A CHARACTERIZABLE SHAPE-FROM-TEXTURE ALGORITHM USING THE SPECTROGRAM

John Krumm
Intelligent Systems and Robotics Center
Sandia National Laboratories
MS 0949, P.O. Box 5800
Albuquerque, NM 87185
jckrumm@sandia.gov

Steven A. Shafer
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
sas@cs.cmu.edu

ABSTRACT

Perspective-induced deformations on otherwise uniformly textured surfaces can be used to compute surface normals of objects from monocular images. This is shape-from-texture. Traditional shape-from-texture algorithms are based on image features like blobs and lines, and it is hard to predict how well the algorithms will work on real data. Newer algorithms are based on local spatial frequency representations, which can be characterized mathematically from beginning to end. We summarize our spectrogram-based algorithm, and show how we can characterize the performance of the algorithm based on the program parameters and the underlying texture.

ANALYZING SHAPE-FROM-TEXTURE RESULTS

Texture is an important monocular shape cue for automatic, computerized measurement and understanding of 3D scenes from 2D images. Shape-from-texture algorithms compute surface orientations of uniformly textured objects from images. The advantages of using this cue are that the method is based on passive sensing, it can be insensitive to lighting variations, and no stereo matching is required (in fact, stereo matching is especially difficult in large, highly textured regions). The disadvantages are that uniformly textured regions must first be segmented from the image for the algorithm to work, and that it is generally hard to predict how well the algorithms will work on a given texture with given program parameters. We address the first problem in [4], and this paper addresses the second problem.

The most promising new approach to shape-from-texture is local spatial frequency analysis, illustrated in Figure 1 and Figure 4. This idea was begun by Bajcsy and Lieberman[1] in 1976, when they showed that local Fourier transforms can be used to characterize the perspective-induced frequency changes on a uniform texture. Further work on actually computing surface normals was done by Jau and Chin[3], Brown and Shvaytser[2], Super and Bovik[9], Malik and Rosenholtz[6], and Krumm and Shafer[5].

The main advantage of these local spatial frequency shape-from-texture algorithms over their older, feature-based counterparts, is that the new algorithms use a low-level representation of the image (e.g. spectrogram, wavelets, Gabor decomposition). Older algorithms used inherently error-prone and often heuristic-dependent feature detectors. Feature-detection is hard to characterize mathematically, which makes the algorithms themselves hard to characterize. The local spatial frequency approach, however, requires no feature detection, so simple mathematics can be applied all the way from the image to the final result.

This "analyzability" presents new opportunities for characterizing and improving shape-from-texture algorithms. We show in this paper how to predict how well our spectrogram-based algorithm works (in terms of surface normal variance) as a function of the underlying texture and the relative position of the texture patches we choose to analyze. This ability to characterize the algorithm's performance means that we can optimize the algorithm's parameters for best results and that we can pass uncertainly information to higher-level image understanding modules. This is important, because older shape-from-texture algorithms could not be analyzed in closed form, and therefore there was little warning when a result was seriously wrong.

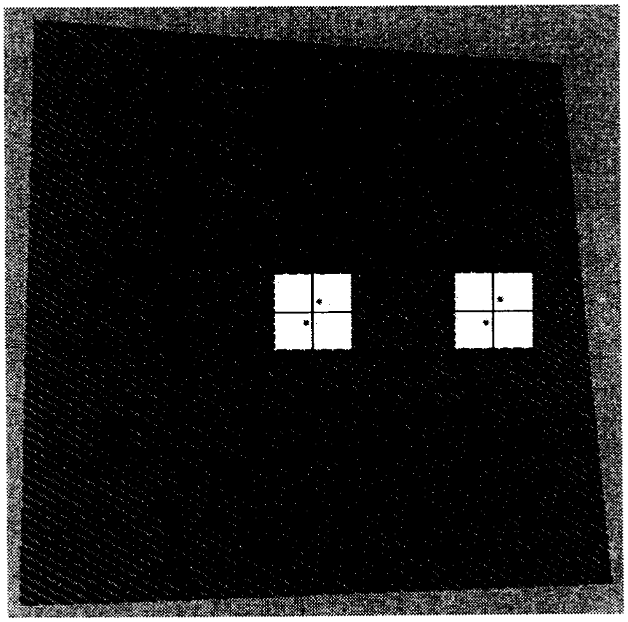

Figure 1: A texture image used for variance experiments. The light patches show the power spectra of the underlying image region. The slight shift in frequencies between the two patches is used to find the object's surface normal. The noisy sinusoid is at 60° .

