MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Automatic Configuration of Systems for Texture Analysis

Christian Kueblbeck, Christoph Schittko, Thomas Wagner *
Fraunhofer Institute for Integrated Circuits Erlangen

Abstract

Automatic visual inspection plays a growing role in today's efforts to reduce the costs of industrial production. Three to ten months of man power have to be invested for the development of a typical industrial application. In this context, a reduction of time and thus of costs will make it easier to introduce new systems. We show a solution by using several algorithms for two kinds of selection routines, feature selection and training pattern selection. This method offers a convenient way to easily adapt a given system to a special texture analysis task.

1 Introduction and Background

Image processing plays a leading role in solving automatic quality control tasks for industrial production.

Image processing systems usually need to be customized to the given application, since the methods used for preprocessing or feature extraction tend to depend very sensitively on the particular problem. This causes considerable costs for engineering.

This paper wants to present an approach of reducing these costs by applying optimization methods to the setup and the configuration of an image processing system. Figure 1 one shows the application of automatic optimization at several points of the process. In detail we chose two starting points for the optimization:

- feature selection: A with respect to speed and classification success best feature subset is selected from a feature pool.
- training pattern selection: The partition of the given patterns in training and test set is being optimized for best classification results.

The results obtained with these methods are compared to those using given training and test sets of conventional textural feature extractors. An unbiased judgement of the selection hereby is only possible by using a separate verification sample set which is only used for the final performance calculation. This applies to all optimization methods that reuse classification results as input.

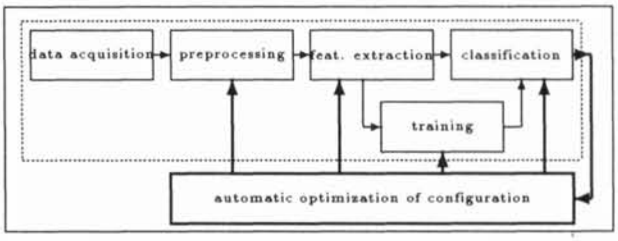

Figure 1: Automatic optimization of system configuration: based on the approach of Niemann (inside the dashed box), the setup can be submitted to a global optimization procedure in order to determine the optimal configuration.

2 The implemented Feature Extractors

A set of six methods of feature extraction has been implemented:

- statistical methods as the Haralick parameters [5] (based on the grey level difference matrix), the Unser parameters [10] (based on the grey level difference histogram), the Galloway parameters (based on the run length matrix), the Chen parameters [1] (statistical geometrical features) and local features
- spectral methods such as the Laine [7] parameters (calculated from the wavelet package decomposition)

Each method returns feature vectors of a length between 13 and 21. A total a pool of 100 features is available.

3 Feature Selection

For industrial applications it is important to reduce the classification time to a minimum. Therefore it is necessary to pick only a subset of the pool of features with a minimum in calculation time and classification errors. In order to find such a subset,

*Address: Am Weichselgarten 3, D-91058 Erlangen, Germany. Email: kue@iis.fhg.de

several methods for quality measurement have been proposed. One possibility is to select the features with a minimum of correlation [9], but this algorithm returns rather poor results. Another method consists in calculating the Bhatthacharrya measure for each feature [8]. Those procedures only focus on the features themselves and do not use the classification results. The following methods use these results as additional input:

- Global random search:

  The best but very time expensive method is to calculate the classification results for all possible combinations of features. A complete search costs too much time, because the number of necessary classifications grows exponentially ($\sum_{i=1}^{N} \binom{i}{n} = 2^N - 1$) with the number N of features.

Two methods that reduce the number of calculations from $2^N - 1$ to $\frac{N(N-1)}{2}$ are the "best with all" and the "sequential backward selection" [6].

- The "best with all" method

  This alghorithm selects the feature that optimally classifies the given sample set. Then successively one feature after another is added always using the one returning the best result.

- The "sequential backward" method

  That algorithm starts with all N features. The feature that is needed the least for an optimal classification result is dropped in the next step. This step is then repeates as often as needed.

