Automated window size determination for texture defect detection

Jörg Amelung+ and Kai Vogel*

+ Technical University of Darmstadt
Control System Theory and
Robotics Department
Landgraf-Georg-Str.4
D-64283 Darmstadt

* ISRA Systemtechnik GmbH
Industriestr.14
D-64297 Darmstadt

Abstract

Texture defect detection methods are used for quality control purposes. Inspection systems fullfilling this task should be able to determine the necessary parameter sets automatically in case of a task change. One of these parameters is the size of the image window from which texture features are calculated. We present an approach for the calculation of an appropriate window size for statistical texture analysis methods. Our approach is based on the goal to choose the window size as small as possible to improve defect discrimination but to choose it also as big as necessary to achieve a sufficient texture representation within the window. The window size is calculated using a degree of feature deviation being defined. Experimental results are presented to show the relevance of the calculated window size.

Keywords : texture analysis, texture defect detection, automated window size determination, surface inspection

1 Introduction

The use of image processing systems for texture or surface inspection purposes can be considered as one of the main industrial vision tasks beside object recognition, identification and geometric measurement.

For inspection of carpets or fabric (fig.1) texture analysis methods have to be used to calculate appropriate surface features. But whenever a surface inspection system will be designed which is based on texture feature extraction two questions have to be considered :

1. Which texture analysis method is appropriate for describing the actual texture and is the most efficient one concerning its computational effort ?
2. Is it possible to determine well suited parameters for the chosen method and how is it possible to calculate them automatically in case of a task change ?

Concerning the first question for example the periodical fabric in fig.1(a) can be analysed with low computation effort by using eigenfilters [DEWA88]. On the other hand statistical texture analysis methods (e.g. co-occurrence matrix [HARA73]) yield respectable results analysing the carpet in fig.1(b).

In the following we will focus on question two. In our defect detection system we developed the ability to determine the parameters of a variety of statistical texture analysis methods which are in detail :

BMVC 1994 doi:10.5244/C.8.10

(a) (b)

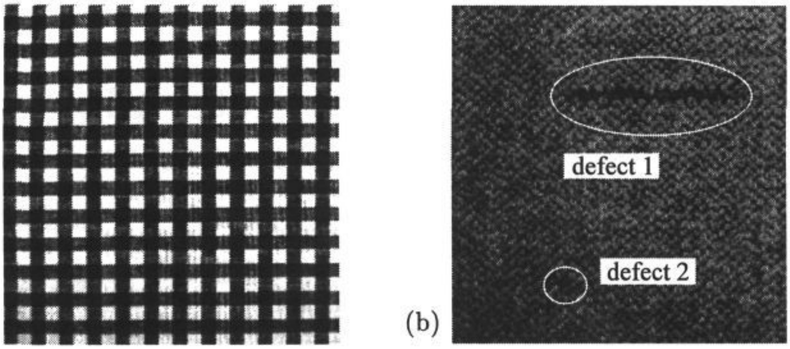


Figure 1: Texture examples. **(a)** periodical fabric, **(b)** carpet with marked defects.

analysis of grey level histogram (HIST), analysis of gradient histogram (GRAD), spatial gray tone dependence matrix (SGTDM) [HARA73], grey level run length matrix (GLRLM) [GALL75], neighborhood grey level dependence matrix (NGLDM) [SUN82]), grey level difference dependence matrix (GLDDM) [DAPE86], neighborhood grey tone difference matrix (NGTDM) [AMAD89].

Using these methods the following parameters have to be set:

1. the grey level, run lengths and grey level differences quantisation levels necessary for an efficient matrix calculation [AMEL92]
2. the size of the texture region (window) from which texture features are calculated
3. an appropriate feature selection for classification [AMEL93]

In this paper we will discuss how the window size can be determined correctly to achieve good classification results.

In section 2 we describe our approach for window size determination. In section 3 we present achieved results applying the window size calculation to the texture examples shown in fig.1. Section 4 shows how the window size calculation is integrated in our texture defect detection system. Achieved classification results will be presented.

