UIUCDCS-R-73-566

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SYNTHESIS OF TEXTURE*

by

B. H. McCormick
S. N. Jayaramamurthy

July 1973

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

* Supported in part by contract US AEC AT(11-1)2118.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION.	1
1.1 <u>Texture Synthesis: Its importance</u>	1
1.2 <u>Texture Synthesis: Earlier work</u>	2
1.3 <u>Seasonal Time Series Model</u>	2
2. SEASONAL TIME SERIES MODEL FOR TEXTURE.	4
2.1 <u>Stationary Models</u>	5
2.1.1 <u>Auto Regressive Models</u>	6
2.1.2 <u>Moving Average Models</u>	6
2.1.3 <u>Mixed Models</u>	6
2.2 <u>Nonstationary Models</u>	6
2.3 <u>Seasonal Model</u>	7
2.4 <u>Identification and Fitting of Models</u>	8
2.5 <u>Forecasting</u>	9
2.5.1 <u>Generation of Time Series</u>	10
2.6 <u>Synthesis of 2-D Texture</u>	12
2.7 <u>Redundancy Reduction</u>	12
3. SYNTHESIS OF TEXTURE: A CASE STUDY	14
3.1 <u>Selection of an Appropriate Model</u>	14
3.2 <u>Estimation of the Parameters</u>	15
3.2.1 <u>A Method to Obtain Starting Values for</u> <u>Recursive Calculations of Residuals</u>	20
3.3 <u>Checking the Adequacy of the Fit</u>	21
3.4 <u>Generation of Texture</u>	22
3.5 <u>Conclusion</u>	23
APPENDIX A. Description of Computer Programs	26
<u>Program 1</u>	26
<u>Program 2</u>	29
<u>Program 3</u>	33
LIST OF REFERENCES.	35

ACKNOWLEDGMENT

The authors wish to thank the entire ILLIAC III staff, especially Mr. Richard Borovec and Mr. Joe Wenta, for their invaluable help through many phases of our work. We also thank Mrs. Barb Armstrong and Mrs. Evelyn Huxhold for an excellent typing job, Mr. Stan Zundo for the drafting, and Mr. Dennis Reed for the offset printing.

1. INTRODUCTION

Visual texture is known to play an important role in the field of pattern recognition. There are three problems which are under study in the context of automated image processing, viz. texture discrimination/recognition, texture analysis and texture synthesis. In the literature many methods are proposed to deal with the above problems [7, 10, 15, 17, 21, 25, 26, 27, 28]. Most of these, however, have little or nothing in common and lack generality.

In this paper we intend to propose a general method for the synthesis of texture. This method is based on a mathematical model which treats pixels (picture elements) of a digitized textural scene as a two-way seasonal time-series.

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 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 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 Texture Synthesis: Earlier work

We find that many authors proposed methods [7, 8, 13, 20, 26] that are based on the subpattern/placement rule model for texture. Among them only Rosenfeld, et.al [26] and Conroy [7] attempt to synthesize natural-looking textures. While others merely generate scenes with pre-specified statistical properties to be used in their experiments concerning the visual perception of texture. In their papers, Rosenfeld and Conroy demonstrate that a rich variety of textures can be generated using only simple subpatterns and placement rules. In these methods the success of synthesizing a natural-looking texture depends strongly on the choice of the subpatterns and the appropriate placement rules. These important choices normally are not based on the analysis of the 'parent' texture, instead involve from human judgment on a trial and error basis. This is the main drawback of these schemes based on the subpattern/placement rule model. It is difficult if not an impossible task even to extract the subpatterns (whose size and shape are unknown) from the analysis of any textural scene, not to speak of decoding the rules of their placement.

1.3 Seasonal Time Series Model

The subpattern/placement rule model, though easy to visualize, creates many practical difficulties. Here we adopt a different approach: Our scheme for texture synthesis is based on a model that views the pixels of a digitized two-dimensional textural scene as a two-way time series. We will then resort to the well-developed methods of the time series analysis. By analyzing the 'parent' texture we identify the process that generates the time series, and estimate its parameters. Then with proper initial conditions we can generate the time series which in essence resemble the pixels of the synthesized textural scene.

In the subsequent chapters the time series strategy is discussed in detail. The effectiveness of this method is illustrated by examples.

2. SEASONAL TIME SERIES MODEL FOR TEXTURE

Any digitized visual scene can be viewed as a two-way time series. The orderly repetitiveness of a subpattern which is an essential characteristic of texture 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 is 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 sub-subpatterns 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 the models can be used to obtain forecasts of the future values of the time series. Our objective, from the engineering view point is to obtain models that possess maximum simplicity and the minimum no. of parameters

consonant with representational adequacy. Precisely the same approach is taken by Box and Jenkins while dealing with this subject in their book "Time Series Analysis - Forecasting and Control" (Holden-Day, 1970).

In their book they discuss various stochastic models for time series. They also present methods for identifying, fitting and checking the models for a given time series. They also extend their methods to deal with seasonal time series. We have presently implemented these methods in our texture analysis programs. In order to preserve continuity we will briefly review some of the models and methods, while omitting much of the detail which can be obtained from their book.

2.1 Stationary Models

In these models the process that generates the time series is assumed to be in equilibrium about a constant mean level.

Let us introduce some notation and operators that are used extensively. Let

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

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

A series of values of a_t ("shocks") are assumed to be generated from a white noise process with mean '0' and variance σ_a^2 .

'B' is the "backward shift" operator such that

$$B Z_t = Z_{t-1}; \text{ hence } B^m Z_t = Z_{t-m}$$

' ∇ ' is the "backward difference" operator

$$\begin{aligned} \nabla Z_t &= Z_t - Z_{t-1} \\ &= (1-B) Z_t. \end{aligned}$$

2.1.1 Auto Regressive Models

In this model, the current value of the process is expressed as a finite, linear aggregate of previous values of the process and a 'shock' a_t

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t$$

This is called an 'autoregressive (AR) process' of order 'p'. Let

$$\tilde{\phi}_p(B) = (1 - \phi_1 B - \phi_2 B^2 \dots - \phi_p B^p)$$

Then

$$\tilde{\phi}_p(B) Z_t = a_t$$

$\tilde{\phi}_p(B)$ is known as 'autoregressive operator' of order p. There are p+1 unknown parameters, $\phi_1, \phi_2, \phi_3 \dots \phi_p, \sigma_a^2$ which in practice have to be estimated from the data.

