M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 381
Also appearing: IEEE Int. Conf. on Image Proc., Lausanne, Sep. 1996.

Temporal Texture Modeling

Martin Szummer and Rosalind W. Picard

MIT Media Lab Rm E15-384; 20 Ames St; Cambridge MA 02139; USA
`szummer@media.mit.edu`, `picard@media.mit.edu`
`http://www-white.media.mit.edu/~szummer/`

Abstract

Temporal textures are textures with motion. Examples include wavy water, rising steam and fire. We model image sequences of temporal textures using the spatio-temporal autoregressive model (STAR). This model expresses each pixel as a linear combination of surrounding pixels lagged both in space and in time. The model provides a base for both recognition and synthesis. We show how the least squares method can accurately estimate model parameters for large, causal neighborhoods with more than 1000 parameters. Synthesis results show that the model can adequately capture the spatial and temporal characteristics of many temporal textures. A 95% recognition rate is achieved for a 135 element database with 15 texture classes.

1 Introduction

Temporal textures are textures with motion. Good examples are fire, wavy water and leaves fluttering in the wind. They are characterized by an indeterminate extent both in space and time [1]. This class of motions can be contrasted with two others: *activities* are temporally periodic but spatially restricted (such as a person walking or swimming). *Motion events* are single events that do not repeat in space or time (such as opening a door or throwing a ball).

Temporal textures have previously been studied for recognition applications (e.g. detecting forest fires) [1] and for synthesis in computer graphics (e.g. artificial fire and smoke). Unlike previous work, we focus on a *representation* that can be acquired directly from image sequences, and that is effective both for recognition and synthesis.

Our representation is the linear spatio-temporal autoregressive model (STAR) [2]. It is a three-dimensional extension of autoregressive models (AR), which are among the best models for recognition and synthesis of image textures [3, 4]. Autoregressive models are also widely used in speech modeling and time series analysis. The STAR model has the form

$$s(x,y,t) = \sum_{i=1}^{p} \phi_i s(x+\Delta x_i, y+\Delta y_i, t+\Delta t_i) + a(x,y,t).$$

We model the signal $s(x,y,t)$ as a linear combination of lagged values of itself plus a Gaussian white noise process $a(x,y,t)$. The lags $\Delta x_i, \Delta y_i$ and Δt_i specify the neighborhood structure of the model. We have used causal neighborhoods, since parameter estimation and synthesis are easier to perform. Examples of causal neighborhoods include nonsymmetric half-spaces, such as the (x,y,t) subset defined by $t < 0 \vee (t = 0 \wedge y < 0) \vee (t = 0 \wedge y = 0 \wedge x < 0)$.

The STAR model makes several assumptions. The data should have a multivariate Gaussian distribution and be wide-sense stationary (constant mean and covariance). Only first and second-order statistics are exploited, hence curved lines cannot be modeled. The noise process is assumed to be uncorrelated (if it is not, use a STARMA model which has moving-average terms). Fortunately, many temporal textures satisfy these conditions approximately.

The neighborhood causality constraint is another restriction that is somewhat unnatural for spatial processes. It introduces an arbitrary directional bias, which depends on the orientation of the nonsymmetric half-space neighborhood (or any other causal neighborhood used). For spatio-temporal processes, the spatial asymmetry is not as severe as for purely spatial process. The spatial asymmetry arises only from restrictions for neighbors at $t = 0$, whereas neighbors at $t < 0$ can be symmetric. In fact, the spatial asymmetry can be completely eliminated by conditioning only on neighbors at $t < 0$. Thus, we can trade off spatial asymmetry against temporal asymmetry. Since time has a clear direction, and the physical world is believed to be causal, temporal asymmetry is easily justified.

This work was supported in part by BT, NEC and Hewlett Packard Labs.

