4

Local Spatial Frequency Analysis for Computer Vision

John Krumm and Steven A. Shafer

CMU-RI-TR-90-11

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

May 1990

This research was supported by the Defense Advanced Research Projects Agency, DoD, through ARPA Order Number 4976, monitored by the Air Force Avionics Laboratory under Contract F33615-87-C-14.) and by the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract 957989 and by the National Aeronautics and Space Administration under the Graduate Student Researcher's Program, Goddard Space Flight Center, for the first author, grant NGT-50423. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the United States Government or the Jet Propulsion Laboratory.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
CMU-RI-TR-90-11

5 MONITORING ORGANIZATION REPORT NUMBER(S)
F33615-87-C-1499

6a. NAME OF PERFORMING ORGANIZATION
The Robotics Institute
Carnegie Mellon University

6b OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Air Force Avionics Laboratory
Jet Propulsion Laboratory

6c. ADDRESS *(City, State, and ZIP Code)*
Pittsburgh, PA 15213

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
DARPA / NASA

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
4976 / NASA Contract 957989

8c ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO.	WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)*
Local Spatial Frequency Analysis for Computer Vision

12 PERSONAL AUTHOR(S)
John Krumm and Steven A. Shafer

13a. TYPE OF REPORT
Technical

13b TIME COVERED
FROM __________ TO __________

14. DATE OF REPORT *(Year, Month, Day)*

15 PAGE COUNT

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A sense of vision is a prerequisite for a robot to function in an unstructured environment. However, real-world scenes contain many interacting phenomena that lead to complex images which are difficult to interpret automatically. Typical computer vision research proceeds by analyzing various effects in isolation (e.g., shading, texture, stereo, defocus), usually on images devoid of realistic complicating factors. This leads to specialized algorithms which fail on real-world images. Part of this failure is due to the dichotomy of useful representations for these phenomena. Some effects are best described in the spatial domain, while others are more naturally expressed in frequency. In order to resolve this dichotomy, we present the combined space-frequency representation which, for each point in an image, shows the spatial frequencies at that point. Within this common representation, we develop a set of simple, natural theories describing phenomena such as texture, shape, aliasin and lens parameters. We show how these theories lead to algorithms for shape from texture and for dealiasing image data. The space-frequency representation should be a key aid in

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

(19 cont'd)

untangling the complex interaction of phenomena in images, allowing automatic understanding of real-world scenes.

Contents

List of Figures

Abstract

A sense of vision is a prerequisite for a robot to function in an unstructured environment. However, real-world scenes contain many interacting phenomena that lead to complex images which are difficult to interpret automatically. Typical computer vision research proceeds by analyzing various effects in isolation (*e.g.* shading, texture, stereo, defocus), usually on images devoid of realistic complicating factors. This leads to specialized algorithms which fail on real-world images. Part of this failure is due to the dichotomy of useful representations for these phenomena. Some effects are best described in the spatial domain, while others are more naturally expressed in frequency. In order to resolve this dichotomy, we present the combined space/frequency representation which, for each point in an image, shows the spatial frequencies at that point. Within this common representation, we develop a set of simple, natural theories describing phenomena such as texture, shape, aliasing and lens parameters. We show how these theories lead to algorithms for shape from texture and for dealiasing image data. The space/frequency representation should be a key aid in untangling the complex interaction of phenomena in images, allowing automatic understanding of real-world scenes. (KR) ←

1 Introduction

In order to function in the real world, robots need to be able to perceive what is around them through a visual sense. Unfortunately, the world is very complex, and current approaches to machine vision have not proven successful at dealing with this complexity. Because of this, most "real systems" for machine vision are actually based on many very specialized assumptions about the world; on the other hand, researchers doing theoretical work study just one simple phenomenon at a time, but cannot deal with the interactions that are always present in realistic scenarios. These circumstances have led to very slow progress in developing real vision systems that have generality and a sound theoretical foundation.

In this paper, we examine the area of *spatial vision* – all of the 2D and 3D geometric factors that combine t result in the arrangement of features in the image. The factors of spatial vision include:

2D Texture: Patterns "painted" on a flat, smooth surface show up as patterns in the image.

3D Texture: Roughness and topography of the surface interact with lighting to produce additional patterns in the image.

Surface Shape and Perspective: The 3D orientation of a surface causes its patterns to project in a particular way onto the image plane.

