2-D ROBUST RECURSIVE LEAST SQUARES LATTICE ALGORITHM FOR DEFECT INSPECTION OF TEXTILE PRODUCTS*

Ruşen Meylani[1], Ayşın Ertüzün[1], Aytül Erçil [2]
[1]Department of Electrical and Electronic Engineering, [2]Department of Industrial Engineering
Boğaziçi University Bebek, İstanbul, 80815,Turkey
ertuz@ boun.edu.tr, ercil@boun.edu.tr

ABSTRACT

In this paper, a 2-D robust recursive least squares lattice algorithm is introduced and is applied to defect detection problem in textured images. The algorithm combines concepts of 1-D robust regression with the recursive least squares lattice algorithm. The philosophy of using different optimization functions that results in weighted least-squares solutions in the theory of 1-D robust regression is extended to 2-D. With this approach, whatever probability distribution of the estimation error may be, small weights are assigned to the outliers in that distribution so that the least squares algorithm will be less sensitive to the outliers. The results obtained are compared with those of conventional recursive least squares lattice algorithm. The performance evaluation, in terms of defect detection rate, demonstrates the importance of the proposed algorithm in reducing the effect of the outliers that generally correspond to false alarms in classification of textures as defective or nondefective.

1. INTRODUCTION

The field of multidimensional digital signal processing has become increasingly important in recent years due to number of trends in digital signal processing. The need for adaptive algorithms in 2-D lattice filtering problems arises in many different fields and they are mostly useful when the knowledge about the input data is limited. There have been a number of studies on adaptive lattice filters. Moro *et.al* [1] have proposed a gradient-type adaptive lattice algorithm for a six-parameter lattice filter structure. Youlal *et.al.* [2] have developed a 2-D adaptive lattice least mean square (LMS) algorithm to update the lattice parameters and then further developed the normalized version of this algorithm in order to maintain the same adaptive time constant and the same misadjustment at each stage. They have used the basic three-parameter lattice filter structure of Parker and Kayran [3] as 2-D lattice structure for adaptive image restoration and noise removal. Meylani *et.al.*[4] have applied the LMS and the gradient based adaptation algorithms on the eight-parameter lattice structure developed in [5]. Ffrench *et.al.*[6] have developed a recursive least squares lattice (RLSL) type adaptive twelve-parameter 2-D lattice filter and have shown that RLSL algorithm provides the exact least squares solution for a single stage lattice filter.

This paper develops a robust extension of the RLSL algorithm, namely the robust recursive least squares lattice (RRLSL) algorithm, to reduce the effects of outliers and demonstrates the performance of this algorithm for the detection of textural defects. The algorithm is developed for the twelve-parameter 2-D lattice filter structure which is the most general structure in the sense that no spectral symmetry assumptions are imposed on the input data. However with small modifications, this algorithm can easily be applied to various 2-D lattice structures[7].

Quality is a topical issue in manufacturing. The automation and the integration of quality control clearly have vital implications for industry. Quality control is designed to ensure that defective products are not allowed to reach the customer. For this reason, quality control activities form an essential information feedback loop for the whole business, with potential influence on the design, process planning and logistics functions as well as on manufacture. Visual inspection constitutes an important part of quality control in industry. Until recent years, this job has been heavily relied upon human inspectors. Development of fast and specialized equipment, however, has facilitated the application of image processing algorithms to real-world industrial inspection problems.

Since in many areas the quality of a surface is best characterized by its “texture”, texture analysis plays an important role in the automated visual inspection of surfaces. There have been a number of applications of texture processing to inspection problems. Majority of texture defect detection applications is on textile, paper, steel and wood inspection. Some of these are as follows: Erçil and Özüyılmaz [8] have proposed a model-based technique to detect and locate the various kinds of defects that might be present in a given painted surface. Jain *et. al.* [9] have used the texture features computed from a bank of Gabor filters to automatically classify the uniformity of painted metallic surfaces. Chen and Jain [10] have used a structural approach to defect detection

* This work is partially supported by Turkish Technology Development Fund with project number TTGV 169.

in textured images. Conners [11] has utilized texture analysis methods to detect defects in lumber wood automatically. Siew *et.al.* [12] have proposed a method for the assessment of carpet wear. Dewaele *et.al.* [13] have employed signal processing methods to detect point and line defects in texture images. Meylani *et.al* [7,14-15] have applied various 2-D lattice filter structures to perform either supervised or unsupervised defect detection on a defective image. Successful results are reported [7].

