Real-Time Defect Detection on Cloths

Antonio Baldassarre[a], Maurizio De Lucia[a], Paolo Nesi[b], Francesca Rossi[a], and Jacopo Zamberlan[a]

[a]Department of Energy Engineering (DEF), University of Florence, Italy
mdl@gedeone.de.unifi.it; http://gedeone.de.unifi.it
[b]Department of Systems and Informatics (DSI), University of Florence, Italy
nesi@ingfi1.ing.unifi.it, nesi@dsi.unifi.it, http://www.dsi.unifi.it/~nesi

ABSTRACT

The detection and classification of defects is strongly useful for stopping in real time the cloth production when degenerative defects occur; for increasing the efficiency of production by limiting the decrement of price for cloth rolls. The paper describes the work performed for detecting defect of well-known manufacturers of cloths and machine builders for cloths (looms). The main goal has been to obtain a new and innovative production line endowed with a system for detecting defects in real-time. The system is based on image processing techniques with a special attention to the real-time constraints. An architecture separating an on-line defect detection and an off-line classification has been proposed. An intelligent optical head, assembled on the loom, has the duty to acquire images and to detect the defects in real-time. A server has the offline task to classify each defect detected by the head. In the paper, some new algorithms for defect detection have been proposed. These have been compared with a selection of the most interesting algorithms for the same purposed taken from the literature. The comparison has been conducted by on the basis of a large test set with several types of defects and by considering reliability, performance, and complexity.

Keywords: Defect detection, real-time, vision architecture.

1. INTRODUCTION

The problem of detecting defects of cloths is very relevant for increasing the production quality[7-24]. The detection of defects is strongly useful for stopping in real time the cloth production when degenerative defects occur, for increasing the efficiency of production by limiting the decrement of price for cloth rolls. The set of defects that are present in a specific roll and their type influence the final price of the roll. Thus, the defects have also to be classified. In the industries, the quality classification on cloth rolls is performed off line; while only few visits per day are performed directly on cloth machines by strongly qualified personnel for detecting degenerative defects. To perform more frequent verifications is too expensive. Once a roll is produced, it has to be off-line analysed in order to classify the defects. The consumers of rolls for producing dresses need the roll profile for planning the production and the roll cuts. High quality productions produce cloth rolls of 75mt with only 8 defects. Some complex defects may be periodic and very long, along the roll. In such cases, the whole roll may be compromised, producing a high loss. These kinds of defects have to be detected in real-time because each minute of loom in producing an unusable cloth is a lost of gain. Detection of defects for identifying the presence of simple defects and degenerative defects has to be computationally light since the elaboration has to be performed in real-time on the looms. On the contrary, the defect classification must be performed more carefully and thus it is computationally expensive. These declarations will be demonstrated in the paper in terms of real experiments.

The paper describes the work performed with well-known manufacturers of cloths and machine builders for cloths (looms): Marzotto, Benninger and Nuovo Pignone. The main goal has been to obtain a new and innovative production line endowed with a system for detecting defects in real-time and thus for controlling the production process. The system is based on image processing techniques with a special attention to the real-time constraints[1-6].

The paper is organised as follows. In Section 2, an overview of the general architecture is presented. In Section 3, the defect detection framework is reported. In Section 4, the new algorithms are presented. Section 5 presents the comparison methodology. In Section 6, some experimental results are reported and discussed. Conclusions are drawn in Section 7.

2. ARCHITECTURAL OVERVIEW

The architecture defined is based on a system for real-time defect detection for each loom and a system for off-line defect classification shared by a set of looms. The classification is performed at a higher velocity with respect to that of humans,

thus reducing the time to delivering, reducing logistic problem to move produced rolls. Moreover, the automatic classification via computer is more reliable and repeatable than human classification, which may depend on the availability of qualified personnel with huge experience. The architecture presents a moving TV-camera mounted on each loom and a stand-alone image processing-board; this whole set-up is called in the following as IOH (Intelligent Optical Head). The camera moves on the loom transversally to the direction of cloth production. The TV-camera is directly endowed with a stand-alone image processing board, for grabbing images and for the fast defect detection. Degenerative defects are detected by the IOH and thus the loom is stopped and the specialised personnel for solving the detected problem are called. The TV-camera is synchronised with the loom in order to get all cloth segments as soon as the loom produce them, depending on the loom velocity. Once a defect is detected, the IOH sends the image with the defect to the server for classification via a fast Ethernet.
The IOH performs the defect detection for a loom since the frequency on which the defects occur is low (as previously stated). If the frequency becomes too high, above an identified threshold, the IOH is not capable to apply the defect detection algorithms on the whole cloth produced. Thus, IOH has to stop the loom to avoid the production of a very low quality roll.

