Image Restoration, Modelling, and Reduction of Dimensionality

ANIL K. JAIN and EDWARD ANGEL

Abstract—Recursive restoration of two-dimensional noisy images gives dimensionality problems leading to large storage and computation time requirements on a digital computer. This paper shows a two-dimensional second-order Markov process representation can be used for fast recursive restoration of images with small storage requirements. Advantages of this method over existing techniques are illustrated by means of examples.

Index Terms—Dimensionality reduction, image enhancement, image filtering, image modelling, image representation, image restoration, real time image filtering.

I. INTRODUCTION

The distribution of brightness levels of monochromatic two-dimensional images often can be described most conveniently by their second-order statistical properties. Experimentally it has been noted [1]-[4] that a single simple correlation function will suffice to approximate the correlation matrices of a large class of real images, thus leading to a number of possible methods for filtering the noise from an observed noisy image. While techniques such as two-dimensional Weiner filtering have an obvious appeal, the two-dimensional nature of the signal often leads to computational difficulties which make on-line restoration difficult.

Recently there has been considerable interest [2]-[4] in recursive estimation of images from noisy observations. In these techniques, the observed image is viewed as being a time signal such as the output of a scanner. The problem is then to construct the optimal on-line (Kalman) filter. Basic to these methods is the necessity of modelling the correlation properties of the image by a linear system with additive white noise. Although experimental results confirm the validity of this modelling, the models themselves are very simple so as to avoid computational difficulties. In effect, these models consider only small amounts of data at a time, which is a suboptimal procedure.

In this paper we will consider another approach to recursive filtering of two-dimensional images corrupted by white Gaussian noise. While such recursive filtering leads to the well-known Kalman–Bucy filters, special consideration is given to the nature of the two-dimensional data so that application of the standard results to large amounts of data is practically feasible. In particular, the unique features of the approach to two-dimensional filtering presented here are as follows.

1) A special model for two-dimensional filtering which gives an estimator very close to the optimal interpolator.

2) Matrix-vector filtering equations which are easily decomposed into scalar equations. This amounts to a scalar filter for vector scanning of the image.

3) A model which allows fast implementation of the filter, thus making the method suitable for on-line image restoration.

4) An isotropic filter so that it is equally effective for all orientations for images which have nearly symmetric autocorrelation function.

II. MODELLING

The several advantages listed above follow directly from the model derived here. Suppose we have an \( N \times N \) two-dimensional monochromatic image. Let \( x_{n,m} \) represent the intensity at the spatial coordinate \((n,m)\). We assume that the image belongs to the class of images whose autocorrelation function can be represented by

\[
E[x_{n,m}x_{n+i,m+j}]=R_{ij}=\sigma^2 \exp (-\gamma |i|-\gamma |j|). \tag{1}
\]

Although a large number of real world images can be approximated by this type of stationary (and separable) autocorrelation function, this form of the autocorrelation function is not crucial to the subsequent development.

The model for image representation is based on the assumption that the random variable, \( x_{i,j} \), is correlated directly with its nearest neighbors as in Fig. 1. Thus, we may write

\[
x_{i,j} = \alpha_1 x_{i,j+1} + \alpha_2 x_{i+1,j} + \alpha_3 x_{i,j-1} + \alpha_4 x_{i-1,j} + \beta u_{i,j}, \tag{2}
\]

where the \([u_{i,j}]\) are zero mean uncorrelated random variables.

Observe that implicit in this assumption is the fact that each point is correlated with every other point through its nearest neighbors. Thus, while point \( A \) in Fig. 1 is directly correlated only with \( B,C,D \), and \( E \), it is, for example, indirectly correlated with \( F,G \), and \( H \) since \( B \) is directly correlated with \( A,F,G \), and \( H \). This model is in
striking contrast with other models. For instance, in the model presented in [2] a given point is correlated directly only with the point before it as in Fig. 2(a), and thus correlations are only in one direction. The model in [3] correlates a given point only with points above and to the left where the solution has already been obtained as in Fig. 2(b).