2 Calculation of the window size

2.1 Window based processing for texture defect detection

For texture defect detection the digital image of a texture is divided into rectangular sub-images (windows) and for each window co-occurrence features are calculated. The feature vector is classified and the result is either a defect-free window or a defect inside the window.

A feature vector $\underline{f}_d$ which is calculated from a window containing a texture defect has to be disriminated from the defect-free feature vector population with center $\underline{f}_c$ and feature variances σ_i^2 by vector distance

$$d_d = ||\underline{f}_d - \underline{f}_c|| \tag{1}$$

using the feature covariance dependent from classification rule.

Evidently feature vectors calculated from small defects (fig.2(a)) will result in small differences d_{d_s} whereas big defects (fig.2(b)) will result in bigger differences $d_{d_b} > d_{d_s}$ if significant features are utilized. Additionally d_d is dependent from window size because small defects will result in bigger differences if we use a small window size (fig. 2(c)).

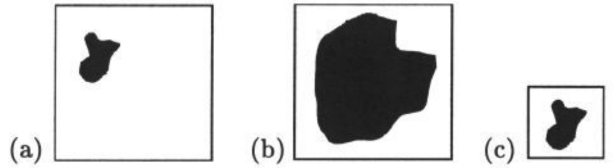


Figure 2: The small defect in **(a)** will effect only a small feature deviation in contrast to the big defect in **(b)**. By use of a small window size in **(c)** the small defect will effect big feature deviations as well.

Due to this the window size depends on defect size as well as on the image resolution and the actual texture itself. If the window size is chosen very small concerning the actual texture (i.e. the actual texture is insufficient represented within the window) the feature values will differ strongly for each window (i.e. σ_i^2 will increase).

Concluding these aspects we define the goal of determining the window size as follows: The window size should be chosen as small as possible to improve defect discriminination but it should be chosen as big as necessary to represent the actual texture.

2.2 Feature deviation function

Since the window size has to be chosen in a way that the actual texture is represented in the window we define a feature deviation function (FDF) $fd_i(w)$ as degree for texture representation using feature m_i with window size w :

$$fd_i(w) = \frac{(\bar{m}_i(w) - min\{m_i(w)\})^2 + (\bar{m}_i(w) - max\{m_i(w)\})^2}{\bar{m}_i^2(w)} \tag{2}$$

The feature deviation is based on the mean value $\bar{m}_i$ of feature m_i calculated from the feature values of a given set of windows with window size w and the minimum and maximum values of feature i, $min\{m_i\}$ and $max\{m_i\}$, within the specified window set.

The feature deviation function resembles feature variance $\sigma_i^2(w)$. We decided to use the FDF instead of a variance based measure to take into account the real size of the feature value interval but a variance based measure yields similar results.

Fig.3 shows the FDF of two features calculated from the carped texture (fig.1(b)) for several window sizes (96×96 pixels to 16×16 pixels in decreasing order, 96 e.g. is the abbreviation for 96×96 pixels.).

For window sizes less than size 48×48 feature deviation increases rapidly. For big window sizes the actual texture is well represented in the window and for sizes beyond the critical size 48×48 we yield an insufficient texture representation.

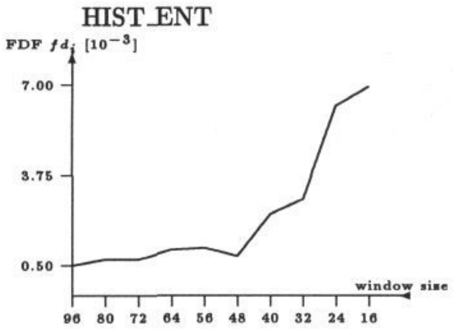
In section 3 we will present in detail deviation measures of the textures shown in fig.1.

2.3 Threshold selection from the feature deviation function

The remaining task in window size determination is the selection of an appropriate threshold to recognize the rapid FDF increase at the critical window size w_T.

