Real-Time Imaging **7**, 507–518 (2001)
doi:10.1006/rtim.2001.0231, available online at http://www.idealibrary.com on IDEAL®

Real-Time Vision-Based System for Textile Fabric Inspection

This paper presents an automatic vision-based system for quality control of web textile fabrics. The general hardware and software platform developed to solve this problem is presented and a powerful algorithm for defect inspection is proposed. Based on the improved binary, textural and neural network algorithms the proposed method gives good results in the detection of many types of fabric defects under real industrial conditions, where the presence of many types of noise is an inevitable phenomenon. A high detection rate with good localization accuracy, low rate of false alarms, compatibility with standard inspection tools and low price are the main advantages of the proposed system as well as the overall inspection approach.

Radovan Stojanovic†, Panagiotis Mitropulos, Christos Koulamas, Yorgos Karayiannis*, Stavros Koubias and George Papadopoulos

*Applied Electronics Laboratory and *VLSI Laboratory, Department of Electrical and Computer Engineering, University of Patras, Patras, Greece*

Introduction

Product inspection is an important aspect of modern industrial manufacturing. The high cost of human visual inspection has led to the development of on-line vision-based systems capable of performing inspection tasks [1,2]. The problem of web inspection is particularly important and complex, and the research in this field is wide open [3]. Web inspection systems are currently used for quality control in numerous production lines such as for cloths and fabrics [4–15], cable insulators [16], paper [17], plastic bags [18], strip steel and metals [19–21], wood [22,23] and leather [24].

The implementation of an automated visual inspection system for defect inspection in the textile industry is of crucial importance. It has been shown [25] that defects reduce the price of the fabrics by 45% to 65%. A typical web material is 1–3 m wide and is driven with speeds ranging from 20 m/min to 200 m/min. Good inspection results can be achieved if the horizontal and vertical resolution is less than 1 mm [4]. In the best case, a man can detect no more than 60% of the present defects, and he cannot deal with fabric wider than 2 meters and moving faster than 30 m/min. On the other hand, in the literature, 235 types of defect and their possible causes are discussed [25]. Their correct detection and classification is a challenging task, and major problems such as extremely high data flow, noise influence, large numbers of defect classes, dynamic defect populations, inter-class similarity and inter-class

†Corresponding author. E-mail: radovan@apel.ee.upatras.gr; Tel.: +30 61 997 312; fax: +30 61 997 333.

diversity, compatibility with standard production lines and economical justification, have yet to be solved.

Many attempts have been made to solve these problems. A simple system, based on adaptive thresholding and binary filtering is described in [4]. Although the influence of noise is observed, its distribution is normal and its amplitude is small. The analyzed defects have a high contrast. The classification procedure is not presented in detail. In [5] a real-time optical system for defect detection is shown. It is based on light scattering and uses electro-optical equipment for defect detection. This approach produces good results for some types of defect, but requires complex and expensive equipment. In [6] a system for detecting one type of defect (''pilling'') in five grades is presented. The Radon transform is used for feature extraction and fuzzy logic is implemented for rating. A method for detecting low contrast defects is described in [7]. Four separate segmentation methods are considered. They are based on direct thresholding as well as thresholding after filtering pre-processing. However, the influence of noise and the classification problem are not considered. In [8] a variety of defects are analyzed for different fabrics and the flaws are finally segmented from the background. Gabor functions are used for feature extraction. In [9] a morphological image processing for gray-level inspections is illustrated and the feature extraction system implemented on a multi-layer artificial neural network is described in [10]. A solution based on a high-resolution linear multi sensor device is presented in [11], while the systems presented in [12] and [13] use standard matrix cameras with associated hardware. They require many cameras due to their inherent low resolution, making the final systems expensive and complicated. The Wavelets transform, fuzzy logic and neural networks are implemented in [14]. A vision-based fabric inspection system that accomplishes on-loom inspection and uses a Wavelet transform and edge fusion for feature extraction is described in [15]. In its final implementation the system is expensive and complicated because even small textile plants have over 20 looms.

The fabric inspection system, developed by the authors and proposed here, differs from existing systems in two crucial ways. Firstly, it uses low cost off-the-shelf components and can be mounted on a standard inspection machine without any adaptation. Secondly, aiming by necessity to reduce data flow, to extract the best features for defect description and to classify them using a reliable and fast classifier, the system is equipped with a suitable mixed inspection technique that is based on the concept of binary image processing, statistical texture analysis and RTC (Run Time Code) neural classification. The efficiency of this algorithm, as well as the overall inspection system, has been tested thoroughly under experimental and realistic conditions.

In what follows, we describe the proposed fabric inspection system in terms of its experimental and realistic acquisition architecture and its inspection algorithms. In the section prior to our conclusions, the emphasis is given to the results of an extensive performance-evaluation test.

Overview of the Proposed Inspection Architecture

The overall system for fabric fault inspection has been developed through its experimental and industrial phases. The architecture of the experimental system is presented in Figure 1(a). It consists of the unwinding machine [Figure 1(b)], lighting system, image processing hardware, and software. The unwinding machine, which simulates the cloth inspection table, consists of the five degrees of freedom mechanical set-up and fabric feeding mechanism.

For the lighting, a specular illumination technique is employed [1]. An air-cooled fluorescent tube provides a 2200–6460 Lux intensity and operates at 40–50 KHz in order to avoid flicker. The tube is about 10 cm away from the fabric. The fabric is scanned in ''warp direction'' by a 2048 element line-scan camera LC1912 (EG & G Reticon), enabling an acquisition speed of up to 7000 lines per second. The power and timing signals are provided by a modular camera controller RS1910 (EG & G Reticon). The line-scan processor board DT2856 (Data Translation) is used for the line acquisition and low-level data pre-processing. A DSP board DT2878 is used for high-level data processing, including a statistical texture features computation and neural classification. An FPGA (Field Programmable Gate Array)-based synchronization trigger board and an incremental encoder enable synchronization between fabric movement and line acquisition rate. A Pentium-based machine is used as a host computer for system coordination and control, binary image processing and industrial network support. Al the software, including board programming and image processing algorithms, has been developed in the ''C'' language in a Microsoft Windows environment.

