyle\sum_i \sum_j P(i,j) \log_2[P(i,j)] \\
\text{Contrast } f_3 &\quad \textstyle\sum_i \sum_j (i-j)^2 P(i,j) \\
\text{Homogeneity } f_4 &\quad \textstyle\sum_i \sum_j P(i,j)/(1+|i-j|) \\
\text{Correlation } f_5 &\quad \textstyle\sum_i \sum_j (i-\mu_x)(j-\mu_y)P(i,j)/\sigma_x\sigma_y
\end{aligned}$$

where μ is the mean value of the co-occurrence matrix P, μ_x, μ_y, σ_x and σ_y are the means and the standard deviations corresponding to the vectors p_x, p_y that are expressed by

$$p_x = \sum_j P(i,j) \qquad \text{and} \qquad p_y = \sum_i P(i,j)$$

3.4 Grayscale Information

Besides the features introduced above, we use additionally the average and standard deviation of grayscale values of the defect image as grayscale features. The grayscale features should be localized considering different size of the various defects. The grayscale information are obtained in four areas in the defect image, respectively, see Fig. 5. In this case, the number of grayscale features is eight, two of each area.

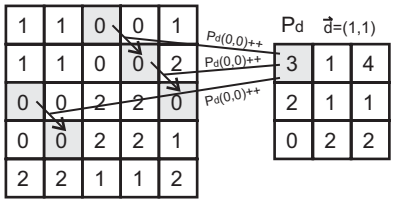

Fig. 4. Get the co-occurrence matrix of the grayscale image

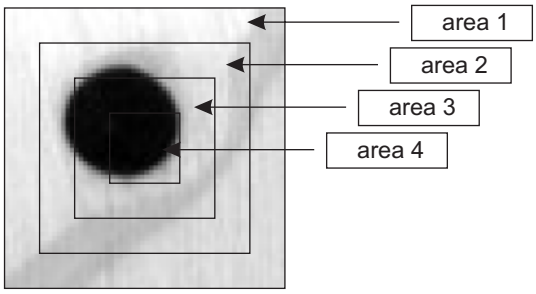

Fig. 5. Grayscale information

Table 1. Training and testing classification rate of varied features

Fea.	Fea. Num.	Tr. cl. rate(%)	Te. cl. rate(%)
Shape	8	87.5	59.5
Laws	25	99.5	69.4
Gabor	16	98.5	70.1
Statistical	15	98.0	75.2
Grayscale	8	98.5	72.3

4 Classification Results

Table 1 shows the classification results using only one kind of features. It can be concluded from the table that the shape feature is not suitable for this application. There are two reasons for that. First, there are no clear differences in the shape between some defects. The second reason is that the geometric information of some kinds of defects cannot be exactly defined. For example, it is not easy to describe the shape of a burned residues and a polishing shade. The pattern information is more effective than the simple geometric information in this sense.

The best result is obtained by using statistical features based on co-occurrence matrix, a 75.2% overall classification rate. The classification efficiency of grayscale information, Gabor features and Laws features are slightly lower than that of statistical features.

The performance of the classification system is improved when features from different technologies are combined. Table 2 shows the classification results of the combined features. The overall classification rate reaches a rate of 81.1% when the statistical features are combined with Gabor features and grayscale information.

5 Optimization of Feature Extraction

Many approaches are available to extract pattern features and many parameters can be adjusted in each approach. Thus, it is usually a troublesome task

Table 2. Training and testing classification rate of combined features

Gabor	Statistical	Grayscale	Num.	Tr. cl. rate(%)	Te. cl. rate(%)
X	X		21	100	77.2
X		X	24	100	78.4
	X	X	13	98.0	77.4
X	X	X	29	98.2	81.1

to select the most appropriate methods and parameters to obtain features that can best separate the samples. Sometimes it can be done by a lot of experiments and then by evaluation of the results of classification. However, there are often demands to have a standard for evaluating features, which does not depend on the classifier that is in use. In fact, features extraction and classification are two separate procedures though they are closely related to each other. Feature extraction is a way to represent the characteristics of a subject, while classification determines how to separate the samples based on the subject representation. The feature extraction should fulfill two principles. One is an indispensable condition of the classification task that a feature should exhibit enough differences among diverse categories to be classified. Otherwise, samples would be impossible to be separated effectively no matter which kind of classifier is applied. The other is a supplementary condition requiring that those features, which do not meet the first principle, are not used. The second principle is to optimize the input to the classifier and ensure the generalization of the model. However, the principles are quite descriptive. They make sense to select suitable features only if we can find an effective way to evaluate the quality and goodness of the features.