Image Resolution: The resolution of the sensor induces sampling and aliasing in the image data, sometimes even causing noticeable moire patterns.

Focus: The optics of the lens induces blurring in the imaging process due to defocus.

Other Factors: There are numerous other factors we shall not address further in this paper, including some whose magnitude is much smaller than the factors listed above (*e.g.* diffraction), and some that involve additional imaging parameters (*e.g.* shadows, motion blur).

For each of the above phenomena, there has already been substantial theoretical vision research and sometimes real systems. However, the theories invariably deal with just one or just two of the above factors; and the real systems work by virtue of the highly limiting assumptions that are imbedded within the algorithms, such as building in a specific size range of textures to be analyzed.

The real world is not so well-behaved. Real images exhibit these factors simultaneously, as we illustrate in Figure 1. This image, synthetically generated, shows two objects with Brodatz [Bro66] textures mapped onto their surfaces. The textures themselves would pose a difficult analysis problem even if they were viewed frontally, as is usually presumed in research into 2D texture analysis. However, in this scene, the textures are mapped onto 3D surfaces, one curved and one polyhedral. Thus, the size and spatial relationships among the repetitive elements may change across an object or a surface. Because the resolution of the imaging sensor is finite, the texture elements or their component features may even become so small that they are blurred out of perceptibility – yet the same texture persists in that place in the real world, even though we can't explicitly see and measure it. The texture patterns themselves are not perfectly repetitive and may vary, and these variations should not be confused with the other sources of variation across a surface. And, this figure doesn't even demonstrate the effects of 3D texture – we mapped the Brodatz intensity patterns onto simulated smooth surfaces – or of defocus, which would cause the texture to blur selectively at some places in the image.

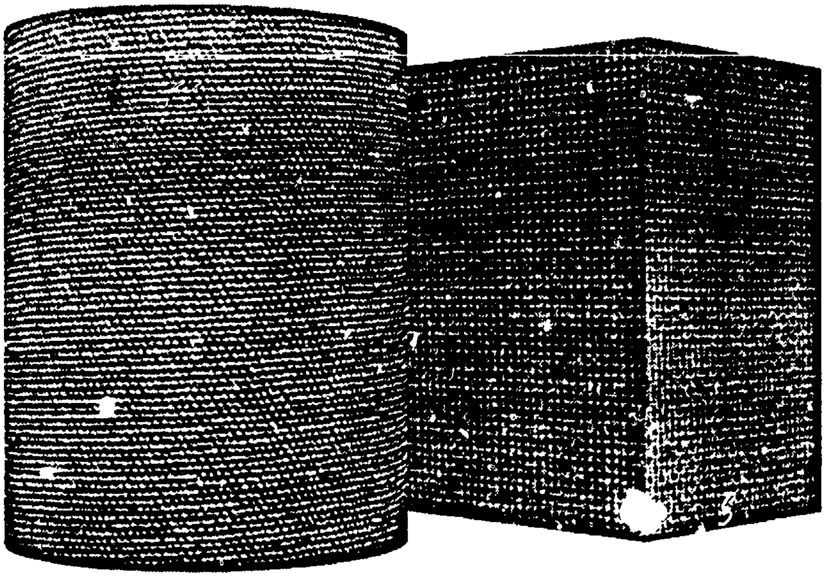

Figure 1: Cylinder and cube with Brodatz textures

Analyzing such combinations of spatial features is far beyond the capability of current robot vision systems. Yet, the real world presents just such interactions, not just on rare occasions, but on virtually every surface in every image that we care to analyze. In order to build reliable, general vision systems, we need to explicitly understand, model, and analyze each of these phenomena and their interactions.

One of the principal reasons for the slow progress in this direction is the lack of even a suitable representation that would allow us to model all of these spatial phenomena in one framework. The use of a single framework is critical, because if each phenomena is described in a different formalism, then their interactions become combinatorially complex even to describe mathematically. But, if a single framework is used, then all of the interactions can be naturally expressed within the same vocabulary.

What framework can be used? The spatial/geometry domain provides elegant descriptions of surface shape and perspective, not-so-elegant descriptions of focus and resolution, and, as the 2D texture community has shown, poor descriptions of 2D texture and repetition. The Fourier domain appears elegant for 2D texture, focus, and resolution. Unfortunately, the frequency domain has great problems with 3D surface shape, multiple surfaces in the scene, and curved surfaces or other sources of local texture variation, because the Fourier transform mixes together frequency information from all across the image without any notion of *locality*. Obviously, no representation can be a general basis for spatial vision if it has no concept of locality within the image.

