Shape from Periodic Texture Using the Spectrogram

John Krumm and Steven A. Shafer

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA

Abstract

We show how local spatial image frequency is related to the surface normal of a textured surface. We find that the Fourier power spectra of any two similarly textured patches on a plane are approximately related to each other by an affine transformation. The transformation parameters are a function of the plane's surface normal. We use this relationship as the basis of a new algorithm for finding surface normals of textured shapes using the spectrogram, which is one type of local spatial frequency representation. We validate the relationship by testing the algorithm on real textures. By analyzing shape and texture in terms of the local spatial frequency representation, we can exploit the advantages of the representation for the shape-from-texture problem. Specifically, our algorithm requires no feature detection and can give correct results even when the texture is aliased.

1. Introduction

Texture has long been considered an important shape cue in monocular images. The corresponding algorithms developed in computational vision exploit the systematic changes in a projected texture's appearance to find the surface normal of the underlying shape. This effect is illustrated in Figure 1, which shows a Brodatz[4] cotton canvas texture synthetically mapped onto a plate. The angle and changing depth of the plate combine to make the texture appear "smaller" as the plate recedes.

A more recent trend in image understanding is local spatial frequency analysis. Here, the image is represented in terms of the local spatial frequencies at every pixel -- the "space/frequency representation". Coherence and changes in local spatial frequency from point to point can be used to understand a rich set of image phenomena that cannot be analyzed easily in the space or frequency domain alone[14].

Since texture is fundamentally a frequency phenomenon, and since shape is fundamentally a spatial phenomenon, it is natural to approach the shape-from-texture problem in terms of this representation. In Figure 1, for example, we show the local Fourier power spectrum (spectrogram) in two places on the image. The frequencies on the right are higher than those on the left, due to perspective and foreshortening. However, there does not exist a theory that relates texture, shape, and the detailed behavior of local spatial frequency. In this paper, we develop a theory that predicts the systematic frequency shifts due to shape and we use the theory in a new shape-from-texture algorithm based on the spectrogram. This has proven to be a simple and intuitive approach to the problem. The method is attractive because it exploits a representation that is useful for understanding other important image phenomena as well.

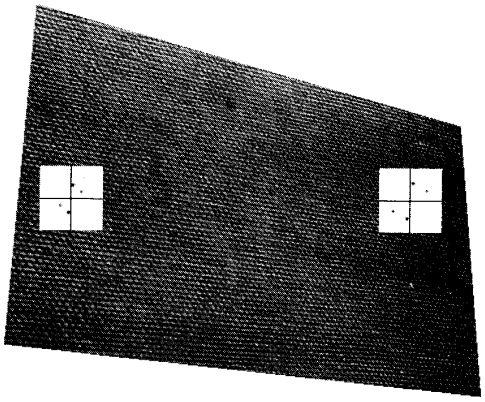

Figure 1: A textured plate with part of its spectrogram superimposed

1.1. The Space/Frequency Representation

The space/frequency representation shows the frequencies of a signal at every point in the signal. The space/frequency representation of a 1-D signal is necessarily a 2-D function of x and frequency u, since it must show a 1-D frequency distribution for every point in the signal. It is like having a little Fourier transform at every point along the x axis. If the original signal were a two-dimensional function of x and y (an image), then the space/frequency representation would be a four-dimensional function of x and y and the two frequencies, u and v.

The space/frequency representation can be computed in many ways, none of them ideal even for all well-behaved signals. We use the image spectrogram as our instantiation of the representation. For each point in the image, we extract a square neighborhood of surrounding pixels and multiply this block of intensities by a window function that falls off at block's edges. We compute the two-dimensional Fourier transform of this product and take the squared magnitude as the local frequency representation, giving the local power spectrum. This is the image spectrogram $S(x, y, u, v)$.

There are several other methods of computing the space/frequency representation. The well-known ones are Gabor functions[9], the Wigner distribution[6], and wavelets[16].