The server satisfies the classification requests of a set of looms, because classification job is not time-critical as defect detection. The server builds the roll profile in terms of list of defects, their type and length.
After an analysis of production and quality control methodology used in a textile factory, the optimal architecture has been defined, for this kind of application having the target of a medium factory.
Main considerations on the described architecture are:

- difficulty to get the compromise between low cost and the high performance hardware needed by classification algorithms to perform them in real-time.
- nowadays textile factory production has very high quality and the probability of defect generation is very low. Statistical data shows a frequency of defects below ten defects for a roll of 75 meters.
- presently textile factories need to know the exact class of each defect to decide if it is severe error to stop the loom. From an analysis, it has been pointed out that a defect is considered serious if it is big and appears with a high frequency along the roll. For this reason, the request for real-time monitoring is only on defect detection maintaining statistical data on position and frequency of defects. For the automatic classification it is important to reduce the overall time needed for human control.

Benefits of the proposed architecture are: the costs for the classification unit are shared among several looms; the overall time for the detection and classification is drastically reduced respect to human operation.
The system bottleneck may be the communication link among the IOHs on looms and the classification unit. The connection can be dimensioned by knowing the quality level of the production. When defected images grow too much the connection can be saturated, but however in that conditions the loom has to be stopped, because it is sure tat serious defect is present.

The defect detection algorithms that have been implemented and tested are characterised by an image decomposition in rectangular windows and use this window as elementary analysis and decision units.

There are two approaches to define the information exchanged between the IOHs and the classification unit: transmission of the defected image window; transmission of features developed by defect detection algorithms. All the algorithms studied and developed are capable to produce for each window a numerical value proportional to the probability that the window contains a defect or a part of a defect. For each image a matrix of features is produced. When the IOH locates a defect, it can transmit only the feature matrix.

Starting from this architecture for defect detection, algorithms for defect detection have been studied and implemented directly on the IOH, while automatic-classification algorithm are under development for the classification workstation which works for more loom. The classification problem is not in the aim of this paper.

3. DEFECT DETECTION FRAMEWORK

The work has been mainly addressed to the definition of fast and reliable defect detection algorithms. In order to identify suitable algorithms for defect detection several well-known algorithms[7] taken from the literature have been implemented and compared, from the point of view of reliability and complexity, among them a modified Golden Template[8,9]. These algorithms are based on spatial local operators. Special implementations have been tested in order to verify their performance as a function of their parameters -- for instance, the dimension of the local area.

Several algorithms have been tested, some have been extracted from the literature other have been specifically defined. The most significant results have been obtained with: *GoldTempAlg*[8]; *MeanAlg*; *Global TextileImaging*; *Local TextileImaging*. The first one comes from algorithms already existent in literature. The others are new algorithms defined by the authors: (a) a method based on the local mean and the standard deviation of the local image brightness (called *MeanAlg*); and (b) a method based on segmentation and labelling techniques (called *TextileImaging*, in two versions). In *MeanAlg* the difference between near values of mean and deviation is considered as a measure of discontinuities and thus of defects. In *TextileImaging* algorithm, the colour information contained in the image is reduced from 256 grey levels to 16, and then segmentation is performed.

Texture classification is generally used in Computer Vision for the segmenting phase. It can also be useful for defect detection. The defect detection can be viewed as a classification problem of defect with respect to the original non-defected texture.
In general, defect detection algorithms taken from literature or which are presented in this work, follow this scheme:

1. each image is divided in a grid of rectangular windows (subimages), W_i, adjacent or partial overlapped, with predefined dimensions;
2. from each window, W_i, a features vector, f_i, (K dimensioned); each component of the vector is a measure of the texture in the window W_i. This phase is named in the following as *feature extraction*;
3. classification algorithms establish if each vector f_i (related to a window) contains or not a defect or a part of it.

This scheme is necessary to relate study of several studied methodologies. Some of algorithms studied in these works benefit from using some pre-elaboration algorithms.

In order to classify features as corresponding to defected and non-defected pieces of cloth, several methods can be used. The system developed uses a method based on a distance Euclidean weighted by standard deviation. The implemented method needs of a preliminary training phase. During the training, several vectors f_i related to non-defected windows are evaluated and the relative mean value of the feature vector, μ_f, and its standard deviation, σ_f, are calculated:

$$\underline{\mu}_{\mathrm{f}} = \sum_{i=1}^{H} \frac{\underline{f}_i}{H}$$

$$\underline{\sigma}_f^{\,2} : \underline{\sigma}_f^{\,2}[j] = \sum_{i=1}^{H} (f_i[j] - \mu_f[j])^2 / H$$

where: H is the number of non-defected windows used as training samples. After the training, for each feature vector that has to be classified, the follow distance is evaluated:

$$d : d^2 = \frac{1}{K} \sum_{j=1}^{K} (f[j] - \mu_f[j])^2 / \sigma_f[j] \tag{1}$$