5.1 Bhattacharyya Distance

Suppose that we have two classes of samples that need to be separated. The feature values for m samples of the first class A are as follows

$$\underbrace{f_{A1}, f_{A2}, f_{A3}, ..., f_{Am}}_{m} \tag{3}$$

and the feature values for n samples of the class B are

$$\underbrace{f_{B1}, f_{B2}, f_{B3}, ..., f_{Bn}}_{n} \tag{4}$$

We also suppose that the feature values are normally distributed. Figure 6 illustrates two different situations of relative distributions of features f_{Ai} and features f_{Bi}. In this first case (left side), the class A can be easily separated from the class B because they are clearly different from each other that there is no overlap between features. In the second example, class A is theoretically hard to be discriminated from class B because the average value of features

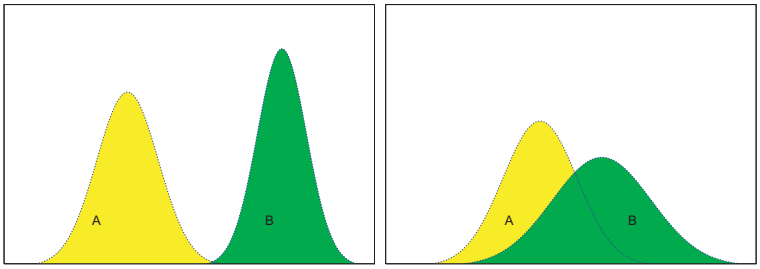

Fig. 6. Separability of two classes

are too close to each other. A good feature drags one class apart from another and the variance of this good feature should be small at the same time. An ideal situation is that the mean error $|\mu_a - \mu_b|$ is very large and two variances σ_a, σ_b are very small. Thus, the separability of a feature relates not only to the difference of the means but also to the deviation of features of each class.

The Bhattacharyya distance (BH distance) [16] is a method to statistically quantify the separability of two classes using a feature which can be written as

$$B_{dis}(A,B) = \frac{1}{4}\left\{\frac{(\mu_A - \mu_B)^2}{\sigma_A^2 + \sigma_B^2}\right\} + \frac{1}{2}\ln\left\{\frac{1}{2}\left(\frac{\sigma_B}{\sigma_A} + \frac{\sigma_A}{\sigma_B}\right)\right\} \tag{5}$$

where μ_A, μ_B, σ_A, σ_B are the features' means and standard deviations of the class A and the class B, which can be written as

$$\mu_A = \frac{1}{m}\sum_{k=1}^{m} f_{Ak} \tag{6}$$

$$\mu_B = \frac{1}{n}\sum_{k=1}^{n} f_{Bk} \tag{7}$$

$$\sigma_A = \sqrt{\frac{\sum_{k=1}^{m}(f_{Ak} - \mu_A)^2}{m}} \tag{8}$$

$$\sigma_B = \sqrt{\frac{\sum_{k=1}^{n}(f_{Bk} - \mu_B)^2}{n}} \tag{9}$$

For simplicity, the first part (Fisher ratio) in (5) can be used instead of BH distance as the measure of separability of two classes with respect to one feature. In the ideal situation, namely a large mean difference and small variances of each class, the BH distance and the Fisher ratio are both large scalars. The smaller the distance, the less separable are the two classes. Therefore, the BH distance or the Fisher ratio can be a criterion for evaluating the goodness of a feature.

5.2 Optimize Features Based on Co-Occurrence Matrix

The statistical features based on the co-occurrence matrix have given good results in the experiments above. In addition, they are flexible to be configured. The statistical features can be thought of as an weighted sum of the co-occurrence matrix elements. The features in (3) and (4) are calculated by

$$f_{A(B)k} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} W(i,j) \cdot P_{A(B)k}(i,j) = \mathbf{w}.p_{A(B)k} \tag{10}$$

where $W(i,j)$ is the weight matrix, P_{Ak} and P_{Bk} are the co-occurrence matrixes, $\mathbf{w}$, p_{Ak}, p_{Bk} are vectors that are formulated from the matrixes W, P_{Ak} and P_{Bk}. The P_{Ak} and P_{Bk} are known. Thus, once the weight matrix $W(i,j)$ is determined, the feature extraction process is sequentially determined. A weight matrix corresponds with a feature extraction strategy.