The problem with these techniques is the fact that the global maximum (the feature set with best classification results) in general can not be obtained in this way. The reason is that the number of operations is predefined. So we propose an other method for searching through the possible combinations:

- Genetic Algorithms Genetic Algorithms (GAs) [3] are a technique that is known to solve given parameter optimization tasks well and therefore also may be used with the problem of feature selection. Here we use a binary coded genetic string of length N (number of features) for the feature selection: at position i a "0" means the ith feature has to be dropped while a "1" indicates selection. A so called fitness function is used to measure the performance of each individual. In this case, a fitness function can be implemented in several ways: just a function returning the individuals's classification result E or this result weighted by several factors like length of the selected feature set and / or classification time.

$$F_1 = E$$
$$F_2 = E \cdot \frac{1}{1 + p_1 \cdot R}$$
$$F_3 = E \cdot \frac{1}{1 + p_2 \cdot M}$$

  In this context R is the calculation time, M is the number of features, p_1 and p_2 are parameters that control the relative influence of R and M. From generation to generation genetic operators like "crossover" and "mutation" are used to create more and more improved combinations. ones. This method provides an algorithm to find a suitable feature combination much faster than the random method. But the search space is not as limited as it is using the "best with all" and the "sequential backward" algorithms. Thus other local maxima or even the global maximum can be found.

↓ Method	test set	ver. set
random	82%	81%
best with all	89%	86%
sequ. backwards	87%	84%
genetic algorithms	86%	84%

Table 1: Comparison of the different methods for feature selection

4 Training Pattern Selection

Obviously the choice of training patterns also has influence on the classification reults. Starting with a pool of patterns we are looking for those ones that are best suitable for training the classifier. Thus typical prototypes for each class can be found while untypical or erratic samples are excluded. Again, the number of necessary classifications increases exponentially with the number of patterns available, so a complete search in general takes too much time.

As a search strategy we propose a modified "best with all" algorithm that works as follows: Given is a pool of feature patterns from K classes. Select randomly one pattern from each of the first $K-1$ classes. Then check out all patterns of class K and select the one returning the best result. Then repeat this operation for all classes.

The "sequential backwards" method as well as genetic selection may also be used for training pattern selection.

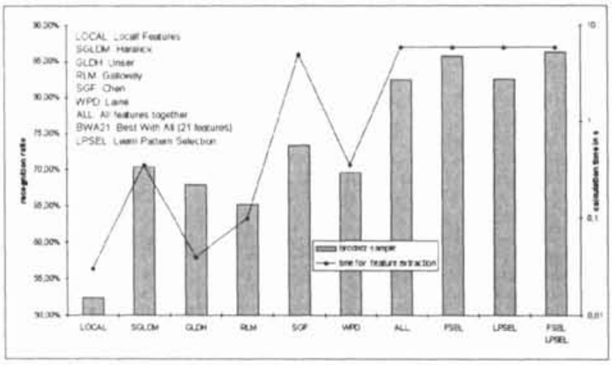

Figure 2: conventional feature extractors compared to all features together and to the results obtained using the selection strategies (FSEL means feature selection while TPSEL means training pattern selection and FSEL LPSEL means both methods applied successively). The calculations were done on a set of 13 Brodatz textures [13], each with 256 training, 256 test and 512 verification patterns.

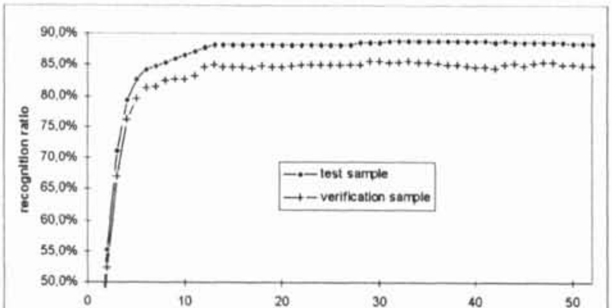

Figure 3: Feature selection: comparison of results on the test and the verification set using the "best with all" method on the test set and applying the selected features on the verification set.

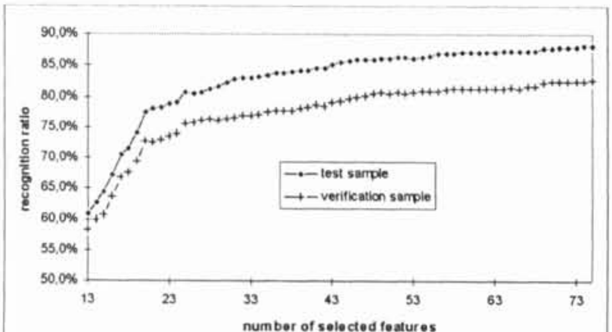

Figure 4: Training pattern selection: comparison of results on the test and the verification set using the "best with all" method on a pool of samples and applying the selected features on the verification sample.