A vector $\underline{f}$, and the corresponding window are considered defected if $d > th$, where th is a threshold value (typically 3.5-4).
All the developed algorithms extract a N-dimensional vector of features, f, from each window. This vector can be identified as describing a non-defected or defected window by estimating the distance d. The decision is taken if the distance is greater than a threshold value the window is considered with a defect. The vectors can be classified with other methods than the estimation of distance d. After the distance has been estimated, it is possible to have another processing session analysing the d distance map, obtaining a final decision on the window defect and also a possible classification. Several alternative methods exist in literature especially based on Neural Network or mixed method

For defect detection the above algorithms use a method based on the distance Euclidean weighted by standard deviation (as it has been shown above), with the exception of the global form of *TextileImagingAlg.*

The following parameters are used in all the above mentioned algorithms: decision threshold, *DX* and *DY* window dimensions, distance between windows called *StepRatio* (*DX StepRatio* pixels is the horizontal distance and *DY StepRatio* pixels is the vertical distance between two adjacent windows). The threshold parameter is very important, because the algorithm indicates a defect on the basis of the value of this parameter. By choosing certain values for the *StepRatio* parameter, it is possible to overlap some windows, thus, optimising the defect detection. For example, if there were two adjacent windows, with a little and narrow defect between them, the single window area occupied by the defect could be too small, and the algorithm could not detect that flaw. On the other hand, if the two windows are overlapped, even if the defect is very narrow, it could be present in both windows; thus, increasing the probability of detecting it. The overlapping of windows has some disadvantages, especially regarding the processing time.

The algorithms taken from the literature that have been used in the comparison are briefly reported:

GoldTempAlg: The Golden Template Algorithm is widely used in literature on several applications of defect detection in industrial productions, especially for silicon wafer in integrated circuit production[8]. This technique is based on using a sample without defect (the gold template): each acquired image is analysed by estimating the difference with respect the sample image. The image obtained for difference has some peak where defects or changes are present. Problems of this technique are due to line up images.

Golden Template Modified method. It should be noted that it is impossible to directly use the Golden Template Algorithm for the defect detection of cloth. In fact, a cloth has a regular and periodic structure. It has an elasticity that may change the typical periodic structure. For this reason, the recording of a reference image to compare the entire cloth image or roll (in order to decide if there are defects), is not a good approach. For this reason, it has been necessary to modify this well-known algorithm to apply it in textile applications. The modified algorithm is comprised of the following steps.

The image is divided in windows whose dimensions are equal or multiple of the textile period, as shown in Fig.1.

W1	W2	W3
W4	W5	W6
W9	W8	W7

Figure 1: Image division in windows.

The point by point distance from two windows is:

$$d(i,j) = dist(Wi,Wj) = \sum_{x}\sum_{y} |(Wi(x,y) - Wj(x,y))| \,/\, NP$$

where: *NP* is the total number pixels of each window, and the sum is performed on all window points. Each window W_i is subtracted to one or more adjacent windows. The firsts to be considered are the upper and that on the left with respect to the considered window: for example for *W6*, *d1 = d(W5, W6)* and *d2 = (W3, W6)* are considered. For each W_i a vector $f_i=[d1, d2]$ is assigned. The vectors are classified with the method of the Euclidean weighted distance by standard deviation. $f_i=[d1, d2]$ is not the only parameter that can be extracted from the windows. In fact other parameters were considered, such as *fi = max (d1, d2); fi =min (d1, d2); fi = (d1+d2)/2*, but the results obtained with these were less satisfactory than that obtained with $f_i=[d1, d2]$.

When windows on the left border of the image are studied, the adjacent windows considered are the ones at their right. In the same way for the windows in upper border of the image are considered the adjacent window at the lower size. In the corners the last two windows connected with the boundaries are considered.

4. NEW ALGORITHMS

In this paper, two new algorithms have been proposed: *MeanAlg* and *Textile ImagingAlg*.

MeanAlg: mean value and standard deviation of grey level distribution. This algorithm is based on the subdivision of the image in windows of *DX* x *DY* pixels, and on the estimation of some image pixel grey level distribution parameters. If *I(x, y)* is the gay level of the pixel in *(x, y)* and *NP=DX DY* is the number of window pixels, the algorithm main parameters are:

$$\text{Mean Value} \qquad \mathrm{M} = \sum_{x=1}^{DX}\sum_{y=1}^{DY} I(x,y)/NP$$

$$\text{(Pseudo) Standard Deviation} \qquad \mathrm{D} = \sum_{x=1}^{DX}\sum_{y=1}^{DY} \left| I(x,y) - M \right|$$

The pseudo standard deviation is not a sum of square (but a sum of absolute values). This has been used instead of the classical standard deviation in order to reduce the processing time. This algorithm extracts from each window a feature vector *f = [M, D]*. Therefore, the windows can be classified as defected or not on the basis of the distance Euclidean weighted by standard deviation, as above described. The mean value highlights the areas in which the image is brighter or darker than the standard image. Thus it is capable for detecting changes. This is also a good approach for detecting defects such as holes or nodes. The standard deviation value gives useful information about the image contrast.