0-7803-2127-8/94 $4.00 ©1994 IEEE

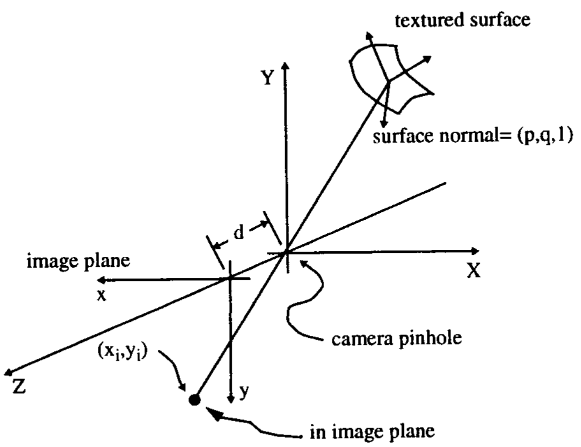

Figure 2: Geometry of camera and surface

SPECTROGRAM-BASED SHAPE-FROM-TEXTURE

This section describes our shape-from-texture algorithm. It is based on the image spectrogram - a set of windowed power spectra taken over the image. Figure 1 shows an image with two power spectra superimposed over the pixels used to compute them. Our algorithm exploits the perspective-induced frequency shifts to compute surface normals, based on the assumption that the frontal texture is uniform.

We assume a pinhole camera pointed at a uniformly textured surface, as shown in Figure 2. Our algorithm begins by computing power spectra in two, 64x64 patches of the image, as shown in Figure 1. We call these power spectra $S_1(u, v)$ and $S_2(u, v)$, and they are centered at image coordinates (x_1, y_1) and (x_2, y_2). We show in [5] that the power spectra are approximately related by an affine transformation:

$$S_1(a_1 u + b_1 v, a_2 u + b_2 v) \approx S_2(u, v) \qquad (1)$$

The affine parameters are functions of the gradient-space variables (p, q) representing the surface normal:

$$(2)$$
$$\begin{aligned}
a_1 &= B[(-d^2 r)(p^2+q^2) + dr(p^3 x_1 + p^2 q y_1 + p q^2 x_2 + q^3 y_2) + dpq(q\Delta x - p\Delta y) - pq(p^2+q^2)(x_1 y_2 - x_2 y_1)] \\
b_1 &= Bq[(-drp)(p\Delta x + q\Delta y) - dq(q\Delta x - p\Delta y) + q(p^2+q^2)(x_1 y_2 - x_2 y_1)] \\
a_2 &= Bp[(drq)(p\Delta x + q\Delta y) - dp(q\Delta x - p\Delta y) + p(p^2+q^2)(x_1 y_2 - x_2 y_1)] \\
b_2 &= B[(-d^2 r)(p^2+q^2) + dr(p^3 x_2 + p^2 q y_2 + p q^2 x_1 + q^3 y_1) - dpq(q\Delta x - p\Delta y) + pq(p^2+q^2)(x_1 y_2 - x_2 y_1)]
\end{aligned}$$

where

$$\begin{aligned}
B &= \frac{p x_1 + q y_1 - d}{dr(p^2+q^2)(p x_2 + q y_2 - d)^2} \\
r &= \sqrt{p^2+q^2+1} \\
\Delta x &= x_1 - x_2 \\
\Delta y &= y_1 - y_2
\end{aligned} \qquad (3)$$

The only unknowns in these equations are (p, q), the surface normal. Our algorithm works by searching over the space of surface normals, finding the one whose corresponding affine transform (Equation (2)) best satisfies Equation (1). We measure the similarity using the sum of squared differences (ssd):

$$\begin{aligned}
\mathrm{ssd}(p, q) = \sum\sum \{ & S_1[a_1(p, q)u + b_1(p, q)v, \\
& a_2(p, q)u + b_2(p, q)v] - \\
& S_2[u, v] \}^2
\end{aligned} \qquad (4)$$

where the sums are taken over the 2D domain of discrete frequencies, and where we have written the affine parameters as explicit functions of the surface normal. Our solution is the (p, q) that generates the smallest ssd. Our tests in [5] show an average error of about 3.6° on images of periodic textures taken in the lab.