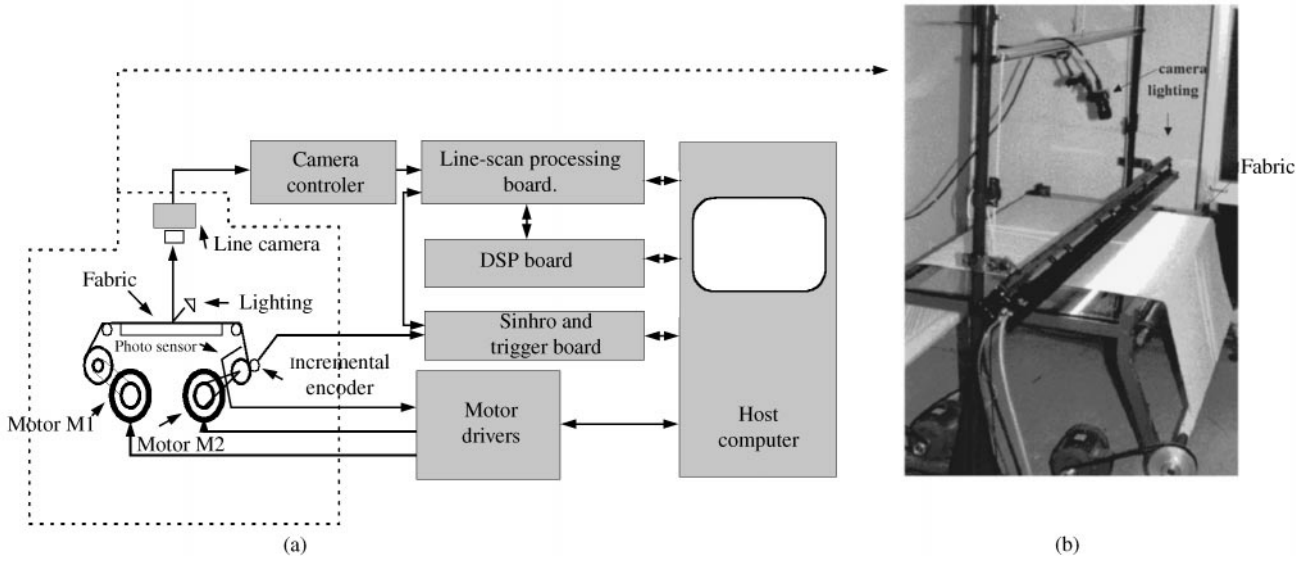

Figure 1. (a) Architecture of the implemented system; (b) unwinding machine.

After experimental evaluation, the main part of the system described above (expect the unwinding machine, lighting and motor drivers) is mounted on the standard fabric inspection machine [Figure 2(a)]. For the lighting, a fibre optics illumination is employed. It consists of two quarts halogen fibre optics illuminators and FIBER-LITE line light configuration with rod lens that can provide a stripe in excess of 50.000 Lux at 1 m. An internal regulation circuit provides constant light with a variation of $+/-2\%$ for a range of mains voltage of 200–280V AC. A Pentium II/200 MHz machine is used as a host. In an industrial environment, as opposed to an experimental situation, the following problems present themselves:

(1) The inspected surface of the fabric is not ideally smooth, and the winding mechanisms are not sufficiently synchronized, resulting in wrinkled fabric [Figure 2(b)], particularly at the beginning

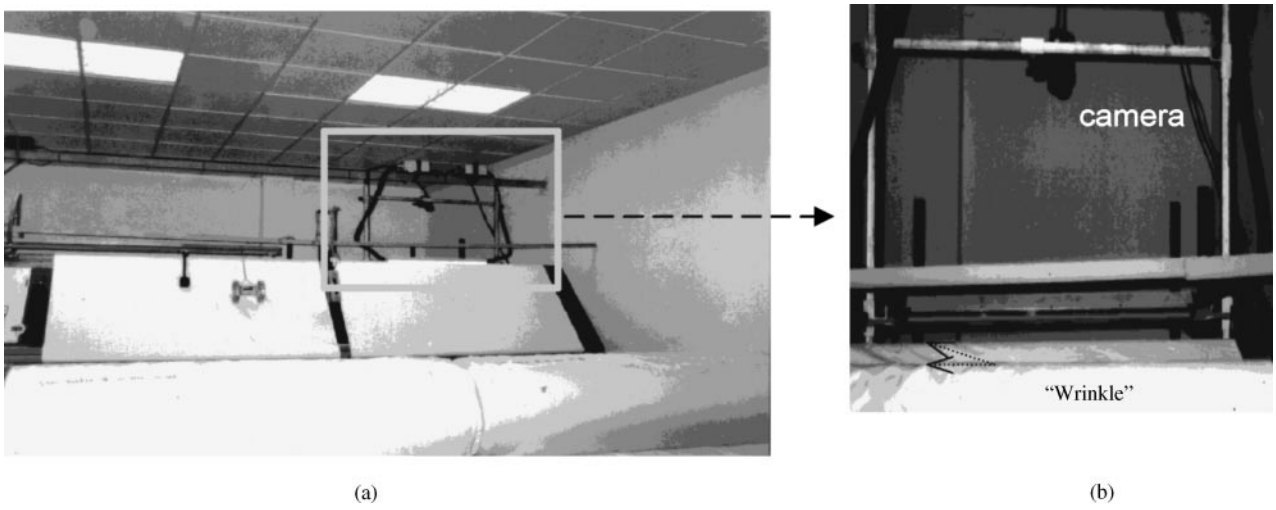

Figure 2. (a) Inspection table with mounted vision system (Courtesy of Siamidis A.E.); (b) fabric on the inspection table (a surface is not smooth).

and the end of the roll. Usually, the wrinkles run in the direction of motion.
(2) The illumination fluctuates with respect to time during machine activity. Sometimes, the voltage drops (generated by motor activity) are so great that the light regulation cannot work properly. Also, ambient light influences the entire light intensity by 1.5–2%.
(3) The electronic motor drivers induce a high frequency noise that sometimes interferes with the camera signals.
(4) The fabric produces impurities (dust). Usually, it flies between the lens and the fabric, or settles on the lens.

In many standard solutions the wrinkling is avoided by using a special machine or mechanisms for stretching the fabric, rapidly increasing the system cost. In order to escape additional mechanical constructions we describe wrinkling as a separate "wrinkle" error and recognize it using appropriate algorithms in the inspection procedure. The illumination problem is solved by adaptive thresholding, described in the next section, while a fan mounted over the scanning area removes the dust.

Image Processing Approach

Process requirements and defect characteristics

As previously mentioned, the processing time and accuracy rate of defect detection and recognition can be considered as crucial, restricting criteria in the final choice of the algorithms for automated fabric inspection.

Consider a 1 m wide fabric moving at a speed of 2 m/s, and containing defects of 1 mm × 1 mm in size. If a defect must be represented by a minimum of two pixels in both directions (i.e. a spatial resolution of 0.5 mm/pixel) then it is necessary, in real-time data acquisition, to realize a data flow of 8 MB/s for a line-scan camera of 2048 pixels. For applications demanding full data processing, defect detection and classification must be also considered.