Textile Imaging Alg. This algorithm is based on detecting and labelling connected regions of adjacent pixels having the same grey level in the image. As a first step the number of the possible grey level is strongly reduced, for instance, from 256 to 16. In Fig.2, it is shown a sample image, in which the regions detected with this method are numbered from 1 to 5.

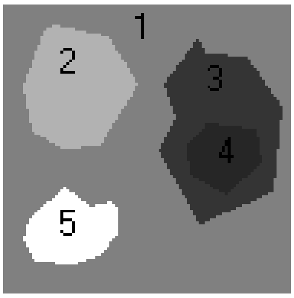


Figure 2 Detection of adjacent pixels regions with the same grey level.

Applying this algorithm to cloth images several very small regions are extracted for the presence of the texture. In the case of a defect the corresponding region with the same grey level results to be greater than those without defect (the dimensions of these good regions depend on the textile basic design).
The algorithm steps are:

- detecting and labelling the regions with the same grey level inside an image;
- measuring the effective area (number of pixels) of the each region;
- extracting f_i parameter, defined as the maximum area of all the image regions;
- if the area is greater than a threshold value (estimated in a training part) the image contains a defect (i.e., it has at least one defect).

From experimental work, it has been observed that the algorithm presents better performance if it is applied on an image presenting only 16 different grey levels, thus with only 4 bits per pixel (instead of using the image with 256 grey levels). For this reason, before applying this algorithm the image is equalised.

The above algorithm is called *Global TextileImagingAlg*, since the whole image is scanned. Also a local form for this algorithm was developed, dividing an image in rectangular windows. In this case, the steps are:

- scanning the whole image by window;
- detecting and labelling the regions inside the *i*-th W_i window;
- measuring the area (number of pixels) of each region;
- extracting the f_i parameter, defined as the maximum area of all the region inside the W_i window;
- if the above value is greater than a defined threshold (estimated in a training work), then the window present a defect (i.e., having at least one defect). This method uses the Euclidean distance, weighted by standard deviation: the properties vector is mono-dimensional and its element is the maximum area among the areas of the regions inside the window under study.

Computational Complexity: In order to compare the algorithms, it is very important to give a measure of the computational complexity. Thus, by the analysis of the algorithms, the following relationships have been identified for the detailed complexity, where *N* x *M* (image dimension) and *DX* x *DY* (windows dimension).

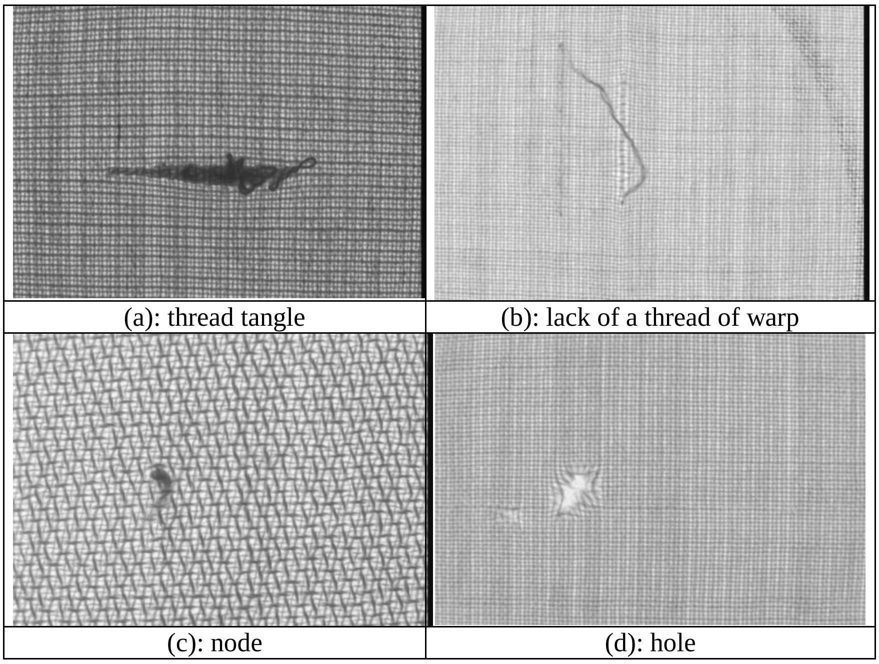
$$C(MeanAlg) = \left(\frac{N}{DX}\frac{M}{DY}\right)2\,DX\,DY = 2N\,M = O(NM)$$
$$C(GoldTempAlg) = \left(\frac{N}{DX}\frac{M}{DY}\right)3\,DX\,DY = 3N\,M = O(NM)$$
$$C(TextileImagingAlg\_Locale) = \left(\frac{N}{DX}\frac{M}{DY}\right)3\,DX\,DY = 3N\,M = O(NM)$$
$$C(TextileImagingAlg\_Globale) = 3\,M\;N = O(NM)$$

