International Journal of Computer and Information Sciences, Vol. 3, No. 4, 1974

Time Series Model for Texture Synthesis

Bruce H. McCormick[1] and Sadali N. Jayaramamurthy[1]

Received March 1974

A general method is proposed for the synthesis of texture. It is based on a model which treats the pixels (picture elements) of a digitized textural scene as a two-way seasonal time series. This method possesses the desirable characteristic that the parameters needed for synthesis are derived directly from the analysis of the "parent" texture (texture to be imitated). With the help of well-developed methods in the time series analysis the process that generates the pixels of the parent texture is identified. From a set of boundary conditions the future values of the time series are generated which in essence are the pixels of the synthesised texture. The effectiveness of this method is illustrated with examples.

1. INTRODUCTION

1.1. Texture Synthesis: Its Importance

The synthesis of natural looking textures deserves considerably more attention than it has received. General methods are needed by which we can generate a scene that bears an acceptable resemblance to the texture to be imitated ("parent" texture). It is highly desirable that the parameters required for the synthesis are derived automatically from the analysis of the "parent" texture.

Texture synthesis procedures of this type can be evolved to solve the ever-pressing problem of image storage. For it requires considerably less storage to save a few parameters and boundary conditions than to store a complete digitized scene. This effectively means redundancy in the image has been reduced, leading to a more efficient analysis of the resultant scene. Using

[1] Department of Information Engineering, University of Illinois at Chicago Circle, Chicago, Illinois.

the values of the texture parameters as features for discrimination/recognition is also suggested. By specifying intervals on values of each parameter, we can potentially generate a family of textures.

1.2. Seasonal Time Series Model for Texture

Many authors have suggested methods for texture synthesis[6–8,10,13,14,16]. Among them only Rosenfeld *et al.*[16] and Conroy[6] attempt to synthesize natural looking textures, while others merely generate scenes with prespecified statistical properties to be used in their experiments concerning the visual perception of texture. However, none of these methods possesses a desirable quality, namely the choice of parameters needed for synthesis be based on the analysis of "parent" texture. We suggest a scheme for the synthesis of texture that has this property. This scheme is based on a model that views the pixels of a digitized two-dimensional textural scene as a two-way seasonal time series.

Any digitized visual scene can be viewed as a two-way time series. Bartels and Wied[2] were among the first to treat a textural scene as ordinary two-way time series. However, the orderly repetitiveness of a subpattern which is an essential characteristic of texture strongly hints at the possible representation of a digitized textural scene by *seasonal time series*. The choice seems quite natural and very appropriate because there is a striking correspondence between some of the problems that occur in textures and those that are considered in seasonal time series analysis. For example, in TV scan of a texture the repeating subpattern is not necessarily identical from line to line though it retains similar characteristics. In the time series analysis the minor variations of the function from one season to another are accounted for by the assumption of the presence of white noise. Well-developed methods are available to estimate the values of the parameters $\{\mu, \sigma^2\}$ of this noise.

It is possible that there can be more than one periodicity present in the textural scene, such as the subpatterns consisting of subsubpatterns and so on. This may be treated as multiple seasonality. In the case of statistical textures these latter periodicities may be absent and they may be treated as an ordinary time series with no seasonal effect.

Spectral analysis, in the frequency domain, comprises one class of techniques for time series analysis. Our interest here is to stay in the time domain and build stochastic models for time series. This way we can gain a better insight into the nature of the system that generates the time series and can better observe regional boundaries, which of course are observed solely in the time domain. The models developed then can be used to obtain forecasts of the future values of the time series. Our objective, from the engineering viewpoint, is to obtain models that possess maximum simplicity and the

minimum number of parameters consonant with representational adequacy. Precisely the same approach is taken by Box and Jenkins.[4]

1.3. Stochastic Models for Time Series

Let us introduce some notation for convenience in representation. Let

$$\cdots Z_{t-2}, Z_{t-1}, Z_t, Z_{t+1}, \ldots$$

be the time series which can be denoted as $[Z_t]$.

A series of values a_t (*shocks*) is assumed to be generated from a white noise process with mean zero and variance σ_a^2.

B is the *backward shift operator* such that $BZ_t = Z_{t-1}$; hence $B^m Z_t = Z_{t-m}$.