Furthermore, a typical defect detection and classification problem involves a large number of defect classes, and the system must be able to deal with a few dozen to a few hundred classes. Sometimes, a single class of defects may vary widely in appearance and may have members that closely resemble defects in another class. On the other hand, small changes in the production process can result in entirely new classes of defects, therefore a useful classification system should be dynamic, with the ability of continuous on-line learning.

The above items make initial system design very difficult, and very fast hardware and suitable software solutions must be implemented to achieve this task. Towards this aim, we examined many approaches using the described experimental set-up [26]. Also, the experience of human inspectors, textile experts and industrial statistics yielded the following conclusion, which can simplify the design task:

Approximately 80% of fabric defects have a preferred orientation, either in the direction of motion (i.e., warp direction) or perpendicular to it (i.e., pick direction). Many defects are caused by machine activity such as holes, oil spots, wrinkle or dirt. On the other hand, warp or pick defects tend to be long and narrow, while slubs can produce point defects. Generally, they change image intensity and apparent texture of the weaving pattern in local fabric areas. Following this logic, the defects can be classified into groups that are formed using different criteria (i.e., Marks & Spencer System, for details see [27]). For example, slubs, knots, tangles and dirt can be considered as a "dark spot," big loops and skipped threads as a "light filling," while double warp thread can be seen as a "dark warp". In our approach, we use eight groups for defect description, which will be discussed in a later section.

Following the above facts, as well as general demands that have already been mentioned, we propose the following inspection algorithm.

Defect inspection algorithm

Seen generally, our inspection algorithm is composed of two algorithmic modules, the defect detection and feature extraction module (Module I) and the defect classification module (Module II), as shown in Figure 3.

Defect detection and feature extraction

This module determines the region covered by the defect, and specifies the information which must be used to identify the defect. The initial stage of this module, as well as of the overall inspection algorithm, is the defect detection.

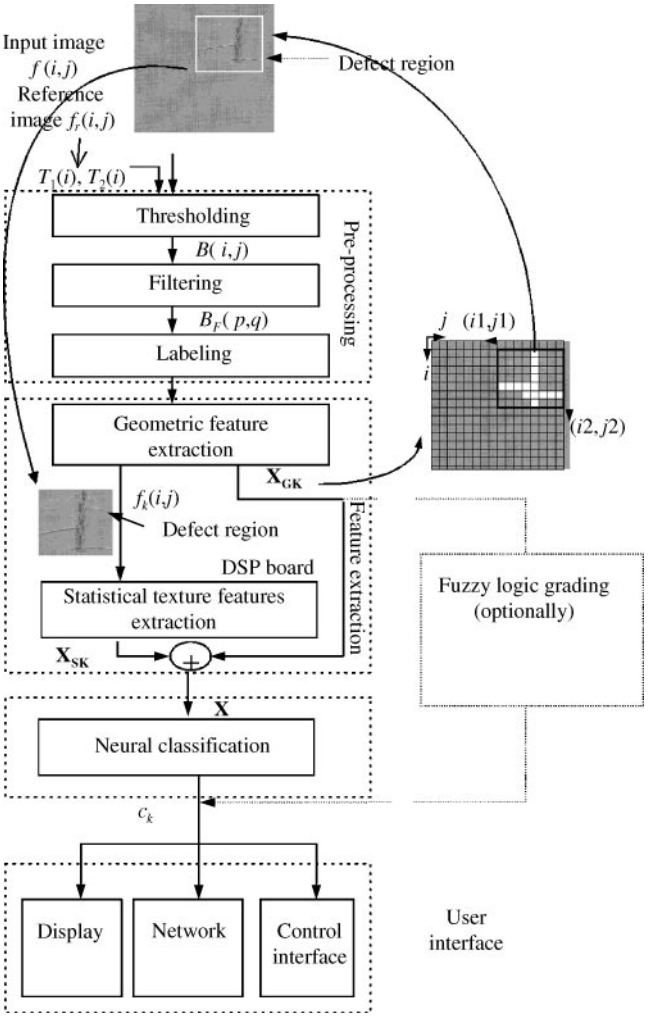

Figure 3. Flow diagram of the proposed inspection algorithm.

Defect detection. It establishes whether a defect of some kind is present or not. Using binary algorithms it reduces the amount of data that needs to be processed. For this stage, consider the following:

- $f(i, j)$ — input image of the fabric with N_g gray-levels and M × N pixels. The parameters i and j indicate the wrap and the pick directions ($0 \le i < M$, $0 \le j < N$);
- $f_r(i, j)$ — reference image, image of a "good fabric" ($0 \le i < M$, $0 \le j < N$);
- $f_k(i, j)$ — sub-image inside the image $f(i, j)$ that covers one of k defect regions ($l \le k < N_L$);
- $l(j)$ — one line of the image $f(i,j)$ ($0 \le j < N$);
- $B(i, j)$ — binary image obtained after thresholding of $f(i, j)$ ($0 \le i < M$, $0 \le j < N$);
- W_τ — window (of size $a \times b$ pixels) with overlapping step τ;
- B_F (p, q) — reduced binary image obtained after filtering image $B(i, j)$ [$0 \le p < r = M/a$, $0 \le q < s = N/(b-\tau)$];
- E_c — morphological erode operator given as;

$$B \quad C = \{\lambda | C_\lambda \subseteq B\} \tag{1}$$

[this means that erosion of a binary image $B(i, j)$ by a binary image C (structuring element) is 1 at a pixel λ if and only if every 1 pixel in the translation of C to λ is also 1 in B];

- *Sum* — arithmetic sum of the binary values inside the window W_τ

$$Sum = \sum_{w} B(i,j) \tag{2}$$

- *Mean* — mean value of the elements in the vector **X**
- *Std* — standard deviation of the vector **X**

The first operation that is applied to the incoming data from the line-scan camera is thresholding (Figure 4). It is a thresholding with two adaptive point-to-point thresholds $T_1(j)$ (upper) and $T_2(j)$ (lower). Their initial values are determined during the set-up procedure (see Figure 4), using statistical parameters of the defect-free image $f_r(i, j)$. During inspection these values are refreshed using statistical parameters from N_s successive defect-free lines $l_s(j)$ ($0 < s \le N_s$) of the inspected fabric. For one point j the thresholds $T_1(j)$ and $T_2(j)$ are determined as

$$T_{1,2}(j) = mean[l_s(j)] \pm a^* mean[std(l_s(j))] \tag{3}$$

where a represents the constant between 1 and 2. The output from this operation is the binary image $B(i, j)$ with zeros (0) for $T_1(j) < f(i, j) < T_2(j)$ and ones (1) elsewhere.