The strength of this algorithm is its simplicity. Basing the algorithm on a local spatial frequency representation means it requires no feature detection and very few "magic numbers." Its simplicity also makes it possible to predict its performance analytically.

VARIANCE EXPERIMENTS

In our experiments with our shape-from-texture algorithm, we noticed that the variance of the result was dependent on the underlying texture and the relative locations of the two power spectrum patches. Not all textures gave good results, and not all placements of the patches gave good results. We conducted two experiments to determine the effects of these two variations.

We performed controlled experiments on a simple, synthetically generated, flat plate rotated to $(p, q) = (0.257, 0.257)$ with a sinusoidal texture mapped on. One of the images is shown in Figure 1. We added uncorrelated, Gaussian noise with a standard deviation of 10 to simulate real camera noise and to induce variations in the results for our tests. (The gray level range was [0,255].) For our first experiment, we tested the effect of varying the texture by rotating the sinusoid angle θ_u through 170° in increments of 10°. At $\theta_u = 0°$ the stripes ran vertically up the plate, while at $\theta_u = 90°$ they ran horizontally across the plate. Our experiment consisted of computing the surface normal 100 times for each texture angle. New random noise was added to the uncorrupted image before each run of the algorithm.

The results of this test are shown in Figure 3. The dots indicate the variance in the results. For both p and q, the maximum variance occurs at $\theta_u = 90°$. At this angle, the stripes run almost horizontally through the power spectrum patches. Any change in surface normal in this configuration will cause only a small change in the power spectra, so noise

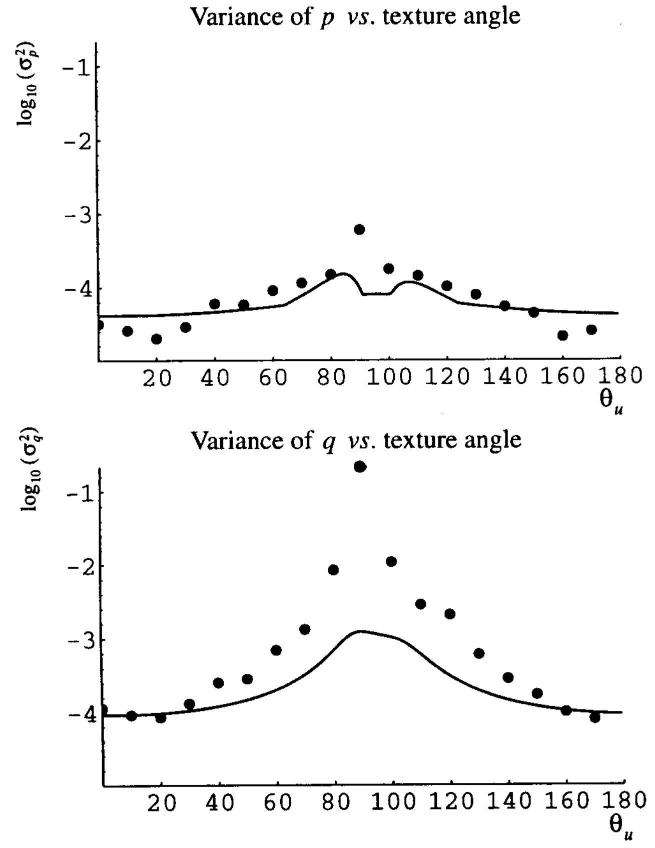

Figure 3: Experimental (dots) and predicted (solid curve) variances for varying texture angle.

dominates the solution. A change in p will be more apparent than a change in q, so the variance for q is higher. This experiment shows, in a controlled way, that some textures give better results than others when doing shape-from-texture.

Our second experiment tested the effect of the relative positions of the two power spectrum patches. The data for this experiment is shown in Figure 4. The textured plate is the same as one of those used for the first experiment, with $\theta_u = 50°$. We put one power spectrum patch at the center of the plate, and we let the other one move along a circle around the first in increments of $10°$. We designated the angle to the second patch as θ_x. We computed surface normal 100 times for each θ_x, adding new random noise to the uncorrupted image before each run.

