Image Processing Techniques and Smart Image Manipulation : Texture Synthesis

Lecture #15: Monday, 26 October 2009
Lecturer: Maneesh Agrawala
Scribe: Wesley Willett

Abstract

Textures - repeatable spatial or temporal patterns - provide a useful mechanism for reusing small samples of text, images, audio, video, or some other media to generate new samples. Generating repeatable and generalizable textures from raw samples presents an interesting set of challenges in computer graphics. This lecture provides some background for the discussion of textures and texture synthesis and then presents several pieces of pioneering work in synthesizing static image textures and video textures.

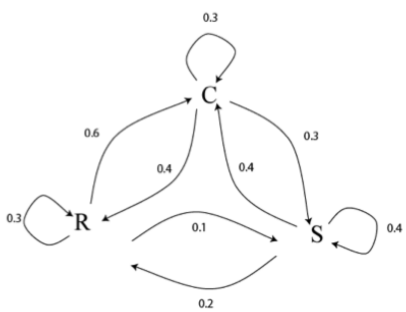
Figure 1: Examples of various sample swatches (left) and larger textured areas synthesized from them (right).

1 An Analogy - "Weather Forecasting for Dummies."

Imagine that, given today's weather, want to know tomorrow's. Assume that the weather on any given day is either "sunny", "cloudy", or "rainy". We can record the weather for each day and use that data to compute the probability of a given weather condition given the weather on the preceding day(s).

$$P(rainy|sunny)$$
$$P(rainy|cloudy)$$
$$P(sunny|rainy)$$
$$\text{etc...}$$

These probabilities form a Markov chain that can be used to make primitive predictions about the weather in the future.



$$\begin{pmatrix} 0.3 & 0.6 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

Figure 2: A Markov chain and corresponding transition matrix).

Extensions of this and other similar prediction techniques can be used to generate patterns based on sampled inputs, as we will demonstrate subsequently.

1.1 Text synthesis

An early analog of texture synthesis can be seen in Shannon's text synthesis work in the late 1940's. [4] Shannon, noted that text could be modeled as a generalized Markov chain. Given a corpus of text, one can compute the probability of a given letter (or word's) occurrence given the previous N-1 terms. Using these probabilities and some initial seed letter or word (depending on the granularity of the probabilities, it is possible to generate strings of plausible-looking (but semantically incorrect) text by adding terms to a sentence probabilistically given its existing content.

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Figure 3: One of Shannon's exemplar text strings generated using word transition probabilities. [4]

The computer program "Mark V. Shaney" [1], created by researchers at Bell Labs utilized this technique to generate novel sentences based on text gleaned from the net.singles newsgroup.

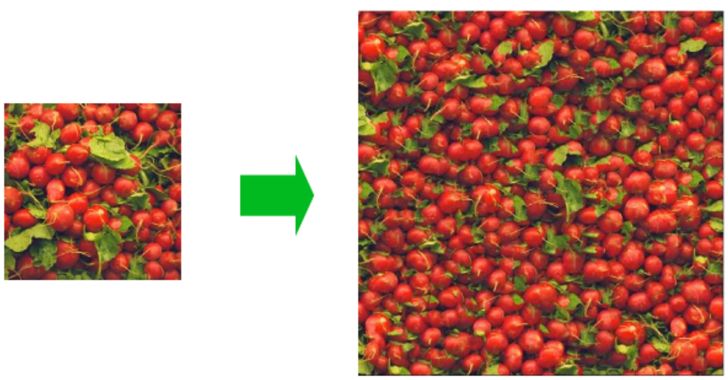
A similar synthesis approach can also be applied to generate visual textures.

2 Static Textures

"What is a texture?" We define texture as a spatially repeating pattern. Many such patterns occur in nature:



The goal of texture synthesis is to create new samples of a given texture given some initial sample.



Texture synthesis has many applications including constructing textures to paint surfaces and hole filling. Here we discuss two general classes of texture synthesis techniques:

- **Non-parametric** - Recombine pieces on an initial texture in a way that minimizes local discontinuities. Do not form a parametric model of synthesis.
- **Parametric** - Generate a "formation model" for the texture by setting model parameters based on a sample texture.

The challenge faced by techniques of both classes is that we ideally want to to be able to model a large spectrum of textures, from strong repetition to stochastic textures.