However, the signals received in practice are not ideal and noise tends to mask the defect indications. Here, two kinds of noise are present: low frequency modulations caused mainly by non-uniform illumination, and high frequency modulations. The latter arise partially from variations in responsivity between the photosites in the CCD array, and partially from the stitch structure of the fabric. Both the illumination variations and the photoside non-uniformity are compensated following camera calibration in the set-up procedure and during adaptive thresholding, but the stitch structure is effectively a random signal, which cannot easily be removed. Towards this aim, the binary image $B(i, j)$ is

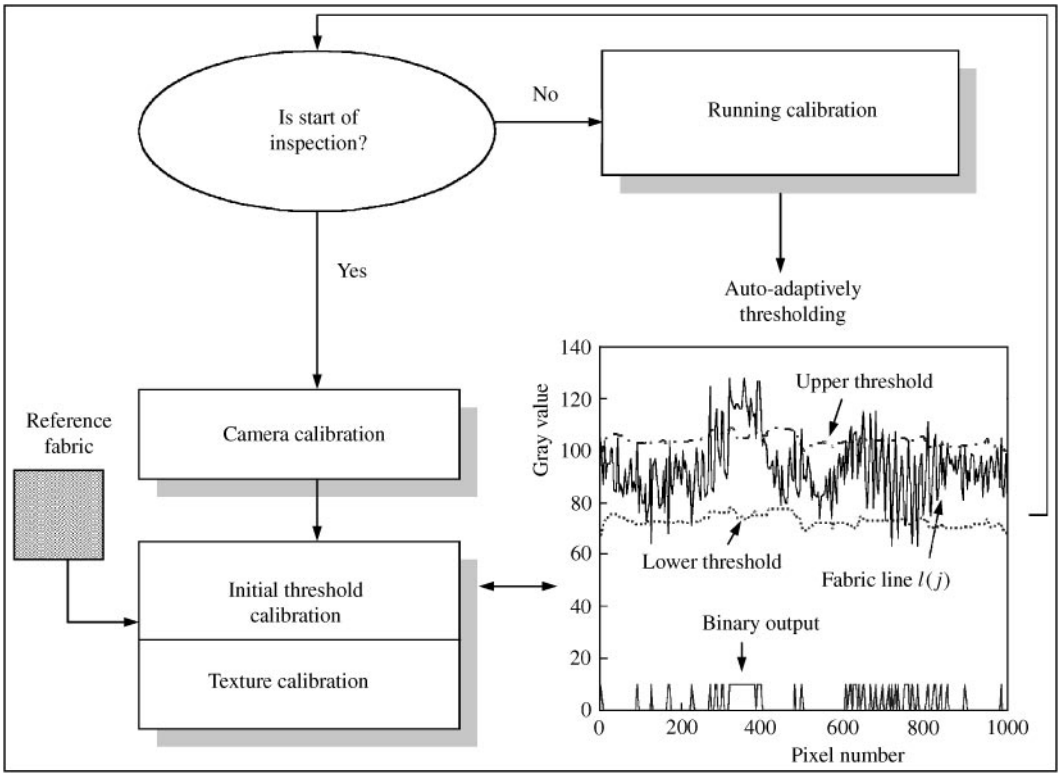

Figure 4. Calibration procedure and thresholding operation.

filtered by operator E_c and then successively averaged in the sliding windows $W\tau$ by operator *Sum*:

$$B_1(i,j) = E_c[(B(i,j))] \tag{4}$$

$$B_F(p,q) = \begin{cases} 1 & Sum[(B_1(i,j))]_{insideW} \geq T_s \\ 0 & elsewhere \end{cases} \tag{5}$$

As a result, the reduced binary image $B_F(p,q)$ is obtained and the possible defects can be considered as non-connected binary objects inside it. The choice of the structuring element C and of the threshold T_s significantly influences the required inspection resolution. For example, for $C = \{1,1\}$ or $\{1,1\}^T$, the minimum value of T_s that provides a resolution of 1 mm in both directions is 2 for $\tau = 0$ and for a sensor 1 array of 2048 pixels with one meter field of view.

To label the above objects L_k ($1 \leq k \leq N_L$) the labeling algorithm from [28] is modified and implemented. Simultaneously, the co-ordinates (i_{k1}, i_{k2}, j_{k1}, j_{k2}) of appropriate gray-level sub-images $fk(m, n)$ that cover the defects are determined. The implemented detection procedures are illustrated in Figure 5. As an example, a fabric image with two defects is shown in Figure 5(a). Figure 5(b) gives its binary representation after thresholding, while Figures 5(c) and 5(d) illustrate the filtering results. For the erode, the structuring element $C = \{1,1\}$ is selected. Averaging is performed in windows of 16×16 pixels. Finally, after labeling, the defects $D1$ and $D2$ are separated in the form of gray-level sub-images $f_{k1}(m, n)$ and $f_{k2}(m, n)$ [Figure 5(e) and Figure 5(f)].

Feature extraction. In this section we specify the features (information), which are used to identify the defect. We use two feature sets: the geometric features $\mathbf{X_{GK}}$ extracted from the binary image B_F (p, q) and statistical texture features $\mathbf{X_{SK}}$ computed from the sub-image fk. These two sets define the feature vector $\mathbf{X_K} = \mathbf{X_{GK}} \cup \mathbf{X_{SK}}$, which becomes the input of a neural network classifier. The geometric features are based on the geometric properties of the objects distinguished from the background (length, width, area etc.). Some of the ones used in our approach are given in Table 1.

