CONVERGENCE BEHAVIOR OF TWO-DIMENSIONAL LEAST-SQUARES LATTICE ALGORITHM

Takayuki NAKACHI* , Katsumi YAMASHITA** and Nozomu HAMADA*

Faculty of Science and Technology, Keio University* Faculty of Engineering, University of the Ryukyus**

Yokohama-shi, 223 Japan Nishihara, Okinawa-ken, 903 Japan

Tel: +81-45-563-1141; Fax: +81-45-563-2773 Tel: +81-98-895-2221; Fax: +81-98-895-2688

e-mail: naka@tkhm.elec.keio.ac.jp e-mail: yamasita@lark.ie.u-ryukyu.ac.jp

ABSTRACT

In this paper, we propose a two - dimensional (2-D) least-squares lattice (LSL) algorithm for the general case of the autoregressive (AR) model with an asymmetric half-plane (AHP) coefficient support. The resulting LSL algorithm gives both order and space recursions for the 2-D deterministic normal equation. The size and shape of the coefficient support region of the proposed lattice filter can be chosen arbitrarily. Although the 2-D signals of the model support are ordered into a one-dimensional (1-D) array, the ordering of the support signal can be assigned arbitrarily. Finally, computer simulation for modeling a texture image is demonstrated to confirm the proposed model gives rapid convergence.

1 INTRODUCTION

In recent years, the development of two - dimensional (2-D) adaptive algorithm has rapidly expanded into a variety of fields such as communication, radar, image restoration and spectral estimation.

On the other hand, one-dimensional (1-D) adaptive lattice filter has been successfully used in a wide range signal processing applications, involving unknown or non-stationary data. Especially, 1-D least-squares lattice (LSL) algorithm based on least-squares concept provides rapid convergence not obtainable with conventional lattice and transversal filters based on statistical concept.

Motivated by the success of 1-D adaptive lattice filter, there have been some reserch effort directed to the development of 2-D adaptive lattice filters [1]-[5]. These studies show that 2-D adaptive lattice filter has rapid convergence property and is useful in many applications such as image restoration and noise cancellation. Most of these lattice filters are formulated based on statistical concept. The development of adaptive lattice algorithms based on least-squares scheme for 2-D problems has been much slower than their development for 1-D problems. Very few lattice filter model for least-squares algorithm have been discussed [2].

In this paper, we propose a 2-D LSL algorithm based on least-squares concept for the autoregressive (AR) model. The proposed LSL algorithm provides a least-squares solution for the 2-D deterministic normal equation. The size and shape of the coefficient support region of the proposed lattice filter can be chosen arbitrarily. This is the validity of the proposed LSL algorithm. Most of previous 2-D lattice filter reported in the references [1]-[4] have been proposed for a quarter-plane (QP) coefficient support. Although the 2-D signals of the model support are ordered into a one-dimensional (1-D) array, the ordering of the support signal can be assigned arbitrarily.

Finally, computer simulation for modeling a texture image is carried out to confirm the proposed method gives rapid convergence.

2 2-D LEAST SQUARES LATTICE (LSL) ALGORITHM

2.1 Some Preliminaries

For a stationary random field model, an asymmetric haif-plane (AHP) support linear prediction of a sample $y(i,j)$ is obtained by the linear combination of its support signals:

$$\boldsymbol{Q}_M(i,j) = \{y(i-s,j-t)|(s,t) \in \boldsymbol{S}_M\} \qquad (1)$$

where $\boldsymbol{S}_M$ is a 2-D index set which is a subset of the following AHP 2-D index set:

$$\hat{\boldsymbol{S}} = \{(s,t)|(s \geq 0, t > 0) \ or \ (s \geq 1, t \leq 0)\} \qquad (2)$$

The subscript M of $\boldsymbol{S}_M$ denotes the number of its element. In this case, a one-to-one mapping $\boldsymbol{I}$ between 2-D index set $\boldsymbol{S}_M$ and an integer set $\boldsymbol{M}$={1,2, $\cdots$,M} are defined.