From the above list of complexities it results evident that all the algorithms present the same asymptotical complexity, an *O(NM).* They are characterised by a different scale factor. The computationally cheapest algorithm results to be the *MeanAlg*. Please note that the number of windows in which the image is divided is *NW:*

$$NW = \lceil (N/DX)(M/DY) \rceil$$

5. ALGORITHM COMPARISON METHODOLOGY

In this section, the methodology that has been defined and used for comparing the above-mentioned algorithms is presented. To this end, a very large database of defects has been created. Typically, the defect influences large areas with respect to the very small areas produced from the texture. In the database, several real images with and without defects have been collected. Typical defects that have been considered are: (i) lack of a thread of warp, (ii) the lack of a thread of weft, (iii) the presence of a hole, (iv) the presence of an external thread, (v) the presence of a node, etc.. In Fig.3 some defects are reported.



(a): thread tangle
(b): lack of a thread of warp
(c): node
(d): hole

Figure 3: Textile defects on different kind of cloths.

The above-discussed algorithms typically depend on more than one parameter. For instance, a threshold, the dimension of the local area in which the algorithm is applied, etc. In order to get the optimal conditions the trend of reliability as a function of algorithm parameters has been studied. The values of the algorithm parameters depend on the cloth under study. For example, if an algorithm depends on *Q* parameters, testing the algorithm means to find the *Q* parameter values (or their range) optimising the algorithm behaviour for defect detection. Therefore, it is very important to define a testing method to compare a single algorithm performance for different parameter values and thus for different algorithm performances.

According to an algorithm testing, it is possible to state that the algorithm is "good" if it detects a great number of defects with a low number of false alarms. To evaluate in which extend an algorithm is good, its behaviour with respect to the test cases has been expressed by means of a specific high-level value. In order to estimate the algorithm reliability the real defects present on an image have to be known. The information on the existence and position of the defects have been inserted from the user and saved in a "Defect Mask". This is a binary image having the same dimensions of the image under studying and in which pixel with a value of 1 mark the defected area, 0 otherwise.

The high-level value is called in the following "Performance Index" and indicated as P. P is defined with the following conditions: it increases with the true positives (correct defect detections), and decreases with the false alarms (not correct defect detection). The P value is extracted on the basis of some parameters giving information on the true positives and false alarms.

In order to define these parameters, the windows in which each image is divided are considered. For each window it is known if it contains a defect or a significant portion of it, on the basis of the "Defect Mask". Two possible states are thus associated with each window: a positive state (if the window contains a defect or part of it); a negative state (if the window doesn't contain a defect). When an algorithm analyses a window, it can give only two possible decisions: the algorithm may consider the window as defected or not. Combining the two real states of the windows and the two possible results of the algorithm analysis, there are 4 different possibilities:

1. Defected window classified as defected: True Positive (TP)
2. Defected window classified as non defected: False Negative (FN)
3. Non defected window classified as defected: False Positive (FP)
4. Non defected window classified as non defected: True Negative (TN)

Considering the total number of TP, FN, FP, TN, it is possible to say that:

a) TP+FN = number of real defected windows;
b) FP+TN = number of real non defected windows;
c) TP+FN+FP+TN = total number of windows.

Once these parameters are evaluated, the algorithm performance is classified in four different cases, described in the following table:

Class	Qualitative valuation	Performance
1	Optimum	No false alarms; detection of the defect
2	Medium	Detection of defects; false alarms
2	Medium	No false alarms; no defect detection
3	Worst	False alarms; no defect detection

Table 1: Definition of algorithm classes on the basis of their reliability

The *medium* cases are grouped in the same class, because an algorithm that is not capable of detecting defects has a unsuitable behaviour such as that gives false alarms (the last one cannot identify a real defect from a false alarm). The definition of the "Performance Index" directly gives the algorithm class. In order to extract only one P value for an entire test set, the classification index mean on each test image is then considered. The "Performance Index" is thus defined as:

$$P = \sum_i \frac{C_i}{N}$$

where C_i is the classification (1, 2 o 3) of the i-th image, and N is the number of images in a test set. A P value close to one means that the algorithm is "good" -- i.e., it belongs to the first class and detect all the defects without false alarms.

6. EXPERIMENTAL RESULTS

The algorithms were tested in two successive sessions. In a preliminary session, four "Test Sets" were used. These are images (defected and not) of four different kinds of cloths, grouped on the basis of some common characteristics. The test sets were used as a starting point for tuning algorithm parameters: changing the fundamental algorithm parameters (such as threshold, StepRatio, windows dimensions) the best P (Performance Index) value was found.
The results obtained were used in a second part of the experimental tests. The algorithms were studied on all cloths and all kind of defects in order to estimate the best values of the parameters characteristic of the algorithms, starting from the those obtained in the first experimental session. Also the processing times were tested. The results obtained were used to compare the different algorithm performances.

7.1 Algorithms studied on Test Set

In order to create the four test sets, the cloths and the defects were grouped in classes, having the same parameters.