In the experiments above, we used only some standard features, e.g., energy, contrast, homogeneity that are general to all applications. The weight matrix of each standard feature is decided beforehand and does not depend on the problem that is being worked on. The idea of the feature extraction optimization is to find the best feature for a specific application, or at least one that is superior to the standard features. As mentioned above, the form of weight matrix defines the final feature. Therefore, obtaining the optimal feature for two classes A and B is equivalent to find a specific weight matrix $W(i,j)$ that can maximize the BH distance or the Fischer ratio between f_{Ak} and f_{Bk}.

This is a nonlinear optimization problem with N_g^2 unknowns. Most of the in-use iteration algorithms, like conjugate gradient method, need not only the function values but also function gradients for a fast convergence rate. The gradients of the objective function (5) with respect to $\mathbf{w}$ can be indirectly calculated by gradients of μ_A, μ_B, σ_A, σ_B with respect to the same $\mathbf{w}$, which can be written as

$$\nabla\mu_A = \frac{\partial\mu_A}{\partial\mathbf{w}} = \frac{1}{m}\sum_{k=1}^{m} p_{Ak} \tag{11}$$

$$\nabla\mu_B = \frac{\partial\mu_B}{\partial\mathbf{w}} = \frac{1}{n}\sum_{k=1}^{n} p_{Bk} \tag{12}$$

$$\nabla\sigma_A = \frac{\partial\sigma_A}{\partial\mathbf{w}} = \frac{\sum_{k=1}^{m}(\mathbf{w}\cdot p_{Ak} - \mu_A)(p_{Ak} - \nabla\mu_A)}{m\sigma_A} \tag{13}$$

$$\nabla\sigma_B = \frac{\partial\sigma_B}{\partial\mathbf{w}} = \frac{\sum_{k=1}^{n}(\mathbf{w}\cdot p_{Bk} - \mu_B)(p_{Bk} - \nabla\mu_B)}{n\sigma_B} \tag{14}$$

with (5–9), (11–14), the optimization problem can be solved.

However, the problem is not as simple as what has been introduced so far. Suppose 256 grayscale levels are used to generate the co-occurrence matrix, i.e., $N_g = 256$. In this case 65,536 unknowns exist in the optimization problem. Even taking the symmetry into consideration, there are still 32,896 unknowns, which means an extremely large optimization problem. The weight matrix W is too flexible to ensure the generalization of the final solution. Even though we obtain a weight matrix, with which the BH distance is a large number for training samples, it cannot be proven that this matrix will also bring a large distance for testing samples. Figure 7 shows a weight matrix that is calculated by maximizing BH distance between pores and polishing shades in the training set with a grayscale level of 32. The BH distance of training set is about 2,347 with this weight matrix, but only about 0.22 for the testing set. It goes back to the generalization problem in the learning theory. The solution of this kind of problems is normally to apply constraints on the over-flexible weight matrix, for example, requiring that the weight matrix surface is smooth and not so chaotic as that in Fig. 7.

Walker et al. [17] presented a strategy to construct a weight matrix. They started with a standard feature, e.g., energy or contrast, and considered every weighted elements in the co-occurrence matrix as a feature. Then the BH distances for each elements were calculated consequently. Therefore another matrix, which was called by them as a discrimination matrix, can be obtained. The discrimination matrix is also disturbed and fragmentary. After that, they used a second order polynomial surface to approximate the discrimination matrix. The polynomial surface was then used as the weight matrix finally. It was reported that the optimized features obtained in this way performs normally a bit better than the original standard features, but not always.