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, Arpa Order No. 7597. This first author is supported by NASA under the Graduate Student Researchers Program, Grant No. NGT-50423.The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

0-8186-2855-3/92 $3.00 ©1992 IEEE

We chose the spectrogram because it gives an intuitive-looking picture, provides a dense sampling in space and frequency, and comes with the well-developed theory of Fourier transforms. The method of computing the representation is really only important at the algorithmic level of our development. The basic theory of projecting frequencies that we develop applies regardless of the particular representation.

1.2. Shape from Texture

There is a lot of work in shape from texture, and the initial assumptions about the texture and imaging process play a significant role in the various algorithms. In our development, we assume the frequencies of the frontally viewed texture remain the same from point to point -- *i.e.* that the frontally viewed texture is stationary. The shifts in local spatial frequencies on the perspectively projected image then give information about the shape of the surface. The resulting algorithm works directly on the spectrogram of the image, requiring no feature detection. This is an important advantage over many other shape-from-texture algorithms, as it is very difficult to reliably find texels in an image.

Local spatial frequency analysis of texture started with descriptions and segmentation of frontally viewed textures. Such work includes the use of the Fourier transform by Bajcsy[1], Gramenopoulos[11] and Matsuyama *et al.*[17], Gabor filters by Turner[21], Fogel and Sagi[8], and Bovik *et al.*[3] and the Wigner distribution by Reed and Wechsler[18].

Starting with Bajcsy and Lieberman[2], one branch of shape-from-texture research has focused on using local spatial frequencies. Brown and Shvaytser[5] use the autocorrelation (Fourier transform of power spectrum) of an entire texture image to determine the slant and tilt of the textured surface. Jau and Chin[12] use the Wigner distribution, and report good results by examining only a scalar measure of the high spatial frequencies. Jones and Malik[13] use spatial frequency disparities in a novel stereo algorithm. Super and Bovik track frequency peaks[19] and frequency moments[20] to get shape from texture. Our formulation and algorithm are different in that we consider the shift of *each* frequency component from point to point in the projected texture. By maintaining a dense representation of the raw frequency data, we can apply basic theory all through the algorithm, allowing us to easily account for complicated phenomena like aliasing.

2. Math

This section contains a derivation of the connection between the surface normal of a textured surface and the local Fourier transform of the projected texture in an image. From this we show that the frequencies of two similarly textured image patches are related by an affine transform.

2.1. Coordinate Systems

Figure 2 shows the coordinate systems used in the derivation. The camera's pinhole is at the origin of the (X, Y, Z) frame. This serves as the world coordinate system, and points defined in it will be referred to with upper-case (X, Y, Z). The $-Z$ axis is coincident with the camera's optical axis and points into the scene being imaged. The image plane is the

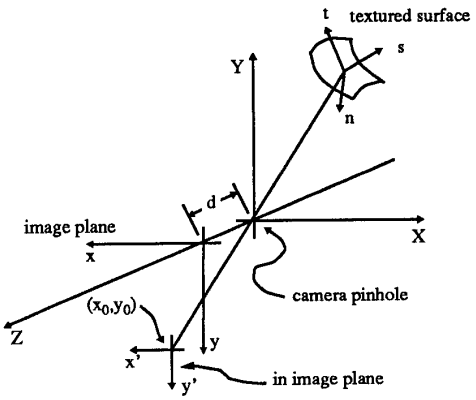

Figure 2: Coordinate systems used in derivation

(x, y) frame at a distance d behind the pinhole.

We imagine that each point on the locally planar textured surface has its own coordinate frame (s, t, n), with the n axis coincident with the surface normal. The surface normal is defined with the gradient space variables (p, q), thus the unit vector along the n axis is $\hat{n} = (p, q, 1)/r$, with $r = (p^2 + q^2 + 1)^{1/2}$, in the world frame. The origin of this surface frame is $(\Delta X, \Delta Y, \Delta Z)$ with respect to the world frame.