Therefore the first derivative of the FDF is calculated and related to the minimum deviation value $min\{fd_i(w)\}$

$$fd'_i(w) = \frac{fd_i(w) - fd_i(w-1)}{min\{fd_i(w)\}} \tag{3}$$



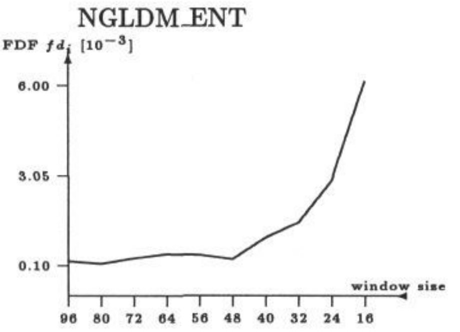


Figure 3: The feature deviation functions of features HIST_ENT (histogram entropy) and NGLDM_ENT (NGLDM entropy) calculated from the texture carpet (fig.1(b)) are shown. The plots show the deviation fd_i over decreasing window sizes – 96 is the abbreviation for 96×96 pixels. For both features a deviation increase can be observed using window sizes beyond the critical window size $w_T = 48$.

In view of a robust recognition the feature deviation function is processed with a low-pass filter before fd'_i is calculated. In our experimental setup best results were achieved by use of a threshold $|fd'_i(w_T)| \geq 0.3$. With this threshold used for each calculated deviation function $fd_i(w)$ the critical window size w_{Ti} has to be determined. The overall window size $\bar{w}_T$ is defined by the mean value :

$$\bar{w}_T = \frac{1}{N} \sum_{i=1}^{N} w_{Ti} \tag{4}$$

2.4 Practical aspects

Our system is capable to calculate up to 78 texture features corresponding to the methods presented above. To save computation time correlated features are removed by calculating the feature covariance.

Furthermore only a given number of different window sizes is investigated which offers the possibility to influence the accuracy of the best window size.

3 Results of window size calculation

First we will show results of the window size calculation for the fabric and the carpet in fig.1.

Actually, an automatic window size calculation is necessary only for the carpet (fig.1(b)). For the fabric (fig.1(a)) an optimal window size can be determined immediately: This is the fabric periodic length. However, we include results of respective calculations, since one can see by this, how the feature deviation function works. Namely, since the fabric has a period of 32 pixels, low deviation will be found with the window sizes 96×96 (3 times the period), 64×64 (2 times the period) and 32×32 (1 time the period), 32×32 being the smallest window representing the whole texture information. Window sizes 80×80, 72×72, 56×56, 48×48 and 40×40 give higher deviation values, because each window test set contains incomplete periods in addition to the complete periods, while window sizes 24×24 and 16×16 lead to lower deviation values due to the fact, that they do not contain any full period. Fig.4 shows the respective feature deviation functions fd_i for six selected features.

For the carpet in fig.1(b) we have no natural period. Therefore the feature deviation function gives higher values only, when the texture description starts to become insufficient. Fig.5 shows the feature deviation functions fd_i for six selected texture features.