For grouping the cloths in classes, three parameters were considered. These are:

1. Horizontal and vertical cloth period: to this parameter is directly connected the minimum defect dimension on the cloth.
2. Directionality: the most part of cloths has orthogonal threads; on some cloths there are also threads inclined with an angle different from 0° or 90°. This can influence the algorithm performance.
3. Regularity: with this concept the real periodicity of a cloth is indicated, or how the cloth is subject to deformations varying its phase or frequency.

The defects have been classified by considering the deformations on the cloths due to the defects; the causes that produced the defects were non considered. Hereafter, there are some defects, each of which can be considered representative of a class.

1. Hole: in the area corresponding to the defect, the cloth is absolutely absent. There is a sharp variation in the pixel brightness and in the cloth geometry;
2. Thread tangle: there is no possibility to indicate different threads; there is a sharp variation in the pixel brightness and in the cloth periodicity;
3. Broken warp: the elementary cloth jersey is deformed; it is similar to the cloth period and presents a localised and sharp increment; it is also possible to value a local variations of the frequency content of the image;
4. Continuous horizontal and vertical thin defect: there is a cloth frequency variation on a stripe traversing horizontally or vertically the whole image, but it is very narrow (in number of pixels);
5. External thread: a thread is coming out the cloth and is overlapped to it; this is generally due to another defect (as a broken warp), but is now considered as a proper defect;
6. Bad worked stitch: a thread is knitted not in a regular way;

The Test Sets used had the following parameters:

Test Set	Cloth Parameters	Defect n°
1	Orthogonal cloth; period = 0.83 x 0.41 mm (15x7 pixels)	2, 3, 4,
2	Orthogonal cloth; period = 0.39 x 0.41 mm (7x7 pixels)	1, 2, 5, 4, 6
3	Orthogonal cloth; period = 2.1 x 2.64 mm (52x45 pixels)	2, 4
4	Orthogonal cloth; period = 0.47 x 0.45 mm (8x8 pixels)	1, 5, 6

Table 2: Test set parameters

As it has been stated, each algorithm was studied on each test set, varying some parameters and searching the values that minimises the Performance Index. The images are classified on the basis of the defect, and a mean Index has been evaluated for each defect. The common parameters considered are the same described in §5.

									Performance per n° Defect Class						
Algorithm	***Test Set***	**Complexity (/ 10000)**	**Pre equalization**	**Best Performance Index**	**Windows Dimensions**	**StepRatio**	**Grey Level Number**	**Minimum threshold range**	**1**	**2**	**3**	**4**	**5**	**6**	**Good Cloth**
	1		YES	1	15x7	0.5		12	-	1	1	1	-	-	1
	2		YES	1.125	6x6	1		4	1	1	-	1.66	1	1	1
MeanAlg	3	7.76	YES	1	30x30	0.5	-	20	-	1	-	1	-	-	1
	4		YES	1	7x7	1		12	1	-	-	-	1	1	1
	Tot			1.042					1	1	1	1.22	1	1	1
	1		NO	1	15x7	1		>40	-	1	1	1	-	-	1
	2		YES	1.1176	6x6	1		>60	1	1	-	1.33	1	1	1
GoldTempAlg	3	11.64	YES	1	14x14	0.5	-	>40	-	1	-	1	-	-	1
	4		YES	1	8x8	1		>250	1	-	-	-	1	1	1
	Tot			1.039					1	1	1	1.11	1	1	1
	1		YES	1	15x7		20	30	-	1	1	1	-	-	1
TextileTrn	2		YES	1.059	6x6		20	>60	1	1	-	1	1	1.33	1
Imaging	3	-	YES	1	22x22	-	20	>60	-	1	-	1	-	-	1
(Locale)	4		YES	1	7x7		20	>110	1	-	-	-	1	1	1
	Tot			1.020			20		1	1	1	1	1	1.17	1
	1		YES	1.28			20	18	-	1	1	2	-	-	1
TextileTrn	2		YES	1.05			20	16	1.66	1	-	1	1	1	1
Imaging	3	-	YES	1	-	-	20	>100	-	1	-	1	-	-	1
(Globale)	4		YES	1			20	>300	1	-	-	-	1	1	1
	Tot			1.110			20		1.33	1	1	1.33	1	1	1

Table 3: Results of the experimental session on the Test Set

As it is possible to see from table 3, *MeanAlg* and *GoldTempAlg* have the best Performance Index if the window dimensions are equal to the cloth period; not always this means the maximum stability threshold range. An exception is the *GoldTempAlg* tested on Test Set 3, in which the window dimensions are 14x14 pixels, quite different from the period cloth (52x45 pixels). This is due to the fact that with the magnification factor used to record these images, the cloth seems to be irregular, and so *GoldTempAlg* (that is based on detecting the cloth regularity) does not work well.