The 4x4 homogeneous transformation matrix that locates and orients the surface frame with respect to the world frame is

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \frac{p^2+rq^2}{p^2+q^2} & \frac{pq(1-r)}{p^2+q^2} & p & r\Delta X \\ \frac{pq(1-r)}{p^2+q^2} & \frac{rp^2+q^2}{p^2+q^2} & q & r\Delta Y \\ -p & -q & 1 & r\Delta Z \\ 0 & 0 & 0 & r \end{bmatrix}. \quad (1)$$

This was derived by making a single rotation of the (s, t, n) frame around the unit vector $(-q, p, 0)/(\sqrt{p^2+q^2})$ by an angle ϕ with $\cos\phi = 1/r$ and $\sin\phi = (\sqrt{(p^2+q^2)})/r$.

2.2. Projected Texture

This subsection concludes with an expression for a perspectively projected texture. We begin by assuming the texture on the surface is "painted" on and not a relief pattern. It is locally characterized in the (s, t, n) surface frame as a pattern of surface markings given by $f(s, t)$. Points on this locally planar surface are given by coordinates $(s, t, 0)$. Applying the transformation matrix, the corresponding world coordinates are

$$\begin{aligned} X &= t_{11}s + t_{12}t + \Delta X \\ Y &= t_{21}s + t_{22}t + \Delta Y \\ Z &= t_{31}s + t_{32}t + \Delta Z \end{aligned} \quad (2)$$

Under perspective, the origin of the (s, t, n) frame projects to $(x_0, y_0) = -\frac{d}{\Delta Z}(\Delta X, \Delta Y)$ on the image plane. In order to avoid carrying a coordinate offset through the calculations, we define another coordinate system, (x', y'), on the image plane that is centered at (x_0, y_0) with its axes parallel to those of the image plane. Given an (x, y) on the surface,

$$\begin{aligned} x' &= x - x_0 = -d\frac{t_{11}s + t_{12}t + \Delta X}{t_{31}s + t_{32}t + \Delta Z} - x_0 \\ y' &= y - y_0 = -d\frac{t_{21}s + t_{22}t + \Delta Y}{t_{31}s + t_{32}t + \Delta Z} - y_0 \end{aligned} \tag{3}$$

Solving these two equations for (s, t) will give equations that give a point in the surface frame for any corresponding point in the (x', y') frame. Doing so, using $(\Delta X, \Delta Y) = (-x_0\Delta Z, -y_0\Delta Z)/d$ and the orthonormality relationships among the vectors in the transformation matrix, we have

$$\begin{aligned} s(x', y') &= -\frac{\Delta Z[d(y' t_{12} - x' t_{22}) + t_{32}(y' x_0 - x' y_0)]}{d[t_{13}(x' + x_0) + t_{23}(y' + y_0) - d\Delta Z]} \\ t(x', y') &= \frac{\Delta Z[d(y' t_{11} - x' t_{21}) + t_{31}(y' x_0 - x' y_0)]}{d[t_{13}(x' + x_0) + t_{32}(y' + y_0) - d\Delta Z]} \end{aligned} \tag{4}$$

Thus, if the brightness pattern on a locally planar patch on a textured surface is $f(s, t)$, then the projected pattern on the image plane is a nonlinear warping of the pattern given by $f(s(x', y'), t(x', y'))$.

2.3. Approximating the Fourier Transform

In order to work with frequencies, we would like to find an expression for the Fourier transform of the projected texture, $f(s(x', y'), t(x', y'))$. But the warpings represented by Equation (4) are too complex to allow us to say anything general. We can make progress by linearizing $s(x', y')$ and $t(x', y')$ using a truncated Taylor series around $(x', y') = (0, 0)$. The approximation is justified since we are only examining a relatively small window of intensities around the point of interest. We have

$$\begin{aligned} s(x', y') &\approx s_x x' + s_y y' \\ t(x', y') &\approx t_x x' + t_y y' \end{aligned} \tag{5}$$

with

$$s_x = \frac{\partial}{\partial x'} s(x', y')|_{(x', y') = (0, 0)} \tag{6}$$

etc. We have left out the actual expressions here to save space. More details can be found in [15].