∇ is the *backward difference operator* such that

$$\nabla Z_t = Z_t - Z_{t-1} = (1 - B)Z_t, \qquad \nabla^m Z_t = (1 - B)^m Z_t$$

1.3.1. *Nonseasonal Time Series*

Box and Jenkins[4] represent the process that generates the nonseasonal time series $[Z_t]$ by the following model:

$$\tilde{\Phi}_p(B)\,\nabla^d Z_t = \tilde{\theta}_q(B) a_t \tag{1}$$

where $\tilde{\Phi}_p(B)$ and $\tilde{\theta}_q(B)$ are polynomials in B of order p and q and are known as the *autoregressive* (*AR*) *operator* of order p and the *moving average* (*MA*) *operator* of order q, respectively. The process is known as ARIMA (autoregressive integrated moving average) process of order (p, d, q). This model is sufficiently powerful to represent time series which show both stationary[2] ($d = 0$) and nonstationary[3] behavior.

There are $p + q + 2$ unknown parameters to be estimated from the data.

It can be seen that the ARIMA process can be generated from white noise a_t by means of three filtering operations as shown in Fig. 1.

[2] The process that generates "stationary" time series is assumed to be in equilibrium about a constant mean level.

[3] Only homogeneous nonstationary behavior which calls for the dth difference of the series to be stationary has been considered.

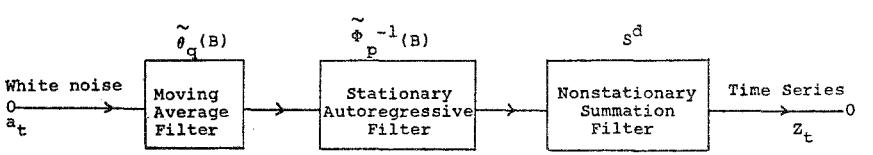

Fig. 1. Block diagram for ARIMA process.

1.3.2. *Seasonal Time Series*

In the same manner if $[Z_t]$ shows seasonal behavior (with period s), it can be represented by the following multiplicative model[4]:

$$\tilde{\Phi}_p(B)\,\tilde{\Phi}_P(B^s)\,\nabla^d\nabla_s{}^D Z_t = \tilde{\theta}_q(B)\,\tilde{\tilde{\theta}}_Q(B^s)a_t \tag{2}$$

where $\tilde{\Phi}_P(B^s)$ and $\tilde{\tilde{\theta}}_Q(B^s)$ are polynomials in B^s of order P and Q, respectively and ∇_s is the seasonal backward difference operator $(1 - B^s)$.

This modified version of the ARIMA process is said to be of the order $(p, d, q) \times (P, D, Q)_s$. This multiplicative model is very useful in that it can be easily extended to take care of multiple seasonalities.

Box and Jenkins[4] and Bacon[1] give detailed procedures for identifying, fitting, and checking the adequacy of the fit of the appropriate model for the given time series. The programs developed by Bacon implementing these procedures have been modified to suit our needs and are described in detail in Refs. 9 and 12. These procedures will become clear when we make a detailed case study with an example in the next section.

1.4. Forecasting

One of the primary objectives in building stochastic models for discrete time series that occur in many practical cases is to utilize their ability to forecast the future values of the series.

Let $Z_t(l)$, $l = 1, 2,...,$ be the function that provides the forecasts at origin t for all future lead times (l). This will be known as the *forecast function* at origin t. For a general ARIMA model (nonseasonal), shown by Eq. (1), the forecast function $\hat{Z}_t(l)$ may be expressed directly in terms of the difference equation by

$$\begin{aligned}\hat{Z}_t(l) = Z_{t+l} = \Phi_1 Z_{t+l-1} + \cdots + \Phi_{p+d} Z_{t+l-p-d} \\ - \theta_1 a_{t+l-1} - \cdots - \theta_q a_{t+l-q} + a_{t+l}\end{aligned} \tag{3}$$

Box and Jenkins[4] show that the minimum mean square error forecast $Z_t(l)$ $(l > 0)$ of Z_{t+l} is the conditional expectation

$$\hat{Z}_t(l) = [Z_{t+l}] = E[Z_{t+l} \mid Z_t, Z_{t-1},...]$$

The conditional expectations of the terms in Eq. (3) are evaluated by inserting actual Z's for future values, actual a's when these are known, and zeros for future a's (because $E[a_{t+j}] = 0$ for $j > 0$).