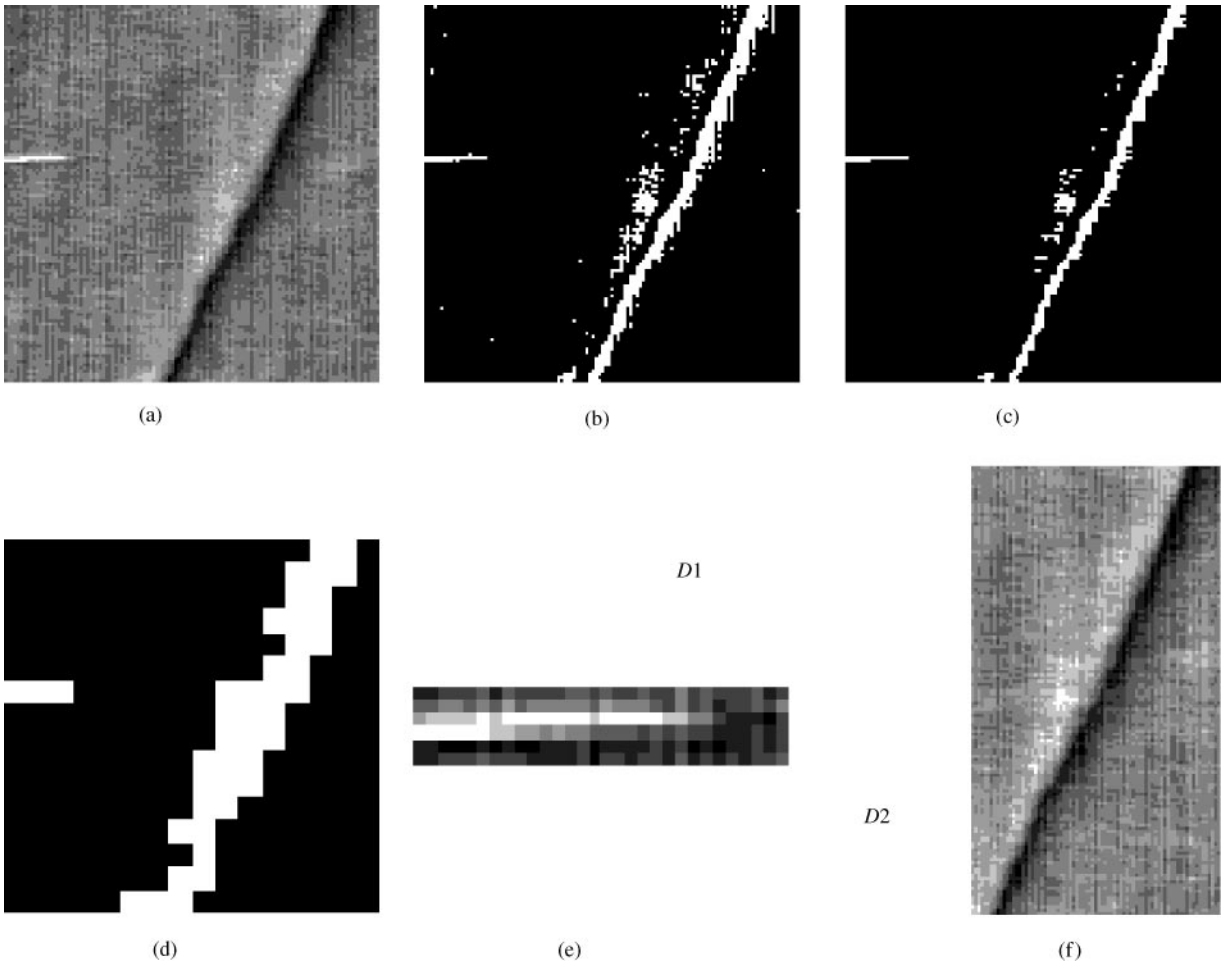

Figure 5. (a) Gray-level fabric image $f(i, j)$ (256×256 pixels, $Ng = 256$) with two defects $D1$ and $D2$; (b) binary image $B(i, j)$ obtained as a result of thresholding operator; (c) binary image $B_1(i, j)$ obtained after filtering of image $B(i, j)$ by erode operation $C = \{1, 1\}$; (d) reduced binary image $B_F(p, q)$ (16×16 pixels) obtained after averaging of image $B_F(p, q)$ in the windows W_τ (16×16 pixels, $\tau = 0$) and its thresholding by $T_s = 4$; (e) defect sub-image $f_{k1}(m, n)$ ($D1$) obtained from input image $f(i, j)$ using geometric coordinates from Figure $B_F(p, q)$; (f) defect sub-image $f_{k2}(m, n)(D2)$.

As a texture feature extractor the Gray Level Difference Method (GLDM) [29,30] is selected. If (m,n) and $(m+\ m, n+\ n)$ present two points of the image $f_k(m,n)$ with difference intensity $g_\delta = |f(m,n) - f(m+\ m, n+\ n)|$ for displacement vector $\delta(\ m,\ n)$ ($\ m$ marks the vertical and $\ n$ the horizontal direction), then

$$p(i, \delta) = \frac{1}{A} \sum_{i=1}^{i=r} \sum_{j=1}^{i=s} (g_\delta(m, n) = i) \tag{6}$$

denotes the estimated probability function for vector δ and normalization factor A. For each of the vectors $V[p(i, \delta)]$ $(0 \le i < Ng)$ a set of features can be calculated for different δ according to the Harallic approach [29] (See Table 2). In our case, we use given texture features for $\delta = (0, 1)$, $\delta = (1, 0)$, $\delta = (1, 1)$, $\delta = (0, 2)$, $\delta = (2, 0)$ and $\delta = (2, 2)$.

The computation speed, memory requirements, immunity to low frequency noises (slow change in the illumination), good results in the implementation after gradient pre-filtering and easy hardware implementation are the major reasons why the GLDM is selected as a feature descriptor. The above advantages became apparent after analysis and comparison of well-known methods of feature extraction, such as The First-Order histogram statistics [31], Co-occurrence matrix [29], Spectral Ring-Angle operators [31], Run-Length matrix [29], Fractal dimension [32], Markov-Random field [33]

Table 1. Geometric features for an isolated object (defect), coordinates of the vector $\mathbf{X_{GK}}$, given for an object L_K

Feature	Equation
Area	$Ar_{Lk} = \sum_{p=0}^{r}\sum_{q=0}^{s}(B(p,q) = L_K)$
Coordinates of the center of the region	$X_{Lk} = \frac{1}{Ar_{Lk}}\sum_{p=0}^{r}\sum_{q=0}^{s}(B(p,q) = L_K)$, $Y_{Lk} = \frac{1}{Ar_{Lk}}\sum_{p=0}^{r}\sum_{q=0}^{s} p \cdot (B(p,q) = L_K)$
X projection and Y projection	$Xpr_{Lk} = length\left(\sum_{q=1}^{s}(B(p,q) = L_k)\right)$, $Ypr_{Lk} = length\left(\sum_{p=1}^{r}(B(p,q) = L_k)\right)$
Orientation	$\alpha_{Lk} = \frac{1}{2}\arctan\frac{2\sum_{p=1}^{r}\sum_{q=1}^{s}(p-X_{Lk})(q-Y_{Lk})(B(p,q)=L_K)}{\left(\sum_{p=1}^{r}\sum_{q=1}^{s}(p-X_{Lk})^2(B(p,q)=L_K)\right)-\left(\sum_{p=1}^{r}\sum_{q=1}^{s}(q-Y_{Lk})^2(B(p,q)=L_K)\right)}$

Table 2. Texture features, the coordinates of the vector $\mathbf{X_{SK}}$, given for one sub-image $f_k(m, n)$

Feature	Equation
Contrast	$CON\_\delta = \sum_{i=0}^{Ng-1} i^2 p(i,\delta)$
Angular Second Moment	$ASM\_\delta = \sum_{i=0}^{Ng-1} [p(i,\delta)]^2$
Mean Value	$MEAN\_\delta = \sum_{i=0}^{Ng-1} ip(i,\delta)$
Entropy	$NTRP\_\delta = \sum_{i=0}^{Ng-1} p(i,\delta_a)\log(p(i,\delta))$
Homogeneity	$HOMO\_\delta = \sum_{i=0}^{Ng-1} \frac{1}{1+i^2} p(i,\delta)$
Average value of features	$x_{FE}\_AV = \sum_{i=1}^{i-m} x_{FE\_\delta i}$

and Multi-resolution Wavelet analysis [15] (For details see [26]).