The supervised defect detection schemes employ model-based methods and they require processing with nondefective and defective images simultaneously. It is shown that the 2-D lattice filters can be successfully used in the context of supervised approach [7]. The lattice filter performs prediction error filtering on the 2-D input data producing reflection coefficients that may be used to estimate the autoregressive (AR) model parameters using the *Levinson-Durbin recursion* assuming that the data can be modeled as an AR process [3]. Since the reflection coefficients can be used to estimate the AR model parameters, they can be used as model parameters, instead, to decrease the computational complexity. That is the main reason behind considering the lattice filter as a model-based method [7].

In this work, a supervised defect detection scheme that employs twelve-parameter 2-D lattice filters is elaborated. The reflection coefficients of the lattice filters are calculated adaptively using the proposed RRLSL algorithm and the results are compared with those obtained by the RLSL algorithm. Satisfactory results, in terms of defect detection ratio, are obtained with the RRLSL algorithm. The proposed algorithm has reduced the false alarm rate, considerably.

2. 2-D LATTICE FILTERS

2-D lattice filter structures consist of concatenated multi-input/multi-output stages that are defined in terms of the reflection coefficients [1-7,14-15]. The inputs and the outputs are the forward and the backward prediction error fields that are generated simultaneously The twelve-parameter lattice filter is the most general structure of the quarter plane filters where no assumptions on spectral symmetry conditions of the input data have been made. Thus each quarter plane filter has to be designed independently [1,6-7]. The input-output relation for the twelve-parameter lattice filter is given as a linear combination of input prediction error fields as follows [1,6-7]:

$$\begin{bmatrix} e_{00}^{(n)}(i,j) \\ e_{10}^{(n)}(i,j) \\ e_{11}^{(n)}(i,j) \\ e_{01}^{(n)}(i,j) \end{bmatrix} = \begin{bmatrix} 1 & -k_1^{(n)} & -k_2^{(n)} & -k_3^{(n)} \\ -k_4^{(n)} & 1 & -k_5^{(n)} & -k_6^{(n)} \\ -k_7^{(n)} & -k_8^{(n)} & 1 & -k_9^{(n)} \\ -k_{10}^{(n)} & -k_{11}^{(n)} & -k_{12}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} e_{00}^{(n-1)}(i,j) \\ e_{10}^{(n-1)}(i-1,j) \\ e_{11}^{(n-1)}(i-1,j-1) \\ e_{01}^{(n-1)}(i,j-1) \end{bmatrix} \quad (1)$$

The vectors on the right and left hand side of Eq. (1) consist of prediction error fields at the input and output of stage (n), respectively. The 4 x 4 matrix consisting of twelve reflection coefficients associated with stage (n). Optimization of the least squares error given as

$$Q^{(n)}(m_1, m_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{m_2} [\mathbf{e}^{(n)}(i,j)^T \mathbf{e}^{(n)}(i,j)] \quad (2)$$

leads to the following four sets of normal equations, one for each quadrant filter [1,6-7]:

$$\mathbf{R}_m^{(n-1)} \mathbf{k}_m^{(n)} = \mathbf{r}_m^{(n-1)} \quad (3)$$

Here $\mathbf{R}_\mathbf{m}^{(\mathbf{n-1})}$ is a 3 x 3 symmetric autocorrelation matrix of stage (n-1), $\mathbf{k}_\mathbf{m}^{(\mathbf{n})}$ is the 3 x 1 reflection coefficient vector of stage (n) corresponding to the m-th quadrant filter and $\mathbf{r}_\mathbf{m}^{(\mathbf{n-1})}$ is the 3 x 1 crosscorrelation vector of stage (n-1). The elements of $\mathbf{R}_\mathbf{m}^{(\mathbf{n-1})}$ and $\mathbf{r}_\mathbf{m}^{(\mathbf{n-1})}$ are the auto- and cross-correlation values between the error fields given as:

$$\Phi_{e_{kl}e_{pq}}^{(n)} = \sum_{i=k}^{N} \sum_{j=l}^{N} [e_{kl}^{(n)}(i-k, j-l) e_{pq}^{(n)}(i-p, j-q)]$$
$$(k, l, p, q = 0,1) \quad (4)$$

