TREIZIÈME COLLOQUE GRETSI - JUAN-LES-PINS DU 16 AU 20 SEPTEMBRE 1991

A New 2D Fast Lattice RLS Algorithm Application for Restoration of Images

Liu Xiang, M. Najim, B. Ténèze* and H. Youlal*

Equipe Signal et Image, ENSERB, 351, Cours de la Libération, 33405 TALENCE Cedex, FRANCE
*LEESA. Faculté des Sciences. BP 1014. Rabat, MOROCCO
GRECO_TDSI

RÉSUMÉ

Résumé. Cet article porte sur le développement d'un nouvel algorithme rapide 2D des Moindres Carrées Récursif en treillis (2D FLRLS) . Nous illustrons son utilisation et ses performances en tant qu' annuleur de bruit 2D pour la restauration d'images bruitées.

ABSTRACT

Abstract. This paper deals with a computational efficient new 2D fast lattice recursive least squares (2D FLRLS) algorithm. We illustrate its implementation as a 2D noise canceller for restoration of noisy images.

In the framework of real time applications, we are carrying on different approaches to perform such a task either in proposing new algorithms, or ad hoc architecture[12]. In such a way, this paper is constructed as follows: In the introduction, we give a short survey about the main contributions in 1 and 2D recursive least squares fast algorithms and, in section 2, we provide a brief outline for the derivation of a new 2D FLRLS algorithm. The application for the restoration of noisy images is included in section 3. Conclusions and further extensions are given in section 4.

1. INTRODUCTION

Computational efficient 1D RLS (Recursive Least Squares) adaptive algorithms have been used successfully in a wide range of applications [1][2][3][8]. Among the numerous papers which has been published in the area of adaptive and fast algorithms, special attention has to be paid to the comprehensive survey from the GRECO_TDSI [3]. One has to notice that only few among these contributions deals with 2D or multidimensional cases. M. Najim and Youlal [4] developed a 2D N_LMS lattice and obtained high improvement in MSE (Minimum Square Error) for adaptive restoration of images. Boutalis et al. [5] introduced a technique for adaptive image estimation based on the multichannel form of the FAEST. Sequeira et al.[6] developed a 2D fast RLS transversal algorithm version by using a similar approach for the 1D FTF derivation. Amoumou[10] has proposed an extension to the multichannel case.

H. Kaufman et al[7], have published a survey on 2D parameter estimation techniques for image restoration and arose the need of fast procedures for updating filter parameters.

In this paper, by using a geometrical approach technique based on the orthogonal projection and by imposing order on the 2D sequence, we transfer a 2D LS problem into a similar 1D multichannel forward and backward filtering problem and get an exact recursive lattice LS solution for the 2D AR model: 2DFLRLS. We also have extended to 2D FLRLS the joint process estimator and applied it to the restoration of noisy images.

2. 2D FAST LATTICE RECURSIVE LEAST SQUARES ALGORITHM(2D FLRLS)

2.1 THE 2D LS PREDICTION PROBLEM

A 2D data sequence with L*K pixels is shown in Fig. 1.

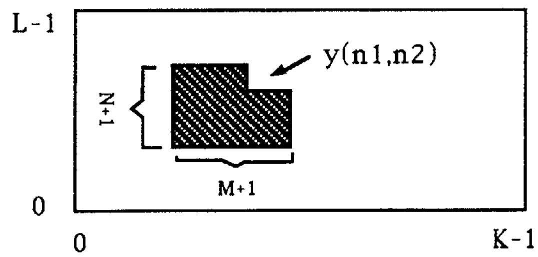


Figure 1. First Quadrant AR Model Support Region

We consider a general 2D AR image model as follows:

$$y(n_1,n_2)= \sum_{(i,j)\in R} a_{i,j}\, y(n_1-i,n_2-j)+u(n_1,n_2) \qquad (1.1)$$

Where R is the model support region, which is defined as follows:

$$R=\{ i,j: (i\geq 0, j>0) \cup (i>0, j\geq 0) \} \qquad (1.2)$$

We choose linear index order along columns, which are described as follows:

$$y(-i,j)=y(L-i,j-1) \quad (i>0,j>0) \qquad (1.3)$$
$$y(i, j)=0 \quad (j<0) \qquad (1.4)$$