2.1.2 Moving Average Models

In this model the present value Z_t is assumed to be linearly dependant on 'q' of previous a's.

$$\begin{aligned} Z_t &= a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} \dots - \theta_q a_{t-q} \\ &= \tilde{\theta}_q(B) a_t \end{aligned}$$

$\tilde{\theta}_q(B)$ is the 'moving average (MA) operator' of order q.

2.1.3 Mixed Models

In many practical cases the parsimony in number of operators is achieved by mixing the autoregressive and moving average terms in the same model.

$$\tilde{\phi}_p(B) Z_t = \tilde{\theta}_q(B) a_t$$

2.2 Nonstationary Models

Many series actually encountered in practice are not stationary in nature. But some of them show homogeneous nonstationary behaviour. Box

and Jenkins show that such cases can be represented by a generalized autoregressive operator $\Phi(B)$ in which one or more of the zeroes of the polynomial $\Phi(B)$ is unity, i.e., $\Phi(B) = \tilde{\phi}_p(B) (1-B)^d$, where $\tilde{\phi}_p(B)$ is the stationery autoregressive operator.

Thus a general model which can represent homogeneous nonstationary behaviour, is of the form

$$\tilde{\phi}_p(B) (1-B)^d Z_t = \tilde{\theta}_q(B) a_t$$

$$\tilde{\phi}_p(B) w_t = \tilde{\theta}_q(B) a_t$$

$$\text{where } w_t = \nabla^d Z_t$$

Homogeneous nonstationary behaviour can therefore be represented by a model which calls for the d^{th} difference of the process to be stationary.

The above representation is a powerful model for describing stationary and nonstationary time series and is called an 'autoregressive integrated moving average (ARIMA) process' of order (p, d, q) . There are $(p + q + 2)$ unknown parameters to be estimated from the data.

Figure 2.1. Block Diagram for ARIMA Process

It can be seen that the ARIMA process may be generated from white noise a_t by means of three filtering operations as shown in Figure 2.1.

2.3 Seasonal Model

It is shown^[5] that the time series which show seasonal effect can be represented by a general multiplicative model:

$$\tilde{\phi}_p(B) \tilde{\phi}_P(B^S) \nabla^d \nabla_s^D Z_t = \tilde{\theta}_q(B) \tilde{\theta}_Q(B^S) a_t$$

where s = Seasonal period

$$\nabla_s = (1-B^s)$$

$\tilde{\phi}_p(B^s)$, $\tilde{\theta}_a(B^s)$ are the polynomials in B^s of the order P , Q respectively.

This is a modified version of ARIMA process, 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.

2.4 Identification and Fitting of Models

Identification methods are rough procedures applied to a set of data to indicate the kind of representational model which is worthy of further investigation. By 'fitting' we mean the estimation of the various parameters. The selection of the right kind of model results from the extensive interaction between theory and practice. The necessary theory is developed in Chapters 2-5 and the methods for identifying and fitting models for the actual time series have been explained and illustrated with examples in Chapters 6-9 in the Box and Jenkins book.^[5]

The programs that have implemented these methods have been described in detail in Appendix A.

It can be observed that the extensive use is made of the behaviour of the autocorrelation function (ACF), ϕ_R , and the partial autocorrelation function (PACF), ϕ_{kk} in the process of identification of the proper model. Tables 6.1 (pp. 176)^[5], Appendix A9.1 (pp. 329)^[5] and Table 3.2 (pp. 32)^[2] are used as reference tables. The initial values of the parameters can also be estimated with the help of these tables.

Chapter 7 of the book^[5] describes a procedure to obtain a final set of parameters for a general ARIMA process from the initial values by minimizing the 'sum of squares function' $S(\phi, \theta)$,

where

$$S(\phi, \theta) = \sum_{t=1}^n [a_t | \phi, \theta, w]^2$$

ϕ and θ are vectors of length p and q that represent the coefficient values of the AR and MA operators, respectively, and

$$w_t = \nabla^d z_t$$

2.5 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. These forecasts provide a basis for a) economic and business planning, b) production planning, c) inventory and production control, d) control and optimization of industrial processes.

Let $\hat{z}_t(l)$, $l=1,2,\dots$ 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 (non-seasonal)

$$\phi_p(B) \nabla^d z_t = \theta_q(B) a_t \quad 2.5.1$$

the forecast function $\hat{z}_t(l)$ may be expressed directly in terms of the difference equation by:

$$\begin{aligned} \hat{z}_t(l) = z_{t+l} = & \phi_1 z_{t+l-1} + \dots + \phi_{p+d} z_{t+l-p-d} - \theta_1 a_{t+l-1} \\ & - \dots - \theta_q a_{t+l-q} + a_{t+l} \end{aligned} \quad 2.5.2$$

Box and Jenkins show that the minimum mean square error forecast

$\hat{z}_t(l)$ ($l > 0$) of z_{t+l} is the conditional expectation

$$\hat{z}_t(l) = [z_{t+l}] = E[z_{t+l} | z_t, z_{t-1}, \dots]$$

The conditional expectations of the terms in 2.5.2 are evaluated by inserting actual Z's when these are known, forecasted Z's for future values, actual a's when these are known, and zeroes 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 zeroes, 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. Especially the eventual disappearance of the stochastic effect is highly detrimental for the textural property of the scene.

2.5.1 Generation of Time Series

As shown in Figure 2.1, the general ARIMA process can be generated from a white noise process with appropriate filtering operations. Figure 2.2 shows the rearrangement of the filters in Figure 2.1 for the generation of time series. 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

FILTER 1

FILTER 2

Figure 2.2. Generation of Time Series $[Z_t]$

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.