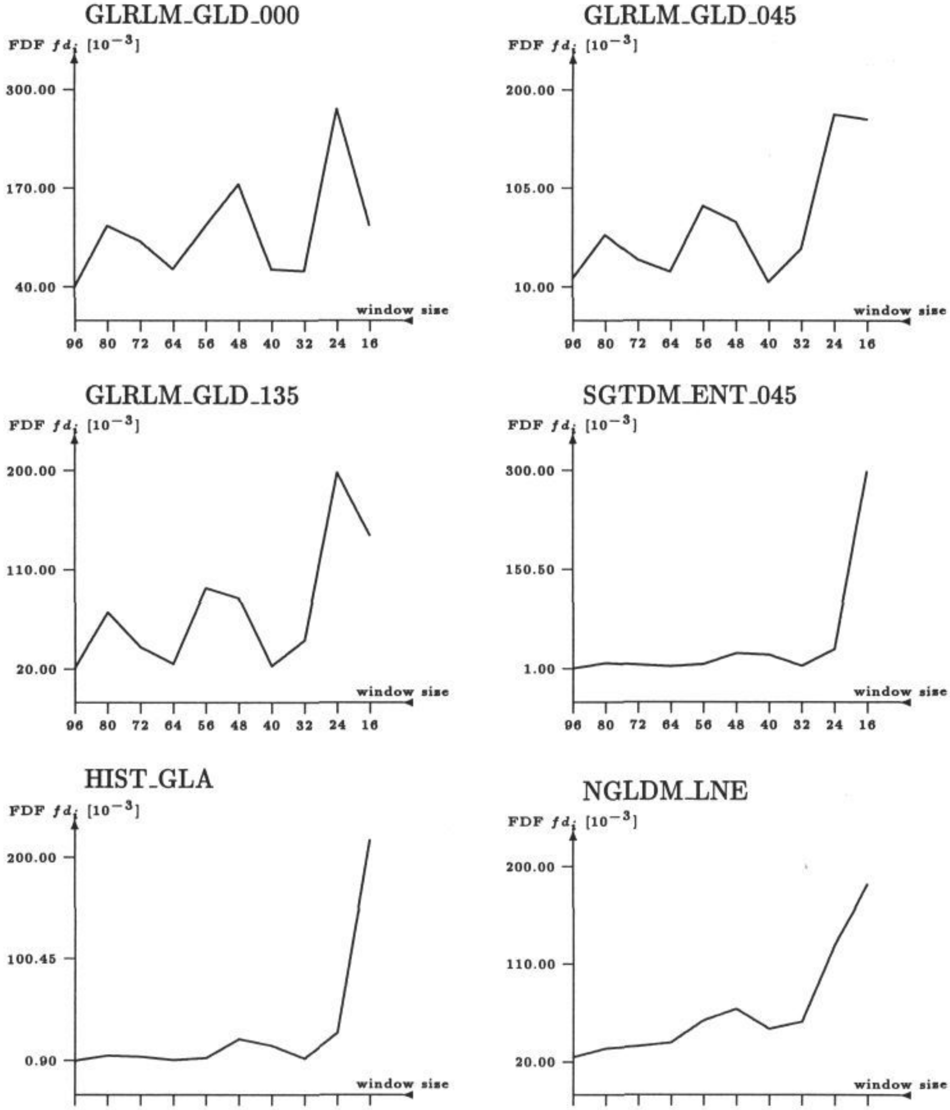


Figure 4: The feature deviation functions of 6 features calculated from the fabric texture in fig.1(a). The plots show the deviation fd_i over decreasing window sizes. Especially the GLRLM features depend on the period length of the fabric texture and for all features a deviation increase using window sizes beyond one period length (size H) can be observed. The features in detail are : GLRLM_GLD_000, GLRLM_GLD_045, GLRLM_GLD_135: GLRLM grey level distribution calculated from directions 0°, 45° and 135°. SGTDM_ENT_045 : SGTDM entropy calculated from direction 45°. HIST_GLA : histogram grey level average. NGLDM_LNE : NGLDM large numbers emphasis.