To illustrate how the above features can describe the fabric defects, look at Figure 5(a) again. Tables 3 and 4 give the geometric and texture features for defects $D1$ and $D2$, extracted from binary image $B_F(p, q)$ and from gray-level sub-images $f_{k1}(m, n)$ and $f_k(m, n)$. The first two rows of Table 4 represent the reference texture features obtained from the same kind of fabric in the set-up procedure. For example, we see that defect $D1$ is a small horizontal defect with $Ar_L = 3$, $Y\_pr_L = 3$, $X\_pr_L = 1$ and $\alpha = 0$ [rad], while defect $D2$ presents a large defect along the whole of the image with specific orientation $\alpha = 0.81 \cong 45^\circ$. Using only this information, defect $D2$ can be recognized as a "wrinkle" defect (specific orientation and length), while to closely describe defect $D1$, additional features must be extracted. This can be done by texture analysis of image $f_{k1}(m, n)$. Now we see that, in comparison with reference features, defect $D1$ has an extremely high contrast in the vertical direction [see parameters for $D1$ in direction $\delta = (1, 0)$ in Table 4] as well as increased values of mean value and entropy. On the other hand, the value of angular second moment in the same direction is small. Using these facts, defect $D1$ can now be recognized as a horizontal compact small white defect. In our classification scheme, which will be explained later, this defect will be seen as "white horizontal error." On the other hand, from a textural point of view, the defect $D2$ differs distinctly from reference parameters only in direction $\delta = (1, 0)$, more like the "black defect."

Note that the geometric features sometimes give enough information for the defects description, and in this case a fuzzy logic grading system (shown as optional in Figure 3) can be used as an effective solution for classification. For example, one warp defect ("double filling") using fuzzy rules can be described as: If *defect size* is **medium** and if *defect orientation* is **small** and if *x_projection* is **big** and if *y_projection* is **small** then *defect* is **horizontal medium**. In order to make a robust inspection system, this approach will be a subject of our future research. To improve computation speed, the host (PC) executes the defect detection and geometric feature extraction, while the DSP board performs the texture features and classification.

Table 3. Geometric features from Table 1 for defects from Figure 5, obtained from Figure 5(d)

	Ar_L	X_L	Y_L	X_pr_L	Y_pr_L	α_L[rad]
Defect $D1$	3	1	5	3	1	0
Defect $D2$	38	9	6	10	16	0.81

Table 4. Texture features extracted from images of defects $D1$ and $D2$

	CON	*ASM*	*MEAN*	*NTRP*	*HOMO*
"good fabric" $\delta = (0, 1)$	5.92e+01	7.03e−02	6.45e+00	−2.74e+00	1.26e−01
"good fabric" $\delta = (1, 0)$	3.34e+0.1	9.65e−0	4.66e+00	−2.49e+00	1.91e−01
Defect $D1$ $\delta = (0, 1)$	5.06e+01	8.52e−02	5.78e+00	−2.62+00	1.48e−01
Defect $D1$ $\delta = (1, 0)$	3.40e+02	3.84e−02	1.44e+01	−3.43e+00	8.99e−02
Defect $D2$ $\delta = (0, 1)$	5.85e+01	7.05e−02	6.20e+00	−2.79e+00	1.38e−01
Defect $D2$ $\delta = (1, 0)$	2.43e+01	1.22e−01	3.82e+00	−2.32+00	2.64e−01

Table 5. Reduced classes of fabric defects

Error class	Error type
c_1	No error
c_2	Black vertical error
c_3	White vertical error
c_4	Wrinkle
c_5	Black horizontal error
c_6	White horizontal error
c_7	Black spot
c_8	White spot

Classification

For defect classification, we implemented a three-layer ANN (Artificial Neural Network) in the form of an RTC (Run Time Code) machine. Several areas, with and without errors, were scanned from the moving fabric. The feature vector $\mathbf{X_K}$ is used as an input. The defect classes are reduced, using a defect point system, into eight classes c_i $(1 <= i <= 8)$, presented in Table 5. The data were treated as noisy and specific transformations were applied to them in order to create the appropriate variables of the ANN. The performance of the network model was evaluated by using the average classification rate, i.e. the average of the class dependent fractions of correct classifications over all fault categories. Several empiric tests were performed in order to fix the size of the hidden second layer: it had to contain only a small number of nodes provided that it could achieve a satisfactory performance in error classification. Thus, several tests were performed with different hidden layers that ranged from four to 12 nodes. Finally, it was decided to use an ANN that consisted of only five nodes. The outputs of the third layer are the general classes of the possible error types. As can be seen in Table 5, the errors are grouped into eight general categories. In each category more than one fabric error can be included. It is obvious that further research has to be performed on the methodology in order to achieve error classification results in more categories, each one of them containing exact fabric errors. The outputs of the third layer are these eight general classes of the possible error types, expressed as percentages of similarity to the learnt patterns. These percentages are derived from the internal mapping of the network, a product of its learning.

The back-propagation weight update rule used for training an ANN is:

$$w_{ij} = \eta \delta_j o_i \tag{7}$$

$$\delta_j = \begin{cases} f(t_j - o_j) \\ f\left(\sum_k \delta_k w_{jk}\right) \end{cases} \tag{8}$$

where:

$f(.)$ is the network's activation function $f(\alpha) = \frac{1}{1+e^{-\alpha}}$;
w_{ij} is the weight of the link from unit i to unit j;
η learning factor eta;
δ_j error;
t_j teaching input of unit j;
o_j output of the preceding unit i;
i index of a predecessor to the current unit j with link w_{ij} from i to j;
j index of the current unit;
k index of a successor to the current unit j with link w_{ij} from i to j.