2.6 Synthesis of 2-D 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*, LIKESURF* an appropriate model is fitted and the values of the parameters are estimated. The generation process shown in Figure 2.2 is simulated in GENTEX* which synthesizes the textural scene from a set of boundary conditions. This method is illustrated with an example in the next chapter.

2.7 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 [30] we find a method known by the name 'optimum linear predictor' in which the

* See Appendix

next value in the series is predicted by expressing it as a linear function of the previous occurrences:

$$\hat{y}_t = \sum_{i=1}^k \theta_i y_{t-i}$$

$$e_t = y_t - \hat{y}_t \quad \text{where } \hat{y}_t = \text{the predicted value}$$

$$y_t = \text{the actual value}$$

$$e_t = \text{the error in prediction}$$

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) from a system that is to be initialized to start with.

In the present scheme that involves 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':

$$\hat{Z}_t = \sum_{i=1}^p \phi_i Z_{t-i} + \sum_{l=1}^q \theta_l a_{t-l}$$

(autoregressive terms) (moving average terms)

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

$$\text{where } \hat{Z}_t = \text{one-step-ahead forecast of } Z_t$$

$$a_t = \text{the 'residual' or 'error'}$$

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 Figure 2.2 to reconstruct Z 's from a 's. Here, of course, 'white noise generator' is replaced by the transmission line.

3. SYNTHESIS OF TEXTURE: A CASE STUDY

3.1 Selection of an Appropriate Model

The textural scenes that are used in the following examples are taken from Brodatz^[6]. A digitized scene of cheesecloth texture (D-105, Brodatz^[6]) with 16 graylevels is considered as the 'parent texture' (Figure 3.1) in example 1. For the sake of analysis, pixels from a 32 x 16 window is 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. $d = 0 \quad D = 0$ [ACF-00]
- (b) for the series differenced once with respect to basic interval only, $\nabla_1 Z_t$
i.e. $d = 1 \quad D = 0$ [ACF-10]
- (c) for the series differenced once with respect to seasonal interval only, $\nabla_s Z_t$
i.e. $d = 0 \quad D = 1$ [ACF-01]
- (d) for the series differenced once with respect to both basic and seasonal intervals, $\nabla_1 \nabla_s Z_t$
i.e. $d = 1 \quad D = 1$ [ACF-11]

Figure 3.3 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

* Appendix A

the identification of the non-seasonal 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 by differencing with respect to the basic interval reducing 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 0 and 1 respectively.

A prospective model is selected for further analysis based on the information furnished by the autocorrelation function by referring to the Table 3.2^[2] and appendix A9.1^[5]. 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:

$$(1 - \phi_1 B^s) W_t = (1 - \lambda_1 B)(1 - \lambda_s B^s) a_t \quad (1)$$

$$\text{where } W_t = \nabla_s Z_t$$

s = length of season (length of row)

3.2 Estimation of the Parameters

We need to estimate the values of four parameters $(\phi_1, \lambda_1, \lambda_s, \sigma_a)$ by analyzing the input data.