It is interesting to note that the model of (2) can represent a large number of physical processes such as steady state diffusion, random walk, birth and death processes, and also corresponds to the discretized version of two-dimensional elliptic boundary value problems. It is from this point of view that the difference between our model and other models is most apparent. The models illustrated in Fig. 2(a) and (b) are first order, and thus lead directly to numerical algorithms which directly compute the solution in an initial value on-line manner. Our model lies in the realm of two-point boundary value problems since every point is correlated with every other point. Thus, the model will require more sophisticated numerical algorithms which fortunately will turn out to be computationally feasible.

Our first task is to find values $a_1, a_2, a_3$, and $a_4$ so that the model will have an autocorrelation matrix close to that given by (1). Writing equations for the regression coefficients $a_1, a_2, a_3$, and $a_4$ by taking the correlation of $x_{i,j}$ with its nearest neighbors, it is found

\[ R_{ii} = a_1 R_{00} + a_2 R_{11} + a_3 R_{22} + a_4 R_{33}, \]

\[ R_{i0} = a_1 R_{10} + a_2 R_{01} + a_3 R_{20} + a_4 R_{30}, \]

\[ R_{i1} = a_1 R_{11} + a_2 R_{10} + a_3 R_{12} + a_4 R_{13}, \]

\[ R_{ii} = a_1 R_{22} + a_2 R_{21} + a_3 R_{20} + a_4 R_{20}, \]

Solving for $a_1, a_2, a_3$, and $a_4$, remembering that $R_{ij}$ is given by (1), we get

\[ a_1 = a_2 = \frac{(1 + r_{00}) r_{00} - 2 r_{11} r_{00}}{(1 + r_{00}) (1 + r_{00}) - 4 r_{11}^2}, \]

\[ a_2 = a_3 = \frac{(1 + r_{00}) r_{00} - 2 r_{11} r_{00}}{(1 + r_{00}) (1 + r_{00}) - 4 r_{11}^2}, \]

where

\[ r_{ij} = \frac{R_{ij}}{R_{00}}, \quad r_{00} = \sigma^2. \]

Squaring both sides of (2) and taking expectations, $\beta$ is found by

\[ \beta^2 = \left| 1 - 2(\alpha_1 r_{00} + \alpha_2 r_{00}) \right|, \]

where each $u_{ij}$ is assumed to be a zero-mean random variable with variance $\sigma^2$.

Note that for any stationary (not necessarily separable) autocorrelation function one can employ the same model by using the appropriate values of $R_{ij}$. It can be shown that the autocorrelation of the model is not separable (see Appendix). Thus, even for separable kernels such as (1), it is convenient, as we shall see, to approximate them by nonseparable kernels in order to do recursive filtering in the spatial domain.

III. TWO-DIMENSIONAL VECTOR FILTERING

Although one could work directly with the model given by (2), it will be convenient to first put the model in matrix-vector form. Define $N$ vectors of order $N, x_i$, as

\[ x_i = [x_{ij}], \quad i, j = 1, 2, \ldots, N, \]

and write (2) as the vector equation

\[ x_{i+1} = Q x_i - x_{i-1} + b u_i, \]

with

\[ u_i = [u_{ij}], \quad E[u_i] = 0, \quad E[u_i u_j'] = \sigma^2 \delta_{ij}, \]

\[ b = \frac{\beta}{\alpha_2}, \]

where $\delta_{ij}$ is the Kronecker delta and $Q$ is the tridiagonal matrix

\[ Q = \frac{1}{\alpha_2} \begin{bmatrix} 1 & -\alpha_1 & 0 \\ -\alpha_1 & -\alpha_1 & -\alpha_1 \\ 0 & -\alpha_1 & 1 \end{bmatrix}. \]

The vector $x_i$ corresponds to considering a row (or column) of scanner output at a time. Assuming the ob-
servations \( y_{ij} \) are corrupted by additive white Gaussian noise, we have
\[
y_i = [y_{ij}] = x_i + \eta_i,
\]
where \( \eta_i \) is zero-mean with covariance
\[
E(\eta_i\eta'_j) = \sigma^2 \delta_{ij}
\]
and \( \eta_i \) and \( u_j \) are uncorrelated.