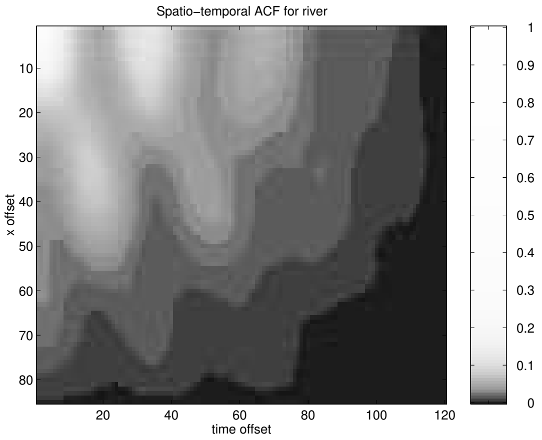

Figure 1: Autocorrelation function for river sequence (x-t slice, y=0). Note periodicity in time.

2 Model selection

Before parameter estimation, we must select neighborhood size and topology and ensure that the data is wide sense stationary.

The autocorrelation function (ACF) is a useful tool for analyzing the correlation structure of autoregressive processes and for model identification [2]. Direct computation of the ACF in the spatio-temporal domain is not feasible due to the large amount of data in an image sequence. Instead, the ACF is computed as the inverse Fourier transform of the power spectrum. Fig. 1 shows the ACF of wavy water for an x-t slice at $y = 0$. Note that there is structure along both the spatial and temporal dimensions. Along the time axis, correlation peaks occur for subsequent waves. Translation along the x-axis is also evident. Thus, full spatio-temporal modeling is necessary to capture all aspects of the signal; purely temporal or purely spatial analysis is not sufficient.

Image data is often nonstationary due to nonuniform illumination of the image. These nonstationarities can be removed using unsharp masking. One approach is to median-filter each frame and subtract the filter output from the frame [3]. We used a purely spatial 21×21 median filter for all image sequences. The illumination gradients were reduced and the ACF decayed to zero exponentially instead of linearly (which indicated nonstationarity).

Finding a good neighborhood size and topology is a difficult task for STAR models. In traditional time series analysis, model selection is done by examining the patterns of the ACF and PACF (partial autocorrelation function). STAR models have large, three-dimensional neighborhoods which generate very complex patterns that cannot be identified easily [2]. Instead, we begin by fitting a large STAR model to the texture. We got the best synthesis results from causal half-sphere neighborhoods with radius between 4 and 7 (with between 128 and 709 parameters). Other attempts included cubic neighborhoods with side length 11 (1270 parameters), and rays of length 21 radiating from the origin in 12 different directions. Such long rays could capture long distance correlations, but produced poor synthesis results.

The large number of parameters is a consequence of modeling three dimensions, as opposed to one or two. Fortunately, our data sets have extents $170 \times 115 \times 120$. Thus, there are at least 2000 data points per parameter, reducing the risk for overfitting.

The large models are already useful, and can be improved by pruning insignificant parameters. The pruning algorithm [3] iteratively discards the least significant parameters as long as the Schwartz's Bayesian Criterion (SBC) decreases. Let $|\Omega|$ be the data set size, p be the number of parameters, and $\hat{\sigma}_a^2$ be the estimated innovation variance. Then

$$\mathrm{SBC} = |\Omega| \ln \hat{\sigma}_a^2 + p \ln |\Omega|.$$

The significance of a parameter is determined by the t-test (the parameter value divided by its standard deviation). For static image textures, the pruning algorithm typically reduces 80 parameter models to 50 parameters while maintaining the visual quality of the simulated texture [3].

3 Parameter estimation

Parameters of the STAR model are determined by minimizing the mean square prediction error. We have used the conditional least squares estimator (CLS). The estimate is conditioned on the unknown values outside the boundary. One can assume that the missing boundary values are equal to the mean of the data (the *correlation* method). Alternatively, one can use only the inner portion of the data, so that all neighborhoods are contained in the data (the *covariance* method). The methods give significantly different results, probably because most visual textures are close to nonstationarity and hence are sensitive to initial conditions. The covariance method gives more accurate estimates [5].

The system of normal equations is then solved using Cholesky decomposition. The accuracy of the estimation can be determined by first estimating parameters from an image sequence, then synthesizing a texture based on them, and finally estimating the parameters of the synthesized texture. The two sets of estimates should be similar. When this test is performed for the

1270 parameter model on a wavy water sequence, the majority of the statistically significant parameters have relative errors less than 20%.