$$\boldsymbol{I} : \boldsymbol{S}_M \mapsto \boldsymbol{M} \qquad (3)$$

For a specific $\boldsymbol{I}$, we introduce the notation

$$\begin{aligned} ((i,j)-p) &\stackrel{\triangle}{=} (i,j) - \boldsymbol{I}^{-1}\{p\} \\ &= (i,j)-(s,t) \\ &= (i-s,j-t) \\ (s,t) \in \boldsymbol{S}_M,&\ p = 1,2,\cdots,M \end{aligned} \qquad (4)$$

where $\boldsymbol{I}^{-1}$ is the inverse mapping of $\boldsymbol{I}$.

Using the notation $((i,j)-p)$, the forward and backward prediction errors with an AHP support are defined by

$$f^q_{((i,j)-p)} = [\ 1\ \ (\boldsymbol{a}^q_{p(m,n)})^T]\boldsymbol{Y}^q_{p(i,j)} \quad (5)$$

$$r^q_{((i,j)-p)} = [(\boldsymbol{b}^q_{p(m,n)})^T\ \ 1\ ]\boldsymbol{Y}^q_{p(i,j)} \quad (6)$$

$$q = 1, \cdots, M,\ p = 0, \cdots, M-q$$

with

$$\boldsymbol{a}^q_{p(m,n)} = [a^q_{p(m,n)}(1), a^q_{p(m,n)}(2), \cdots, a^q_{p(m,n)}(q)]^T \quad (7)$$

$$\boldsymbol{b}^q_{p(m,n)} = [b^q_{p(m,n)}(q), b^q_{p(m,n)}(q-1), \cdots, b^q_{p(m,n)}(1)]^T \quad (8)$$

$$\boldsymbol{Y}^q_{p(i,j)} = [y((i,j)-p), \cdots, y((i,j)-p-q)]^T \quad (9)$$

where $a^q_{p(m,n)}(s)$ and $b^q_{p(m,n)}(s)$ $(s = 1, \cdots, q)$ are the 2-D forward and backward prediction coefficients at the point $((m,n)-p)$, respectively.

Now, the prediction error coefficients are determined to minimize the following sum of weighted prediction error squares:

$$A^q_{p(m,n)} = \sum_{(i,j)=(0,0)}^{(m,n)} \lambda^{S-s(i,j)}\{f^q_{((i,j)-p)}\}^2 \quad (10)$$

$$B^q_{p(m,n)} = \sum_{(i,j)=(0,0)}^{(m,n)} \lambda^{S-s(i,j)}\{r^q_{((i,j)-p)}\}^2 \quad (11)$$

where λ $(0 \le \lambda \le 1)$ is the exponential weighting factor. The $s(i,j)$ is the total number of 2-D input data used by the point (i,j), and S = s(m,n). In the recursion of 2-D input data, raster scan is taken.

If (5) and (6) are substituted into (10) and (11), respectively, the stationary condition of $A^q_{p(m,n)}$ and $B^q_{p(m,n)}$ with respect to the prediction coefficient yields the following 2-D deterministic normal equation:

$$\boldsymbol{R}^q_{p(m,n)} \begin{bmatrix} 1 & \boldsymbol{b}^q_{p(m,n)} \\ \boldsymbol{a}^q_{p(m,n)} & 1 \end{bmatrix} = \begin{bmatrix} \hat{A}^q_{p(m,n)} & \mathbf{o} \\ \mathbf{o} & \hat{B}^q_{p(m,n)} \end{bmatrix} \quad (12)$$

where

$$\boldsymbol{R}^q_{p(m,n)} = \sum_{(i,j)=(1,1)}^{(m,n)} \lambda^{S-s(i,j)}[\ \boldsymbol{Y}^q_{p(i,j)}\{\boldsymbol{Y}^q_{p(i,j)}\}^T\ ], \quad (13)$$

$\hat{A}^q_{p(m,n)}$ and $\hat{B}^q_{p(m,n)}$ are the minimum mean-squares of the forward and backward prediction errors, respectively, and $\mathbf{o}$ is the zero column vector.