The recursive filtering problem is that of finding an estimate \( \hat{x}_i \) of \( x_i \) such that \( E[(\hat{x}_i - x_i)'(\hat{x}_i - x_i) \mid y_i, y_{i-1}, \ldots, y_1] \) is minimal. The optimal interpolation problem is that of minimizing \( E[(\hat{x}_i - x_i)'(\hat{x}_i - x_i) \mid y_{N, y_{N-1}, \ldots, y_1}] \). While the two problems are somewhat similar in form and possess a common theoretical basis, the interpolation problem is computationally more difficult since all the observations are used to make an estimate at any point, thus leading to large storage requirements. For first-order systems, the optimal recursive estimator is given by the well-known Kalman filter. One could use the Kalman equations directly by first converting the second-order equation of (12) to a first-order equation of dimension \( 2N \). We hesitate to do this since then we would be left with a \( 2N \times 2N \) matrix Riccati equation and there would also be difficulty in deriving the optimal interpolator in a simple manner. Both these difficulties are avoided by working with the dual problem [5], [6]. Then both the optimal estimator and interpolator will be obtained directly in terms of \( N \) dimensional matrices and vectors. By converting (12) to a first-order equation, it easy to show that the dual problem is to minimize the functional
\[
J = \frac{1}{\sigma^2} \sum_{i=1}^{N} (y_i - x_i)'(y_i - x_i) + \frac{1}{\sigma^2} [x_0'x_0 + x_1'x_1]
+ \frac{1}{\sigma^2} \sum_{i=1}^{N} f_i'f_i,
\]
over \( \{x_i\} \) and \( \{f_i\} \) subject to \( x_i \) and \( f_i \) being related by
\[
x_{i+1} = Qx_i - x_{i-1} + bf_i.
\]
Introducing Lagrange multipliers, \( \lambda_i \), there are a number of methods [8], [9] of obtaining necessary conditions for minimizing (24) subject to (25). If \( v_i \) and \( w_i \) denote \( N \)-dimensional vectors of multipliers, the necessary conditions are
\[
x_{i+1} = Qx_i - x_{i-1} + \sigma^2 Qv_{i+1},
\]
\[
v_i = Qv_{i+1} + w_{i+1} + \frac{1}{\sigma^2} (y_i - x_i),
\]
\[
w_i = -v_{i+1},
\]
subject to
\[
v_{N+1} = w_{N+1} = 0,
\]
\[
x_1 = \sigma^2 v_1,
\]
and to also satisfy (25) it is found
\[
E_0 = \frac{\sigma^2}{1 + \sigma^2/\sigma^2} I,
D_0 = 0,
\]
\[
g_0 = \frac{\sigma^2/\sigma^2}{1 + \sigma^2/\sigma^2} y_0.
\]
These equations determine both the optimal estimator and optimal interpolator. We reason as follows. For an online-value of \( x_i \), assume the process of (18) has \( j \) rather than \( N \) stages. Hence, the appropriate boundary conditions would be
\[
w_{j+1} = v_{j+1} = 0,
\]
and by (27)
\[
x_j = g_j,
\]
which is the on-line estimate. If the process has \( N \) stages, then \( x_j \) is the interpolated value. Thus, to find the optimal estimate one needs only to solve (28)–(34) but, to determine the optimal interpolated value, one must solve (28)–(34), store the results, and then work backwards.
using relations based on the original dynamics, (20)–(22), to obtain the desired $x_i$. The storage requirements for this problem can be very large but, in general, the interpolated values are much better than the estimated values. To get around this problem we will use a "one-step" interpolator which can be implemented on-line, adding only a fixed delay. Suppose we want the optimal interpolated value of $x_{N-1}$. By (27) we have
\[
x_{N-1} = D_{N}v_{N} + E_{N}w_{N} + g_{N}.
\]
But using (21) and (23) we see
\[
v_{N} = \frac{1}{\sigma_{n}^{2}}(y_{N} - x_{N}), \tag{40}
\]
\[
w_{N} = 0, \tag{41}
\]
and by (26) and (23)
\[
x_{N} = R_{N}v_{N} + T_{N}w_{N} + s_{N}. \tag{42}
\]
Solving these equations, $x_{N-1}$ is given by
\[
x_{N-1} = \sigma_{n}^{2}D_{N}F_{N}(y_{N} - s_{N}) + g_{N}. \tag{43}
\]
This expression needs observations only one stage past the desired stage. Thus the "one-step" interpolator
\[
x_{i} = \sigma_{n}^{2}D_{i+1}F_{i+1}(y_{i+1} - s_{i+1}) + g_{i+1}, \tag{44}
\]
necessitates a fixed delay of one observation so that $y_{i+1}$ can be received and $D_{i+1}, F_{i+1}, s_{i+1}$, and $g_{i+1}$ computed.