* Appendix A

Figure 3.1. Cheesecloth Texture

Figure 3.2. Synthesized Texture

Figure 3.3. Serial Correlations in Example 1

NO. OF DATA USED FOR FITTING= 250

NO. OF PREDICTIONS= 384

CODE 1

PHI=0.150									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	5.6852E 02	5.6058E 02	5.5302E 02	5.4584E 02	5.3903E 02	5.3259E 02	5.2654E 02		
SL1=0.200	5.7093E 02	5.6249E 02	5.5499E 02	5.4819E 02	5.4135E 02	5.3484E 02	5.2870E 02		
SL1=0.150	5.7384E 02	5.6482E 02	5.5611E 02	5.4738E 02	5.3855E 02	5.3004E 02	5.2182E 02		
SL1=0.100	5.8000E 02	5.7112E 02	5.6204E 02	5.5273E 02	5.4326E 02	5.3389E 02	5.2466E 02		
SL1=0.050	5.8866E 02	5.7932E 02	5.6983E 02	5.6047E 02	5.5075E 02	5.4063E 02	5.3012E 02		
SL1=0.000	6.1146E 02	6.0007E 02	5.8780E 02	5.7400E 02	5.5853E 02	5.4133E 02	5.2269E 02		
SL1=0.250	6.2347E 02	6.1254E 02	6.0187E 02	5.9047E 02	5.7831E 02	5.6541E 02	5.5185E 02		
SL1=0.100	6.3562E 02	6.2483E 02	6.1394E 02	6.0243E 02	5.9029E 02	5.7754E 02	5.6417E 02		
PHI=0.100									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	5.8439E 02	5.7601E 02	5.6772E 02	5.5942E 02	5.5110E 02	5.4286E 02	5.3461E 02		
SL1=0.200	5.8713E 02	5.7882E 02	5.7049E 02	5.6213E 02	5.5374E 02	5.4531E 02	5.3684E 02		
SL1=0.150	5.9031E 02	5.8194E 02	5.7353E 02	5.6508E 02	5.5659E 02	5.4806E 02	5.3949E 02		
SL1=0.100	5.9347E 02	5.8503E 02	5.7657E 02	5.6808E 02	5.5956E 02	5.5101E 02	5.4243E 02		
SL1=0.050	6.0314E 02	6.0054E 02	5.9712E 02	5.8867E 02	5.8011E 02	5.7152E 02	5.6290E 02		
SL1=0.000	6.2147E 02	6.2158E 02	6.1294E 02	6.0424E 02	5.9548E 02	5.8674E 02	5.8037E 02		
SL1=0.250	6.2533E 02	6.1714E 02	6.0833E 02	6.0000E 02	5.9148E 02	5.8283E 02	5.7405E 02		
SL1=0.100	6.3747E 02	6.2924E 02	6.2057E 02	6.1235E 02	6.0407E 02	5.9574E 02	5.8735E 02		
PHI=0.050									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	6.0330E 02	5.9436E 02	5.8542E 02	5.7559E 02	5.6715E 02	5.5911E 02	5.5145E 02		
SL1=0.200	6.0642E 02	5.9747E 02	5.8853E 02	5.7870E 02	5.6926E 02	5.6113E 02	5.5344E 02		
SL1=0.150	6.1111E 02	6.0219E 02	5.9327E 02	5.8351E 02	5.7404E 02	5.6600E 02	5.5811E 02		
SL1=0.100	6.2043E 02	6.1050E 02	6.0049E 02	5.9234E 02	5.8344E 02	5.7533E 02	5.6760E 02		
SL1=0.050	6.3374E 02	6.2371E 02	6.1363E 02	6.0472E 02	5.9582E 02	5.8730E 02	5.7917E 02		
SL1=0.000	6.5007E 02	6.4003E 02	6.3013E 02	6.2057E 02	6.1134E 02	6.0241E 02	5.9275E 02		
SL1=0.250	6.7101E 02	6.6030E 02	6.5000E 02	6.4007E 02	6.3058E 02	6.2147E 02	6.1275E 02		
SL1=0.100	6.8303E 02	6.7271E 02	6.6307E 02	6.5368E 02	6.4376E 02	6.3425E 02	6.2514E 02		
PHI=0.000									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	6.2378E 02	6.1328E 02	6.0317E 02	5.9249E 02	5.8435E 02	5.7537E 02	5.6642E 02		
SL1=0.200	6.2701E 02	6.1626E 02	6.0591E 02	5.9546E 02	5.8617E 02	5.7728E 02	5.6849E 02		
SL1=0.150	6.3170E 02	6.2104E 02	6.1090E 02	6.0112E 02	5.9174E 02	5.8236E 02	5.7300E 02		
SL1=0.100	6.3943E 02	6.2877E 02	6.1863E 02	6.0874E 02	5.9897E 02	5.8921E 02	5.7946E 02		
SL1=0.050	6.5007E 02	6.4003E 02	6.3013E 02	6.2057E 02	6.1134E 02	6.0241E 02	5.9275E 02		
SL1=0.000	6.7101E 02	6.6030E 02	6.5000E 02	6.4007E 02	6.3058E 02	6.2147E 02	6.1275E 02		
SL1=0.250	6.7494E 02	6.6428E 02	6.5400E 02	6.4376E 02	6.3352E 02	6.2328E 02	6.1304E 02		
SL1=0.100	6.8696E 02	6.7630E 02	6.6602E 02	6.5578E 02	6.4554E 02	6.3530E 02	6.2506E 02		
SL1=0.050	6.9898E 02	6.8832E 02	6.7804E 02	6.6780E 02	6.5756E 02	6.4732E 02	6.3708E 02		
SL1=0.000	7.1099E 02	7.0033E 02	6.9005E 02	6.7981E 02	6.6957E 02	6.5933E 02	6.4909E 02		
PHI=0.150									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	6.4707E 02	6.3843E 02	6.3108E 02	6.2391E 02	6.1762E 02	6.13628E 02	6.2544E 02		
SL1=0.200	6.4794E 02	6.3939E 02	6.3193E 02	6.2476E 02	6.1847E 02	6.1447E 02	6.2703E 02		
SL1=0.150	6.4881E 02	6.4026E 02	6.3280E 02	6.2563E 02	6.1934E 02	6.1534E 02	6.3258E 02		
SL1=0.100	6.4968E 02	6.4113E 02	6.3367E 02	6.2650E 02	6.2021E 02	6.1621E 02	6.3487E 02		
SL1=0.050	6.5055E 02	6.4200E 02	6.3454E 02	6.2737E 02	6.2108E 02	6.1708E 02	6.3574E 02		
SL1=0.000	6.5142E 02	6.4287E 02	6.3541E 02	6.2824E 02	6.2195E 02	6.1795E 02	6.3661E 02		
SL1=0.250	6.5129E 02	6.4274E 02	6.3528E 02	6.2811E 02	6.2182E 02	6.1782E 02	6.3648E 02		
SL1=0.100	6.5216E 02	6.4361E 02	6.3615E 02	6.2898E 02	6.2269E 02	6.1869E 02	6.3734E 02		
SL1=0.050	6.5303E 02	6.4448E 02	6.3702E 02	6.2985E 02	6.2356E 02	6.1956E 02	6.3800E 02		
SL1=0.000	6.5390E 02	6.4535E 02	6.3789E 02	6.3072E 02	6.2443E 02	6.2043E 02	6.3886E 02		
PHI=0.100									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	6.6707E 02	6.5843E 02	6.5108E 02	6.4391E 02	6.3762E 02	6.33628E 02	6.2544E 02		
SL1=0.200	6.6794E 02	6.5939E 02	6.5193E 02	6.4476E 02	6.3847E 02	6.3447E 02	6.2703E 02		
SL1=0.150	6.6881E 02	6.6026E 02	6.5280E 02	6.4563E 02	6.3934E 02	6.3534E 02	6.3258E 02		
SL1=0.100	6.6968E 02	6.6113E 02	6.5367E 02	6.4650E 02	6.4021E 02	6.3621E 02	6.3487E 02		
SL1=0.050	6.7055E 02	6.6200E 02	6.5454E 02	6.4737E 02	6.4108E 02	6.3708E 02	6.3574E 02		
SL1=0.000	6.7142E 02	6.6287E 02	6.5541E 02	6.4824E 02	6.4195E 02	6.3795E 02	6.3661E 02		
SL1=0.250	6.7129E 02	6.6274E 02	6.5528E 02	6.4811E 02	6.4182E 02	6.3782E 02	6.3648E 02		
SL1=0.100	6.7216E 02	6.6361E 02	6.5615E 02	6.4898E 02	6.4269E 02	6.3869E 02	6.3734E 02		
SL1=0.050	6.7303E 02	6.6448E 02	6.5702E 02	6.4985E 02	6.4356E 02	6.3956E 02	6.3800E 02		
SL1=0.000	6.7390E 02	6.6535E 02	6.5789E 02	6.5072E 02	6.4443E 02	6.4043E 02	6.3886E 02		
PHI=0.050									
LARGE LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	6.8207E 02	6.7343E 02	6.6608E 02	6.5891E 02	6.5262E 02	6.48628E 02	6.4044E 02		
SL1=0.200	6.8294E 02	6.7439E 02	6.6703E 02	6.6086E 02	6.5457E 02	6.5057E 02	6.4313E 02		
SL1=0.150	6.8381E 02	6.7526E 02	6.6790E 02	6.6173E 02	6.5544E 02	6.5144E 02	6.4400E 02		
SL1=0.100	6.8468E 02	6.7613E 02	6.6877E 02	6.6260E 02	6.5631E 02	6.5231E 02	6.4487E 02		
SL1=0.050	6.8555E 02	6.7700E 02	6.6964E 02	6.6347E 02	6.5718E 02	6.5318E 02	6.4574E 02		
SL1=0.000	6.8642E 02	6.7787E 02	6.7051E 02	6.6434E 02	6.5805E 02	6.5405E 02	6.4661E 02		
SL1=0.250	6.8629E 02	6.7774E 02	6.7038E 02	6.6421E 02	6.5792E 02	6.5392E 02	6.4648E 02		
SL1=0.100	6.8716E 02	6.7861E 02	6.7125E 02	6.6508E 02	6.5879E 02	6.5479E 02	6.4735E 02		
SL1=0.050	6.8803E 02	6.7948E 02	6.7212E 02	6.6595E 02	6.5966E 02	6.5566E 02	6.4822E 02		
SL1=0.000	6.8890E 02	6.8035E 02	6.7300E 02	6.6683E 02	6.6054E 02	6.5654E 02	6.4910E 02		
PHI=0.150									
SMALL LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	7.0203E 02	7.0214E 02	7.0225E 02	7.0236E 02	7.0247E 02	7.0258E 02	7.0269E 02		
SL1=0.200	7.0214E 02	7.0225E 02	7.0236E 02	7.0247E 02	7.0258E 02	7.0269E 02	7.0280E 02		
SL1=0.150	7.0225E 02	7.0236E 02	7.0247E 02	7.0258E 02	7.0269E 02	7.0280E 02	7.0291E 02		
SL1=0.100	7.0236E 02	7.0247E 02	7.0258E 02	7.0269E 02	7.0280E 02	7.0291E 02	7.0302E 02		
SL1=0.050	7.0247E 02	7.0258E 02	7.0269E 02	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02		
SL1=0.000	7.0258E 02	7.0269E 02	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02		
SL1=0.250	7.0269E 02	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02		
SL1=0.100	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02		
SL1=0.050	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02		
SL1=0.000	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02		
PHI=0.100									
SMALL LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	7.0269E 02	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02		
SL1=0.200	7.0280E 02	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02		
SL1=0.150	7.0291E 02	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02		
SL1=0.100	7.0302E 02	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02		
SL1=0.050	7.0313E 02	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02	7.0379E 02		
SL1=0.000	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02	7.0379E 02	7.0390E 02		
SL1=0.250	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02	7.0379E 02	7.0390E 02	7.0401E 02		
SL1=0.100	7.0346E 02	7.0357E 02	7.0368E 02	7.0379E 02	7.0390E 02	7.0401E 02	7.0412E 02		
SL1=0.050	7.0357E 02	7.0368E 02	7.0379E 02	7.0390E 02	7.0401E 02	7.0412E 02	7.0423E 02		
SL1=0.000	7.0368E 02	7.0379E 02	7.0390E 02	7.0401E 02	7.0412E 02	7.0423E 02	7.0434E 02		
PHI=0.050									
SMALL LAMBDA 1									
	0.350	0.375	0.400	0.425	0.450	0.475	0.500		
SL1=0.250	7.0324E 02	7.0335E 02	7.0346E 02	7.0357E 02	7.0368E 02	7.0379E 02	7.0390E 02		
SL1=0.200	7.0335E 02	7.0346E 02	7.0357E 02						