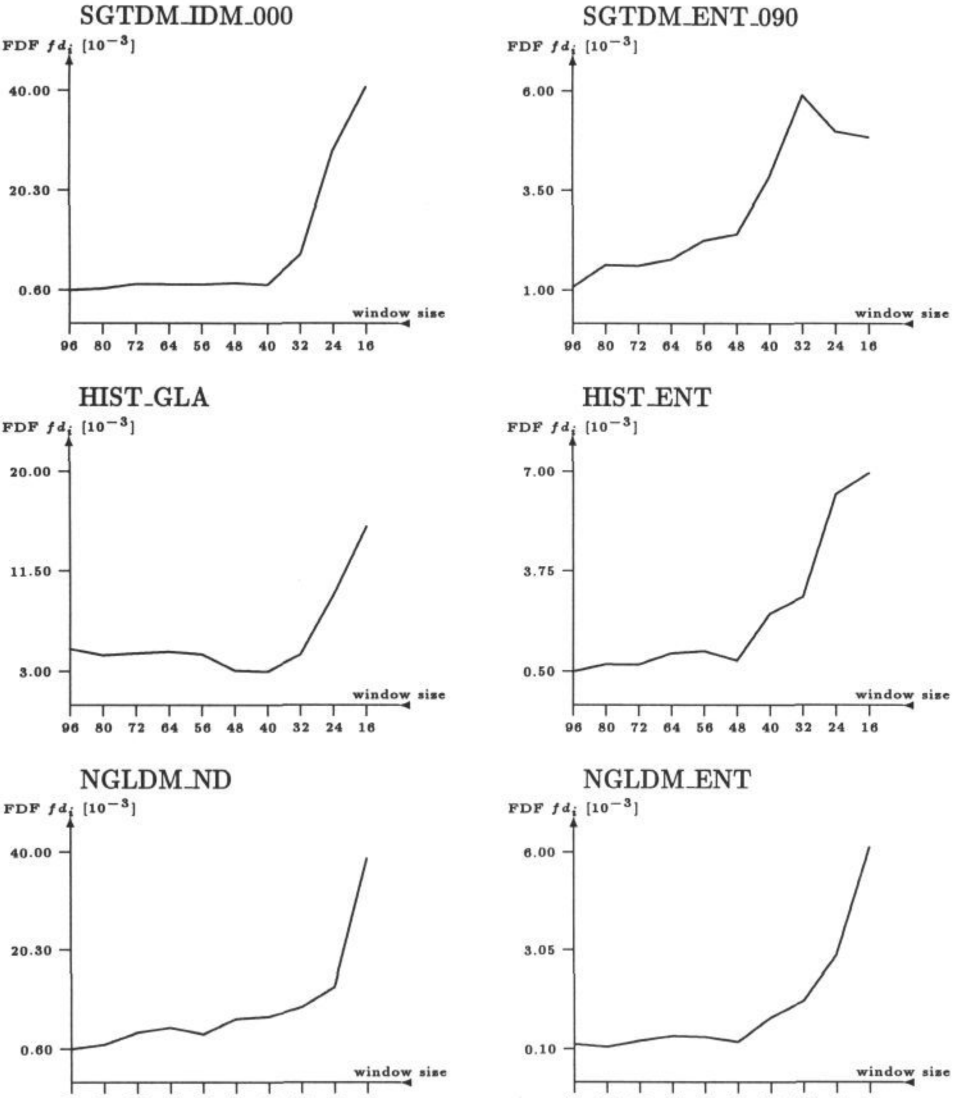
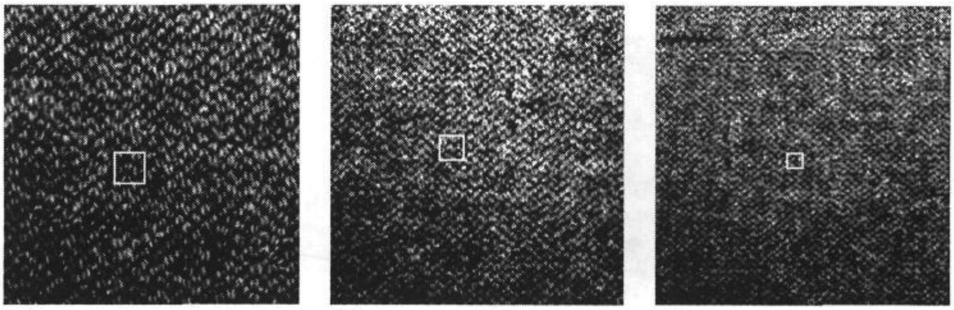


Figure 5: The feature deviation functions of 6 features calculated from the carpet texture in fig.1(b). The plots show the deviation fd_i over decreasing window sizes. For all features a deviation increase using window sizes beyond the critical window size w_T can be observed. The features in detail are : SGTDM_IDM_000 : SGTDM inverse difference moment calculated from direction 0°. SGTDM_ENT_090 : SGTDM entropy calculated from direction 90°. HIST_GLA, HIST_ENT : histogram grey level average and entropy. NGLDM_ND, NGLDM_LNE : NGLDM number nonuniformity and large numbers emphasis.