5 Experiments and Results

In order to measure the performance of the proposed selection algorithms we used a set of 13 images of size 1024x1024 each showing one class of the well known Brodatz textures [13]. Every image was cut (non overlaping) into pieces of size 32x32. Finally three sample sets were built: a training and a test set each consisting of 13x256 images and a verification set consisting of 13x512 images.

The first two sets were used during the optimization process and the third one was only used for the final performance calculation. For the experiments with the genetic algorithms were computed on a cluster of workstations [12] using the two programs Gallops [4] and PVM [2] Firstly let's have a look at an overview of the results. Classification rate and time for different feature sets are shown in fig. 2: The leftmost six bars show the rates obtained by the different feature extractors (explained in section 2) applied to the training and verification set. The next bar shows the rate that was obtained using all 100 features together. The three rightmost bars show the increase of the classification results using the feature selection, the training pattern selection and both procedures together (from left to right).

Now let's have a more detailed look at the results of the feature selection (tab. 1). Of course, the random search method ranks last amoung the different strategies while using the "best with all" method on the test set and using the selected features to classify the verification set yields the best results.

In fig. 3 it can be seen that the plotted recognition ratio reaches its saturation point after selecting about 15 out of 100 features. The results on the verification set are not as good as the ones on the test set but the results are better than the outcomes using all features.

When looking at the training pattern selection (fig. 4) a similar effect of increasing classification ratios can be recognized.

6 Discussion and Outlook

We have proposed several approaches to automate system configuration in the fields of feature selection and training pattern selection. Both methodss provide significant improvements in classification success. Using a seperate verification sample set is essential to get reliable information about the system's capability to generalize.

With respect to the question of finding the most adequate feature subset, genetic selection strategies are outmatched by the intelligent strategy "best with all". Nevertheless, when going to examine genetic programming for the construction of new genetic feature sets, there is no conventional pendant which could be used instead of the computationally challenging genetic strategy. We hope that the experiences from genetic selection will be of some help for this. In addition other selection strategies and other approaches to automated configuration have to be examined.

References

[1] Chen, Y. , Nixon, M. and Thomas, D. (1993), *Statistical Geometrical Features for Texture Classification*, Pattern Recognition, Vol. 28, No.4, pp. 537-552.

[2] Geist, A. (1994), *PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing.* The MIT Press, Cambridge, Massachusetts, 1994.

[3] Goldberg, D. (1989), *Genetic algorithms in search, optimization and machine learning.* Reading, MA. Addison-Wesley, 1989

[4] Goodman, Erik D. (1995), *An Introduction to Galopps.* Michigan State University, 1995 (95-06-01), Tech. Report

[5] Haralick, M., Shanmugam, K. and Dinstein, I. (1973), *Textual Features for Image Classification*, IEEE Trans. Systems Man Cyber., Vol. SMC-3, No.6, pp. 610-621.

[6] Kittler, J. (1986), *Feature Selection and Extraction*, Academic Press, Inc., Pub. 86.

[7] Laine, A. and Fun, J. (1993), *Texture Classification by Wavelet Packet Signatures*, IEEE Trans. Pattern Anal. Machine Intell., Vol. 15, No. 11, pp. 1186-1191.

[8] Niemann, H. (1974), *Methoden der Mustererkennung*, Akademische Verlagsgesellschaft.

[9] Press, W. (1992), *Numerical Recipes in C*, Cambridge University Press, Cambridge, pp. 636-644.

[10] Unser, M. (1986), *Sum and Difference Histograms for Texture Classification*, IEEE Trans. Pattern Anal. Machine Intell., Vol. 8, No. 1, pp.

[11] Wagner, T. (1996), *Texturanalyse für industrielle Oberflächenprüfaufgaben*, 3. Heidelberger Bildverarbeitungsforum, 1996

[12] Wagner, T., Kueblbeck C., Schittko C. (1996), *Genetic Selection and Generation of Textural Features with PVM*, Third EuroPVM User's Group Meeting, TU München, 1996

[13] Weber, A. (1992), *The USC-SIPI Image Database Vers. 3*, University of Southern California.