2.2 Order Recursions

2.2.1 Recursion for prediction errors

In this section, order recursion for the prediction errors is derived. At first, order recursion for prediction coefficients is discussed. Prediction coefficient vectors $\boldsymbol{a}^q_{p(m,n)}$ and $\boldsymbol{b}^q_{p(m,n)}$ of order q can be updated from the $\boldsymbol{a}^{q-1}_{p(m,n)}$ and $\boldsymbol{b}^{q-1}_{p+1(m,n)}$ of order $q-1$. The update equation is defined by

$$\begin{bmatrix} 1 & \boldsymbol{b}^q_{p(m,n)} \\ \boldsymbol{a}^q_{p(m,n)} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \boldsymbol{a}^{q-1}_{p(m,n)} & \boldsymbol{b}^{q-1}_{p+1(m,n)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & K^r_{p,q(m,n)} \\ K^f_{p,q(m,n)} & 1 \end{bmatrix} \quad (14)$$

where $K^f_{p,q(m,n)}$ and $K^r_{p,q(m,n)}$ are the so-called reflection coefficients at the point $((m,n)-p)$.

Substitution of (14) into (12) yields the reflection coefficients such that

$$K^f_{p,q(m,n)} = -\Delta^{q-1}_{p(m,n)}/\hat{B}^{q-1}_{p+1(m,n)} \quad (15)$$

$$K^r_{p,q(m,n)} = -\Delta^{q-1}_{p(m,n)}/\hat{A}^{q-1}_{p(m,n)} \quad (16)$$

and minimum mean square errors

$$\hat{A}^q_{p(m,n)} = \hat{A}^{q-1}_{p(m,n)} + K^f_{p,q(m,n)}\Delta^{q-1}_{p(m,n)} \quad (17)$$

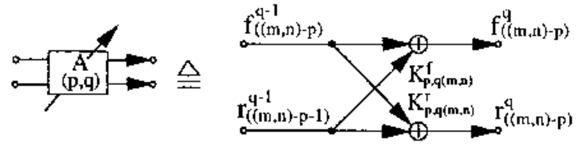
$$\hat{B}^q_{p(m,n)} = \hat{B}^{q-1}_{p+1(m,n)} + K^r_{p,q(m,n)}\Delta^{q-1}_{p(m,n)} \quad (18)$$

where

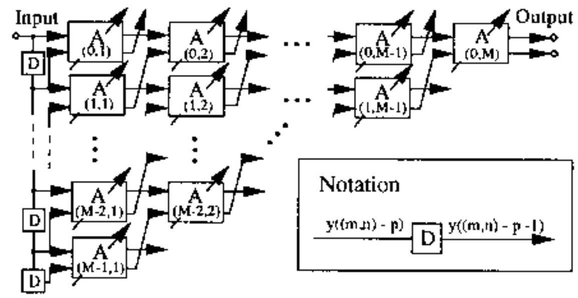
$$\Delta^{q-1}_{p(m,n)} = \begin{bmatrix} 1 \\ \boldsymbol{a}^{q-1}_{p(m,n)} \\ 0 \end{bmatrix} \boldsymbol{R}^q_{p(m,n)} \begin{bmatrix} 0 \\ \boldsymbol{b}^{q-1}_{p+1(m,n)} \\ 1 \end{bmatrix}. \quad (19)$$

Furthermore, premultiplication of $\boldsymbol{Y}^q_{p(m,n)}$ by the transpose of both side of (14) yields the recurrence formula of the forward and backward prediction errors:

$$\begin{bmatrix} f^q_{((m,n)-p)} \\ r^q_{((m,n)-p)} \end{bmatrix} = \begin{bmatrix} 1 & K^f_{p,q(m,n)} \\ K^r_{p,q(m,n)} & 1 \end{bmatrix} \begin{bmatrix} f^{q-1}_{((m,n)-p)} \\ r^{q-1}_{((m,n)-p-1)} \end{bmatrix} \quad (20)$$



(a) Internal structure.



(b) Overall structure.