4 Synthesis

To examine how well the STAR model can capture temporal textures, we synthesize textures based on parameters estimated from real sequences. The initial conditions for the synthesis are Gaussian random noise, and new values are recursively computed as a linear combination of past values plus Gaussian random noise. The synthesized sequence is histogram-matched to the original to get the same grey-level distribution. The perceptual quality of some textures is very good (Fig. 2). The raw and synthesized image sequences are available online at `http://www-white.media.mit.edu/~szummer/icip-96/`. The examples of steam and boiling water are convincing, and river is also fairly realistic. However, rotational motion (e.g. spiraling water flow of a toilet) cannot be captured by the STAR model, because it violates the stationarity assumption. The specularity of water is also difficult to model.

The STAR model offers a very compact representation of temporal textures. For comparison, the sequences were compressed by taking the three-dimensional DCT and discarding the smallest magnitude coefficients. Then the sequences were reconstructed by the inverse DCT. The DCT reconstruction looks like a blurry version of the original. In contrast, the STAR model looks like a somewhat noisy version of the original. When the same number of coefficients are used in both representations (for a 2000:1 compression ratio), the STAR synthesis subjectively looks significantly better.

5 Recognition

We tested recognition of temporal textures in a database with 15 classes and 9 examples from each class, taken at different times. We used still images of the textures and applied a purely spatial autoregressive model (SAR) at three different scales [4]. Thus, the recognition is motion-invariant, which is desirable in many applications. For a given texture, we find other examples with the most similar autoregressive parameters, according to the Mahalanobis distance metric.

The recognition performance is very good. 95% of the top 8 matches belong to the correct texture class. In other words, we usually manage to retrieve all the other 8 examples of a texture class when querying on any instance of it. In addition to recognizing images from the same class, the algorithm is also good at finding other perceptually similar textures. Given water with big waves, it also returns wavy water with smaller waves. Similarly, for boiling water, it gets other boiling water filmed from a different angle and illumination. A query on steam first retrieves other steam and then the next closest matches are smoke.

6 Conclusion and Future Work

The STAR model can successfully represent several temporal textures, and enables good synthesis and compression. A subset of STAR (SAR) achieves excellent recognition. As a general three-dimensional texture model, STAR has a wealth of other applications, such as segmentation of medical MRI imagery.

In future work, we hope to build a multi-scale STAR model. The neighborhood would be hierarchically decomposed, achieving the effect of very large neighborhoods but with fewer parameters and less computation. However, the different scales are not necessarily independent. Hence, estimation and synthesis must be coordinated across scales.

For recognition applications, we must design features invariant to motion direction and magnitude. One possibility is to use features of STAR parameters, e.g. averages of parameters at the same distance from the origin [4].

A challenging problem is to model nonstationary temporal textures. For this task, nonlinear models are likely to be needed.

7 References

[1] Randal C. Nelson and Ramprasad Polana. Qualitative recognition of motion using temporal texture. *Comp. Vis., Graph., and Img. Proc.*, 56(1):78–89, July 1992.

[2] Phillip E. Pfeifer and Stuart Jay Deutsch. A three-stage iterative procedure for space-time modeling. *Technometrics*, 22(1):35–47, February 1980.

[3] Michael Grunkin. *On the Analysis of Image Data Using Simultaneous Interaction Models.* PhD thesis, Inst. of Mathematical Modeling and Operations Research, Technical University of Denmark, Lyngby, August 1993. no. 67.

[4] Jianchang Mao and Anil K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[5] Martin Szummer. Temporal texture modeling. Technical Report 346, MIT Media Lab Perceptual Computing, 1995. `http://www-white.media.mit.edu/vismod/cgi-bin/tr_pagemaker#TR346`.