2.1 Pyramid-Based Texture Synthesis

Heeger and Bergen's seminal paper [2] introduced texture synthesis to the graphics community (although it was not actually the first to propose this kind of texture synthesis). They proposed an algorithm for iteratively synthesizing new stochastic texture patches based on image statistics from a given sample patch. An overview of their algorithm is given below.

Algorithm

Given two images, one containing a texture (I) and one empty(j):

- Initialize J to noise
- Create mul multi-resolution pyramids for I and J (steerable filter image pyramid)
- Given a set of derivative filters (multiple scales and orientations), and apply to an input image to pull our pieces of input image that
- match the histograms of J's pyramid levels with I's pyramid levels
- loop until convergence

This approach avoids any direct copying of patches, instead focusing on matching image histograms. It is automated and requires no human intervention. While this technique performs well from for stochastic textures, it fails when larger-scale cohesion is necessary (as seen in figure 4).

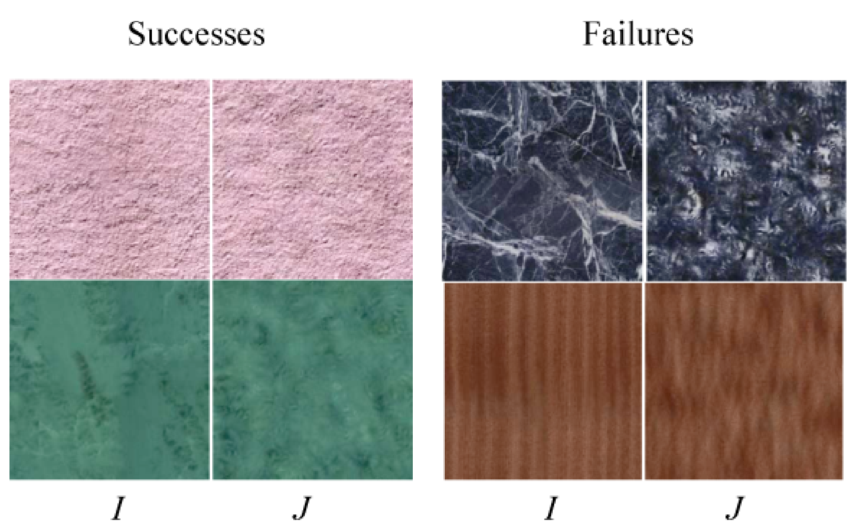


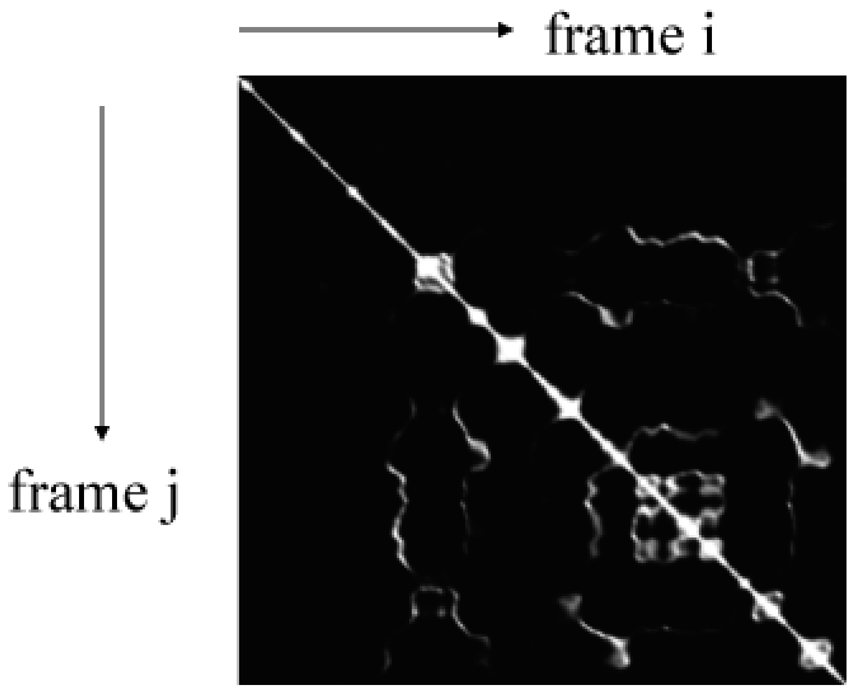
Figure 4: Example results from [2]. For each pair, I is the original image and J the generated texture. Fine-grained stochastic textures are well-preserved, but more complex global details are not.