Fig. 1 2-D lattice structure.

with the initial conditions as

$$f^0_{((m,n)-p)} = r^0_{((m,n)-p)} = y((m,n)-p) \quad (21)$$

$$y((m,n)-0) = y(m,n). \quad (22)$$

The proposed 2-D lattice structure generated by (20) is illustrated in Fig. 1.

2.2.2 Recursion for conversion factor

To complete order recursion required for the LSL algorithm, we need another recursion for the so-called conversion factor. The gain vector is defined by

$$\boldsymbol{g}^q_{p(m,n)} = [\boldsymbol{R}^{q-1}_{p(m,n)}]^{-1}\boldsymbol{Y}^{q-1}_{p(m,n)}. \quad (23)$$

It can be viewed as the solution of the 2-D deterministic normal equation for a special desired response such as

$$d((i,j)-p) = \begin{cases} 1 & (i,j)=(m,n) \\ 0 & Otherwise. \end{cases}$$

The conversion factor, defined by

$$\gamma^q_{p(m,n)} = 1 - [g^q_{p(m,n)}]^T \boldsymbol{Y}^{q-1}_{p(m,n)}, \quad (24)$$

has important property that is bounded within zero and one:

$$0 \le \gamma^q_{p(m,n)} \le 1. \quad (25)$$

When the filter operates for a stationary input signal, $\gamma^q_{p(m,n)}$ rapidly increases to 1. For a non-stationary input signal, $\gamma^q_{p(m,n)}$ rapidly decreases to zero.

On the other hand, the inverse of the correlation matrix $\boldsymbol{R}^q_{p(m,n)}$ can be expressed as follows:

$$[\boldsymbol{R}^q_{p(m,n)}]^{-1} = \begin{bmatrix} [\boldsymbol{R}^{q-1}_{p(m,n)}]^{-1} & \mathbf{o} \\ \mathbf{o} & 0 \end{bmatrix} + \frac{1}{\hat{B}^q_{p(m,n)}} \begin{bmatrix} \boldsymbol{b}^q_{p(m,n)} \\ 1 \end{bmatrix} [\ (\boldsymbol{b}^q_{p(m,n)})^T \ \ 1\ ]. \quad (26)$$

Using (23) and (26), the order recursion of the gain vector $\boldsymbol{g}^q_{p(m,n)}$ can be obtained by

$$\boldsymbol{g}^{q+1}_{p(m,n)} = \begin{bmatrix} \boldsymbol{g}^q_{p(m,n)} \\ 0 \end{bmatrix} + \frac{r^q_{((m,n)-p)}}{\hat{B}^q_{p(m,n)}} \begin{bmatrix} \boldsymbol{b}^q_{p(m,n)} \\ 1 \end{bmatrix}. \quad (27)$$

Premultiplication of $\boldsymbol{Y}^q_{p(m,n)}$ by the transpose of both side of (27) yields the order recursion of the conversion factor $\gamma^q_{p(m,n)}$:

$$\gamma^{q+1}_{p(m,n)} = \gamma^q_{p(m,n)} - \{r^q_{((m,n)-p)}\}^2 / \hat{B}^q_{p(m,n)}. \quad (28)$$

2.3 Space Recursion

In this section, space recursion for the cross-correlation function $\Delta^{q-1}_{p(m,n)}$ is derived. From (13), the correlation matrix $\boldsymbol{R}^q_{p(m,n)}$ can be rewritten as

$$\boldsymbol{R}^q_{p(m,n)} = \lambda \boldsymbol{R}^q_{p(m,n-1)} + [\boldsymbol{Y}^q_{p(m,n)}][\boldsymbol{Y}^q_{p(m,n)}]^T. \quad (29)$$

Substituting (29) into (19) yield the space recursion for $\Delta^{q-1}_{p(m,n)}$:

$$\Delta^{q-1}_{p(m,n)} = \lambda \Delta^{q-1}_{p(m,n-1)} + \eta^{q-1}_{p(m,n)} r^{q-1}_{((m,n)-p-1)} \quad (30)$$

where $\eta^{q-1}_{p(m,n)}$ is the so-called forward a priori prediction error defined by

$$\eta^q_{p(m,n)} = [\ 1 \ \ (\boldsymbol{a}^q_{p(m,n-1)})^T] \boldsymbol{Y}^q_{p(m,n)}. \quad (31)$$