Figure 3.4. Output of LIKESURF in Example 1

Figure 3.5. Serial Correlations of Residuals in Example 1

Figure 3.6. Forecast Values (shown by x) in Example 1

$$\begin{aligned}
Z_t = (1+\phi) Z_{t-s} - \phi Z_{t-2s} + a_t - \lambda_1 a_{t-1} \\
- \lambda_s a_{t-s} + \lambda_1 \lambda_s a_{t-s-1}
\end{aligned} \tag{2}$$

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

$$\begin{aligned}
\hat{Z}_{t+1} = (1+\phi) Z_{t+1-s} - \phi Z_{t+1-2s} - \lambda_1 a_t - \lambda_2 a_{t+1-s} \\
+ \lambda_1 \lambda_s a_{t-s}
\end{aligned} \tag{3}^{**}$$

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.

3.2.1 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^[5]. It is as follows: For a given set of parameters the series may be forecast backwards starting somewhere in the middle of the series. To begin with 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 back-forecast 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 $t < 1$ is assumed to be zero.

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

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

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

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

The program 'LIKESURF' plots the sum of squares function over a given parametric space and obtains the values of the parameters for which it is minimal.

The output of the program is shown in Figure 3.4. The estimated values of the parameters are:

$$\hat{\phi} = -0.15 \quad \hat{\lambda}_1 = 0.25 \quad \hat{\lambda}_s = 0.5$$

and min. sum of squares = 526.538

So the variance of the residuals is $\sigma_a^2 = 1.03$,

3.3 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 3.5 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 chapter 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 different equation approach). The values of a_t for $t > t_{\text{origin}}$ is set to zero.

Figure 3.6 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 $t \gg t_{\text{origin}}$.

3.4. Generation of Texture

We can create a white noise generator having a mean = 0 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 3.2 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.

The effectiveness of this scheme can be tested by attempting to fill the holes in the 'parent' texture. The results are shown in Figure 3.7. It can be seen that the edge effects are clearly visible. This can be expected in the case of textures which have an associated structure.