In the forward propagation phase of training a feed-forward ANN with supervized learning, an input pattern is presented to the network, then propagated forward in the net until activation reaches the output layer. The output of the output layer is then compared with the teaching input, and the difference (delta) i.e. error, between the output and the teaching input of a target output unit is then used together with the output of the source unit to compute the necessary changes of the link. The backward propagation phase occurs if there is no teaching input available from computing the deltas of inner units (units of hidden layers). In this case, the deltas of the following layer are used. Thus, the errors (deltas) are propagated backward.

A learning algorithm with an adaptive gradient learning rule, using back-propagated gradient information for guiding an iterative line search algorithm, was used in the ANN's learning procedure of the application [35,36]. The weight changes are applied to the network after each training pattern, i.e. after each forward and backward pass. It is performed off-line, prior to the fabric detection procedure, for an appropriate number of input features. The ANN training is performed on a Pentium 133 MHz PC, taking less than two minutes. 50 samples from each of the eight categories are used as an input in the training procedure.

Results and Discussion

A database of 128 images (2048×2048 pixels), randomly collected from 16 different rolls of the same fabric, was used to determine the performance of the system and to select an appropriate feature subset. The images are acquired in working conditions and stored without any post-processing. In the first phase, the human expert analyzed the images, off-line, and all defects were extracted and described.

In the second phase, the system performance was evaluated for recognition rate and inspection speed on the same database. For each of $Ns = 32$ subsets (with a different number of features $10 \leq n = dim(Ns) \leq 24$) the system is evaluated for recognition rate and processing speed using developed classifiers (each of which is trained for a specific feature subset). The definition of recognition rate is:

$$R_R = \frac{C.D. - F.A.}{T.D.} \times 100 \qquad (9)$$

where $C.D.$ is the number of defects that are correctly detected, $F.A.$ is the number of false alarms and $T.D.$ is the total number of defects in the database. The processing time is corrected by a factor $= tr - tb$, where tr is the time for image acquisition from the camera, and tb is the time for image reading from the database. Using conventional search techniques, the subset having the best recognition rate for required maximum speed is selected. Generally, the detection rate increases with the number of features, while the inspection speed decreases, as shown in Figure 6. For the required maximum processing speed of 120 m/min, the maximum number of features is 16 for a classification accuracy of 86.2%. This result was later confirmed in real-time testing. The fabric is first scanned, and random image sequences (about 64 from different rolls) are stored by appropriate defect reports. The scanning speed is increased to that necessary for storage of selected images. Then, the previous results are checked off-line

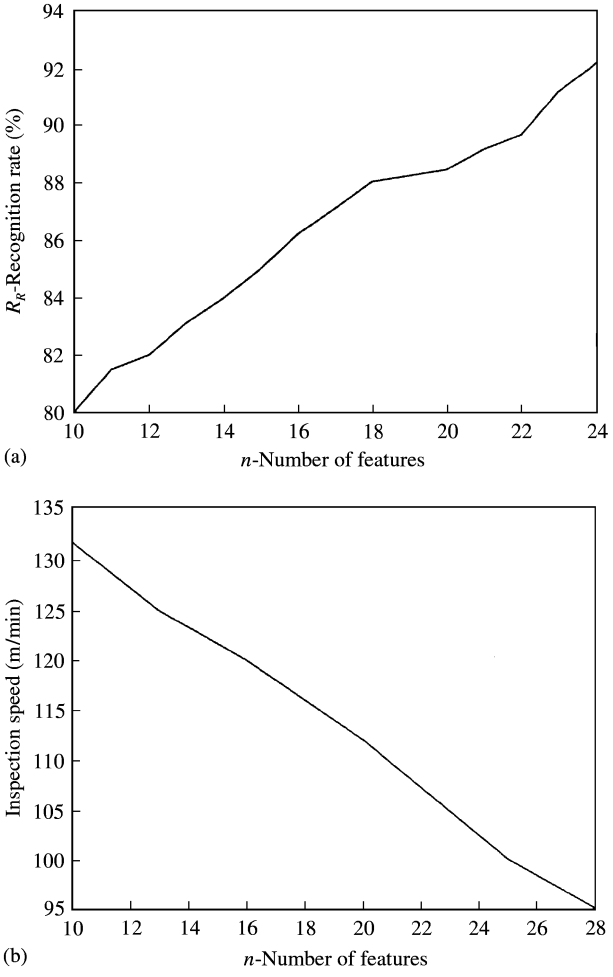

Figure 6. Recognition rate (a) and operation speed (b) as functions of number of features.

Table 6. General results of the proposed inspection method

Inspection	Maximal Inspection speed	Width of Testing Fabric (Field of view)	Space inspection resolution	Recognition Rate	Rate of False alarms	Localization
Manual	30 m/min	2 m	1 mm	50%	2.5%	—
Proposed	120 m/min	1 m	0.5 mm	86.2%	4.3%	5–8 mm

by a human expert. The error in detection rate between the two mentioned criteria for the selected set of features was about 0.78%. For lower processing speeds, the number of features can be increased, resulting in a better recognition rate. Also, by using a higher performance DSP board and host, the number of features and the recognition rate can be increased significantly. The performance results of the proposed system, including some auxiliary evaluation criteria (width of testing fabric, inspection resolution, and localization accuracy) are presented and compared with a human inspector in Table 6.

The results obtained in the evaluation and testing period indicate that a reliable automatic visual inspection system for moving web textiles can be created. However, the accuracy rate of inspection cannot be 100% at the required speed, using the simple and commercial hardware components. But, using suitable defect detection and classification algorithms that are based on the reduced number of defect classes (''defect point system'') this system can be designed. Otherwise, the complex and slow expert system such as in [25] must be incorporated for full classification. In this case the major problems are the complex hardware and software required, and the economic justification.

Conclusion

The problem of replacing the off-line inspection of textile fabrics by an automated visual system is very difficult and still unresolved. The major problem is to create the necessary image analysis and pattern recognition methods to quickly and accurately locate, identify, and determine the classes of the defects in the fabric using low cost standard equipment. A pilot approach for detecting and classifying defects of textile fabrics has been demonstrated. It uses low-cost off-the-shelf components, and can be mounted on any inspection machine without any adaptation. Also, it takes advantage of a suitable inspection technique that is based on the concept of binary image processing, statistical texture analysis and neural networks. The results obtained in the evaluation and testing phases indicate that a reliable visual inspection system for industrial requirements, based on the given approach, can be created.