From this experimental session, it is possible to say that the best global performance has been obtained with *local TextileImaging*. Its mean Performance Index on the four test sets has been equal to 1.02. *GoldTempAlg* and *TextileImaging* (local and global) have a good stability against the threshold parameter. *Global TextileImaging* has the great advantage that has only one important parameter (beside the threshold parameter): the grey level number. Therefore, the dimensions of the windows are not relevant, and the algorithm reliability results independent from the cloth (and the most simple to calibrate).

7.2 Algorithms studied on whole cloth and defects

In a second testing session, all the algorithms were tested on all cloths and defects. The cloths can be grouped in three classes, depending on the cloth framework: simple, complex and coloured framework.
Using the results of the study described in §7.1, the test was made in order to determine the algorithm with the Performance Index more close to 1 (and the value of its parameters), following these steps:

1. <u>Optimum Threshold Value:</u> varying the threshold value for each defect, the Performance Index has been estimated. The windows dimensions were fixed at the same value of the cloths period and its multiple: varying the windows dimensions in these values, the threshold for which the Index Parameter is more equal to 1.
2. <u>Optimum Windows Dimensions:</u> The dependency from the cloth period have been removed, because the estimation with good accuracy of this parameter is not always easy and in order to find an algorithm with good performances on all cloths. The algorithms were tested on all cloths using square windows having the same dimensions on different materials. The best dimension of the windows (i.e., the one for which the Performance Index is closer to 1) was then chosen.
3. <u>Algorithms Comparison:</u> once the best values for the threshold and the windows dimensions were found for each algorithm, it was possible to compare different algorithms in their best conditions.

The results obtained in this experimental session are described in the following. For what concern the coloured cloths, the Performance Index is the parameter used to test the algorithm characteristics. This has been possible since the studied defects were not on the colour information, but on the cloth framework. Therefore, it has been decided to work and record even for coloured cloth BW images.

GoldTempAlg: in this test adjacent windows and pre-equalised images were considered.

	Simple Framework	**Complex Framework**	**Coloured Cloths**
Threshold	100	40	150
Windows Dimension (DX*DY)	10x10	10x10	10x10
Performance Index	1.39	1.39	1.6

MeanAlg: the *StepRatio* parameter has been set to 1 (adjacent windows were not overlapped); also the mean and standard deviation parameters were used; the images were pre-equalised and the grey level number is 256.

	Simple Framework	**Complex Framework**	**Coloured Cloths**
Threshold	20	20	20
Windows Dimension (DX*DY)	16x16	10x10	30x30
Performance Index	1.35	1.36	1.5

Local TextileImaging: The *StepRatio* parameter was fixed to 1. It was then observed that reducing the grey level number there was an improvement in the algorithm performances: the images were then used with 20 and 16 grey levels, instead of 256.

	Simple Framework	**Complex Framework**	**Coloured Cloths**
Threshold	150	300	75
Windows Dimension (DX*DY)	24x24	30x30	20x20
Performance Index	1.23	1.32	1.5

Global TextileImaging: this algorithm analyses the whole image grey level, without dividing it in windows. The parameters *DX* and *DY* have been for this reason not considered.

	Simple Framework	**Complex Framework**	**Coloured Cloths**
Threshold		100	50
Grey Level Number	16	20	10
Performance Index	1.31	1.54	2.39

Processing time: The estimation of the elapsed CPU-time for each algorithm was made on a Pentium 133, with algorithms implemented in C++ and partially optimised measuring the time necessary for analysing a single image. For the simple framework cloths the processing time could be valued, for all algorithms except Global TextileImaging. From this study resulted that GoldTempAlg has the greater processing time, varying from 322 ms in case of large windows to 21 ms for decreasing windows dimensions. The MeanAlg has lower processing time, always decreasing for decreasing windows dimensions: the values are 157 ms for the bigger windows and 77 ms for the little ones. The Local TextileImaging processing times are comparable to those of MeanAlg, but in this case the minimum time (about 90 ms) is for intermediate dimensions of the window, not for the minimum windows dimensions. Hereafter, there is the table with the experimental results. The values are a mean value of the processing time needed to analyse each image.

	GoldTempAlg	MeanAlg	Local TextileImaging
Windows Dimensions (pixels)	6x6	7x7	24x24
Mean Processing Time (ms)	325.5	158.1	94.6
Windows Dimensions (pixels)	15x15	20x20	32x32
Mean Processing Time (ms)	246.9	90.6	88.0
Windows Dimensions (pixels)	45x45	40x40	40x40
Mean Processing Time (ms)	216.2	77.2	153.9

7.3 Algorithm comparison

From the experiments made, it is possible to say that:

- In the case in which the windows dimension is a fundamental parameter, the best performances are for small windows. .
- Local TextileImaging has good behaviour on all kind of framework cloths: the Performance Index is very close to 1, and the processing time is very short; its behaviour is better for simple cloth framework.
- Global TextileImaging has good performances, as the previous one from which it derive from. The processing time is even better, because the image is not divided in windows. On complex cloth framework the results are not as good.
- MeanAlg has good performances on simple framework cloths: the Performance Index is very close to 1, and the processing time is very short; using it on complex framework its performances are very bad.
- GoldTemp has the worst performances on simple framework cloths, both for Performance Index and processing time. On complex framework it has good results, especially on little windows.