This method depends on standard features because the standard weight matrix is needed to calculate the discrimination matrix. In contrast, the constrained weight matrix strategy we introduced above is more general and more configurable. The problem now is what kinds of constraints should be imposed on the weight matrix in advance. We suggest two options. One is adopting polynomials as the form of the weight matrix. In this case, Walker's method can be considered a special implementation of the strategy we put

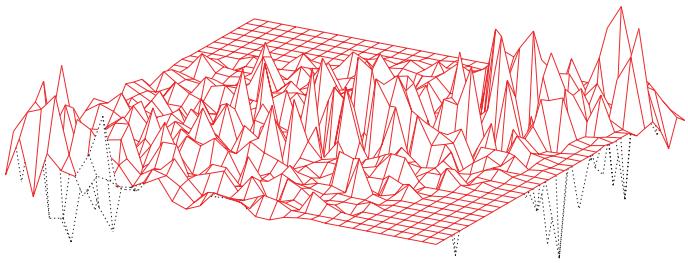

Fig. 7. Nonconstrained weight matrix

forward here. Another is using B-Spline surface representation. The optimization unknowns are the coefficients of the polynomials for the first case, while the coordinates of control points become the optimization objective when the B-Spline representation is adopted. Apparently, the B-Spline is a more adaptable representation because both the continuity of the surface and the number of control points are configurable. However, the optimization problem is much more complicated than polynomial representation because it is not easy to calculate the gradients of the control points coordinates with respect to the unknowns $\mathbf{w}$.

6 Summary

In this paper, an industrial vision system is introduced to identify and classify defects on free-form surfaces during grinding and polishing processes. The classification is based on grayscale images taken by a vision system. Some features, shape features, filter banks, statistical features and grayscale information are adopted for the classification task. SVM is served as a multiclass classifier, receiving the features as input and determine the category of the defect. In this application, the statistical features, grayscale features, and Gabor filter bank have shown better results than other kinds of features. The result is even better when these three kinds of features are combined together. With the combined features, an overall classification rate 81.1% can be reached, which is comparable to a trained operator. In addition, the optimization of the statistical features based on the co-occurrence matrix is also discussed in this paper. The statistical features based on the co-occurrence matrix can be considered as a weighted sum of the elements of the co-occurrence matrix. A general weight matrix can be adopted instead of the standard matrixes to construct a new feature. An optimized weight matrix should generate a feature, with respect to which the BH distance among defects is as large as possible. Constraints must be imposed on the weight matrix to guarantee the generalization of the weight matrix which is generated through the optimization process.

References

1. Bernd Kuhlenkötter and Thorsten Schüppstuhl. *VDI Berichte 1892, Mechatronik 2005, Innovativ Produktentwicklung*, chapter Vollautomatisierung durch innovative Robotersysteme. VDI, 2005
2. Bernd Kuhlenkötter and Xiang Zhang. *Cutting Edge Robotics*, chapter A Robot System for High Quality Belt Grinding and Polishing Processes, pp. 755–770, 2005
3. Vladimir Vapnik. *The nature of statistical learning theory.* Springer, New York, second edition, 2000

4. Bernhard Schölkopf, Chris Burges, and Vladimir Vapnik. Extracting support data for a given task. In *First International Conference on Knowledge Discovery and Data Mining*, pp. 252–257, 1995
5. Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML*, pp. 137–142. Springer, 1998
6. Chih-Wei Hsu and Chih-Jen Lin. A comparsion of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002
7. Trygve Randen and John Håkon Husøy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999
8. Isaac Ng, Tele Tan, and Josef Kittler. On local linear transformation and gabor filter representation of texture. In *International Conference of Pattern Recognition*, volume III, pp. 627–631, 1992
9. K. Laws. Rapid texture identification. In *Proceedings of SPIE: Image Processing for Missile Guidance*, vol. 238, pp. 367–380, 1980
10. K. Laws. *Textured Image Segmentation.* PhD thesis, University of Southern California, 1980
11. John G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847–856, 1980
12. John G. Daugman. Complete discrete 2d gabor transformation by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169–1179, 1988
13. Xiang Zhang, Carsten Krewet, and Bernd Kuhlenkötter. Automatic classification of defects on the product surface in grinding and polishing. *International Journal of Machine Tools and Manufacture*, 46(1):59–69, 2006
14. Robert M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on System, Man and Cybernetics*, 3(6):610–621, 1973
15. Robert M. Haralick. Statistical and structural approaches to texture. In *Proceeding of IEEE*, vol. 67, pp. 786–804, 1979
16. Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1972
17. Ross F. Walker, Paul Jackway, and I.D. Longstaff. Improving co-occurrence matrix feature discrimination. In *Proceedings DICTA-95 Digital Image Computing: Techniques and Applications*, pp. 643–648, 1995