This forecast function is highly useful when the future values are needed for very short lead times as required in many business and industrial applications. For longer lead times the forecast error will be cumulatively larger.

Also, as the future values of a's are replaced by zeros the eventual forecast function takes on a deterministic mathematical form as dictated by the solution of the homogeneous difference equation containing autoregressive terms only.

Our intention of fitting a time series model for the pixels from a textural scene is to be able to forecast the series for longer lead times and thus effect the texture synthesis from a set of boundary conditions. The forecast function will be unsuitable for this purpose for the reasons mentioned above. In particular, the eventual disappearance of the stochastic effect is highly detrimental for the textural property of the scene.

1.5. Generation of Time Series

As shown in Fig. 2, the general ARIMA process can be generated from a white noise process with appropriate filtering operations. Figure 2 shows the rearrangement of the filters in Fig. 1 for the generation of time series.

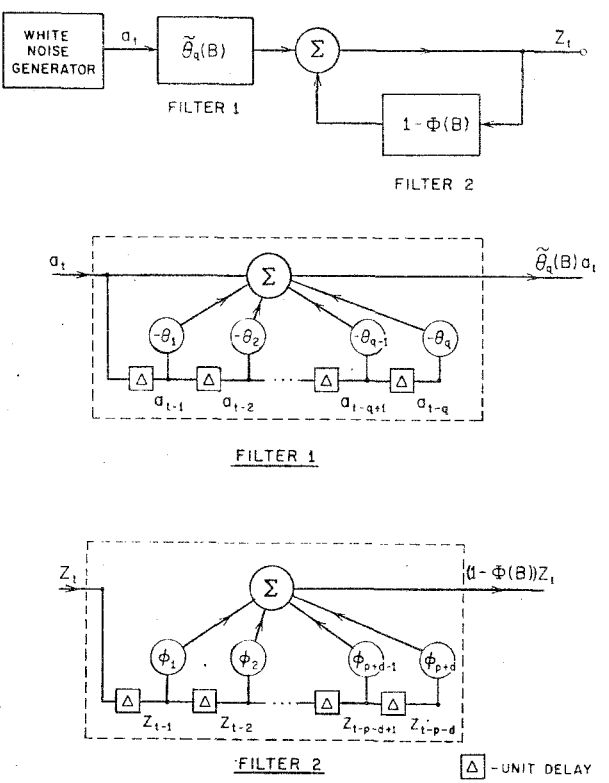

Fig. 2. Generation of time series $[Z_t]$.

Here $\Phi(B)Z_t = \tilde{\Phi}_p(B)\,\nabla^d Z_t$ and a_t is the output of the white noise generator. The present scheme requires $p + q + d$ delays and registers to store the previous occurrences of Z's and a's which are required in the calculation of the present value. The multiplicative constants (Φ's and θ's) are the parameters of the model. To start with, the registers are loaded with the "initial" conditions and the values of Z_{t+l} $(l > 0)$ are successively regenerated.

The above discussion can be extended to seasonal time series.

1.6. Synthesis of 2D Texture

So far we have considered one-dimensional time series. By concatenating either successive rows or successive columns, a two-way series can be treated as a one-way time series. Of course, by doing this we are introducing one more (known) periodicity.

The pixels from a digitized textural scene to be imitated ("parent" texture) are row- (or column-) concatenated to form a one-way time series $[Z_t]$. With the help of programs USID[4] and LIKESURF,[5] an appropriate model is fitted and the values of the parameters are estimated. The generation process shown in Fig. 2 is simulated in GENTEX,[6] which synthesizes the textural scene from a set of boundary conditions. This method is illustrated with an example in the next section.

1.7. Redundancy Reduction

The digitized TV scan of a picture is a row-concatenated one-way time series. When an appropriate model is fitted for this series we come up with an attractive scheme for redundancy reduction in the transmission of the TV scan of the picture. In the literature[17] we find a method known by the name "optimum linear predictor" in which the next value in the series is predicted by expressing it as a linear function of the previous occurrences:

$$\hat{Y}_t = \sum_{i=1}^{k} \theta_i Y_{t-i}\,, \qquad e_t = Y_t - \hat{Y}_t$$

[4] Program USID: Univariate Stochastic Model Identification. This program inputs a time series $[Z_t]$ and plots the autocorrelation function (acf) r_k and the partial autocorrelation function (pacf) Φ_{kk} which help in the identification of the series. For details see Refs. 1, 4, and 12.