V. REDUCTION OF DIMENSIONALITY

As a first step toward reduction in dimensionality, we have obtained two coupled $N \times N$ matrix Riccati equations rather than one $2N \times 2N$ matrix Riccati equation which would have been required by the direct application of Kalman filtering. A little reflection will indicate that this reduction is not sufficient for most purposes. For images with $N > 64$, the implementation of our equations requires matrix inversions of order $N$ and various other matrix manipulations of this order. For implementation of the optimal interpolator, one not only has these difficulties but one must also store $R_{i}, T_{i}$, and $s_{i}$ for all $i$. Fortunately further reductions are possible and easily realized.

The idea behind the reduction in dimensionality is that any symmetric-tridiagonal matrix has the same orthonormal basis as any other matrix of this form. As is shown in the Appendix, a matrix of orthogonal eigenvectors for symmetric tridiagonal matrices of order $N$ is given by
\[
H = (h_{ij}) = \left( \sin \frac{i\pi}{N + 1} \right). \tag{45}
\]
Let $M$ denote the matrix of orthonormal vectors obtained from $H$. Hence,
\[
MM' = I, \tag{46}
\]
and, if $Q$ is given by (15), then
\[
\Lambda = M'QM, \tag{47}
\]
is diagonal. Suppose we transform the vectors $x_{i}, v_{i}, w_{i}$, and $y_{i}$ by $M'$ and define the new vectors as
\[
\hat{x}_{i} = M'x_{i}, \tag{49}
\]
\[
\hat{v}_{i} = M'v_{i}, \tag{50}
\]
\[
\hat{w}_{i} = M'w_{i}, \tag{51}
\]
\[
\hat{y}_{i} = M'y_{i}. \tag{48}
\]
Then, using (20)–(25) and (47), we find these quantities are determined by the equations
\[
\hat{x}_{i+1} = \Lambda \hat{x}_{i} - \hat{x}_{i-1} + \sigma^{2}h_{i+1} \hat{v}_{i+1}, \tag{49}
\]
\[
\hat{v}_{i} = \Lambda \hat{v}_{i+1} + \hat{w}_{i+1} + \frac{1}{\sigma_{n}^{2}}(\hat{y}_{i} - \hat{x}_{i}), \tag{50}
\]
\[
\hat{w}_{i} = - \hat{v}_{i+1} \tag{51}
\]
satisfy $x_{N-1}$ is given by
\[
\hat{x}_{i} = \sigma_{n}^{2}D_{i+1}F_{i+1}(y_{i+1} - s_{i+1}) + g_{i+1} \tag{44}
\]
necessitates a fixed delay of one observation so that $y_{i+1}$ can be received and $D_{i+1}, F_{i+1}, s_{i+1}$, and $g_{i+1}$ computed.

V. REDUCTION OF DIMENSIONALITY

As a first step toward reduction in dimensionality, we have obtained two coupled $N \times N$ matrix Riccati equations rather than one $2N \times 2N$ matrix Riccati equation which would have been required by the direct application of Kalman filtering. A little reflection will indicate that this reduction is not sufficient for most purposes. For images with $N > 64$, the implementation of our equations requires matrix inversions of order $N$ and various other matrix manipulations of this order. For implementation of the optimal interpolator, one not only has these difficulties but one must also store $R_{i}, T_{i}$, and $s_{i}$ for all $i$. Fortunately further reductions are possible and easily realized.