What we seek is a representation for image data that provides frequency data, but does so within the context of surfaces and other local neighborhoods of the image. There exists a class of representations that does just this: the so-called *space/frequency* distributions. These have been proposed specifically for analysis of 2D textures on flat surfaces in the past, but as shown above, that is a small part of the total problem of spatial vision. In this paper, we show that this same class of representations can be used as an elegant representation for all of the phenomena described above, in 3D as well as 2D. We concentrate on a particular space/frequency distribution, the *image spectrogram*, because it has properties that appear most desirable

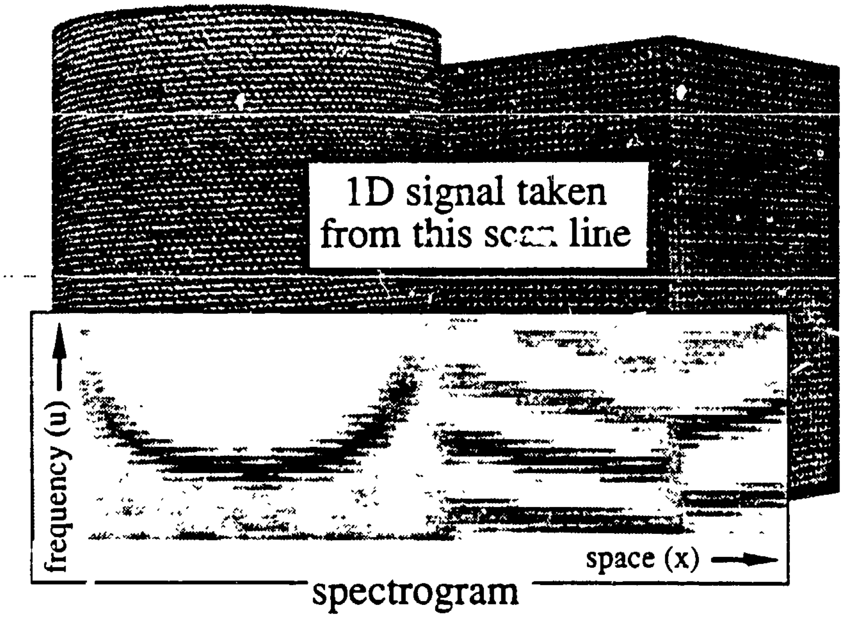

Figure 2: Figure 1 with spectrogram of center row

for general robot vision.

We show the spectrogram of the center scan-line of Figure 1 superimposed in Figure 2. The spectrogram is a two-dimensional function of space (horizontal axis) and frequency (vertical axis). Because the underlying patterns on the two objects are periodic, there are dark, frequency peaks in the spectrogram where the objects occur. The large, "U"-shaped frequency peak on the left shows that the frequency of the texture pattern projected from the cylinder appears higher near the edges than in the middle, as one would expect. At the extreme edges of the cylinder, the projected frequency is so high it cannot be adequately reproduced in the image. This is shown in the spectrogram as the frequency peak bumping into the Nyquist frequency at the top. On the left side of the cube, we see a slowly decreasing fundamental frequency and overtones which are likewise decreasing. This decrease continues to the corner of the cube, where the fundamental and harmonics begin to increase as the side recedes into the distance. This is a sample of the kind of analysis possible with the spectrogram.

The remainder of this paper explores in more detail the connections between the image spectrogram and the 3D scene. Although we do not present any "real" vision algorithms, we see to present the space/frequency representation as an important, unifying framework for future work in computer vision. Our research is in its early stages, so our opinion of the representation remains speculative but optimistic.

1.1 Previous Work

Because local spatial frequency analysis is especially well-suited to investigating repetitive patterns, most of the work similar to ours has been in image texture. There is a large set of work on texture, so much so that at least three survey papers have been published on the topic [Har79] [Wec80] [VGDO85]. We will restrict our comments to those efforts in which local spatial frequency analysis plays a dominant role. While much of the work we review is aimed at analyzing texture, other concerns the issue of image representation.