A 2D linear predictive filter is of the form:

$$\hat{y}(n_1,n_2)=\sum_{(i,j)\in R}\hat{a}_{i,j}\,y(n_1-i,n_2-j) \qquad (1.5)$$

Space recursive least squares (RLS) algorithms can be defined as a technique, which estimates coefficients of model by minimizing the accumulated squared error

$$\xi(n_1,n_2)=\sum_{i=0}^{n_1}[e(i,n_2)]^2+\sum_{i=0}^{L-1}\sum_{j=0}^{n_2-1}[e(i,j)]^2 \qquad (1.6)$$

where: $e(i,j)=y(i,j)-\hat{y}(i,j)$ (1.7)

2.2 Presentation of the Algorithm

First, let us introduce a linear scanning index n such that n=n2*L+n1. As shown in figure 2, the 2D data y(n1,n2) can be expressed as :

$$y(n_1,n_2)=y_1(n) \qquad (2.1)$$

For the other channels $y_i(n)$ is given in terms of $y_1(n)$ as follows:

$$y_i(n)=y_1(n-(i-1)L) \qquad (2.2)$$

Let us define:

a) J×1 vector as follows:(J>> K×L)

$$\underline{x}_1(n)=[y_1(n)\ y_1(n-1),\ldots,y_1(0)\ \ 0,\ldots,0]$$
$$\underline{x}_2(n)=[y_2(n)\ y_2(n-1),\ldots,y_2(0)\ \ 0,\ldots,0]$$
$$\ldots\ldots$$
$$\underline{x}_{M+1}(n)=[y_{M+1}(n),\ldots,y_{M+1}(0)\ 0,\ldots,0] \qquad (2.3)$$

b) J×K_2 forward/backward prediction vector spaces at mth stage: (K_2=(N+1)(M+1)-1)

$$X_m^{F,n}=[\underline{x}_2(n)\cdots\underline{x}_{M+1}(n)\ \underline{x}_1(n-1)\cdots\underline{x}_{M+1}(n-1)\cdots \underline{x}_1(n-m)\cdots\underline{x}_{M+1}(n-m)] \qquad (2.4)$$

$$X_m^{B,n}=[\underline{x}_2(n+1)\cdots\underline{x}_{M+1}(n+1)\ \underline{x}_1(n)\cdots\underline{x}_{M+1}(n)\cdots\underline{x}_1(n-m+1)\cdots\underline{x}_{M+1}(n-m+1)] \qquad (2.5)$$

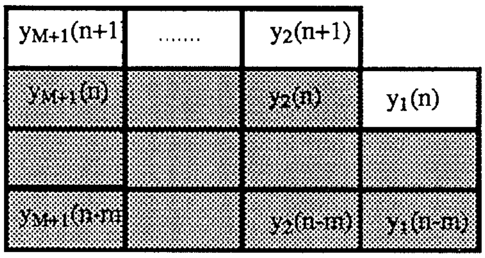


Figure 2. Forward Prediction Matrix

Now, the normal eqations of 2D LS problem can be expressed in a matrix form:

$$[(X_m^{F,n})^T(X_m^{F,n})]\hat{\underline{a}}(n)=(X_m^{F,n})^T\underline{x}_1(n) \qquad (2.6)$$

and an error vector is introduced as :

$$\underline{e}_1(n)=\underline{x}_1(n)-(X_m^{F,n})\hat{\underline{a}}(n) \qquad (2.7)$$

The sum of the squared errors in the RLS adaptive filter is interpreted as the squared length of an appropriate defined error vector. Elementary geometrical concepts are than used to minimize this length[2].

In the vector space, $\underline{e}_1(n)$ is orthogonal to the column space of $X_m^{F,n}$. Let us introduce the following projection operators:

$$P_U=U(U^TU)^*U^T \quad \text{and} \quad P_U^{\perp}=I-P_U \qquad (2.8)$$

From (2.7), we can get

$$\underline{e}_1(n)=P^{\perp}_{X_m^{F,n}}\underline{x}_1(n) \qquad (2.9)$$

we will now discuss the mth stage forward/backward prediction problems.