[5] Program LIKESURF: This program plots the likelihood surface over the specified parameter space and outputs the maximum likelihood estimates of parameters for the selected model for a given time series. See Refs. 1, 4, and 12.

[6] Program GENTEX: This program synthesizes the textural scene by generating the pixels for a set of boundary conditions from a given time series model. Using PAX[11] subroutines the output scene is printed as a gray-level picture. See Ref. 12.

where $\hat{Y}_t$ is the predicted value, Y_t is the actual value, and e_t is the error in prediction; θ's are selected to minimize the mean square error $[\sigma^2(e_t)]$. Data compression (redundancy reduction) is achieved by transmitting the error values only. At the receiving end the actual values (y's) can be reconstructed from the "errors" (e's) once the system has been properly initialized.

In the present scheme derived from the time series model a one-step-ahead forecast is made by expressing the next value as a linear function of not only the previous values in the series, but also the previous "errors":

$$\hat{Z}_t = \underbrace{\sum_{i=1}^{p+d} \Phi_i Z_{t-i}}_{\text{(autoregressive terms)}} - \underbrace{\sum_{i=1}^{q} \theta_i a_{t-i}}_{\text{(moving average terms)}}$$

$$a_t = [Z_t - \hat{Z}_t]$$

where $\hat{Z}_t$ is the one-step-ahead forecast of Z_t and a_t is the "residual" or "error." The values of Φ's and θ's are selected to minimize the mean square error $[\sigma_a^2]$. We can see that this is an improved version of the "optimum linear predictor" method for achieving redundancy reduction in the transmission of a TV scan of a picture. At the receiving end we can perform a similar "filtering" operation as shown in Fig. 2 to reconstruct Z's from a's, which are transmitted.

2. SYNTHESIS OF TEXTURE: A CASE STUDY

2.1. Selection of an Appropriate Model

The textural scenes that are used in the following examples are taken from Brodatz.[5] A digitized scene of cheesecloth texture (D-105, Brodatz[5]) with 16 gray levels is considered as the "parent texture" (Fig. 3) in Example 1. For the sake of analysis, pixels from a 32×16 window are treated as a two-way input time series. It is row-concatenated ($s = 32$) and is given as an input one-way time series ($[Z_t]$, $t = 1{,}512$) to the program USID. The output of this program consists of the estimated autocorrelations for $[W_t]$ where $W_t = \nabla_1^d \nabla_s^D Z_t$:

(a) For the original series, Z_t, i.e., $d = 0$, $D = 0$; [ACF-00].
(b) For the series differenced once with respect to basic interval only, $\nabla_1 Z_t$, i.e., $d = 1$, $D = 0$; [ACF-10].
(c) For the series differenced once with respect to seasonal interval only, $\nabla_s Z_t$, i.e., $d = 0$, $D = 1$; [ACF-01].
(d) For the series differenced once with respect to both basic and seasonal intervals, $\nabla_1 \nabla_s Z_t$, i.e., $d = 1$, $D = 1$; [ACF-11].

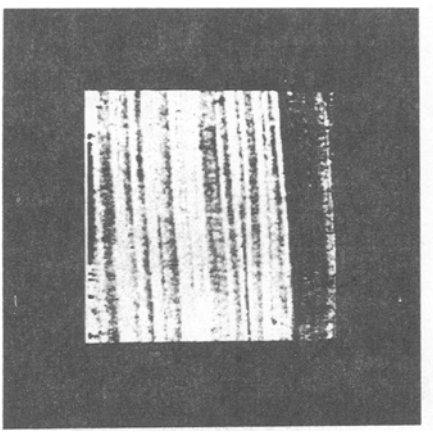

Fig. 3. Cheesecloth texture.