Previous work with windowed Fourier transforms in computer vision reveals some of the potential utility of local spatial frequency analysis. Image spectrograms have been used for a variety of image analysis work, including texture segmentation and shape from texture. In one method of statistical texture segmentation, a small number of features is extracted from windowed Fourier transforms taken over the image. Fourier transform methods are frequently included in comparisons of statistical texture segmentation techniques, although they are generally outperformed by other methods [DR76]. Bajcsy and Lieberman [BL76] recovered information about the shape of textured surfaces by examining the shape and behavior of peaks in windowed Fourier transforms over the image. Because they used non-overlapping windows, their analysis was based on a coarse sampling in space of the spectrogram. Matsuyama *et al.* [MMN83] used Fourier transforms taken over regions of uniformly distributed texture elements in order to find the two spatial vectors which characterize the placement of the elements. The Fourier transform has also been considered for calculating the point of best focus for an entire image by Horn [Hor68], and for a subsection by Krotkov [Kro87]. Pentland uses the Fourier transform for both shape from focus [Pen85] and shape from shading [Pen88].

All of these approaches use the Fourier transform over either the whole image or a fairly large region. The Fourier transform, however, hides the spatial coherence of the image. Thus, although one can identify the component frequencies of an image, their location in the image is a mystery. Large-support Fourier transforms tend to smear the frequency peaks of signals whose frequency is changing (*e.g.* a periodic pattern on a tilted plane) and confound the analysis of signals with spatially distinct subcomponents (*e.g.* two adjacent textures). A solution to this problem is the space/frequency representation which shows the frequency content of only small, local regions of the signal.

One popular space/frequency representation is the Wigner Distribution (WD), introduced by Wigner for use in quantum mechanics. Like the spectrogram, the WD produces a function of both space and frequency from a function of space alone.[1] An informative introduction to the WD can be found in a three-part series by Claasen and Mecklenbräuker [CM80a] [CM80b] [CM80c]. Practically speaking, the WD can effectively deal with signals whose frequency is changing, giving a clear indication of their instantaneous frequency. It has been applied to texture segmentation by Reed and Wechsler [RW90] and to shape from texture by Jau and Chin [JC88]. Both the spectrogram and WD are joint representations of space and spatial frequency. Such representations are reviewed and compared by Jacobson and Wechsler [JW88]. A description of the WD and our reasons for not using it are presented below in Section 2.2.

An early effort aimed at creating a joint representation was that of Gabor [Gab46], who proposed the use of one-dimensional, Gaussian-modulated sinusoids as basis functions which are maximally compact in both time (space) and frequency. Marčelja [Mar80] found that these functions describe the response of visual cortex cells. The theory was extended to two dimensions by Daugman [Dau85], who showed that the two-dimensional Gabor functions can describe the cells of the visual cortex. Gabor-function filtering has been applied to the tasks of texture segmentation by Turner [Tur86] and Bovik *et al.* [BCG90], and to optical flow extraction by Heeger [Hee88]. Fogel and Sagi [FS89] found that Gabor function texture segmentation closely paralleled human performance. Most work in image analysis of this type uses the Gabor functions as convolution filters, but not as a form of complete image representation. The Gabor functions are a complete, but not orthogonal, set of basis functions. Nonorthogonal basis functions complicate the process of decomposition, although it has been achieved with a neural network by Daugman [Dau88].

Mallat [Mal89] has developed a theory for the multiresolution representation of images called an "orthogonal wavelet representation". It is composed of a low resolution image and successively higher resolution "difference" images which fill in the details of the previous images. The representation falls between the space and frequency domains, and gives an idea of the predominant frequencies at every point in the image.

[1] We note that much of the work in space/frequency representations is presented in terms of time rather than space.

A significant difference between the wavelet and Gabor representations is that the wavelet representation has orthogonal basis functions, making the representation easy to compute.

1.2 This Paper

Our work is distinguished from most of that above, not by the particular representation we have chosen, but by how we propose to analyze the local spatial frequencies. Most of the work in texture analysis above uses just a small set of frequencies, usually for segmentation. Our work demonstrates how a denser set of frequencies at each point can be used not only for segmentation, but to chart other space-varying properties in the scene.