The results of this second experiment are shown in Figure 5. The variance is maximum when a line connecting the two patches is parallel to the stripes of the texture. This is expected, since, at this angle, the power spectra will be relatively insensitive to changes in the surface normal.

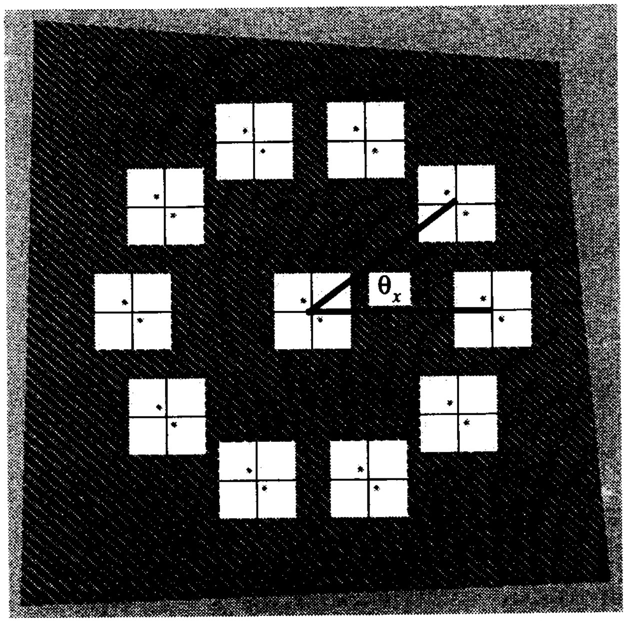

Figure 4: Data used for second experiment. The position of one of the power spectrum patches was moved around a circle.

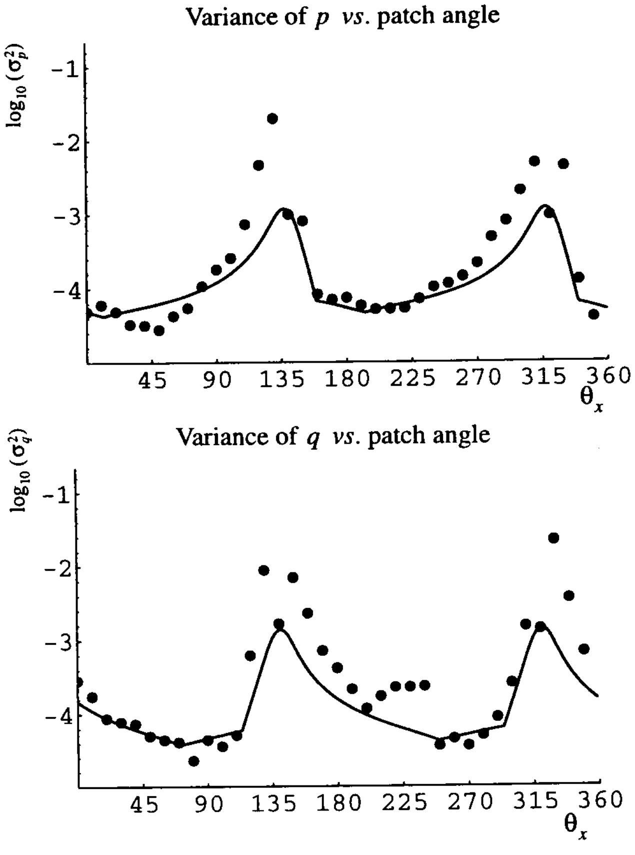

Figure 5: Experimental (dots) and predicted (solid curve) variances for varying angle between power spectrum patches.

VARIANCE PREDICTION

This section shows how we can predict the variance of the solutions in order to attach an uncertainty to the result. In order to predict the variance, we examine the shape of the ssd surface around the minimum point. A sharp or shallow depression corresponds to a low or high variance, respectively.