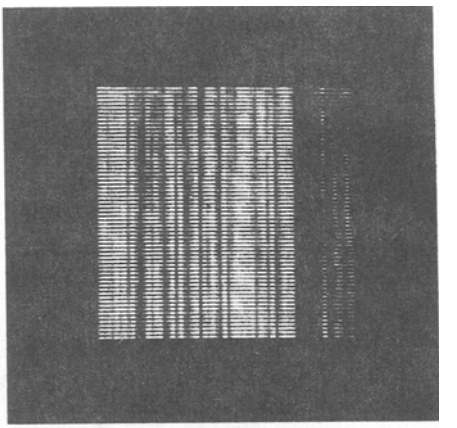

Fig. 4. Synthesized texture (see Section 2.4).

Figure 5 shows the plots of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for all the four cases mentioned above. The partial autocorrelation function is more useful in the identification of the nonseasonal models and hence is not further discussed.

The autocorrelation values in case (a) are large and highly periodic, as might be expected. In case (b) it is seen that differencing with respect to the basic interval reduces the correlation in general while a heavy periodic component remains. We obtain stationarity by differencing with respect to the seasonal interval, as seen in cases (c) and (d), where the correlation values diminish very rapidly. The values of d and D which produce stationarity in the present case are zero and one respectively.

A prospective model is selected for further analysis based on the information furnished by the autocorrelation function by referring to Table 3.2 of Ref. 1 and Appendix A9.1 of Ref. 4. In practice we can pick up many tentative models and even probably overparameterize them. The program LIKESURF is used to estimate the parameters of a given model from analyzing the input data. This program is capable of checking many models at a time and its output indicates the adequacy of any model as well as any redundancy in the choice of number of parameters.

By the procedure mentioned above, we select the following model to represent the given data:

$$(1 - \Phi B^s)W_t = (1 - \lambda_1 B)(1 - \lambda_s B^s)a_t \tag{4}$$

where $W_t = \nabla_s Z_t$ and s is the length of season (length of row).

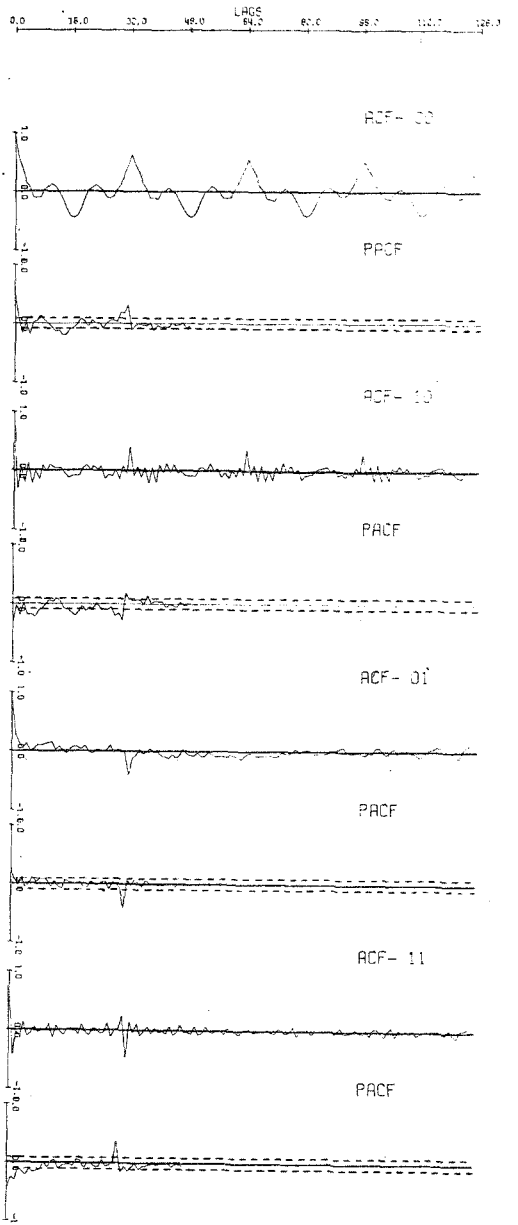

Fig. 5. Serial correlations in Example 1.