The projected version of $f(s, t)$ is then approximately $f(s_x x' + s_y y', t_x x' + t_y y')$, which is just an affine transformation (without translation) of the coordinates. A similar relationship holds in the Fourier domain given by the following Fourier transform pairs[10]:

$$\begin{aligned} &f(x', y') \Rightarrow F(u, v) \\ &f(s_x x' + s_y y', t_x x' + t_y y') \Rightarrow \frac{1}{|D|} F\left(\frac{t_y}{D}u - \frac{t_x}{D}v, -\frac{s_y}{D}u + \frac{s_x}{D}v\right) \end{aligned} \tag{7}$$

where $D = s_x t_y - s_y t_x$. Here (u, v) are spatial frequency coordinates in cycles/unit distance.

The significant conclusion is that the Fourier transform of a perspectively projected texture patch is approximately an affine transformation of the Fourier transform of the frontally viewed texture. The affine transformation parameters are a function of the camera focal length, the pixel coordinates of the point of interest, and the depth and orientation of the patch.

2.4. Relation Between Fourier Transforms of Two Patches

Since there is usually no way to determine what the frontally viewed texture looks like, we resort to comparing patches of the same texture at different locations in the image. We showed above that the Fourier transform of each patch is related to the Fourier transform of the frontally viewed texture by an affine transformation. This means that the Fourier transforms of patches themselves are related by affine transformations.

Suppose the two patches $f_1(s, t)$ and $f_2(s, t)$ are related to the frontally viewed texture by the affine parameters $(s_{x1}, s_{y1}, t_{x1}, t_{y1})$ and $(s_{x2}, s_{y2}, t_{x2}, t_{y2})$. In Fourier space, an affine transformation of the first into the second means that

$$\begin{aligned} &\frac{1}{|D_1|} F_1\left(a_1\left(\frac{t_{y1}}{D_1}u - \frac{t_{x1}}{D_1}v\right) + b_1\left(-\frac{s_{y1}}{D_1}u + \frac{s_{x1}}{D_1}v\right), a_2\left(\frac{t_{y1}}{D_1}u - \frac{t_{x1}}{D_1}v\right) + b_2\left(-\frac{s_{y1}}{D_1}u + \frac{s_{x1}}{D_1}v\right)\right) \\ &= \frac{1}{|D_2|} F_2\left(\frac{t_{y2}}{D_2}u - \frac{t_{x2}}{D_2}v, -\frac{s_{y2}}{D_2}u + \frac{s_{x2}}{D_2}v\right) \end{aligned} \tag{8}$$

where $F_1(u, v)$ and $F_2(u, v)$ are the Fourier transforms of the two patches, $D_1 = s_{x1}t_{y1} - s_{y1}t_{x1}$, $D_2 = s_{x2}t_{y2} - s_{y2}t_{x2}$, and (a_1, b_1, a_2, b_2) are the affine transformation parameters connecting the two Fourier transforms. Note that we have ignored phase differences here. In reality, the Fourier phases of the two patches will be different. This difference is masked because each patch is defined with respect to its own local coordinate system. In our formulation, phase would only complicate the derivation, and we discard it by computing the Fourier transform's magnitude in our algorithm.

Equating coefficients on (u, v) in Equation (8) leads to

$$\begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix} = \frac{1}{D_2} \begin{bmatrix} s_{x1}t_{y2} - s_{y1}t_{x2} \\ t_{x1}t_{y2} - t_{x2}t_{y1} \\ s_{x1}t_{y2} - s_{y1}t_{x2} \\ s_{x2}t_{y1} - s_{y2}t_{x1} \end{bmatrix}. \tag{9}$$

Thus, the affine parameters connecting the two Fourier transforms are functions of the affine parameters connecting

the two patches to the frontally viewed texture. In order to relate this equation to the physical parameters of the camera and the textured surface, we take the values of $(s_{x1}, s_{y1}, t_{x1} t_{y1})$ and $(s_{x2}, s_{y2}, t_{x2}, t_{y2})$ from Equation (6). Before doing this, however, we will make the assumption that the two texture patches have the same surface normal, i.e. $(p_1, q_1) = (p_2, q_2) = (p, q)$, and that both patches are on the same plane, i.e.

$$\frac{\Delta Z_2}{\Delta Z_1} = \frac{d - px_{0_1} - qy_{0_1}}{d - px_{0_2} - qy_{0_2}}. \tag{10}$$

This makes the depth variables drop out.