The method of least squares (LS) estimates the unknown parameters directly using Eq. (3) or recursively using RLSL algorithm [6]. The LS estimator, whether calculates the unknown parameters directly or recursively, is known to be unreliable when the observations contain outliers and/or when there is collinearity between the independent variables [16]. The outliers may be present as a result of nonnormal errors. Robust estimation provides methods to detect outliers and reduce their effect.

3. ROBUST RECURSIVE LEAST SQUARES LATTICE ALGORITHM

RRLSL algorithm [7] is a novel approach that extends the idea of using weights in an iterative manner from the 1-theory of *robust ridge regression* [16] to 2-D. The goal in the RRLSL algorithm, is to minimize an objective function of the following form:

$$J^{(n)} = \sum_{i,j} \rho\, (\mathbf{e}^{(n)}(i,j)^T \mathbf{e}^{(n)}(i,j)) \quad (5)$$

where ρ is an appropriately chosen function. This performance index is used to reduce the effect of outliers when the error distribution is not close to the normal distribution. Different types of ρ functions can be used to reduce the effects of outliers[7].

In the RRLSL algorithm, it is desired to calculate the correlation values recursively, in other words the correlation at each pixel (i,j) is calculated based on previous pixels (i-1,j) and (i,j-1). If an image is processed by scanning it in the horizontal direction, this can be accomplished by defining a sum of vertical correlation components and a recursive horizontal sum of these summed vertical correlation values. The vertical sum, $\varphi^{(n)}_{e_{kl}e_{pq}}(i,m_2)$, can be updated recursively by [7]

$$\varphi^{(n)}_{e_{kl}e_{pq}}(i,m_2) = \lambda\,\varphi^{(n)}_{e_{kl}e_{pq}}(i-1,m_2) + e^{(n)}_{lk}(i-k,m_2-l)\,w(s)\,e^{(n)}_{pq}(i-p,m_2-q) \tag{6}$$

where w(s) is a weight term. s stands for the value of the forward prediction error field (when only forward optimization is done) at the current pixel position (i,j). w(s) is a weight function that is designed to make sure that smaller weights are given to outliers. For any given objective function ρ, there corresponds a weight function w(s). For each w(s), there is a corresponding objective function ρ(s), which gives an idea on the general behavior of the weight function in comparison to the mean-squared error. The weight function that corresponds to the squared error is constant 1. Introducing the forgetting term, λ, which is a constant in the interval (0,1), allows the algorithm to converge to new image statistics or new image features for nonstationary data. The autocorrelation and crosscorrelation values at pixel location (i,j), namely, $\Phi^{(n)}_{e_{kl}e_{pq}}(i,j)$'s, are recursively calculated as [7]:

$$\Phi^{(n)}_{e_{kl}e_{pq}}(i,j) = \lambda\,\Phi^{(n)}_{e_{kl}e_{pq}}(i,j-1) + \varphi^{(n)}_{e_{kl}e_{pq}}(i,j) \tag{7}$$

The true correlations are *totally independent* of the scanning scheme used. In this algorithm [7], the correlation values are calculated recursively and since the sizes of the autocorrelation matrices are small, their inverses are taken directly, like in [3].

The RRLSL algorithm is iterative in the following manner [7]:

(a) Within each stage, the reflection coefficients are calculated using no weights. In other words, the elements of autocorrelation matrix and the crosscorrelation vector are calculated setting w(s) =1 in Eq. (6) and using Eq. (7). The normal equations given by Eq. (3) are solved for the reflection coefficients.

(b) For the same stage, the output prediction error fields are calculated using the input prediction error fields and the reflection coefficients calculated in step (i) using Eq (1).