Naturally the question arises, whether the determined optimal window size is depending on the used resolution. Therefore the investigations were performed with three different resolutions. All cases gave the same relative window size — see fig.6 —. Due to this the window is bigger for the high resolution image and smaller for the low resolution image — in accordance with our expectations.

The approach for automated window size calculation was applied to several textures. In the following we want to present condensed results for three different textures (fig. 7) For each texture the investigated window sizes (upper left frames in texture images, see section 2.4) and the resultant window size (lower right frame in texture images) are demonstrated. Furtheron for each texture the summed feature deviation function $\sum_{i=1}^{N} fd_i$ is given to show the FDF increase for window sizes beyond the critical window size w_{Ti}. The summed FDF only serves as indicator for the applicability of the approach, the optimal window size itself is calculated from the FDF's of the single features.



(a) (b) (c)

Figure 6: The results for window size calculation for the carpet using three different resolutions. The windows highlighted in **(a)**, **(b)** and **(c)** correspond to the calculated window sizes. In accordance with our expectations the window size decreases if the image resolution decreases.

4 Window size determination for defect detection systems

As mentioned above an automated parameter determination for industrial texture inspection systems is desired. The presented window size determination has been integrated in our experimental defect detection system as well as approaches developed for range determination [AMEL92] and feature selection [AMEL93]. The resulting system architecture is shown in fig.8.

The advantage of an automated parameter determination is not only flexibility in case of a task change but delivers us an optimal parameter set with respectable classification results. The carpet texture was processed using three window sizes : (a) a large window size, (b) the calculated window size and (c) a smaller window size.

The result for the large window size (fig.9(b)) shows that defect 2 (compare fig.9(a)) does not effect the statistical features and is not recognized. For the optimal window size (fig.9(c)) defect 1 and defect 2 are recognized correctly. The smaller window (fig.9(d)) results in classification errors in defect-free regions since the defect-free texture is insufficiently represented in the window.

The results have been achieved using a maximum likelihood classifier (MLH) and the following features (in descending order of significance):

(a) NGLDM_angular second moment, NGLDM_entropy, GLDDM_grey level distribution, HIST_angular second moment