The affine parameters connecting the Fourier transforms of the two patches are then

$$\begin{aligned}
a_1 &= A\left[-d^2 r(p^2+q^2) + dpq(p\Delta x - q\Delta y) + dr(pq^2 x_{0_1} + q^3 y_{0_1} + p^3 x_{0_2} + p^2 q y_{0_2}) + pq(p^2+q^2)(x_{0_1} y_{0_2} - x_{0_2} y_{0_1})\right] \\
b_1 &= pA\left[dp(q\Delta x - p\Delta y) - dqr(p\Delta x + q\Delta y) - p(p^2+q^2)(x_{0_1} y_{0_2} - x_{0_2} y_{0_1})\right] \\
a_2 &= qA\left[dq(p\Delta y - q\Delta x) - dpr(p\Delta x + q\Delta y) + q(p^2+q^2)(x_{0_1} y_{0_2} - x_{0_2} y_{0_1})\right] \\
b_2 &= A\left[-d^2 r(p^2+q^2) + dpq(q\Delta x - p\Delta y) + dr(p^3 x_{0_1} + p^2 q y_{0_1} + pq^2 x_{0_2} + q^3 y_{0_2}) - pq(p^2+q^2)(x_{0_1} y_{0_2} - x_{0_2} y_{0_1})\right]
\end{aligned} \tag{11}$$

where

$$A = \frac{px_{0_1} + qy_{0_1} - d}{dr(p^2+q^2)(px_{0_2} + qy_{0_2} - d)^2} \tag{12}$$

and $\Delta x_0 = x_{0_1} - x_{0_2}$, and $\Delta y_0 = y_{0_1} - y_{0_2}$.

These equations are not easy to interpret intuitively. The notable feature is that the only unknowns are (p, q).

To summarize this section, we first showed how a locally planar surface patch projects by perspective into the image. We approximated this projection with a truncated Taylor series. This gave an affine relationship between the frontally viewed texture and the projected texture. A property of the Fourier transform says that an affine transformation in space is an affine transformation in frequency. Since the Fourier transform of each image patch is related by an affine transformation to the Fourier transform of the frontally viewed texture, the Fourier transforms of the image patches are also related by an affine transformation. If we assume the two patches are on the same plane, the affine parameters that connect their Fourier transforms are functions of known camera parameters and the unknown surface normal.

3. Algorithm

Here we discuss our core shape-from-texture algorithm. The five major steps involved in computing a surface normal from an image of a textured surface are

1. Manually pick two test points on the surface that would have the same texture when viewed frontally.
2. Multiply the neighborhood of each point by a window function.
3. Compute the 2D Fourier transform of each windowed patch.
4. Compute the squared magnitude of each Fourier transform, giving the local power spectrum at each point (part of the spectrogram).
5. Search for the (p, q) that gives the best affine warping from one local power spectrum to the other.

The window function we use is the "Blackman-Harris minimum 4-sample" window defined in [7]. In our experiments, we settled on a window size of 64x64 pixels in images that are typically 512x512 pixels. In Figure 1, the size of the two light-colored squares is equal to the window size.

Because of varying phase, the Fourier transforms at any two general points even on a frontally viewed texture would be different. For the same reason, strictly speaking, Equation (8) would not hold. In order to match the phases of two patches, we would have to use a six-parameter affine transformation (including translation) rather than the four-parameter version (no translation) that we use now. By using only magnitude, we can reduce the complexity of the affine transformation and speed up the program.