Let us define a multichannel forward prediction matrix (J×(M+1))

$$X^F(n)=[\underline{x}_2(n+1)\ldots\underline{x}_{M+1}(n+1)\ \underline{x}_1(n)] \qquad (2.10)$$

and forward prediction error matrix at mth stage:

$$E_m^F(n)=X^F(n)-\hat{X}_m^F(n) \qquad (2.12)$$

A multichannel forward prediction problem can be defined as :

$$E_m^F(n)=P^{\perp}_{X_m^{F,n}}X^F(n) \qquad (2.13)$$

It projects the forward prediction matrix into the same column space as for solving the 2D LS problem.

The multichannel forward error power is expressed as follows:

$$\varepsilon_m^{F,n}=(X^F(n))^TP^{\perp}_{X_m^{F,n}}X^F(n) \qquad (2.14)$$

Similarly, we have:

$$X_m^B(n)=[\underline{x}_1(n-m)\cdots\underline{x}_{M+1}(n-m)] \qquad (2.15)$$
$$E_m^B(n)=X_m^B(n)-\hat{X}_m^B(n) \qquad (2.16)$$
$$E_m^B(n)=P^{\perp}_{X_m^{B,n}}X_m^B(n) \qquad (2.17)$$
$$\varepsilon_m^{B,n}=(X_m^B(n))^TP^{\perp}_{X_m^{B,n}}X_m^B(n) \qquad (2.18)$$

Now, we introduce the pinning vector as J×1 vector :

$$\sigma=[1\ 0\cdots 0]^T \qquad (2.19)$$

Thus, all the relationships at space position n=(n_1,n_2)(m^{th} stage):

$$\underline{x}^f(n)=[y_2(n+1)\ \cdots\ y_{M+1}(n+1)\ \ y_1(n)] \qquad (2.20)$$
$$\underline{x}_m^b(n)=[y_1(n-m)\ \cdots\ y_{M+1}(n-m)] \qquad (2.21)$$
$$\underline{e}_m^f(n)=\underline{x}^f(n)-\hat{\underline{x}}^f(n) \qquad (2.22)$$
$$\underline{e}_m^b(n)=\underline{x}_m^b(n)-\hat{\underline{x}}_m^b(n) \qquad (2.23)$$

can be expressed with the pinning vector:

$$\underline{x}^f(n)=\sigma^T X^F(n)\ ,\quad \underline{x}_m^b(n)=\sigma^T X_m^B(n)$$
$$\underline{e}_m^f(n)=\sigma^T P^{\perp}_{X_m^{F,n}}X^F(n)\ ,\quad \underline{e}_m^b(n)=\sigma^T P^{\perp}_{X_m^{B,n}}X_m^B(n)$$

We introduce the squared cosine of angle between the columns space at space positions n and n-1(m^{th} stage)

$$\gamma_m(n)=\cos^2(\theta)=\sigma^T P^{\perp}_{X_m^{B,n}}\sigma \qquad (2.24)$$

For the joint process predictor, which models a 2D process d(i,j) from a related process y(i,j), the extension is easy. It can be interpreted as an orthogonal projection, that projects the vector $d_1(n)$(defined as $y_1(n)$) into the column space $X_m^{F,n}$.

$$\underline{e}_m^{d,y}(n)=P^{\perp}_{X_m^{F,n}}\underline{d}_1(n)$$

Here, we have merely defined the components of the algorithm. Due to limited space, we will only give a brief outline of formulas. A complete derivation will be provided in [11].

2.3 Summary of the 2D FLRLS Algorithm

We initialize the algorithm at n=0. The reflection coefficients matrix $K_m^f(-1)$, $K_m^b(-1)$ and the joint lattice reflection coefficients matrix $H_m^{d,y}(-1)$ are set to zero, The inverse error covariance matrix $(\varepsilon_m^{F,-1})^{-1}$,$(\varepsilon_m^{B,-1})^{-1}$ are set to the diagonal matrix with error elements 1/a, a is a small positive constant.

For n=0, 1, ...