The idea behind the reduction in dimensionality is that any symmetric-tridiagonal matrix has the same orthonormal basis as any other matrix of this form. As is shown in the Appendix, a matrix of orthogonal eigenvectors for symmetric tridiagonal matrices of order $N$ is given by
\[
H = (h_{ij}) = \left( \sin \frac{i\pi}{N + 1} \right). \tag{45}
\]
Let $M$ denote the matrix of orthonormal vectors obtained from $H$. Hence,
\[
MM' = I, \tag{46}
\]
and, if $Q$ is given by (15), then
\[
\Lambda = M'QM, \tag{47}
\]
is diagonal. Suppose we transform the vectors $x_{i}, v_{i}, w_{i}$, and $y_{i}$ by $M'$ and define the new vectors as
\[
\hat{x}_{i} = M'x_{i}, \tag{49}
\]
\[
\hat{v}_{i} = M'v_{i}, \tag{50}
\]
\[
\hat{w}_{i} = M'w_{i}, \tag{51}
\]
\[
\hat{y}_{i} = M'y_{i}. \tag{48}
\]
Then, using (20)–(25) and (47), we find these quantities are determined by the equations
\[
\hat{x}_{i+1} = \Lambda \hat{x}_{i} - \hat{x}_{i-1} + \sigma^{2}h_{i+1} \hat{v}_{i+1}, \tag{49}
\]
\[
\hat{v}_{i} = \Lambda \hat{v}_{i+1} + \hat{w}_{i+1} + \frac{1}{\sigma_{n}^{2}}(\hat{y}_{i} - \hat{x}_{i}), \tag{50}
\]
\[
\hat{w}_{i} = - \hat{v}_{i+1} \tag{51}
\]
satisfy $x_{N-1}$ is given by
\[
\hat{x}_{i} = \sigma_{n}^{2}D_{i+1}F_{i+1}(y_{i+1} - s_{i+1}) + g_{i+1} \tag{44}
\]
where
\[ \hat{a}_{ji} = \hat{f}_{ji}(\hat{R}_{ji} - \hat{l}_{ji}), \quad (65) \]
\[ \hat{b}_{ji} = \sigma_n^2(1 - \hat{f}_{ji}), \quad (66) \]
\[ \hat{f}_{ji} = \frac{1}{1 + (1/\sigma_n^2)\hat{R}_{ji}}, \quad (67) \]
\[ \hat{g}_{ji} = \hat{f}_{ji} \left( \frac{1}{\sigma_n^2} \hat{R}_{ji} \hat{g}_{ji} + \hat{d}_{ji} \right), \quad (68) \]
and these equations are subject to
\[ \hat{R}_{ji} = \sigma^2, \quad \hat{l}_{ji} = 0, \quad \hat{d}_{ji} = 0, \quad (69) \]
and
\[ \hat{d}_{ji} = \frac{\sigma^2}{1 + \sigma^2/\sigma_n^2}, \quad \hat{d}_{ji} = 0, \quad \hat{g}_{ji} = \frac{\sigma^2/\sigma_n^2}{1 + \sigma^2/\sigma_n^2} \hat{y}_{ji}. \quad (70) \]

If \( \hat{R}_i \) denotes the diagonal matrix of elements \( \hat{R}_{ji} \), i.e.,
\[ \hat{R}_i = \text{diag} \left( \hat{R}_{i1}, \hat{R}_{i2}, \ldots, \hat{R}_{IN} \right), \quad (71) \]
it is easy to show
\[ \hat{R}_i = M' \hat{R}_i M. \quad (72) \]

Similar results hold for the other scalar quantities.

VI. EXAMPLES AND IMPLEMENTATION

A count of the operations required to compute the solution of (62)–(70) will show that the transformation of the observations and the transformation of the solution require \( O(N^3) \) operations while all other computations can be done in \( O(N^2) \) operations. This is a considerable savings over the solution of (28)–(36) which requires \( O(N^4) \) operations. Fortunately, even further reductions are possible. The elements of the transformation matrix \( H \) (or \( M \)) contain terms
\[ h_{ij} = \sin \frac{i \pi}{N + 1}. \quad (73) \]