The last step of our core algorithm is an exhaustive search for the (p, q) that best transforms the power spectrum of one patch into another. Our current implementation searches over a 64x64 grid, with $(|p|, |q|) \leq (2, 2)$. This corresponds to a maximum slant of about 63^o. Given a (p, q) to try, we compute the corresponding affine parameters from Equation (11), use these to transform the power spectrum of the first patch using bilinear interpolation, and compute the sum of squared differences (ssd) between the two power spectra. We take the (p, q) that generates the minimum ssd as the solution. The ssd surface from the image in Figure 1 is shown in Figure 3, where we have scaled so the minimum ssd is one. We do the gradient space search on a 64x64 processor parallel computer, using one processor for each candidate surface orientation.

This algorithm is better than other shape-from-texture algorithms in several ways. It requires no feature-finding, which is normally an unreliable step. We make no strong assumptions about the frontally-viewed texture, only that it is

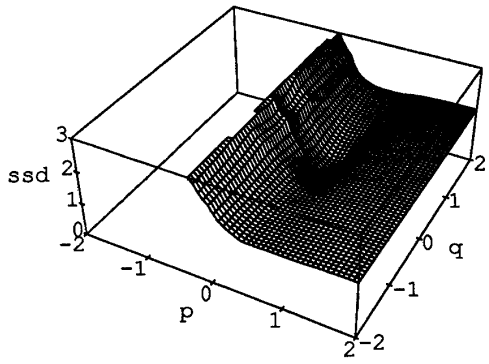

Figure 3: SSD surface from comparing patches in Figure 1

stationary. Specifically, we don't require that the texture be isotropic. Theoretically, the method should work for both periodic and random textures. We will have to find a better spectral power estimator before we can make it work on random textures, however. Finally, by formulating and solving the problem with the space/frequency representation, we can easily account for other frequency phenomena such as focus and aliasing in the same framework. We show how the method successfully deals with aliasing in the next section.

4. Results

4.1. Flat Plate

The plates in Figure 1 and Figure 4 are geometrically identical with different textures mapped on by a computer graphics program. The surface normal of the plates is $(p, q) = (0.614, 0.364)$. In terms of slant and tilt, $(\sigma, \tau) = (35.5^o, 30.7^o)$. The first plate in Figure 4 has a simple intensity function of two crossed cosines. The next two are Brodatz[4] textures: wire screen (D14) and straw cloth (D52). The last is another pair of cosines that show aliasing. This is discussed in the next section. Each of the images is shown with two patches removed and replaced by their Fourier power spectra, which are part of the total image spectrogram. These patches are size 64x64.

We ran our algorithm on each of these images and also on the canvas image in Figure 1. The results are shown in Table 1. The method works best on textures that are closest to being purely periodic, like the cosine and canvas textures. It loses some accuracy for textures with slightly more random spacing like the wire screen and straw cloth. Considering that the algorithm is only examining data from about 5.5% of the pixels on each textured region, these results are good. Most algorithms for shape-from-texture examine an entire image of a plane covering the whole field of view.

We investigated the effect of window size by running our program on the same five textures with different window sizes. The angle error generally decreased with increasing window size up to a window size of 40x40, where it leveled off.

4.2. Aliasing

The last image in Figure 4 shows moire patterns that result from aliasing. The corresponding effects in frequency can be seen in the series of local power spectra. The frequencies move toward the edge of the boxes, which represent one half the sampling rate. When a projected frequency moves beyond the boundary, it reenters at another point. This is aliasing. Assuming a sampling period of one, a frequency

texture	computed (p, q)	error
cosines	(0.603, 0.349)	0.7^o
wire screen	(0.540, 0.286)	4.4^o
canvas (Figure 1)	(0.603, 0.349)	0.7^o
straw cloth	(0.540, 0.349)	3.1^o
aliased cosines	(0.603, 0.286)	3.5^o