Initialize:

$\bar{x}_0^b(n)=[\, y_2(n)\cdots y_{M+1}(n)\,]$

$\bar{x}_0^f(n)=[\, y_2(n+1)\cdots y_{M+1}(n+1)\,]$

$$(\bar\varepsilon_0^{F,n})^{-1}=(\tfrac{1}{\lambda}\bar\varepsilon_0^{F,n-1})^{-1}-\frac{(\frac{1}{\lambda}\bar\varepsilon_0^{F,n-1})^{-1}(\bar x^f(n))^T\bar x^f(n)(\frac{1}{\lambda}\bar\varepsilon_0^{F,n-1})^{-1}}{1+\bar x^f(n)(\frac{1}{\lambda}\bar\varepsilon_0^{F,n-1})^{-1}(\bar x^f(n))^T}$$

$$(\bar\varepsilon_0^{B,n})^{-1}=(\tfrac{1}{\lambda}\bar\varepsilon_0^{B,n-1})^{-1}-\frac{(\frac{1}{\lambda}\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T\bar x_0^b(n)(\frac{1}{\lambda}\bar\varepsilon_0^{B,n-1})^{-1}}{1+\bar x_0^b(n)(\frac{1}{\lambda}\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T}$$

$$\bar K_0^f(n)=\bar K_0^f(n-1)+\frac{(\frac1\lambda\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T}{1+\bar x_0^b(n)(\frac1\lambda\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T}\times(x^f(n)-\bar x_0^b(n)\times\bar K_0^f(n-1))$$

$$(\bar K_0^b(n))=(\bar K_0^b(n-1)+\frac{(\frac1\lambda\bar\varepsilon_0^{F,n-1})^{-1}(\bar x^f(n))^T}{1+\bar x^f(n)(\frac1\lambda\bar\varepsilon_0^{F,n-1})^{-1}(\bar x^f(n))^T}\times(x_0^b(n)-\bar x^f(n)\times\bar K_0^b(n-1))$$

$e_0^f(n)=x^f(n)-\bar x_0^b(n)\times\bar K_0^f(n)$

$e_0^b(n)=x_0^b(n)-\bar x_0^f(n)\times\bar K_0^b(n)$

$\gamma_0(n-1)=1-\bar x_0^b(n)(\bar\varepsilon_0^{B,n})^{-1}(\bar x_0^b(n))^T$

For m=1, 2, ..., N

$$(\varepsilon_{m-1}^{F,n})^{-1}=(\tfrac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}-\frac{(\frac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}(e_{m-1}^f(n))^Te_{m-1}^f(n)(\frac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}}{\gamma_{m-1}(n-1)+e_{m-1}^f(n)(\frac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}(e_{m-1}^f(n))^T}$$

$$(\varepsilon_{m-1}^{B,n})^{-1}=(\tfrac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}-\frac{(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^Te_{m-1}^b(n-1)(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}}{\gamma_{m-1}(n-1)+e_{m-1}^b(n-1)(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^T}$$

$$K_{m-1}^f(n)=K_{m-1}^f(n-1)+(e_{m-1}^f(n)-e_{m-1}^b(n-1)\times K_{m-1}^f(n-1))\times\frac{(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^T}{\gamma_{m-1}(n-1)+e_{m-1}^b(n-1)(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^T}$$

$$K_{m-1}^b(n)=K_{m-1}^b(n-1)+(e_{m-1}^b(n-1)-e_{m-1}^f(n)\times K_{m-1}^b(n-1))\times\frac{(\frac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}(e_{m-1}^f(n))^T}{\gamma_{m-1}(n-1)+e_{m-1}^f(n)(\frac1\lambda\varepsilon_{m-1}^{F,n-1})^{-1}(e_{m-1}^f(n))^T}$$

$e_m^f(n)=e_{m-1}^f(n)-e_{m-1}^b(n-1)\times K_{m-1}^f(n)$

$e_m^b(n)=e_{m-1}^b(n-1)-e_{m-1}^f(n)\times K_{m-1}^b(n)$

$\gamma_m(n)=\gamma_{m-1}(n-1)-e_{m-1}^f(n)(\varepsilon_{m-1}^{F,n})^{-1}(e_{m-1}^f(n))^T$

For the joint process predictor, we add the order recur sions as follows:

Initialize:

$$(\bar H_0^{d,y}(n))=(\bar H_0^{d,y}(n-1)+(d(n)-\bar x_0^b(n)\times\bar H_0^{d,y}(n-1))\times\frac{(\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T}{1+\bar x_0^b(n)(\bar\varepsilon_0^{B,n-1})^{-1}(\bar x_0^b(n))^T}$$