2.2. Estimation of the Parameters

We need to estimate the values of four parameters (Φ, λ_1, λ_s, σ_a) by analyzing the input data,

$$Z_t = (1 + \Phi)Z_{t-s} - \Phi Z_{t-2s} + a_t - \lambda_1 a_{t-1} - \lambda_s a_{t-s} + \lambda_1\lambda_s a_{t-s-1} \quad (5)$$

With the knowledge of a_t, a_{t-1}, a_{t-2}, etc., we can make a one-step-ahead forecast[7]:

$$\hat{Z}_{t+1} = (1 + \Phi)Z_{t+1-s} - \Phi Z_{t+1-2s} - \lambda_1 a_t - \lambda_s a_{t+1-s} + \lambda_1\lambda_s a_{t-s} \quad (6)$$

The "residual" $a_{t+1} = Z_{t+1} - \hat{Z}_{t+1}$, i.e., the residuals are considered on the one-step-ahead forecast errors. With appropriate starting values we can see that the values of a_t can be recursively calculated for a given set of parameters.

2.2.1. *A Method to Obtain Starting Values for Recursive Calculations of Residuals*

A procedure to obtain the starting values is described in detail and illustrated with an example in Chapter 9 of the book by Box and Jenkins.[4] It is as follows: For a given set of parameters the series may be forecast backward starting somewhere in the middle of the series. Initially the unknown a's are assumed to be zero. This introduces transients which will hopefully die down by the time Z_0, Z_{-1}, Z_{-2}, etc. are estimated, provided the starting point is chosen sufficiently far along the series. The backforecast-continues till we obtain sufficient number of terms to be able to forward-forecast the first value (Z_1):

$$\hat{Z}_1 = (1 + \Phi)Z_{1-s} - \Phi Z_{1-2s}$$

Here we need the values till Z_{1-2s} and the values of a_t for $t < 1$ are assumed to be zero.

Now we can calculate the first of the residuals:

$$a_1 = Z_1 - \hat{Z}_1$$

2.2.2. *Sum of Squares Function*

From now on we can recursively estimate the values of residuals a_1, a_2,..., a_n. We define a sum of squares function S as

$$S(\Phi, \lambda_1, \lambda_s) = \sum_{t=1}^{n} a_t^2$$

[7] While forecasting, the unknown values of a_t are replaced by their expected value. In this case $E(a_{t+1}) = 0$.

We obtain the least squares estimate of the parameters by picking a set of values $(\hat{\Phi}, \hat{\lambda}_1, \hat{\lambda}_s)$ which provides the minimum sum of squares $S_{\min}$. It is shown in Refs. 1 and 4 that likelihood estimates are the same as the least squares estimates if we assume that the a's are normally distributed. Under the assumption program LIKESURF actually plots the sum of squares function and obtains a set of values that make it minimal.

Using the program LIKESURF, the following model is fitted to Example 1:

$$(1 + 0.15B^s)\,\nabla_s Z_t = (1 - 0.25B)(1 - 0.5B^s)a_t$$

where $\sigma_a^2 = 1.03$.

2.3. Checking the Adequacy of the Fit

According to the model, the residuals a_t are generated by a white noise process. If the fit were to be adequate, the calculated residuals should be uncorrelated. Figure 6 shows the correlation values of the residuals. It can be seen that the values are fairly small and there is no periodic component either. Thus, it offers no significant departures from randomness among residuals, confirming the adequacy of the fit.

As mentioned in the previous section, we can also check the adequacy by comparing the actual values to the forecast values from any origin. Forecasting is done on the same principle as before (i.e., using the difference

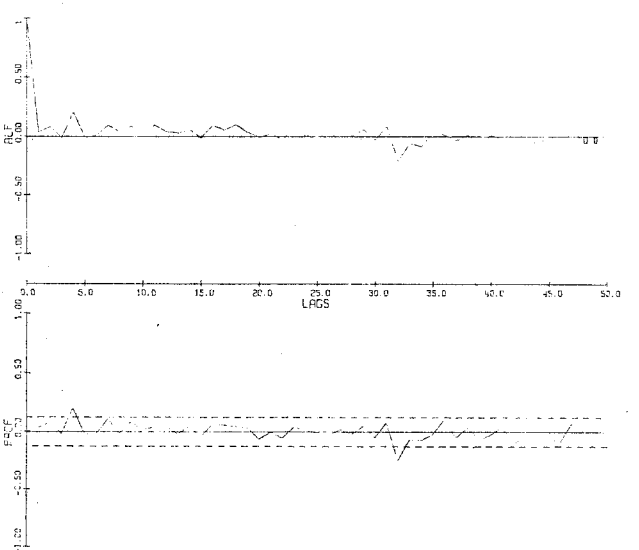

Fig. 6. Serial correlations of residuals in Example 1.