(c) Then a distance measure is defined in terms of the forward prediction errors if the lattice filter is optimized in the forward direction:

$$d = \text{mean}\,|e^{(n)}_{00}(i,j) - \text{mean}(e^{(n)}_{00}(i,j))| \tag{8a}$$

$$s = e^{(n)}_{00}(i,j)/d \tag{8b}$$

(d) The weights are employed and the weighted correlations are calculated using Eqs. (6) and (7). The weight function w(s) is evaluated using the value of s as defined in Eq. (8b). The autocorrelation matrix and the crosscorrelation vector in the normal equations (Eq. (3)) are, now, formed by these weighted correlations and the reflection coefficients are recalculated using the weighted correlations.

(e) The steps (b)-(d) are repeated until there is no change in the reflection coefficients, or a predetermined number of iterations are performed to assure convergence.

(f) When convergence within one stage is achieved, the stage number is updated and steps (a)-(e) are performed for the new stage.

4. APPLICATION TO DEFECT DETECTION PROBLEM

RRLSL algorithm is applied to the defect detection problem in textured images to alleviate the undesirable effects of outliers.

Texture defect detection can be defined as the process of determining the location and/or extend of a collection of pixels in a textured image with remarkable deviation in their intensity values or spatial arrangement with respect to the background texture.

The defect detection system used in the experiments consists of two stages:

(i) The feature extraction part utilizes prediction error filtering of the textured images and calculates the reflection coefficients of the twelve-parameter lattice filter using the proposed algorithm.

(ii) The detection part is a mahalanobis distance classifier being trained by defect-free samples.

The algorithms for each are provided below:

(i) Feature Extraction: Each 256 x 256 image is subdivided into non-overlapping subwindows of size 32 x 32 and each subwindow is processed using the twelve-parameter lattice filter and the reflection coefficients are adaptively calculated using either the RLSL or the RRLS algorithms. Window size chosen, in scanning the images depends both on the resolution of the camera used for image acquisition and the textural properties of the fabrics as well as how localized the defects are. In the experiments, the highest performance is obtained by using non-overlapping subwindows of size 32 x 32 [7]. For each subwindow, the feature vector that consists of the reflection coefficients calculated in step (b) of the RRLSL algorithm is constructed. For this approach, the reflection coefficients of greatest significance are those of the first stage. For this reason, only the reflection coefficients of the first stage are used for the analysis and the feature vectors consist of the twelve reflection coefficients of the first stage.

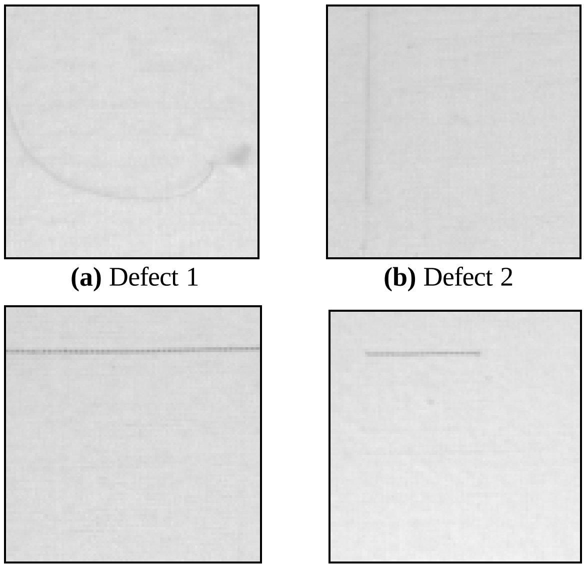
(ii)Detection: The detection part of the system consists of a learning phase and a classification phase: In the learning phase, *k* defect-free256 x 256 fabric images are used as the training images and the true feature vectors

for each subwindow are calculated using the feature extraction scheme given above. In the classification phase, the feature vectors of a test image of size 256 x 256 is calculated for each subwindow using the feature extraction scheme given above and the mahalanobis distance between each feature vector and the true feature vectors are calculated. Then each subwindow is classified as defective if the mahalonobis distance exceeds a threshold value or else it is identified as nondefective.

For the experimental justification of the algorithm, real fabric images acquired by a CCD camera in a laboratory environment are used [7]. The database consists of 256x256 sized 8-bit long gray level images. Front lighting has been used during the acquisition of the images, that is the camera and the light source are placed on the same side of the fabrics. Each of the acquired images corresponds to 8.53 cm x 8.53 cm fabric with the resolution of 3.33 pixels/mm, which is the same resolution required in the factory environment. Effort has been made to include various textures and different types of defects. Examples of defective images used in the experiments may be observed in Fig. 1.