In this paper we show how a joint space/frequency representation can be used to effectively examine a variety of important phenomena in computer vision. In the next section, we examine two of the most popular joint representations – the spectrogram and the Wigner distribution – and we compare their usefulness for 3D image understanding. In Section 3 we show how the spectrogram maintains coherence over regions of similar texture, even if the texture is changing in frequency. Making this coherence explicit means that the spectrogram can be used for segmentation on textures other than just those on a plane viewed frontally, which is an implicit limitation in most texture segmentation algorithms. In Section 4 we show how 3D object shapes affect the spectrogram. We examine in detail the spectrogram of a texture along a line and demonstrate how we can accurately extract shape parameters in this simple case. Section 5 shows how spatial aliasing (moire patterns) affects the spectrogram. In Section 6 we show how changes in a camera's lens parameters (zoom, focus, and aperture) affect the spectrogram in a predictable way. The zoom analysis, combined with the development on aliasing, leads to an algorithm for dealiasing images of simple textures. We examine other issues in Section 7.

2 Space/Frequency Representations

Contiguous texture patterns in a scene normally do not appear as constant frequency patterns in an image, because the underlying shape is usually not planar. Even if it were, the frequency would only appear constant if the texture were veiwed along the plane's normal. Thus, frequency analysis of texture in nontrivial scenes requires a method which can account for changes in frequency with position. This is beyond the ability of conventional, large support, Fourier transforms, so other methods have been devised.

We show two examples of idealized space/frequency representations in Figures 3 and 4. Figure 3-a shows a simple sinusoidal wave, and Figure 3-b shows the magnitude of its Fourier transform. The ideal space/frequency representation appears in Figure 3-c, and shows that the signal's frequency u is constant with respect to the spatial variable x. Figure 4-a shows two sinusoidal waves in which the higher-frequency wave occupies the center quarter of the signal. The Fourier transform of this signal is shown in Figure 4-b. Although it shows two pairs of frequency peaks, it does not show *where* in space the subsignals of corresponding frequency occur. The structure of the signal is made clear in the space/frequency representation of Figure 4-c, which shows that a relatively low-frequency component exists at the ends of the signal in question, while a higher-frequency part occurs in the middle one quarter. This localization is the power of the space/frequency representation.

Signals whose frequency changes with position are called *nonstationary*. A simple example is $\cos(2\pi u_o x^2/2)$. The *instantaneous frequency* of such a signal is defined as the derivative of the argument with respect to the spatial variable – in this example, $u_o x$ (in cycles/unit distance). Certain frequency-based, texture

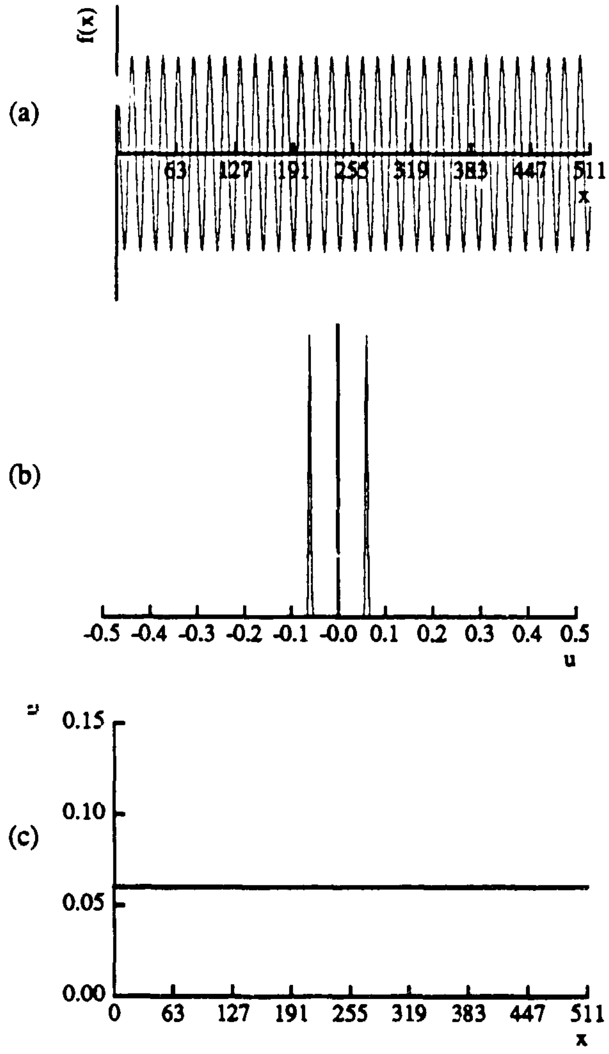

Figure 3: (a) Single sinusoid (b) Fourier transform (c) Space/frequency representation shows constant frequency