$e_0^{d,y}(n)=d(n)-\bar x_0^b(n)\times\bar H_0^{d,y}(n)$

For m=1, 2, ..., N

$$H_{m-1}^{d,y}(n)=H_{m-1}^{d,y}(n-1)+(e_{m-1}^{d,y}(n)-e_{m-1}^b(n-1)\times H_{m-1}^{d,y}(n-1))\times\frac{(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^T}{\gamma_{m-1}(n-1)+e_{m-1}^b(n-1)(\frac1\lambda\varepsilon_{m-1}^{B,n-2})^{-1}(e_{m-1}^b(n-1))^T}$$

$e_m^{d,y}(n)=e_{m-1}^{d,y}(n)-e_{m-1}^b(n-1)\times H_{m-1}^{d,y}(n)$

For the joint process predictor, the total number of operations (multiplications and divisions) required per iteration of the algorithm at m=N, is

$$11(N+1)(M+1)^2+9(N+1)(M+1)+11M^2+15M+1$$

3. APPLICATION TO THE RESTORATION OF NOISY IMAGE

A specific application of the 2D joint process estimator is a 2D noise canceller. For the image restoration problem, we assume that the primary image d(i,j) is an ideal image s(i,j) corrupted by noise:

$$d(i,j)=s(i,j)+v(i,j)$$

The reference input field y(i,j) is a 2D random field which is correlated with the noise v(i,j). It is assumed to be such as:

$$B(Z_1^{-1},Z_2^{-1})y(i,j)=v(i,j)$$

where $B(Z_1^{-1},Z_2^{-1})$ is a stable 2D polynomia.

The corresponding restored image is

$$\hat s(i-1,j)=e_{N+1}^{d,y}(i,j)$$

The stimulation of the proposed 2D FLRLS noise canceller is performed by using artificial blurred image data for several signal to noise ratio(SNR). The primary image "Lena" at a resolution of 256*256 pixels, is corrupted by an additive Gaussian noise or a spatial periodic noise v(i,j). The input signal to noise ratio is defined as:

$$SNR(dB)=10\log_{10}\frac{\text{Variance of the original image}}{\text{Variance of the noise}}$$

The performances of this technique are expressed by using the improvement in the SNR:

$$P_{MSE}(dB)=10\log_{10}\frac{\text{MSE of degraded image}}{\text{MSE of restored image}}$$

The results reported inTable 1, 2 are obtained by setting:

$$B(Z_1^{-1},Z_2^{-1})=1-0.2Z_1^{-1}+0.6Z_2^{-1}+0.14Z_1^{-1}Z_2^{-1}$$

The results reported in table 1 are obtained by running the proposed algorithm and the 2D LNLMS[4] algorithm 10 times using different Gaussian noise sequences.

From these results, we can see that high improvement in MSE has been achieved, even at very low input SNR, by using the technique developed in this paper. Comparison with the 2D

LNLMS noise canceller[4], it provides better improvement in MSE.

4. CONCLUSIONS

A new 2D FLRLS algorithm and application for the restoration of noisy image were described in this paper. This algorithm requires about 11(M+1)K1 operations (multiplications or divisions) per iteration, where (M+1) is the number of channels, K1 is the total number of data used in the 2D filter. A reduction in the computational cost is obtained when compared with the standard RLS algorithm($1.5K1^2$). The 2D lattice NLMS algorithm is known to be more economical than this algorithm in computational cost, but we can expect better convergence of this new algorithm compared to the 2D NLMS algorithm. Though the new algorithm has more complexity in computation than its transversal form, it has nice numerical properties. Further application for the restoration of images in various context is on way. Results will be reported in a future paper[11].