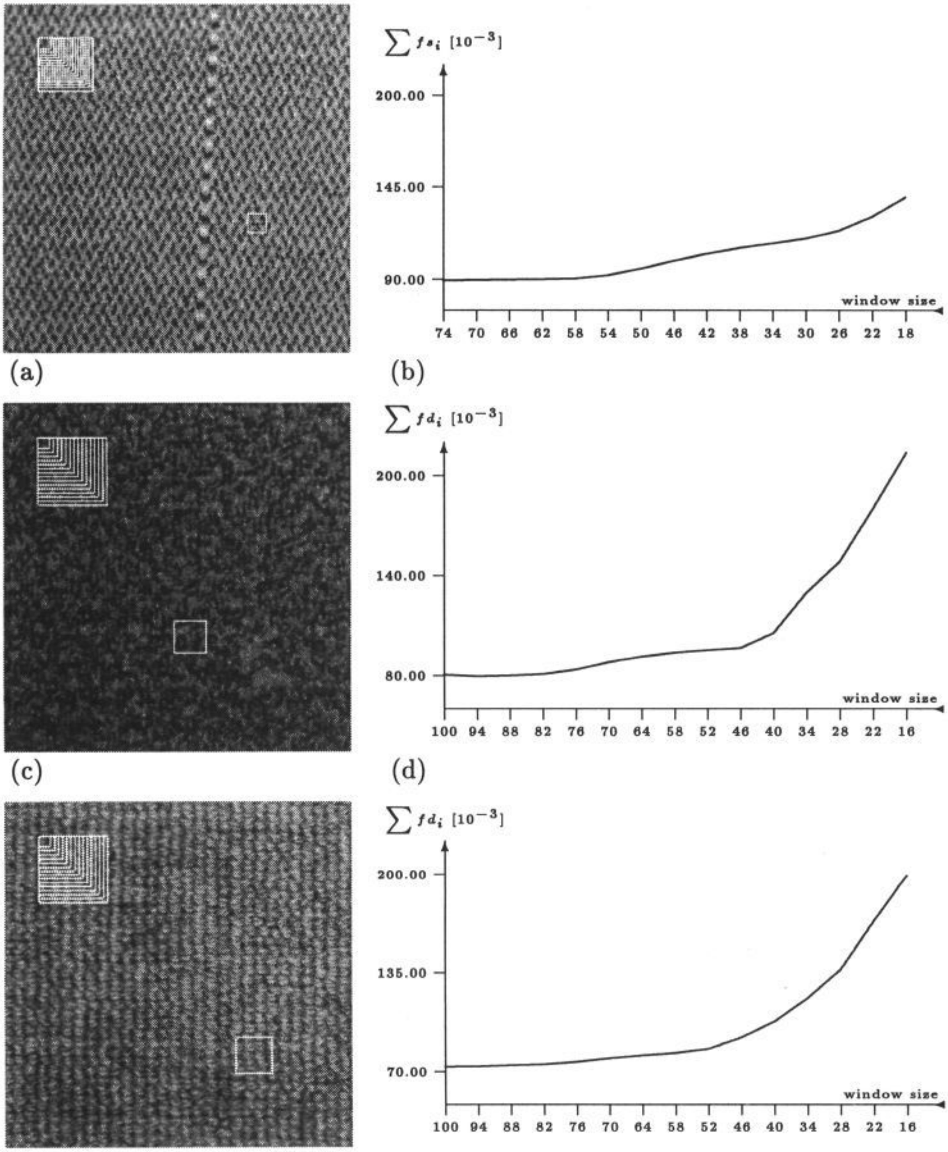


Figure 7: **(a),(c),(e)** The texture images on the left contain the investigated window sizes (upper left frames) and the resultant window size (lower right frame). **(b),(d),(f)** The summed FDF $\sum_i fd_i$ of the textures. For summation the low pass filtered deviation functions of all features were utilized. Since the texture in **(a)** is a kind of periodic the FDF increases two times, for the single period length 26 and the double period length 52. The resultant window size is 26 $\times$ 26. The resultant window size is 46 $\times$ 46 for **(c)** and 52 $\times$ 52 for **(e)**.

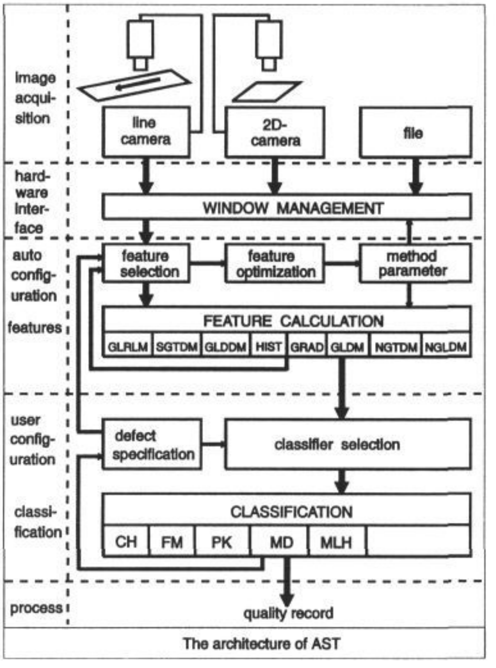
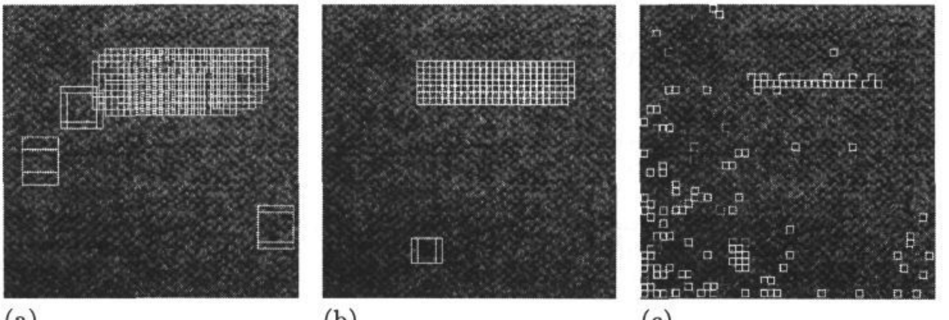


Figure 8: The architecture of the texture defect detection system AST (analysis of statistical textures).