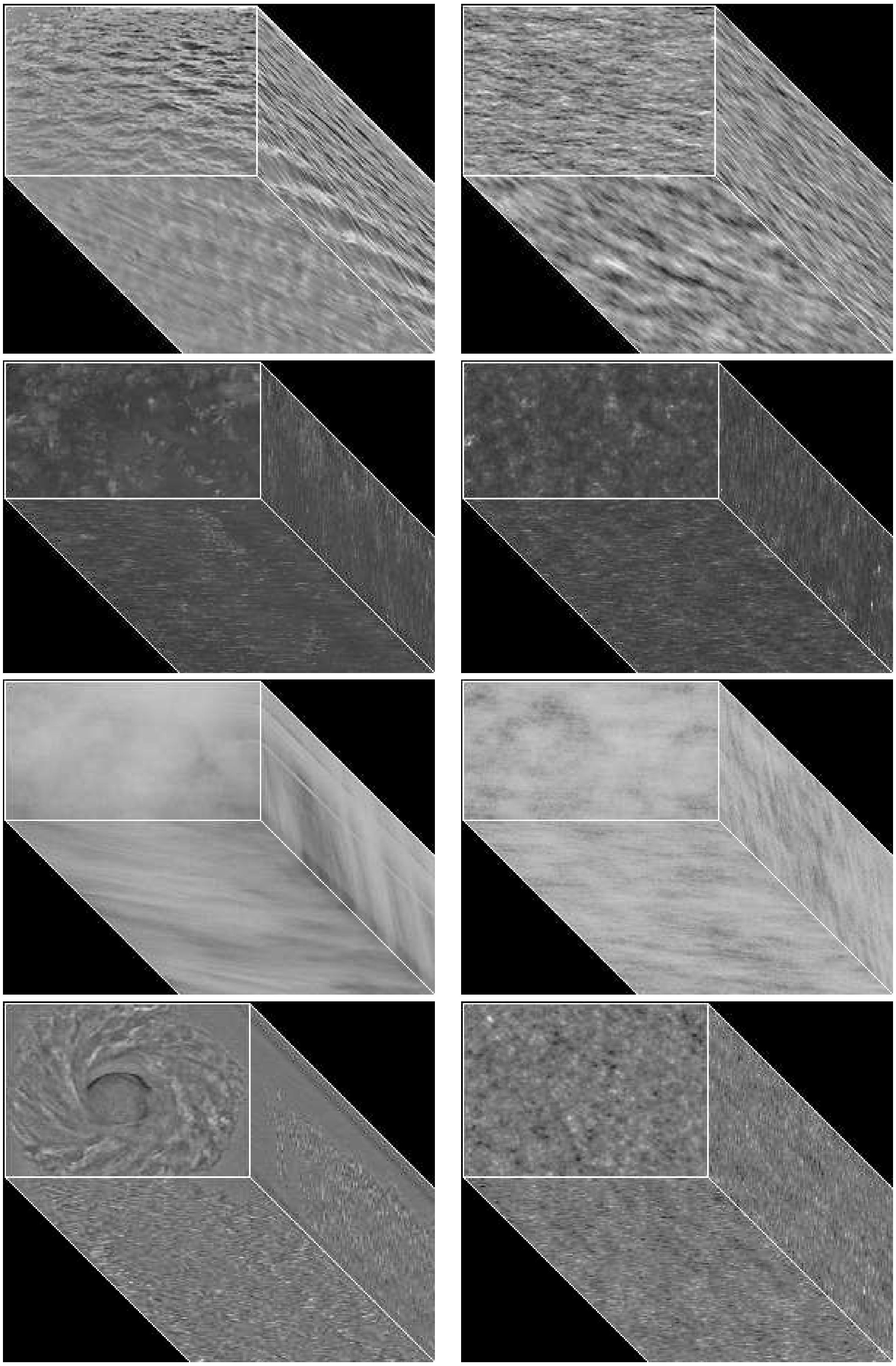

Figure 2: Synthesis results displayed as xyt-volumes. Originals (left column) and synthesized (right). The sequences and the size of neighborhoods are `river` (1270), `boiling water` (128), `steam` (1270) and `spiraling water` (128) (number of parameters in parentheses).