Table 1: Results on textured plates

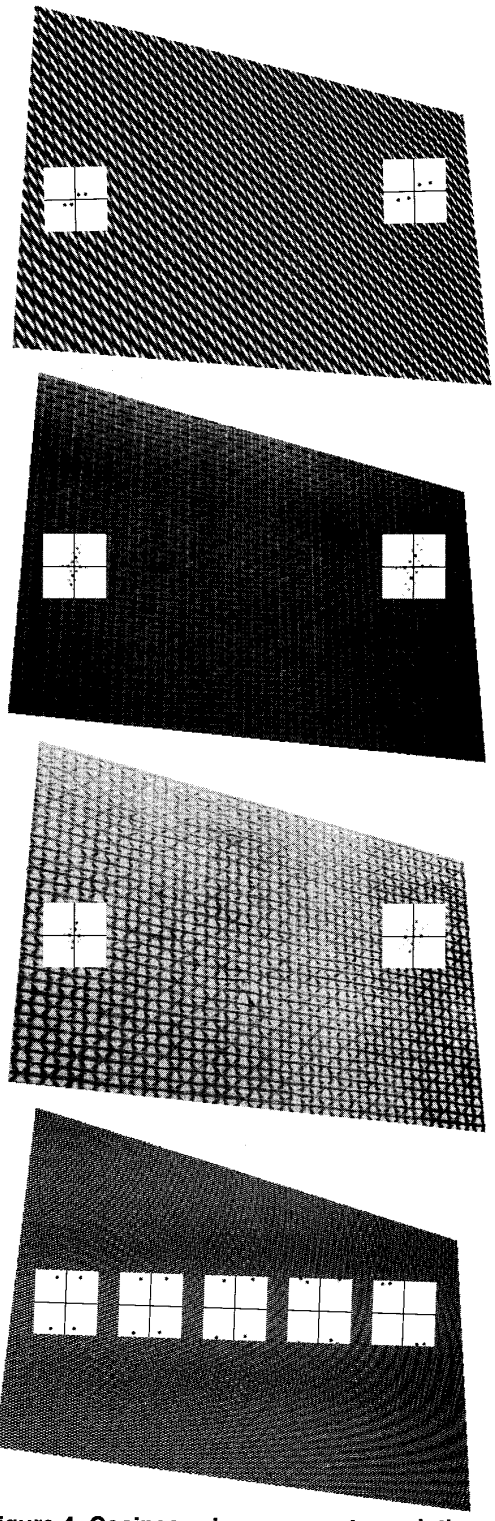

Figure 4: Cosines, wire screen, straw cloth, and aliased cosines

(u,v) is aliased according to

$$(u_{alias}, v_{alias}) = (u - \left\lfloor u + \frac{1}{2} \right\rfloor, v - \left\lfloor v + \frac{1}{2} \right\rfloor) \quad (13)$$

with $\lfloor x \rfloor$ being the "floor" function, returning the largest integer not exceeding x.

Our algorithm allows us to account for aliasing very easily. When we test a given (p, q), we warp the frequency coordinates in one power spectrum by an affine transformation. We simply put all the transformed (u, v)'s through Equation (13) to adjust them for aliasing. This way, if a given (p, q) causes frequencies to be transformed outside the half-sampling-frequency limits, they will be aliased back in at the proper coordinates. This is also a convenient way of making sure both frequency patches overlap exactly, instead of having one skewed off the other with no corresponding frequencies in the other patch after the affine transformation.

We ran our algorithm on the left and right spectra of the aliased texture in Figure 4 and got an error of about 3.5^o. Thus the method successfully accounts for aliasing. There are two restrictions. First, it is assumed that the first patch is not aliased. Second, we cannot yet account for the fact that aliased frequencies actually sum with nonaliased frequencies.

We know of no other shape-from-texture algorithm that can account for aliasing even in this simple case. We attribute the ability to the fact that the spectrogram preserves essentially all the data in the original signal and that frequency is the natural domain for the analysis of aliasing.

5. Conclusion

We have advocated the use of the space/frequency representation, which shows an image's spatial and local spatial frequency characteristics simultaneously. One natural application for such a representation is the shape-from-texture problem. If we assume that the frontally viewed texture is stationary, we can expect to see systematic shifts in frequency from point to point due to shape and perspective projection. We developed a new theory that predicts the detailed behavior of spatial frequencies in the image of a projected surface. Because it makes predictions at a low level, this theory can be applied to any space/frequency representation of an image. Using this math, we developed an algorithm based on the spectrogram that successfully finds surface normals of textured surfaces with a search in gradient space. The algorithm requires no feature-finding, working instead on a low-level representation that is still convenient for analysis. Because the representation is low-level, it should support other kinds of image analysis as well. For instance, the algorithm can easily handle simple cases of aliasing.