We follow the recipe in *Numerical Recipes*[8] for predicting the variance of our results based on the ssd surface. We designate (p^*, q^*) as the location of the minimum. The Hessian of the ssd surface at this point is

$$\alpha = \frac{1}{2}\begin{bmatrix} \frac{\partial^2}{\partial p^2}\mathrm{ssd}\,(p^*, q^*) & \frac{\partial^2}{\partial p \partial q}\mathrm{ssd}\,(p^*, q^*) \\ \frac{\partial^2}{\partial p \partial q}\mathrm{ssd}\,(p^*, q^*) & \frac{\partial^2}{\partial q^2}\mathrm{ssd}\,(p^*, q^*) \end{bmatrix} \qquad (5)$$

If the errors in the power spectra are normally distributed, then the covariance matrix of the solution will be the inverse of the Hessian. This covariance matrix defines an ellipse in (p, q) that is approximately the same shape as the constant-ssd contours of the ssd surface around the minimum point. The directions of the axes of the ellipse are given by the eigenvectors of the inverse Hessian, and the lengths of the axes are the eigenvalues. The variances along p and q are given by the projection of the ellipse onto the p and q axes. Our power spectra errors are not normally distributed, so this analysis is only approximate. This also means we can only approximate the covariance up to a scale factor. This is still useful for predicting the relative variance of solutions for different situations.

We model the sinusoidal texture as a unit amplitude, complex sinusoid with frequency (u_0, v_0), whose Fourier transform is $\delta(u - u_0, v - v_0)$. This frequency will be warped to (u_1, v_1) and (u_2, v_2) on the two patches we analyze in the image. We showed that these two frequencies are approximately related by an affine transformation. If we use a Gaussian windowing function $\left(\exp\left[-\frac{\pi}{l^2}(x^2 + y^2)\right]\right)/l^2$, then the power spectrum patches will be

$$S_i(u, v) = \exp[-2\pi l^2((u - u_i)^2 + (v - v_i)^2)]$$

We can write out the full ssd function using the above relationship along with Equations (2), (3), and (4). Space limitations and common sense prevent us from writing out this equation. The partial derivatives for the Hessian would take many pages. We expanded the equations with Macsyma, converted them to Mathematica, and used it to produce C-language expressions for our program.

We numerically computed eigenvalues and eigenvectors of the resulting Hessian matrices. The projections of the eigenvalue-scaled eigenvectors gave the predicted variances up to a common scale factor, which we estimated with a robust estimation algorithm.

The results of our variance prediction are given as solid curves in Figure 3 and Figure 5. It is clear that the predictions show the same trends as the actual variances.

The same analysis could be done for textures that are more complicated than our simple sinusoids. Our procedure illustrates that the analysis can be done and that it successfully predicts the algorithm's performance.

This work is similar to that of Malik and Rosenholtz[7], in that they also predict variances in their spectrogram-based, shape-from-texture algorithm. Whereas they do a sensitivity analysis on the parameters connecting the power spectrum patches, we do our analysis on the full algorithm. Therefore, our results account for the texture itself, while theirs ignores this effect. As our analysis shows, the texture has a significant effect on the variance of the solution.

REFERENCES

[1] Bajcsy, Ruzena and Lawrence Lieberman. "Texture Gradient as a Depth Cue." *Computer Graphics and Image Processing* 5 (1976): 52-67.

[2] Brown, Lisa Gottesfeld and Haim Shvaytser. "Surface Orientation from Projective Foreshortening of Isotropic Texture Autocorrelation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (June 1990): 584-588.

[3] Jau, Jack Y. and Roland T. Chin. "Shape from Texture Using the Wigner Distribution." *Computer Vision, Graphics, and Image Processing* 52 (1990): 248-263.

[4] Krumm, John and Steven A. Shafer. "Segmenting Textured 3D Surfaces Using the Space/Frequency Representation." *Spatial Vision*, 1994, to appear.

[5] Krumm, John. "Space/Frequency Shape Inference and Segmentation of Textured 3D Surfaces." Ph.D. dissertation, Carnegie Mellon University, December 1993. Also as technical report CMU-RI-TR-93-32.

[6] Malik, Jitendra and Ruth Rosenholtz. "A Differential Method for Computing Local Shape-From-Texture for Planar and Curved Surfaces." Computer Vision and Pattern Recognition Conference, June 1993, 267-273.

[7] Malik, Jitendra and Ruth Rosenholtz. "Computing Local Surface Orientation and Shape From Texture for Curved Surfaces." University of California, Computer Science Division (EECS), Report No. UCB/CSD 93/775, March 1994.

[8] Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*, First Edition, Cambridge University Press, 1986.

[9] Super, Boaz J. and Alan C. Bovik. "Shape-from-Texture by Wavelet-Based Measurement of Local Spectral Moments." *IEEE Conference on Computer Vision and Pattern Recognition*, 296-301, June 1992.