Between $\eta^q_{p(m,n)}$ and $f^q_{((m,n)-p)}$, we have the relationship as follows:

$$\gamma^q_{p+1(m,n)} = \frac{f^q_{((m,n)-p)}}{\eta^q_{p(m,n)}}. \quad (32)$$

Using (32), eq. (30) can be rewritten as

$$\Delta^{q-1}_{p(m,n)} = \lambda \Delta^{q-1}_{p(m,n-1)} + \frac{f^{q-1}_{((m,n)-p)} r^{q-1}_{((m,n)-p-1)}}{\gamma^{q-1}_{p+1(m,n)}}. \quad (33)$$

The correction term $f^{q-1}_{((m,n)-p)} r^{q-1}_{((m,n)-p-1)}$ in (33) is amplified by the reciprocal of the conversion factor $\gamma^{q-1}_{p+1(m,n)}$. This parameter enables the proposed LSL algorithm to adapt rapidly to sudden changes in the 2-D input data.

3 EXAMPLE

Computer simulation for modeling a texture image (100×100), shown in Fig. 2, is carried out to confirm the proposed method has superior convergence over the LMS algorithm [6] and the conventional LMS lattice algorithm [5]. The recursion of the algorithm use a raster scan type of 2-D input data. Each model has AHP coefficient support with 12 stages ($M = 12$).

To measure the performance of the adaptive algorithm, we use the mean square error

$$MSE_k = E[\{f^M_{((m,n)-0)}\}^2] \quad (34)$$

and the mean square deviation

$$MSD_k = E[(\boldsymbol{a}^M_{0(m,n)})^T(\boldsymbol{a}^M_{0(m,n)})] \quad (35)$$

where the subscript k is the number of iterations given by $100(m-1)+n$. Furthermore, to examine the behavior of the conversion factor, we use the average of $\gamma^q_{0(m,n)}$ ($q = 1, \cdots, M$):

$$\gamma_k = \sum_{q=1}^{M} \gamma^q_{0(m,n)}. \quad (36)$$

Figs. 3 and 4 show MSE_k and MSD_k versus the number of iterations, respectively. Each curve is obtained by averaging 20 independent texture images. The step-size parameters of the LMS algorithm and the conventional LMS lattice algorithm are 1.0×10^{-6} and

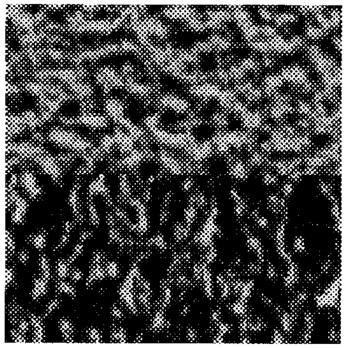


Fig. 2 Texture image.

1.0×10^{-5}, respectively. The weighting factor λ of the proposed algorithm is 0.99. Figs. 3 and 4 show that the proposed method has superior convergence property.

Fig. 5 depicts the trajectory of γ_k obtained by modeling one of the samples. The value of γ_k is small during the initialization period. Thereafter, γ_k rapidly increases to 1, and it decreases rapidly at the boundary portion of two textures in the image.

4 CONCLUSION

In this paper, 2-D LSL algorithm for the general case of the AR model with an AHP support is presented. It is shown that the proposed model has superior convergence property through computer simulation for modeling a texture image.

ACKNOWLEDGMENT

This work is supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant No. 07650444, and the participation expense for the conference is supported in part by NTT DoCoMo.

References

[1] Moro, H., Watanabe T., Taguchi, A. and Hamada, N., *Proc. ECCTD'95*, pp. 430-434, 1989.