Therefore, the elements \( h_{ij} \) are related to the Fourier transform and hence a fast Fourier transform (FFT) algorithm can be employed to perform all the necessary transformations. Thus, the \( O(N^2) \) operations are reduced to \( O(N^2 \log_2 N) \) operations, which is a tremendous reduction from the original \( O(N^4) \). Figs. 3 and 4 show the implementation of the scalar filter on \( 32 \times 32 \) images. The difference between the one step interpolator and the optimal interpolator is less than 1.0 db (Table I). Thus, a one step interpolator gives an estimate very close to the optimal interpolator. Even closer estimates may be obtained by implementing a two step interpolator. The computation time on an IBM 360/44 computer for a \( 32 \times 32 \) image was found to be reduced by a factor of 30 without use of the FFT algorithm. Further reduction in computation can of course be achieved by using the FFT algorithm. Implementation details on \( 256 \times 256 \) images via the fast algorithm for the transformation \( M \)

![Image](https://example.com/image1)

**Fig. 3.** (a) Original "S" image. (b) "S" image with additive noise of variance 9 and signal-to-noise ratio 2.33. (c) One step interpolative estimate of the noisy "S" signal. (d) Optimal interpolation of the noisy "S" signal.

![Image](https://example.com/image2)

**Fig. 4.** (a) Original square image. (b) Square image with additive noise of variance 9 and signal-to-noise ratio 2.27. (c) One step interpolative estimate of the noisy square signal. (d) Optimal interpolation of the noisy square signal.

| TABLE I |
| --- | --- | --- | --- |
| | 1-Step | Optimal |
| | Interpolator | Interpolator |
| | Improvement in dB | Improvement in dB |
| Square Signal | 7/3 | 9 | 7.98 | 8.94 |
| S Signal | 6/3 | 9 | 7.08 | 7.57 |

will be presented in a future paper. The coefficients \( \alpha_0, \alpha_2, \alpha_3, \) and \( \alpha_4 \) were determined using the square picture. The same model was used for the "S" picture.

Also, it was found the results were unchanged (in terms of improvement in \( S/N \) ratio) when the orientation of the pictures were changed (i.e., for horizontal, vertical, left to right, right to left scans). This is in contrast with other models such as in Fig. 2(a) and (b), where the estimator or interpolator [2]–[4] tends to stretch the image in the direction of the scan. This somewhat distorted visual appearance is due to the inherent directional bias in the structure of such models even when the images have a symmetric correlation function (i.e.,
where each $R_{ij} = R_{ji}$). The isotropic property in our case follows from (2) since the model is the same when $i$ and $j$ are reversed. For a class of images where the horizontal and vertical correlations have a large difference, a $90^\circ$ rotation of the image would correspond to interchanging the coefficients $a_i$ and $a_j$.

From the $32 \times 32$ results, we see that the results are poorest at the edges. To some extent this is to be expected from the nature of the one-step interpolator and from the nature of the initial approximation which determines the initial conditions of (59) and (60). We can improve the results slightly by including an “edge” term in our equations. For instance, if we know, as in the $32 \times 32$ examples, that the picture lies in a background of given brightness, we can consider, instead of (19), the equation

$$x_{i+1} = Qx_i - x_{i-1} + b_i + c_i,$$

where the vector $c_i$ is a given vector which accounts for the known brightness at the edges.

Perhaps the most important extensions are to more complex models. For instance, one might consider correlations with six or eight nearest neighbors as in Fig. 5(a) and (b) or unsymmetric models as in Fig. 5(c). The model of Fig. 5(b) could be written as

$$Q(x_{i+1} = Qx_i + Qx_{i-1} + b_i + c_i),$$

while that at Fig. 5(c) would be described by the equation

$$Q(x_i = Qx_{i-1} + b_i + c_i),$$

where all the matrices $Q_i$ will be symmetric and tridiagonal. In general, models of the following form can be considered

$$\sum_{j=-1}^{k} Q_{ij}x_{i+j} = u_i,$$

where each $Q_j$ is symmetric and tridiagonal. All these models would, due to the results in the Appendix, lead to fast implementation.

VII. SUMMARY

In summary, first we developed a second-order vector Markov model for restoration of $(N \times N)$ images. For this model, we then obtained scalar filter equations leading to a reduction in dimensionality by the order $N$. Further reduction in dimensionality is achieved via the fast algorithm. Finally, the filter is isotropic and the optimal interpolator is nicely approximated by the one step interpolator, thereby minimizing the storage requirements. Questions regarding the application of this comparison with off-line techniques need further investigation.