References

1. Batchelor, B.G. & Whelan, P. F. (1994) Selected Papers on Industrial Vision Systems. SPIE Milestone Series.
2. Newman, T.S. & Jain, A.K. (1995) A survey of automated visual inspection, *Comput. Vis. Image Understanding*, **61**: 321–262.
3. Brzakovic, D. & Vujovic, N. (1996) Designing a defect classification systems: a case study, *Pattern Recognition*, **29**: 1401–1419.
4. Northon, L., Bradshaw, M. & Jewell, A.J. (1992) Machine vision inspection of web textile fabric, *Proc. British Machine Vision Conference*, pp. 217–226.
5. Harris, J.S. (1996) Optical system for real-time web-process defect inspection, *Proc. SPIE*, **2908**: 18–28.
6. Dar, I. M., Mahmood, W. & Vachtsevanos, G. (1997) Automated pilling detection and fuzzy classification of textile fabrics, *Proc. SPIE*, **3029**: 26–36.
7. Laitinen, J. (1997) Image quality in automated visual web inspection, *Proc. SPIE*, **3029**: 78–89.
8. Escofet, J., Navarro, R., Millan, M. & Pladellorens, J. (1996) Detection of local defects in textile webs using Gabor filters, *Proc. SPIE*, **2785**: 163–170.
9. Mueller, S. & Nickolay, B. (1994) Morphological image processing for the recognition of surface defects, *Proc. SPIE*, **2249**: 298–307.
10. Lepage, R., Laurendeau, D. & Gagnon, R. Extraction of texture features with a multiresolution neural network, *Proc. SPIE*, **1709**: 64–75.
11. Huart, J. & Postaire, J. (1994) Integration of computer vision on to weavers for quality control in the textile industry, *Proc SPIE*, **2183**: 155–163.
12. Takatoo, M., Takagi, Y. & Mori, T. (1988) Automated Fabric Inspection Using Image Processing Techniques, *Proc. SPIE*, **1004**: 151–158.
13. Virk, G.S., Wood, P.W. & Durkacz, I. D. (1990) Distributed Image Processing for the Quality Control of Industrial Fabrics, *Computing and Control Engineering Journal*, **1**: 241–246.
14. Research News, Georgia Institute of Technology (1997) On line fabric inspection system uses neural networks, fuzzy logic and wavelets to help improve textile quality.
15. AMTEX, Vision System for On-Loom Fabric Inspection, WWW presentation and technical report, http://amtex.-sandia.gov.

16. Mustafa, M., Bernard, R. & Rouse, M. (1995) Real-Time detection of defects on high tension cables insulator, *Proc. of The International Conference on Quality Control by Artificial Vision*, *QCAV95, June 1995, Le Creuset, France,* pp. 215–219.
17. Roberts, J.W., Rose, S.D., Jullien, G.A., Nichols, L., Jenkins, P., Chamberlain, S.G. & Maroscher, G. (1993) PC based real-time defect imaging system for high speed web inspection, *Proc. SPIE*, **1907**: 164–176.
18. Brown, C.C., Olsson, O.J., Palmer, G.T., & Penman, D.W. (1995) *On-line high resolution inspection of multi-layered plastic bags*, Document No. AB0031-1. 0 of Datacube, Datacube Inc., Danvers, MA 01923, August, 1995.
19. Rautrarukki, R. (1995) *Defect classification in surface inspection of strip steel*, Document No. AB0019-1. 0 of Datacube, Datacube Inc., Danvers Inc., Danvers, MA 01923, August 1995.
20. Don, H.S., Fu, K.S., Liu, C.R. & Lin, W.C. (1984) Metal surface inspection using image processing techniques, *IEEE Transaction on SMC*, Vol. SMC-14, No. 1, pp. 139–146.
21. Olsson, L.J. & Gruber, S. (1993) Web process inspection using neural classification of scattering light, *IEEE Transaction on industrial electronics*, **40**: 228–234.
22. Conners, R.W., McMillin, C.W., Lin, K.A., Espinosa, R.E. (1983) Identifying and locating surface defects in wood: part of an Automated Lumber Processing System. *IEEE Transaction on PAMI*, Vol. PAMI-5, No. 6, pp. 573–583.
23. Penman, D., Olsson, O. & Bowman, C. (1992) Automatic inspection of reconstituted wood panels for surface defects, *Proc. SPIE*, **1823**: 284–292.
24. Hoang, K., Wen, W., Nichimuthu, A. & Jiang, X.L. (1997) Achieving automation in lather surface inspection, *Computer in industry*, **34**: 43–54.
25. Srinivasan, K., Dastor, P.H., Radhakrishnaihan P. & Jayaraman, S. (1992) FDAS: A Knowledge-based Frame-detection work for Analysis of Defects in Woven textile Structures, *J. Text. Inst.*, **83**: 431–447.
26. Stojanovic, R., Koubias, S. & Papadopoulos, G. (1999) A Comparison of Statistical and Spectral Analytical Methods for Feature Extraction in the Process of Web Defect Detection, *Proceedings of IIA'99*, Genova, June 1999, pp. 350–356.
27. Nicolay, B., Schicktanz, K. & Schamalfuβ, H. (1993) *Automatic Textile Inspection*, Internal Study, Berlin, June 1993.
28. Jain, R., Kasturi, R. & Schunk, B.G. (1995) *Machine Vision*. New York: McGraw-Hill.
29. Haralic, R.M., Shanmugam, K. & Deinstein, I. (1973) Textural Features for Image Classification, *IEEE Transactions on Systems, Man and cybernetics*, Vol. SMC-3, No. 6, pp. 610–621.
30. Conners, R.W. & Harlow, C.A. (1980) A Theoretical Comparison of Texture Algorithms, *IEEE Transaction on PAMI*, Vol. PAMI-2, No. 3, pp. 204–222.
31. Gonzalez, R.C. & Woods, R.E. (1993) Digital Image Processing. New York: Addison-Wesley Publishing Company.
32. Dennis, T. & Dessipris, N. (1989) Fractal Modeling in Image Texture Analysis, *Proc. IEEE*, **136**: 227–235.
33. Cross, G. & Jain, A.K. (1983) Markov Random Field Texture Models, *IEEE Transaction on PAMI*, Vol. PAMI-5, No. 1, pp. 25–39.
34. Guantilake, P., Siegel, M., Jordan, A. & Paduar, G. (1997) Image Understanding for Remote Visual Inspection of Aircraft Surfaces, *Proc. SPIE*, **3029**: 1–13.
35. Lippmann, R.P. (1982) An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, pp. 4–2.
36. Fahlman, S.E. (1988) Faster-learning variations on back-propagation: An empirical study. In: Sejnowski, G.E., Hinton, G.E. & Touretzky, D.S. (eds), *Connectionist Models Summer School*. SanFrancisco, USA: Morgan Kaufmann.