In the experiments, the lattice filters are optimized in the forward field only and the RRLSL algorithm is employed using various weight functions w(s). The weight functions used are $w(s) = (1/s)\sin(s)$, $w(s) = (1+s^2)^{-1}$ and $w(s) = (1-s^2)^2$ and these correspond to the RRLSL algorithm type **a**, type **b** and type **c**, respectively. These are the weight functions associated with the objective functions $\rho(s)=1-\cos(s)$, $\rho(s)=s(1+s^2)^{-1}$ and $\rho(s)=s(1-s^2)^2$, respectively. The weight functions can be classified according to the behavior of the first derivative of the objective function. The **a** and **b** type weight functions are examples of hard redescenders whose first derivatives are zero for sufficiently large s. The **c** type weight function is a soft redescender and is asymptotic to zero for large |s|. The parameter λ used in Eqs. (6) and (7) is chosen to be 0.99 in the experiments.

The RRLSL algorithms give better results compared to the RLSL algorithm and among the RRLSL algorithms, the best performance is given by type **c**. The results obtained by these algorithms are presented in Table 1. The correctly labeled defective subwindows sum up to the number defined as PP (actually **present** and labeled as **present**). The number of false alarms sum up to the number AP (actually **absent** but labeled as **present**). The undetected defective subwindows sum up to PA (actually **present** but labeled as **absent**). This is the number of missed subwindows. And finally the number AA represents the number of correctly classified non-defective subwindows (actually **absent** and labeled as **absent**). The performance is evaluated in terms of the false alarm rate (the AP column). YES at the status column indicates that the defect is detected, and NO indicates otherwise. For comparison purposes, the detection ratio is calculated as



(a) Defect 1 **(b)** Defect 2

(c) Defect 3 **(d)** Defect 4

Figure 1 Examples of defective textile images.

the ratio of the truly identified defective and non-defective subwindows to the total number of subwindows, numerically being equal to (PP+AA)/(defective + nondefective). The experiments on the actual defective images reveal that the best performance among all the algorithms is given by RRLSL algorithm type **c** with all the defects being successfully detected and the least number of false alarms (see the AP and the status columns). Then come the RRLSL algorithms type **b**, type **a** and the RLSL algorithm.

5. CONCLUSIONS

In this work, a 2-D robust recursive least squares lattice algorithm is introduced to handle the adaptive defect detection problem in textured images. The algorithm is developed for the twelve-parameter 2-D lattice filter structure which is the most general structure in the sense that no spectral symmetry assumptions are imposed on the input data. However with small modifications, this algorithm can easily be applied to various 2-D lattice structures. Success of the algorithm is verified by computer examples employing images acquired from real textile products containing various defects. Satisfactory results, in terms of defect detection ratio, are obtained with the RRLSL algorithm. The proposed algorithm reduced the false alarm rate, considerably at the expense of increased computational complexity.

6. REFERENCES

[1] H. Moro, T. Watanabe, A. Taguchi and N. Hamada, “On the Adaptive Algorithm and its Convergence Rate Improvement of 2-D Lattice Filter”, *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 430-434, 1988.

[2] H. Youlal, M. Janati-I and M. Najim, “Two-Dimensional Joint Process Lattice for Adaptive Restoration of Images”, *IEEE Trans. Image Proc.* vol.1, pp.366-378, July 1992.

[3] S. R. Parker, and A.H.Kayran, “Lattice Parameter Autoregressive Modelling of Two-Dimensional Fields Part I: Quarter-plane Case” *IEEE Trans. Acoust., Speech and Sig. Proc.,* vol.. 32, pp. 872-885, Feb. 1984.

[4] R. Meylani, S. Sezen, A. Ertüzün, and Y. Istefanopulos, “LMS and Gradient Based Adaptation Algorithms for the Eight-Parameter Two-Dimensional Lattice Filter”, *Proc. European Conf. Circuit Theory and Design*, İstanbul, Turkey, Aug. 1995, pp.741-744,

[5] H.M. Bilgin, “Modeling of 2-D Fields with The Lattice Method”, *M.S. Thesis*, Istanbul Technical University, İstanbul, Turkey, 1988 (in Turkish).