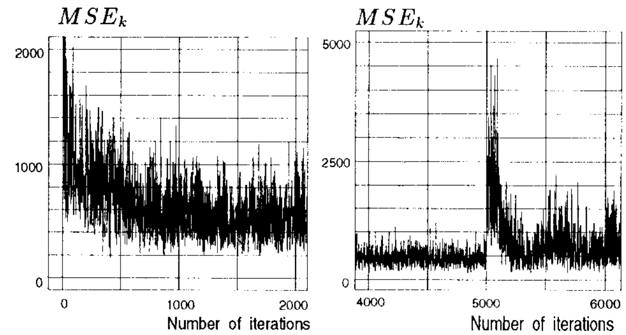
[2] Liu, X., Baylou, P. and Najim, M., *Proc. ICASSP'92*, vol. 3, pp. 329-332, Apr. 1992.

[3] Youlal, H., Janati-I, M. and Najim, M., *IEEE Trans. on Image Process.*, vol. 1, no. 3, pp. 366-378, July 1992.

[4] Meylani, R., Sezen, S., Ertüzün, A., and İstefanopulos, Y., *Proc. ECCTD'95*, vol. 2, pp. 741-744, 1995.

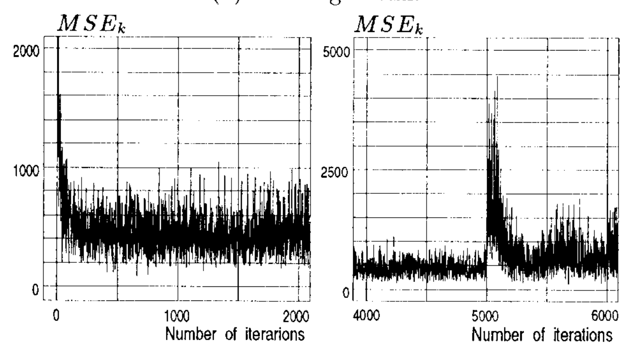
[5] Nakachi, T.,Yamashita, K. and Hamada, N., *Proc. of DSP Symp.*, pp. 349-354, Nov. 1995.

[6] Hadhoud, M. M. and Thomas D. W., *IEEE Trans. on Circuits and Systems*, vol. CAS-35, no. 5, pp. 485-494, May 1988.



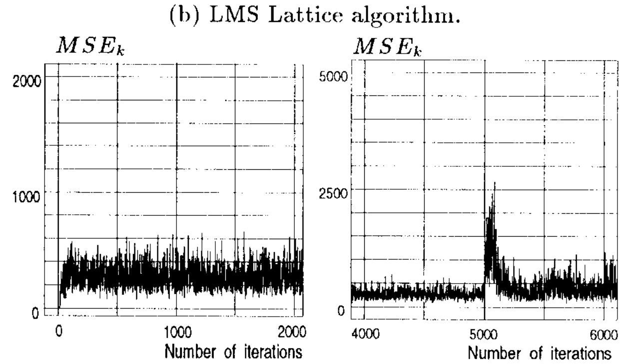
Initialization period. Transition period.

(a) LMS algorithm.



Initialization period. Transition period.

(b) LMS Lattice algorithm.



Initialization period. Transition period.

(c) Proposed algorithm.

Fig. 3 MSE_k versus the number of iterations.

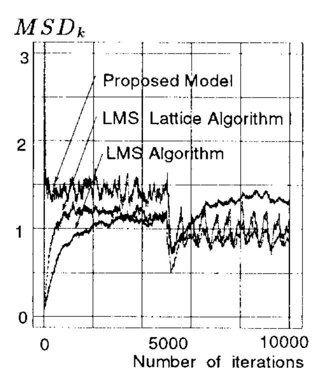


Fig. 4 MSD_k versus the number of iterations

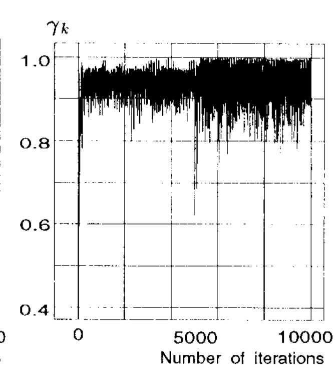


Fig. 5 γ_k versus the number of iterations.