References

[1] Bajcsy, Ruzena. "Computer Description of Textured Surfaces." *Proceedings*, Third International Joint Conference on Artificial Intelligence, August, 1973, 572-577.

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

[3] Bovik, Alan Conrad and Clark, Marianna and Geisler, Wilson S. "Multichannel Texture Analysis Using Localized Spatial Filters." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 1, January 1990, 55-73.

[4] Brodatz, Phil. *Textures: A Photographic Album for Artists and Designers*, Dover, 1966.

[5] Brown, Lisa Gottesfeld and Shvaytser, Haim. "Surface Orientation from Projective Foreshortening of Isotropic Texture Autocorrelation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12,6, June 1990, 584-588.

[6] Claasen, T. A. C. M. and Mecklenbrauker, W. F. G. "The Wigner Distribution -- A Tool for Time-Frequency Signal Analysis, Part I: Continuous-Time Signals", *Philips Journal of Research*, 35, 1980, 217-250.

[7] DeFatta, David J. and Lucas, Joseph G. and Hodgkiss, William S. *Digital Signal Processing: A System Design Approach*. John Wiley & Sons, 1988, p. 270.

[8] Fogel, I. and Sagi, D. "Gabor Filters as Texture Discriminator." *Biological Cybernetics* 61, 2, June 1989, 103-113.

[9] Gabor, D. "Theory of Communication", *The Journal of the Institute of Electrical Engineers*, Part III, 93, 21, January 1946, 429-457.

[10] Gaskill, Jack D. *Linear Systems, Fourier Transforms, and Optics*. John Wiley & Sons, 1978.

[11] Gramenopoulos, Nicholas. "Terrain Type Recognition Using ERTS-1 MSS Images." *Proceedings*, Symposium on Significant Results Obtained from the Earth Resources Technology Satellite-1, Volume 1: Technical Presentations, Section B, March 1973, 1229-1241.

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

[13] Jones, David G. and Jitendra Malik. "Determining Three-Dimensional Shape from Orientation and Spatial Frequency Disparities II - Using Corresponding Image Patches." Computer Science Division, University of California Berkeley, Technical Report No.UCB-CSD 91-657, October 1991.

[14] Krumm, John and Steven A. Shafer, "Local Spatial Frequency Analysis of Image Texture." *Proceedings*, Third International Conference on Computer Vision, December 1990, 354-358.

[15] Krumm, John and Steven A. Shafer, "Shape from Periodic Texture Using the Spectrogram." Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-91-29, November 1991.

[16] Mallat, Stephane G. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 7, July 1989, 674-693.

[17] Matsuyama, Takashi and Miura, Shu-Ichi and Nagao, Makoto. "Structural Analysis of Natural Textures by Fourier Transformation." *Computer Vision, Graphics and Image Processing*, 24, 1983, 347-362.

[18] Reed, Todd R. and Wechsler, Harry. "Segmentation of Textured Images and Gestalt Organization Using Spatial/Spatial-Frequency Representations." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 1, January 1990, 1-12.

[19] Super, Boaz J. and Alan C. Bovik, "Three-Dimensional Orientation from Texture Using Gabor Wavelets." *Proceedings*, SPIE Conference on Visual Communications and Image Processing, November 1991.

[20] Super, Boaz J. and Alan C. Bovik, "Solution to Shape-from-Texture by Wavelet-Based Measurement of Local Spectral Moments." University of Texas at Austin, Computer and Vision Research Center, Technical Report TR-92-05-80, November 1991.

[21] Turner, M.R. "Texture Discrimination by Gabor Functions." *Biological Cybernetics* 55, 1, October 1986, 71-82.