equation approach). The values of a_t for $t > t_{\text{origin}}$ are set to zero. Figure 7 shows the actual values (solid line) and forecast values (crosses) from the origin $t = 128$. Considering the limited number of parameters steering the model, the forecast function follows actual values rather closely even for $t \gg t_{\text{origin}}$.

2.4. Generation of Texture

We can create a white noise generator having a mean zero and the variance σ_a^2 which generates the residuals a_t. To start the recursive procedure for the generation of future values, we need at least $2s$ values of Z_t. These can be considered as the boundary conditions. Figure 4 shows the texture synthesized in this manner. Here, using two rows of length 64 ($s = 64$) from the original scene as the starting values, the future values of the pixels have been generated by the procedure mentioned above.

One of the ways of testing the effectiveness of the scheme of synthesis is by attempting to fill the holes in the "parent" texture. The results are shown in Figs. 8a and 8b. We notice some edge effects here. This can be expected in the case of structural textures. It is possible to minimize these effects. In the present case we have used pixels from two rows parallel to the top edge of the hole as the starting values. Instead we can incorporate pixels from a couple of rows and columns which are parallel to *each* edge of the hole as boundary conditions. Then we can proceed to patch the hole by *forward* forecasting from the top edge and backward forecasting from the bottom edge [as suggested in Eq. (5)]. In the center we can average out the values forecasted from either side, thus reducing the edge effects.

Let us consider a different class of textures in the second example. The texture of *handmade paper* (Fig. 9a) belongs to general class of statistical

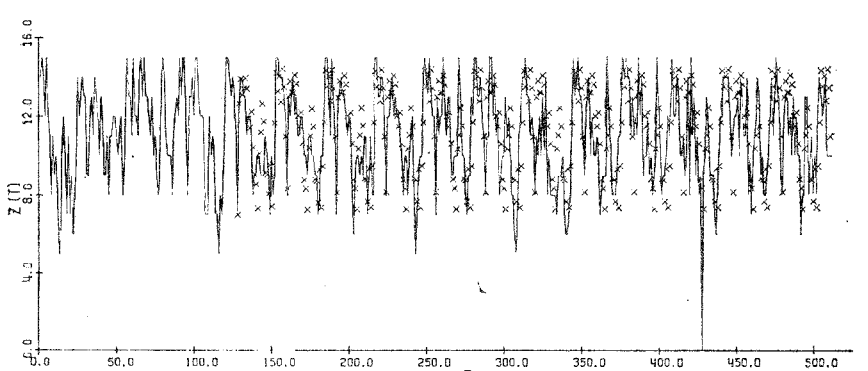

Fig. 7. Forecast values (shown by ×'s) in Example 1.

Fig. 8a. Holes in cheesecloth texture.

Fig. 8b. Resultant scene after filling the hole

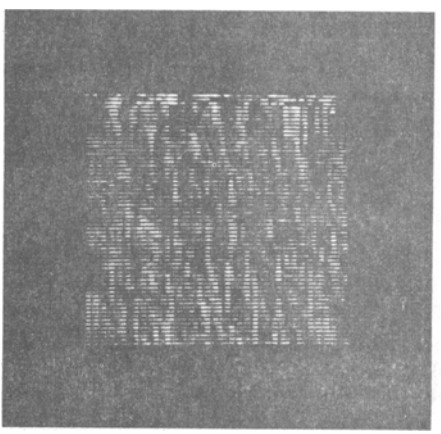

Fig. 9a. Texture of a handmade paper.

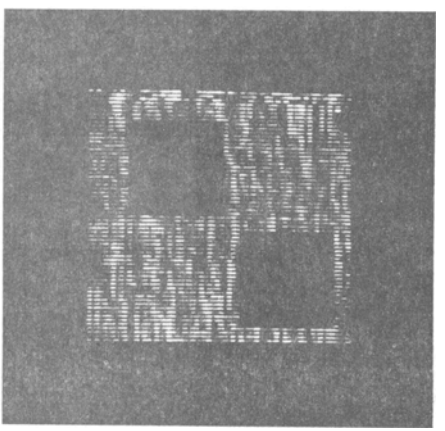

Fig. 9b. Holes.