Let us consider a different class in the second example. The texture of a handmade paper (Figure 3.8a) belongs to a general class of statistical textures. It has no structure. The 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 s = seasonal period (length of the row)

$$\sigma_a^2 = 3.27$$

Figure 3.8b and Figure 3.8c show the results of hole-filling operation using the above model. Here edge effects are hardly noticeable.

3.5 Conclusion

It has been demonstrated that the natural looking textures can be synthesized using a time series model. This procedure appears to be very effective while dealing with the statistical textures. Many biological textures belong to this general category.

We may add here that the motivation for this work has been provided by our desire to obtain a procedure by which we can make a just choice of the size and shape of template for the R.O.C. analysis of texture^[18]. It was our belief that the number of autoregressive and moving average terms and the values of the parameters in a time series model for a given texture influence that choice. At the present time, work is being done to establish a link between the present work and the R.O.C. analysis of texture.

(a)

(b)

Figure 3.7. Filling the Holes in a Textural Scene (Example 1)

(a)

(b)

(c)

Figure 3.8.

- a. Texture of a Handmade Paper
- b. Holes
- c. Resultant Scene after Filling the Holes

APPENDIX A

Description of Computer Programs

Acknowledgment

The description, flowcharts and the listing of the programs found in Bacon's thesis [2] and in the appendix of Box and Jenkins book [5] have been very helpful in the development of the following programs. Several modifications have been done to Bacon's programs to suit our needs. Efforts have been made to retain the same nomenclature and interpretation of several parameters in order to conform as far as possible to the description of Bacon's programs.

Program 1USID (Univariate Stochastic Model Identification)1.1 General Description

This program is developed to assist in the identification stage. It calculates the serial correlations and their standard errors for a series, $[Z_t]$ and any of its differences of the form $[\nabla_1 \nabla_s^D Z_t]$. The main program USID controls the input data and performs the required differencing while the subroutine SER is used for estimating the serial correlations.

1.2 Description of Input

A total of m cards are required by program as input for each series, m depending on the number of observations.

Card 1

FORMAT (8I4)

NC	:	The series identification number
NT	:	The number of data in the series
NPER	:	The length of the period (s)
NLG	:	If NLG=0, the original data is analyzed If NLG≠0, the mean of the series is subtracted from each term Z_t .
NM	:	The highest degree of differencing by basic intervals (i.e.d.)
NK	:	The highest degree of differencing by seasonal intervals (i.e.D.)
NL	:	The maximum lag of autocorrelations to be computed
NLP	:	The maximum lag of partial autocorrelations to be computed

Cards 2-m

These cards contain the original series data. The present input format is 32Z1. This may be changed as per the convenience.

Last Card

After the last card of the final series, a card should be placed with 0000 in NT field (columns 5-8) which causes the program to stop.

1.3 Computation

Mean of the series, $\bar{z} = \frac{1}{n} \sum_{t=1}^n z_t$

If $NLG \neq 0$, $z'_t = (z_t - \bar{z})$, otherwise $z'_t = z_t$

$$W_t = \nabla_1^d \nabla_s^D z_t \text{ for } t=1,2,\dots,n$$

$$\text{Mean } \bar{W} = \frac{1}{n} \sum_{t=1}^n W_t$$

The autocovariance function for lag $k(C_k)$ is:

$$C_k = \frac{1}{n} \sum_{t=1}^{n-k} (w_t - \bar{w})(w_{t+k} - \bar{w})$$

The autocorrelation function for lag $k(r_k)$ is:

$$r_k = C_k / C_0$$

The standard error for the estimated autocorrelation r_k is: (Bartlett's approx.)

$$\delta(r_k) = \frac{1}{\sqrt{n}} \{1 + 2(r_1^2 + r_2^2 + \dots + r_q^2)\}^{1/2} \quad k > q$$

The partial autocorrelation function for lag $\ell(\phi_\ell)$ is:

$$\phi_\ell = \begin{cases} r_1 & \text{for } \ell = 1 \\ \frac{r_1 - \sum_{J=1}^{\ell-1} \phi_{\ell-1,J} r_{\ell-J}}{1 - \sum_{J=1}^{\ell-1} \phi_{\ell-1,J} r_J} & \text{for } \ell > 1 \end{cases}$$

where $\phi_{\ell J} = \phi_{\ell-1,J} - \phi_\ell \phi_{\ell-1,\ell-J} \quad J=1,2,\dots,\ell-1$

The standard error for the estimated partial autocorrelation formula is:

$$\delta(\phi_{kk}) = 1/\sqrt{n} \quad \text{for } k > p$$

1.4 Description of Output

For each series, for all values of $d = 0$ through NM and $D = 0$ through NL , the autocorrelation function and partial autocorrelation function has been plotted for the specified number of lags. The output also consists of the standard errors of each function at every lag.

Program 2

LIKESURF

2.1 General Description

This program computes a grid of residual sum of squares over a specified region of parameter values and provides confirmation test results on the residuals from the best fitting model. This is a modified version of Bacon's program and accommodates any model of the form:

$$\begin{aligned} \nabla_1^d \nabla_s^D (1 - \phi_1 B^r) z_t = & (1 - \lambda_1 B - \lambda_2 B^2 - \lambda_s B^s - \lambda_{s+1} B^{s+1} \\ & - \lambda_{s+2} B^{s+2} - \lambda_{2s} B^{2s} - \lambda_{2s+1} B^{2s+1} \\ & - \lambda_{2s+2} B^{2s+2}) a_t \end{aligned} \quad (1)$$

This would include many commonly occurring seasonal models. Provision has been made to accept a constrained form of equation (1) such as:

$$\nabla_1^d \nabla_s^D (1 - \phi_1 B^s) z_t = (1 - \lambda_1 B)(1 - \lambda_s B^s) a_t$$

The program locates the minimum sum of squares and identifies the parameter values which produce that minimum. The serial correlations of the residuals are computed and plotted. Subroutine WES computes the periodogram plot and also computes the value of Bartlett's statistic M/C [2]. These values are used for checking the adequacy of the fit of the model. For more details the reader is referred to Bacon's thesis.