7. CONCLUSIONS

The proposed algorithms (MeanAlg and TextileImaging) are quite better ranked for detecting defects of those selected from the literature. The computational complexity of the algorithms proposed are lower than those compared and thus the velocity is suitable for the application considered. For these reasons they have been selected for detecting defects in real time in the above mentioned architecture and pipeline of production.

8. ACKNOWLEDGE

The authors would like to thank the components of CAT consortiums and in particular the following industries: Marzotto, Benninger, Sulzer.

9. REFERENCES

1. P.Nesi, M.Trucco, "Real-Time Detection of Texture Defects", special issue, Journal on Real-Time Imaging, Academic-Press, London, UK, in press, 1999.
2. A.Del Bimbo, P.Nesi, J.L.C.Sanz, "Analysis of Optical Flow Constraints", IEEE Trans. on Image Processing, USA, Vol.4, N.4, pp.460-469, April 1995.
3. A.Del Bimbo, P.Nesi, J.L.C.Sanz, "Optical Flow Computation Using Extended Constraints", IEEE Trans. on Image Processing, Vol.5, pp.720-739, May 1996.
4. A.Baldassarre, M.De Lucia, P.Nesi, "Real-Time Detection of Cavitation for Hydraulic Turbomachines", Journal on Real-Time Imaging, Academic-Press, London, UK, Vol.4, N.4, 1998.
5. A.Del Bimbo, P.Nesi, "Blackboard-Based Concurrent Object Recognition Using an Object-Oriented Database", Proc.of the 11th IEEE International Phoenix Conference on Computers and Communications, IPCCC'92, IEEE Press, Scottsdale, Arizona, pp.172-180, USA, 1-3 April 1992.
6. P.Nesi, F.Innocenti, P.Pezzati, "RETIMAC: Real-TIme Motion Analysis Chip", IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, IEEE press, USA, Vol. 45, N. 3, pp.361-375, March 1998.
7. M. Borghesi, V. Cantoni, M. Damiani, "An Industrial Application of Texture Analysis", in Proc. 7th ICPR, Montreal, Canada, pp.420-423, 1984.
8. P. B. Chou, "Automatic Defect Classification for Semiconductor manufacturing", Machine vision and Applications, Vol.9, pp.201-214, 1997.
9. D. Dunn, "Texture Segmentation Using 2-D Gabor Elementary Function", IEEE Trans. on Patt. Anal. And Machine Intell., Vol.16, n.2, 1994.
10. Choen S, Fan Z, Attali S, "Automated Inspection of Textile Fabrics Using Textural Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.13 NO.8, August 1991, pp. 803-808
11. Conners RW, Harlow CA, "A Theoretical Comparison of Texture Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2 NO.3, May 1980, pp. 204-222
12. Davis LS, Johns SA, Aggarwal JK, "Texture Analysis Using Generalized Cooccurences Matrices", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1 NO.3, July 1979, pp. 251-259
13. Dom BE, Brecher V, "Recent advances in the automatic inspection of integrated circuits for patterns defects", Machine Vision and Applications, 8, 1995, pp. 5-19
14. Ferretti M, "Analisi di trame e tessiture" in Cantoni V, Levialdi S, "La visione delle macchine", Tecniche Nuove, 1985 , pp.71 e segg.
15. Jain AK, "Fundamentals of Digital Images Processing", Prentice Hall International, 1985

16. Jain AK, Karu K, "Learning texture discrimination masks", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.18 NO.2, February 1996, pp. 195-205
17. Kashyap RL, Chellappa R, "Estimation and Choice of Neighbours in Spatial-Interaction Models of Images", IEEE Transactions on Information Theory, VOL. IT-29 NO.1, January 1983
18. Laws KI, "Texture energy measures", Proc. Image Understanding Workshop, 1979, pp. 47-51
19. Laws KI, "Textured Image Segmentation", USCIPI Rep.940, Image Processing Institute University of Southern California, 1980
20. Manjunath BS, Ma WY, "Texture features for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.18 NO.8, August 1996, pp. 195-205
21. Perner P, "A knowledge based image inspection system for automatic defect recognition, classification, and process diagnosis", Machine Vision and Applications, 7, 1994, pp. 135-147
22. Pietikäinen M, Ojala T, "Texture Analysis in Industrial Applications", Image Technology, 1996, pp.337-359
23. Rajasekaran S, "Training Free Counter Propagation Neural Network for Pattern Recognition of Fabric Defects", Textile Research Journal, 67(6), June 1997, pp. 401-405
24. Zhang YF, Bresee RR, "Fabric Defect Detection and Classification Using Image Analysis", Textile Research Journal, 65(1), January 1995, pp. 1-9