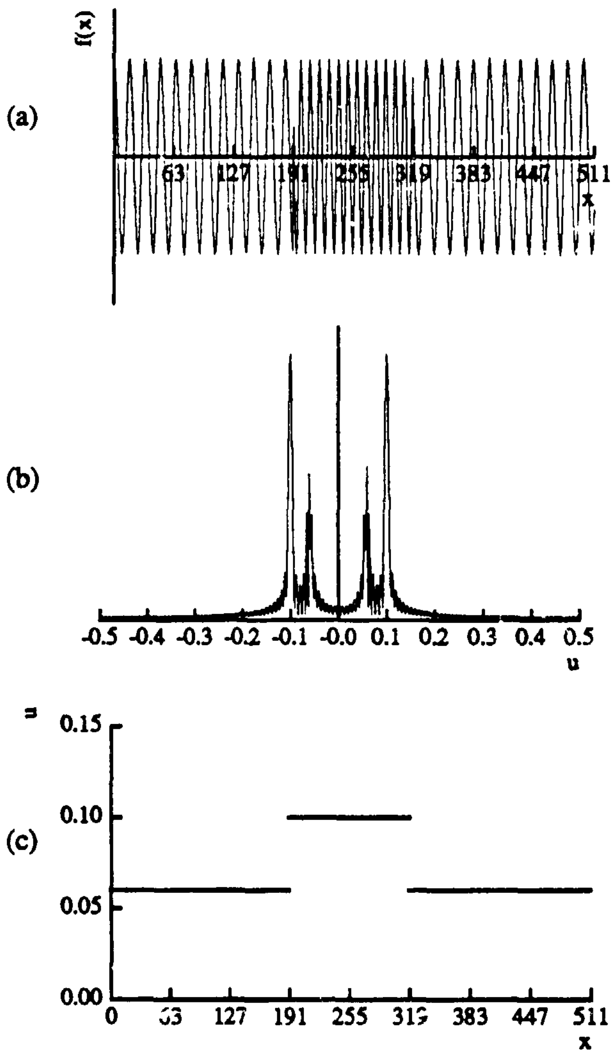

Figure 4: (a) Two sinusoids (b) Fourier transform shows only frequency (c) Space/frequency representation shows structure

segmentation algorithms [DR76] do not require an accurate estimate of the instantaneous frequency, only one which is sensitive to significant differences in frequency. Thus, they can work with only a coarse sampling in frequency. In our work, however, we are concerned with small changes in frequency, due to, for instance, surface slope or variations in zoom. Thus, we require a high resolution, accurate estimate of the instantaneous frequency.

We consider in this section the two primary means of calculating space/frequency representations: the spectrogram and the Wigner distribution. A third method is to fit sinusoids to the signal over small windows; although it is slow, it leads to high resolution estimates. Both this method and the spectrogram are based on the assumption that the signal is locally stationary. The WD relaxes this assumption.

Our analysis in this section and the rest of this paper will be limited to one-dimensional signals. This not only simplifies understanding the mathematics, but makes visualization of the representation much easier. For a 1D signal, the space/frequency representation is two-dimensional, while for a 2D signal (an image), it is four-dimensional. Our example spectrograms are superimposed on 2D images. In these figures, the spectrogram was computed from the center row of the image. We include the entire image to illustrate more clearly the various effects we are considering.

2.1 The Spectrogram

The spectrogram of a signal is a series of small-support, Fourier transforms of the signal, each centered around a different point of the signal. For a one-dimensional signal $f(x)$, the spectrogram is $S_f(x,u)$, where u is frequency in cycles/unit distance. $S_f(x,u)$ is an estimate of the power of frequency u at the point x. The continuous spectrogram of the one-dimensional function $f(x)$ is given by

$$S_f(x,u) = \left| \int_{-\infty}^{\infty} w_l(\alpha - x) f(\alpha) e^{-j2\pi u\alpha} d\alpha \right|^2 ,$$

where $w_l(x)$ is a window function with support length l.