The random shocks ' a_t ' are generated by a subroutine GUASS when the appropriate parameters are supplied. Using this subroutine, the future values of the series are generated with the help of the fitted model.

The subroutine SHWPAX, treating the time series as the pixels of the generated scene, rearranges and quantizes appropriately and points a two-dimensional gray-level picture of the resultant scene with the help of the 'PAX' subroutines.

2.2 Description of Input Data

It is possible to analyze several series sequentially. It is also possible to check on several models for the same series.

Card 1

FORMAT(5I4,2F4.1)

NC	:	The series identification number
NT	:	The number of data in the series
NPER	:	The length of the period s
NPH	:	The autoregressive lag r in (1)
NLG	:	If NLG is nonzero, the mean value is subtracted from each term of the series; otherwise, the original series is analyzed
U	:	The value of d (0 or 1)
V	:	The value of D (0 or 1)

Cards 2-m

Contains the original series data. The present format is (32Z1). May be changed according to the requirements.

Card m+1

FORMAT 4I4

ND : The number of series data to be used
to estimate the parameters

NPR : The number of data to be forecast

NORG : The value of the origin from which
forecasts are made. The values of
 z_t for $t=1$ through NORG are used as
boundary conditions

NXX : Is a code variable

Nxx=0 for models of general form (1)

Nxx=1 for models of the form

$$\nabla_1^d \nabla_s^D (1 - \phi_1 B^r) z_t = (1 - \lambda_1 B)(1 - \lambda_s B^s) a_t \quad (2)$$

Nxx=2 for models of the form

$$\nabla_1^d \nabla_s^D (1 - \phi_1 B^r) z_t = (1 - \lambda_1 B)(1 - \lambda_s B^s - \lambda_{2s} B^{2s}) a_t \quad (3)$$

Nxx=3 for models of the form

$$\nabla_1^d \nabla_s^D (1 - \phi_1 B^r) z_t = (1 - \lambda_1 B - \lambda_2 B^2)(1 - \lambda_s B^s) a_t \quad (4)$$

Cards m+2, m+3, m+4

These cards contain the ranges of the values of parameters to be considered. The range is specified for each parameter by the lowest value, the highest value and the step change. Each card specifies the ranges of three parameters with the format (9F6.3). Card m+2 specifies the ranges for ϕ_1 , λ_1 and λ_2 . Card m+3 for λ_s , λ_{s+1} and λ_{s+2} . Card m+4 for λ_{2s} , λ_{2s+1} , λ_{2s+2} . The greatest number of values of any parameter

that can be tested in one fit is 16. If a parameter is to be restricted to one quantity, say 0.0, the three entries should be 0.0, 0.0, 1.0.

The last four cards only need be repeated for subsequent models to be tested for the same series.

Last Card for Each Series

A card containing zeros in ND field notifies that another series is to be processed.

Final Card

At the end of final series, a card containing zeros in NT field causes the program to stop.

2.3 Description of Output

The output contains the sum of squares function plotted over the defined parametric space. The set of parameters which produce the minimum sum of squares is shown. The values of the residuals for this set is also shown. The serial correlations of these residuals are plotted. The forecast function is also plotted starting from NORG.

NPR values of the series are generated using the values of $'a_t'$ generated from the subroutine GAUSS starting from NORG. These generated values are passed on to the subroutine SHWPAX and its output contains the gray-level picture of the two-dimensional scene, as described earlier.

Program 3GENTEX3.1 General Description

This program generates the future values of a seasonal time series from a given model for a specified boundary condition. In a limited sense, in the present context it generates the pixels of a digitized texture scene. The model and the values of its parameters are embedded in the program and can be changed for various cases.

3.2 Description of Input Data

Only the necessary details of the series and boundary conditions are fed in as input data.

Card 1

Contains NC, NT, NPER, NPH, NLG, U, V in the same format as that of LIKESURF. All parameters have same interpretation, except NT indicates the number of data included as boundary conditions.

Cards 2-m

Contains boundary conditions. Input format at present is (32Z1) which may be changed as per convenience.

We will illustrate with an example a method of embedding any model in the program:

Suppose the model to be implemented is of the form:

$$\nabla_1^d \nabla_s^D (1 - \phi_1 B^r) z_t = (1 - \lambda_1 B)(1 - \lambda_s B^s) a_t \quad (2)$$

(Values of d and D are restricted to be 0 and 1). Using this model, a one-step-ahead forecast is:

$$\begin{aligned}
\hat{z}_t = & d * z_{t-1} + D * z_{t-s} - d * D * z_{t-s-1} \\
& + \phi_1 (z_{t-r} - d * z_{t-r-1} - D * z_{t-r-s} + d * D * z_{t-r-s-1}) \\
& - \lambda_1 a_{t-1} - \lambda_s a_{t-s} + \lambda_1 \lambda_s a_{t-s-1}
\end{aligned} \tag{2'}$$

The values of the parameters of this model ϕ_1 , λ_1 , λ_s , σ_a are specified in the beginning of the program. The equation 2' is appropriately coded and placed in the loop as a one-step-ahead forecast function, which is the only thing that changes for every model.

3.3 Description of Output

The output consists of the generated values of the time series. As mentioned before, these are passed on to the subroutine SHWPAX to point a gray-level picture of the resultant scene.