APPENDIX

1) Suppose $Q$ is an arbitrary symmetric tridiagonal matrix of order $n$

$$Q = \begin{bmatrix}
\alpha & \beta \\
\beta & \alpha \\
& & \ddots & \beta \\
& & \beta & \alpha
\end{bmatrix}. \quad (A1)$$

$Q$ can be written as

$$Q = \alpha I + \beta P, \quad (A2)$$

where

$$P = \begin{bmatrix}
0 & 1 \\
1 & 1 \\
& & 1 \\
& & & & 0
\end{bmatrix}. \quad (A3)$$

Obviously, if

$$P = TAT', \quad (A4)$$

where $A$ is diagonal and $T$ orthogonal, then

$$Q = TDT', \quad (A5)$$

where the diagonal matrix $D$ is given by

$$D = \alpha I + \beta A. \quad (A6)$$

Let us define a vector $h$ where

$$h = [h_x] = \begin{bmatrix}
\sin \frac{kix}{n+1}
\end{bmatrix} \quad (A7)$$

for $k$ an arbitrary integer. Then, using some trigonometric identities, it is easy to show that the vector $d$

$$d = Ph, \quad (A8)$$

is given by
where using
\[ d = [d_i] = \left[ 2 \sin \frac{k \pi}{n+1} \cos \frac{k \pi}{n+1} \right]. \]

Thus \( Ph = \lambda h, \) \( \text{(A9)} \) if and only if
\[ \lambda = 2 \cos \frac{k \pi}{n+1}. \]
(A10)

Since \( k \) is arbitrary, a complete set of eigenvalues is given by
\[ \lambda_j = 2 \cos \frac{j \pi}{n+1}, \quad j = 1, 2, \ldots, n, \]
(A11)
and the corresponding eigenvectors are given by
\[ h_i = [h_{ij}] = \left[ \sin \frac{ij \pi}{n+1} \right]. \]
(A12)

\( T \) is then constructed from \( \{h_j\} \) by normalizing each vector and is given by
\[ T = \left( \frac{2}{n+1} \right)^{1/2} \{h_1, h_2, \ldots, h_n\}. \]
(A13)

2) To demonstrate the model autocorrelation is not separable, consider
\[ x_{i+1} = -\alpha P x_i - x_{i-1} + \beta u_i, \quad i = 1, \ldots, n \]
(A14)
where
\[ E[u_i] = 0, \quad E(u_i u'_j) = \sigma^2 \delta_{ij}. \]
(A15)

The model considered in the text is of this form. Defining \( n^2 \)-dimensional vectors \( u \) and \( v \) by
\[ u = \begin{bmatrix} u_1 \\ u_{n^2} \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_{n^2} \end{bmatrix} \]
(A16)
and using Kronecker products \([7]\), (A14) can be written as
\[ [P \otimes I + \alpha I \otimes P]x = \beta u. \]
(A17)

Using (4) we have
\[ T \otimes T[\Lambda \otimes I + \alpha I \otimes \Lambda]T' \otimes T' x = \beta u. \]
(A18)

Hence
\[ x = T \otimes T \Psi T' \otimes T' \beta u, \]
(A19)
where
\[ \Psi = [I \otimes \Lambda + \alpha \Lambda \otimes I]^{-1}. \]
(A20)

and \( \Psi \) is diagonal with its elements given by the \( n^2 \) numbers
\[ 2 \left[ \cos \frac{i \pi}{n+1} + \alpha \cos \frac{j \pi}{n+1} \right]^{1 \over 2}, \quad i, j = 1, \ldots, n. \]
(A21)

Taking the autocorrelation of \( x \), we find
\[ E[xX'] = \beta \sigma^2 T \otimes T \Psi T' \otimes T'. \]
(A22)

From the definition of \( \Psi \) in (A21), it can be seen that \( \Psi \) cannot be factored into a product of terms in \( i \) and terms in \( j \). Therefore (A22) is nonseparable.

REFERENCES