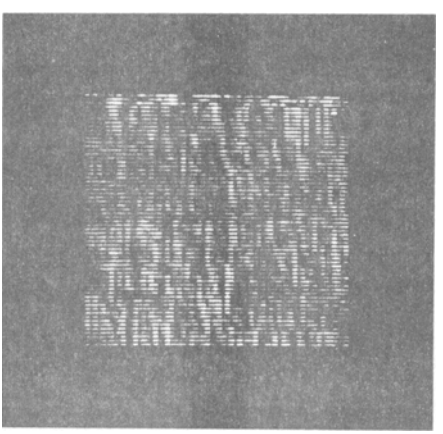

Fig. 9c. Resultant scene after filling the holes.

textures. It has little global structure. The seasonal time series model fitted for this texture is of the form

$$(1 - 0.8B^s)Z_t = (1 + 0.25B)(1 - 0.5B^s)a_t$$

where $\sigma_a^2 = 3.27$. Figures 9b and 9c show the results of hole-filling operation using the above model. Here the edge effects are hardly noticeable.

3. CONCLUSION

We have shown that it is possible to synthesize natural-looking textures using the seasonal time series model. This method is particularly suitable in the case of statistical textures. Many biomedical textures of interest are statistical in nature, as are many textures occurring in remote sensing of the environment. It can be anticipated that the parameterization of texture proposed here will be of value in monitoring the dynamics of change with time of such textured areas.

REFERENCES

1. D. W. Bacon, "Seasonal time series," Ph.D. Thesis, University of Wisconsin, Madison (1965).
2. Peter Bartels and G. L. Wied, "Tumor cell diagnosis based on stochastic properties of digitized images," in *Proc. 2-D Dig. Signal Proc. Conf.* (Columbia, Missouri, 1971).
3. P. K. Bhattacharya, "Order of dependence in a stationary normally distributed two-way series," *Ann. Math. Statist.* **43**:1792–1807 (1972).
4. J. E. P. Box and G. M. Jenkins, *Time Series Analysis* (Holden-Day, 1970).
5. P. Brodatz, *Textures* (Dover, New York, 1966).
6. P. Conroy, "Simulation of texture by computer graphics," TR-14, Department of Computer Science, University of Toronto (1969).
7. J. J. Gibson, "The reproduction of visually perceived forms," *J. Exp. Psychol.* **12**:1–39 (1929).
8. J. J. Gibson, "The perception of visual surfaces," *Am. J. Psychol.* **63**:367–84 (1950).
9. S. N. Jayaramamurthy, "Computer methods for analysis and synthesis of visual texture" (Ph.D. Thesis) DCS Report 601, Department of Computer Science, University of Illinois, Urbana (October 1973).
10. B. Julesz, "Texture and visual perception," *Sci. Am.* **1965** (February):38–48.
11. L. E. Lipkin, W. C. Watt, and R. A. Kirsch, "The analysis and description of biological Images," *Ann. N. Y. Acad. Sci.* **128**:984–1012 (1966).
12. B. H. McCormick and S. N. Jayarmamurthy, "Synthesis of texture," DCS Report 566, Department of Computer Science, University of Illinois, Urbana (July 1973).
13. R. M. Pickett, "Response latency in a pattern perception situation," *Acta Psychologica* **27**:160–69 (1967).
14. R. M. Pickett, "The perception of random visual texture," in *Models for the Perception of Speech and Visual Form*, W. Walter Dunn, ed. (MIT Press, Cambridge, Massachusetts, 1967).

15. E. A. Robinson, *Multichal Time Series Analysis with Digital Computer Programs* (Holden-Day, San Francisco, 1967).
16. A. Rosenfeld and B. S. Lipkin, "Texture synthesis," in *Picture Processing and Psychopictorics*, Lipkin and Rosenfeld, ed. (Academic Press, New York, 1970).
17. Special Issue on "Redundancy reduction," *Proc. IEEE* **1967** (March).