This approach can also be generalized to 3D and can be used to generate 3D-suitable textures from 2D.

3 Video Textures

While image textures are patches of imagery that can be spatially repeated, video textures are segments of video that can be repeated indefinitely over time. As presented by Schdl, et al [3], video textures can be derived by analyzing and restructuring pieces of input video with definite beginning and end points into shorter collections of video that can be looped indefinitely.

This process is accomplished by first processing a video clip and computing the L2 distance D_{ij} between all pairs of frames (i,j) in the video.



The image above plots the similarity of all pairs of frames for a sample of video. Most frames are similar to their temporal neighbors (as evidenced by the strong line along the diagonal). However, some pairings further from the diagonal are also quite similar to one another. These frames are good candidates for looping or jumping within the texture, since little visual discrepancy exists between them that would expose the cut. In addition to accounting for pixel similarity, it is also important to consider similarity of dynamics. A pendulum swinging back and forth, for example, could have two nearly-identical frames at the base of a left swing and a right swing (figure 3), but cutting between these would not be appropriate.

Because similar frames make good transitions, one can build a Markov chain in which the probability of visiting one frame from another is based on the similarity between those frames. In most cases, a general forward flow of the video is desirable, so those frames following a given frame are weighted more heavily than those preceding it. The likelihood of transitioning between non-adjacent frames can be varied parametrically by either decreasing or increasing the cost of such jumps

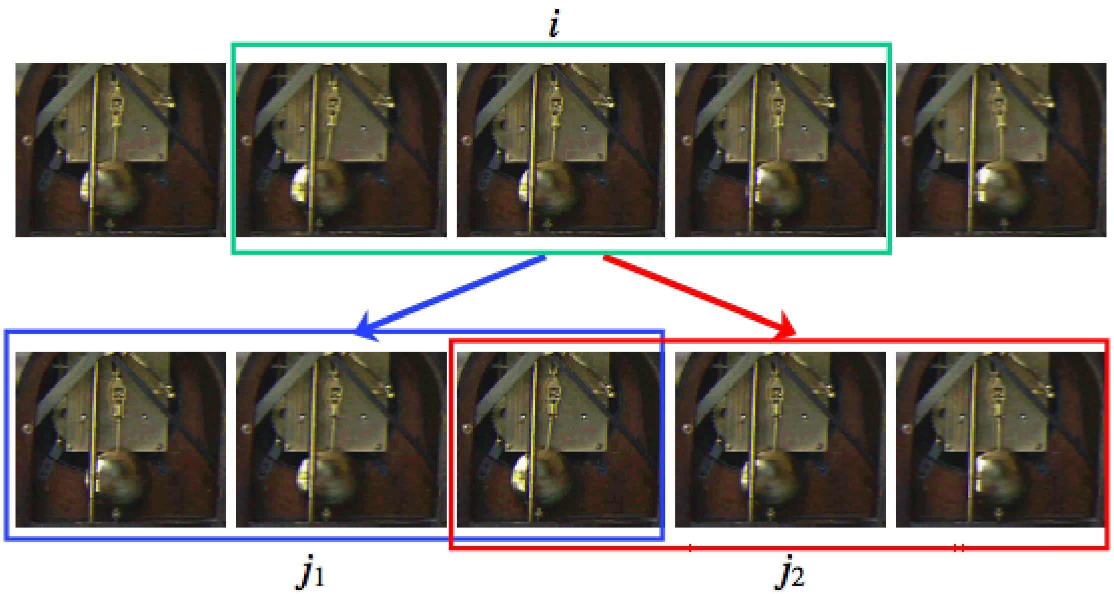


Figure 5: An example of a video in which dynamics must be preserved. Here both the frames j_1 and j_2 are good matches for frame i, but only j_2 matches the current motion of the pendulum.

based on a coefficient σ. Choosing a high value for σ produces textures with jumpier transitions, while low σ produces smoother ones. Dynamics are preserved by preferentially considering a small neighborhood of adjacent frames and by down-weighting frames that appear particularly distinct. Dead ends - sets of frames with no good way to transition out of them - also need to be avoided. sometimes t are no good transitions out of some nodes/sets of nodes. Schdl, et al address this by propagating future transition costs backward and iteratively compute new costs. In practice, the actual sequencing of frames and selection of transitions is handled using a dynamic programming algorithm to combine the smaller loops created by transitions into larger compound loops and generate the final textures.