LIST OF REFERENCES

1. Anderson, T. W. An Introduction to Multivariate Statistical Analysis. John Wiley & Sons, 1958.
2. Bacon, D. W. "Seasonal Time Series," Ph.D. Thesis, University of Wisconsin, Madison, 1965.
3. Bartels, Peter and G. L. Wied, "Tumor Cell Diagnosis Based On Stochastic Properties of Digitized Images," Proc. 2-D Dig. Signal Proc. Conf., Columbia, Mo. 1971.
4. Bhattacharya, P. C. "Order of Dependence in a Normally Distributed Two-Way Series." University of Arizona, 1972 (unpublished note).
5. Box, J. E. P. and G. M. Jenkins. Time Series Analysis. Holden-Day Inc., 1970.
6. Brodatz, P. Textures. New York: Dover Publications, 1966.
7. Conroy, P. Simulation of Texture by Computer Graphics. TR-14. Toronto: Department of Computer Science, University of Toronto, 1969.
8. Gibson, J. J. "The Reproduction of Visually Perceived Forms." Journal of Experimental Psychology, 12 (1929), 1-39.
9. Gibson, J. J. The Perception of the Visual World. Boston: Houghton Mifflin, 1950.
10. Hawkins, J. K. "Textural Properties for Pattern Recognition." Picture Processing and Psychopicturics. Edited by Lipkin and Rosenfeld. New York: Academic Press, 1970.
11. IEEE Transactions on Computers, July, 1972. (Special Issue on Two-Dimensional Digital Signal Processing).
12. Julesz, B. "Visual Pattern Discrimination." IRE Transactions, IT-8 (1962), 86-92.
13. Julesz, B. "Texture and Visual Perception" Scientific American, (February, 1965),
14. Julesz, B. "Cluster Formation at Various Perceptual Levels." Methodologies of Pattern Recognition. Edited by W. Walter Dunn. New York: Academic Press, 1968.
15. Ledley, R. S. "Texture Problems in Biomedical Pattern Recognition," Proceedings of the 1972 IEEE Conference on Decision and Control and 11th Symposium in Adaptive Processes, 1972.

16. Lipkin, B. S. and A. Rosenfeld, eds. Picture Processing and Psychopictorics. New York: Academic Press, New York, 1970.
17. Muerle, J. L. "Some Thoughts on Texture Discrimination by Computer." Picture Processing and Psychopictorics. Edited by Lipkin and Rosenfeld. New York: Academic Press, 1970.
18. McCormick, B. H. and S. N. Jayaramamurthy. "Analysis of Texture," Report 531, Department of Computer Science, University of Illinois, Urbana, (July 1972).
19. Pickett, R. M. "The Perception of a Visual Texture." Journal of Experimental Psychology, 68 (1964), 13-20.
20. Pickett, R. M. "The Perception of Random Visual Texture." Models for the Perception of Speech and Visual Form. Edited by W. Walter Dunn. Cambridge: M. I. T. Press, 1967.
21. Pickett, R. M. "Visual Analyses of Texture in the Detection and Recognition of Objects." Picture Processing and Psychopictorics. Edited by Lipkin and Rosenfeld. New York & London: Academic Press, 1970.
22. Puzin, M. "Simulation of Cellular Patterns by Computer Graphics." Unpublished M.S. Thesis, University of Toronto, 1969.
23. Read, J. S. and S. N. Jayaramamurthy. "Automatic Generation of Texture Feature Detectors." IEEE Transactions on Computers (July 1972).
24. Rosenfeld, A. "On Models for the Perception of Visual Texture." Models for the Perception of Speech and Logical Form. Edited by W. Walter Dunn. Cambridge: M. I. T. Press, 1967.
25. Rosenfeld, A., Y. H. Lee, and R. B. Thomas. "Edge and Curve Detection for Texture Discrimination." Picture Processing and Psychopictorics. Edited by Ripkin and Rosenfeld. New York: Academic Press, 1970.
26. Rosenfeld, A. and B. S. Lipkin. "Texture Synthesis." Picture Processing and Psychopictorics. Edited by Lipkin and Rosenfeld. New York: Academic Press, 1970.
27. Rosenfeld, A. and E. B. Troy. Visual Texture Analysis. (Computer Abstracts, TR 70-116), Computer Science Center, University of Maryland, 1970.
28. Trout. "The Description of Texture." File No. 832, Department of Computer Science, University of Illinois, Urbana, 1970.
29. Two Dimensional Digital Signal Processing Conference. Proceedings. Columbia, Mo., October, 1971.
30. Proceedings of IEEE, March, 1967 (Special Issue on Redundancy Reduction).

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-73-566	2.	3. Recipient's Accession No.
4. Title and Subtitle SYNTHESIS OF TEXTURE		5. Report Date July 1973	
		6.	
7. Author(s) B. H. McCormick, S. N. Jayaramamurthy		8. Performing Organization Rept. No.	
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois		10. Project/Task/Work Unit No.	
		11. Contract/Grant No. US AEC AT(11-1)2118	
12. Sponsoring Organization Name and Address US AEC Chicago Operations Office 9800 South Cass Avenue Argonne, Illinois		13. Type of Report & Period Covered Technical	
		14.	
15. Supplementary Notes			
16. Abstracts A general method has been proposed for the synthesis of texture. In this method, the pixels (picture elements) from a textural scene are considered as a two-way seasonal time series. With the help of well-developed methods in time series analysis, a model is fitted for the above time series. From a set of boundary conditions, with the help of this model, the future values of the time series are generated which in essence are the pixels of the synthesized texture. The effectiveness of this procedure is illustrated with examples.			
17. Key Words and Document Analysis. 17a. Descriptors visual texture seasonal time series autoregressive terms moving average terms time series analysis forecasting generation of time series texture synthesis			
17b. Identifiers/Open-Ended Terms			
17c. COSATI Field/Group			
18. Availability Statement unlimited		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
		20. Security Class (This Page) UNCLASSIFIED	22. Price

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.

COO-2118-0039

2. TITLE

SYNTHESIS OF TEXTURE

3. TYPE OF DOCUMENT (Check one):

- ☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:

Title of conference _____

Date of conference _____

Exact location of conference _____

Sponsoring organization _____

- ☐ c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- ☒ a. AEC's normal announcement and distribution procedures may be followed.
☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
☐ c. Make no announcement or distribution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

S. N. Jayaramamurthy, Research Assistant
B. H. McCormick, Principal Investigator

Organization

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Signature

Date

July 1973

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

- ☐ a. AEC patent clearance has been granted by responsible AEC patent group.
☐ b. Report has been sent to responsible AEC patent group for clearance.
☐ c. Patent clearance not required.