[6] P. A. Ffrench, Z. A. Zeidler, and W. H. Ku, “Enhanced Detectability of Small Objects in Correlated Clutter Using an Improved 2-D Adaptive Lattice Algorithm”, *IEEE Trans. On Image Proc*., vol. 5, Dec. 1996.

[7] R. Meylani, “Texture Analysis Using Adaptive Two-Dimensional Lattice Filters”, *M.S. Thesis*, Boğaziçi University, İstanbul, Turkey, 1997.

[8] A. Erçil, and B. Özüyılmaz, “Automated Visual Inspection of Metalic Surfaces”, *The Proc. The Third International Conference on Automation, Robotics and Computer Vision (ICARCV'94),* Singapore, 1994, pp.1950-1954.

[9] A. K. Jain, F. Farrokhina and D. Alman, “Texture Analysis of Automotive Finishes", *Proc. SME Machine Vision Applications Conference*, 1990, pp.1-16,

[10] J. Chen, and A. K. Jain, “A Structural Approach to Identify Defects on Textural Images”, *Proc. IEEE International Conference on Systems, Man, and Cybernetics*,1988, pp. 29-32.

[11] R. Conners, “Identifying and Locating Surface Defects in Wood”, *IEEE Trans. Pattern Anal. and Mach. Intell***.** Vol.5. 1983.

[12] L. Siew, R. M. Hodgson and E. J. Wood, “Texture Measures for Carpet Wear Assessment”, *IEEE Trans. Pattern Anal. and Mach. Intell*. vol.10, pp. 92-105, 1988.

[13] P. Dewaele, L. Van Gool, A. Oosterlinchk, “Texture Inspection with Self-Adaptive Convolution Filters”, *Proc. 9th International Conference on Pattern Recognition*, 1988, pp. 14-17.

[14] R. Meylani, A. Ertüzün and A Erçil, “Texture Defect Detection Using the Adaptive Two-Dimensional Lattice Filter”, *Proc., IEEE Int. Conf. on Image Proc. (ICIP’96),* Laussanne, Switzerland, Sept.1996, pp. 165-168.

[15] R. Meylani, A. Ertüzün and A. Erçil, “A Comparative Study on the Adaptive Lattice Structures in the Context of Texture Defect Detection”, *Proc. The Third IEEE Int. Conf. On Electronics, Circuits and System (ICECS’96),* Rhodes, Greece, Oct. 1996, pp. 976-979.

[16] A. Erçil, “Robust Ridge Regression”, *Research Publication, General Motors Research Laboratories*, GMR-5349,1986.

TABLE 1.

Simulation Results

IMAGE	PP	AP	PA	AA	defective blocks	non-defective blocks	status	detection ratio
Recursive Least Squares Lattice Algorithm (RLSL)								
defect 1	1	3	11	49	12	52	YES	0.78
defect 2	0	2	8	54	8	56	NO	0.84
defect 3	8	4	0	52	8	56	YES	0.93
defect 4	4	2	0	58	4	60	YES	0.96
Robust Recursive Least Squares Lattice Algorithm Type a (RRLSL-Type a)								
defect 1	4	2	8	50	12	52	YES	0.84
defect 2	1	4	7	52	8	56	YES	0.82
defect 3	8	4	0	52	8	56	YES	0.93
defect 4	4	2	0	58	4	60	YES	0.96
Robust Recursive Least Squares Lattice Algorithm Type b (RRLSL-Type b)								
defect 1	2	1	10	51	12	52	YES	0.82
defect 2	0	2	8	54	8	56	NO	0.84
defect 3	7	0	1	56	8	56	YES	0.98
defect 4	4	6	0	60	4	60	YES	1.00
Robust Recursive Least Squares Lattice Algorithm Type c (RRLSL-Type c)								
defect 1	3	1	9	51	12	52	YES	0.84
defect 2	2	3	6	53	8	56	YES	0.85
defect 3	8	3	0	53	8	56	YES	0.95
defect 4	4	0	0	60	4	60	YES	1.00