3.0.1 Parametric motion control

Other features of video textures can also be parametrically varied. For example, it is possible to speed up video playback by omitting frames and instead skipping to similar frames in their neighborhood based on the aforementioned weighings. It is also possible to decrease the speed by jumping to similar frames in a close window using smooth transitions.

Figure 6: An example of a video texture using parametric motion control. The speed of the runner can be adjusted smoothly and frames are recombined accordingly.

3.0.2 Video sprite extraction

Using similar techniques, it is also possible to extract video "sprites" (masked pieces of video, containing a single object rather than the entire scene) from source video. By computing probable transitions between frames with respect to a proscribed direction of movement, it is possible to produce controllable animations based on a video clip. Schdl, et al [3] demonstrate an animated, mouse controlled fish that uses this approach (figure 3.0.2). Given the video of the swimming fish they precompute the costs of transitioning between directions of motion (in addition to the previous costs based on frame similarity) and switch between precomputed directions of motion based on user input.

4 Summary

Texture synthesis provides a useful mechanism for reusing small samples of text, images, audio, video, or some other media to generate new samples. Numerous methods for analyzing and synthesizing static textures exist - some parametric and some non-parametric. Here, we have covered a few pioneering applications of texture synthesis in computer graphics, including non-parametric image-based synthesis of static textures and video textures.

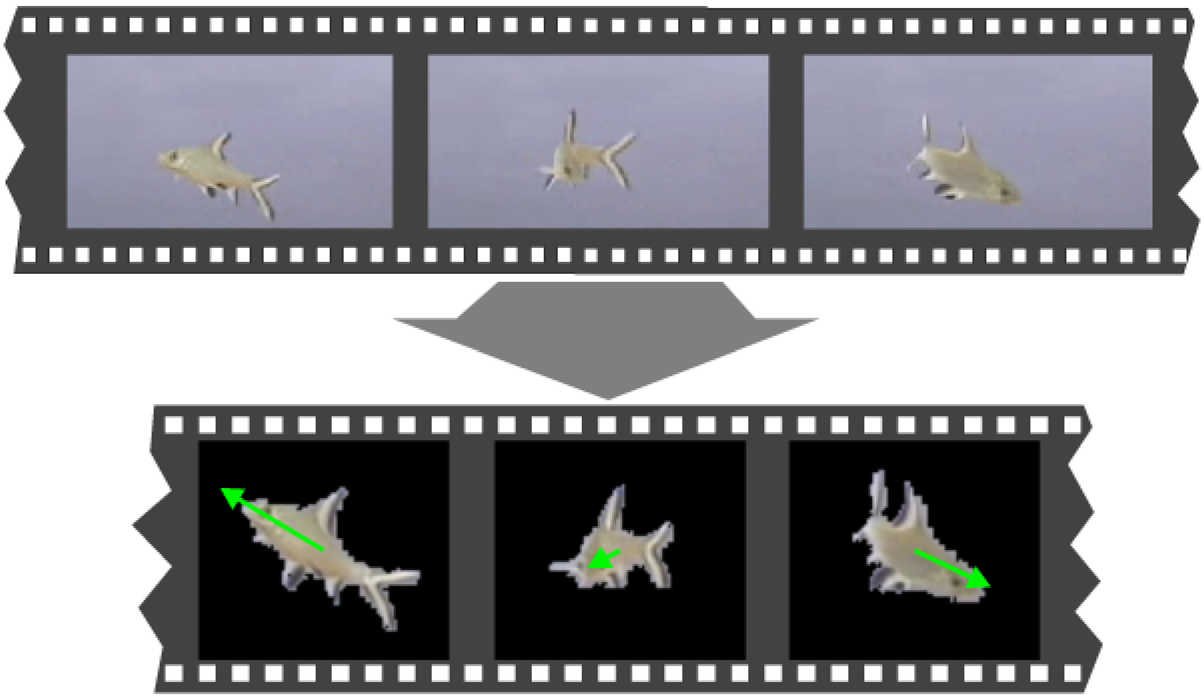


Figure 7: Extracted video sprite of a fish derived via blue screen matting and veloctiy estimation..

References

[1] Alexander Keewatin Dewdney. A potpourri of programmed prose and prosody. *Scientific American*, pages 122–TK, June 1989.

[2] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, New York, NY, USA, 1995. ACM.

[3] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[4] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, 2001.