References

[1] B. Friedlander, D.T.L. Lee and M. Morf: " Recursive least squares ladder estimation algorithms" IEEE. Trans. on ASSP, vol. ASSP.29, pp.627-641, June 1981.
[2] J. M. Cioffi and T. Kailath: " Fast, recursive-least-squares transversal filters for adaptive filtering" IEEE Trans. on ASSP, vol. ASSP.32, pp. 304-337, April 1984.
[3] Numéro spécial revue " Traitement du Signal" vol. 6, No. 5, Decembre 1989.
[4] H. Youlal, M. Janati-I. and M. Najim: " Two-dimensional joint process lattice for adaptive restoration of images" IEEE-EURASIP Workshop on Algorithms and Parallel VLSI Architectures, Pont à Mousson,(1990)
A comprehensive version is under review for the IEEE Trans. on ASSP.
[5] Y.S. Boutalis and S.D. Kollias: "A fast multichannel approach to adaptive image estimation" IEEE Trans. on ASSP, vol.37, pp. 1090-1098, July 1989.
[6] A.M. Sequeira and C.W. Therrien: " A new 2-D fast RLS algorithm" Proc. IEEE ICASSP, Albuquerque.(1990).
[7] H. Kaufman and A. M. Tekalp: "Survey of estimation Techniques in Image Restoration" IEEE Control System, pp. 16-24, January,1991.
[8] P. Strobach: Linear Prediction Theory. Springer-Verlag Berlin Heidelberg 1990
[9] A. K. Jain: Fundamental of Digital Image Processing. Englewood Cliffs: Prentice Hall, 1989
[10] A. Amoumou, D. Aboutajdine, M. Najim: "Stabilisation de L'Algorithme FTF Multicanal Application a la Modélisation ARMA" GRETSI_91
[11] X. Liu et al: "A New 2D Fast Lattice RLS Algorithm" to be puplished.
[12] B. Ténèze, P. Baylou et M. Najim: "Parallélisation de données séquentielles pour le traitement d'image par système multitransputers" GRETSI_91

Table 1: Result of 2D Noise Canceller

(Gaussian noise, Unit: dB, *_FLRLS,**_LNLMS)

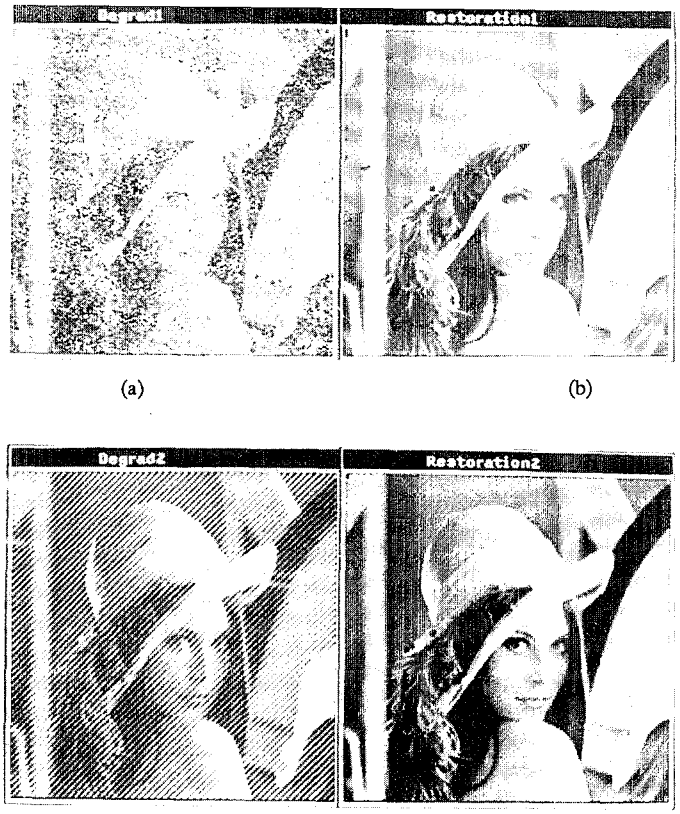
INPUT SNR	10.12	7.19	5.06	0.67	-1.80
OUTPUT SNR*	24.73	24.30	23.36	21.73	19.62
IMPROVEMENT*	14.61	17.11	18.80	21.06	21.42
OUTPUT SNR**	19.06	16.41	14.40	10.22	8.83
IMPROVEMENT**	11.85	9.22	9.34	9.55	9.63

Table 2: Result of 2D Noise Canceller

(Periodic noise, Spatial Frequency=0.01, Unit: dB)

INPUT SNR	11.38	5.51	3.65	-0.17	-1.88
OUTPUT SNR*	30.38	25.70	23.59	18.96	16.63
IMPROVEMENT*	19.00	20.19	19.94	19.13	18.51
OUTPUT SNR**	23.23	15.55	13.77	8.55	5.80
IMPROVEMENT**	11.85	10.05	10.13	8.72	7.68

*--2D FLRLS, **--2D LNLMS



(a) (b) (c) (d)

Fig. 3 (a) Noisy image with SNR=-2dB(Gaussian) (b) Restoration with SNR=19dB(2D FLRLS) (c) Noisy image with SNR=-2dB(Periodic noise) (d) Restoration with SNR=17dB (2D FLRLS)