The process by which a spectrogram is calculated is shown in Figure 5. To calculate one vertical slice of the spectrogram for a given value of x, say x_o, the signal is first multiplied by a window offset by x_o. This product is Fourier transformed; the magnitude is calculated from the complex values of the Fourier transform; and the non-negative half of the magnitudes serve as $S_f(x_o,u)$, which is one column of the spectrogram. This process is repeated for every x. We only consider the non-negative half of the magnitudes since the Fourier transform of a real signal (the only kind we have) is symmetric in magnitude. The discrete version is computed using the discrete Fourier transform (DFT), which is discrete in both space and frequency. The window function controls how much of the rest of the signal contributes to the spectrogram at the point x. In terms of $W_l(u)$ and $F(u)$, the Fourier transforms of $w_l(x)$ and $f(x)$, the spectrogram is

$$S_f(x,u) = \left| \left(e^{-j2\pi xu} W_l(u) \right) * F(u) \right|^2 , \tag{1}$$

where "$*$" is convolution.

The spectrogram of a two-dimensional function $f(x,y)$ is a straightforward extension of the equation above, giving a four-dimensional spectrogram, $S_f(x,y,u,v)$, with two spatial variables and two frequency variables.

There are ongoing questions about the best shape and size of the window $w_l(x)$. Many window shapes are considered by Harris in [Har78]. He illustrates the compromises involved in the selection, and concludes by

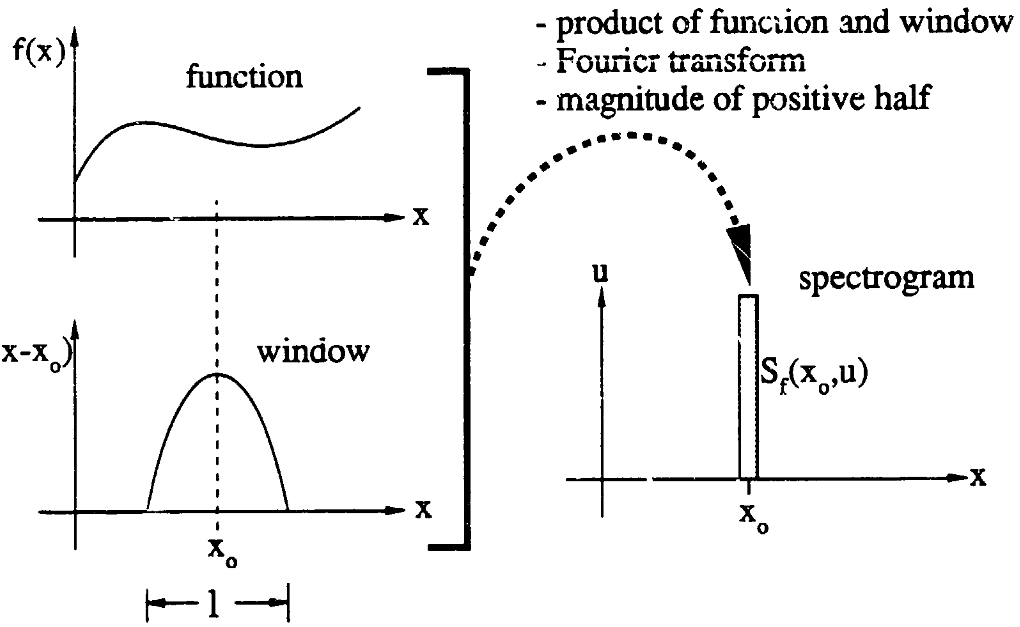

Figure 5: Computing the spectrogram

recommending the 4-sample Blackman-Harris window. We use the minimum, 4-sample Blackman-Harris window, which for a discrete set of n points is given by

$$w_n(k) = a_0 - a_1 \cos\left(\frac{2\pi}{n-1}k\right) + a_2 \cos\left(\frac{2\pi}{n-1}2k\right) - a_3 \cos\left(\frac{2\pi}{n-1}3k\right) \qquad (2)$$

for $k = 0.1.\ldots.n-1$ and $(a_0.a_1.a_2.a_3) = (0.35875.0.48829.0.14128.0.01168)$.