(a) (b) (c)

Figure 9: The results for processing the carpet using three different window sizes. Highlighted windows correspond to windows classified as defects (see fig.1). In **(a)** a large window size, in **(b)** the calculated optimal window size and in **(c)** a small window size were used.

(b) NGLDM_angular second moment, NGLDM_entrpy, GLDDM_grey level distribution, HIST_3rd moment
(c) HIST_3rd moment, SGTDM_inverse difference moment from direction 0°, GLDDM_grey level distribution, GLDDM_grey level average

The computation of one window of the correct size takes 0.4 seconds using a MC68020-16MHz processor.

5 Conclusions

We presented an approach for window size determination using statistical texture analysis methods for texture defect detection which is based on the introduced feature deviation function.

Results have been presented to show the relevance of the calculated window size. Feature deviation functions have been shown for a periodical fabric texture and a statistical carpet texture. For the carpet classification results using different window sizes have shown the relevance of window size determination.

The analysis of a carpet texture of 15cm×15cm size takes 60 seconds using a MC68020-16MHz processor. To reduce computation time we realized parallel processing by use of transputers (Inmos T800). In this case computation time depends on window size, link speed and the actual number of transputers, e.g. using 4 transputers computation time has been ruduced 5 times ([AMEL94]).

References

[AMAD89] Moses Amadasun and Robert King: Textural Features Corresponding to Textural Properties; IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-19 (1989), No.5, pp.1264-1274.

[AMEL92] Jörg Amelung, Kai Vogel: Automatische Konfiguration für bildpunktbezogene Texturanalyseverfahren; Proc.37.Int.wiss. Kolloquium, Ilmenau, 1992, pp.533-538.

[AMEL93] Jörg Amelung: Ein Verfahren zur effizienten Merkmalsauswahl für die Texturfehleranalyse; Proc.15.DAGM Symp.Mustererkennung, Lübeck, 1993, pp.552-559.

[AMEL94] Jörg Amelung, Michael Tönsmann, Steffen Leonhard: Einsatz paralleler Rechnerstrukturen für die bildgestützte Fehlererkennung in texturierten Oberflächen; Proc.39.Int.wiss. Kolloquium, Ilmenau, 1994.

[DAPE86] Zhang Dapeng and Li Zhongrong: Digital image texture analysis using gray level and energy cooccurrence; Proc. SPIE Int. Soc. Opt. Eng. Conference on Applications of Artificial Intelligence IV, Innsbruck, Austria, 15-16 April 1986, Vol.657 (1986), pp.152-156.

[DEWA88] P.Dewaele, L.Van Gool, P.Wambacq and A.Oosterlinck: Texture Inspection with Self-adaptive Convolution Filters; Proc.IEEE 9th Intern.Conf.on Pattern Recognition, Washington DC, 1988, pp.56-60.

[GALL75] Mary M. Galloway: Texture Analysis Using Gray Level Run Lengths; Computer Graphics and Image Processing, Vol.CGIP-4 (1975), pp.172-179.

[HARA73] Robert M.Haralick, K.Shanmugam and I.Dinstein: Textural Features for Image Classification; IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-3 (1973), No.6, pp.610-621.

[HE88] Dong-Chen He, Li Wang und Jean Guibert: Textural Discrimination Based On An Optimal Utilization Of Texture Features; Pattern Recognition, Vol.21 (1988), No.2, pp.141-146.

[SUN82] Sun,C.and Wee,W.G.: Neighboring Gray Level Dependence Matrix for Texture Classification; Computer Vison,Graphics and Image Processing, Vol.23 (1982), pp.341-352.