The window size l (or in the discrete case n) affects how much of the signal is included in the Fourier transform at each point. Equation 1 above shows that the effect of windowing is to convolve the Fourier transform of the signal, $F(u)$, with the Fourier transform of the window, $W_l(u)$. This can be thought of as a blurring of the signal's spectrum with $W_l(u)$. As the width of the window decreases, the width of W_l grows, meaning that the spectrum will be more smeared. Thus, a large window is desirable for a sharp spectrum. However, a large window will compromise the localization ability of the spectrogram, as it will include components of the signal which are distant from the point of interest. In practice, we have found $n = 63$ to be satisfactory on discrete signals of length 512 (one image scan-line). We investigate a more sophisticated windowing technique in Section 7.1.

2.2 Wigner Distribution

An alternative method of calculating a joint space/frequency representation of a signal is the Wigner distribution. The Wigner distribution has been used in the computer vision community for both texture segmentation[RW90] and shape from texture[JC88]. For a one-dimensional function $f(x)$, the Wigner distribution is

$$W_f(x,u) = \int_{-\infty}^{\infty} f(x+\alpha/2) f^*(x-\alpha/2) e^{-2\pi u\alpha} d\alpha .$$

In words, the way to compute $W_f(x,u)$ is to first calculate the product $f(x+\alpha/2)f^*(x-\alpha/2)$, which is the original signal multiplied by a conjugated version of the original signal flipped around the point x. This product is Fourier transformed to get the WD at x. In practice, $f(x)$ is first windowed, leading to the pseudo-Wigner distribution (PWD) [CM80a]. The open questions pertaining to the window function for the spectrogram also apply to the PWD.

The WD generally works best on analytic signals, *i.e.* signals whose Fourier transforms contain no negative frequencies [Boa88]. It is fairly straightforward to calculate an analytic signal which corresponds to a real signal defined by samples. Thus, our two examples will be for analytic signals.

The example to which many WD advocates point is the WD of the chirp signal $f(x) = e^{j2\pi u_o x^2/2}$. This nonstationary, complex sinusoid is the analytic extension of $\cos(2\pi u_o x^2/2)$, whose instantaneous frequency is $u_o x$ (frequency proportional to x). The WD is

$$\begin{aligned} W_f(x,u) &= \int_{-\infty}^{\infty} e^{j2\pi u_o(x+\alpha/2)^2/2} e^{j2\pi u_o(x-\alpha/2)^2/2} e^{-j2\pi u\alpha} d\alpha \\ &= \int_{-\infty}^{\infty} e^{j2\pi u_o x\alpha} e^{-j2\pi u\alpha} d\alpha \\ &= \delta(u - u_o x). \end{aligned}$$

In (x,u) space, this is a δ-ridge which tracks at exactly the instantaneous frequency of $f(x)$. For any x, the position of the ridge is at $u_o x$, which is exactly what we would like to see for this signal.

Most textures are not simple sinusoids, however. They are, rather, sums of sinusoids in the sense of Fourier series. It is desirable that the joint representation show multiple frequency peaks at the constituent frequencies of the texture. This means that the representation should be linear – that the representation of the sum of two sinusoids should be the sum of the representations of the two sinusoids by themselves. Unfortunately, the WD is not linear. That is, $W_{f+g}(x,u) \neq W_f(x,u) + W_g(x,u)$. We show in Figure 6 the spectrogram (on the left) and the Wigner distribution (on the right) of a sum of two sinusoids. Let $f(x) = e^{j2\pi u_f x}$ and $g(x) = e^{j2\pi u_g x}$, both constant-frequency, complex sinusoids with frequencies u_f and u_g respectively. We have

$$\begin{aligned} W_f(x,u) &= \delta(u - u_f), \\ W_g(x,u) &= \delta(u - u_g), \\ W_{f+g}(x,u) &= W_f(x,u) + W_g(x,u) + 2\cos[2\pi x(u_f - u_g)]\, \delta\left(u - \frac{u_f + u_g}{2}\right). \end{aligned}$$

Thus the WD of a single, complex sinusoid is what we would expect, but the WD of a sum of sinusoids has a cross term. This term is a δ in u at the mean frequency of the two original sinusoids, modulated in x at a frequency which is the difference in frequencies of the two original sinusoids. The WD gives cross terms for every pair of constituent sinusoids. The cross term of the WD is clearly visible in Figure 6.

The analysis that follows in this paper depends on accurately finding the frequency peaks in the joint representation. Noise in some of the images complicates this task. The cross terms introduced by the WD

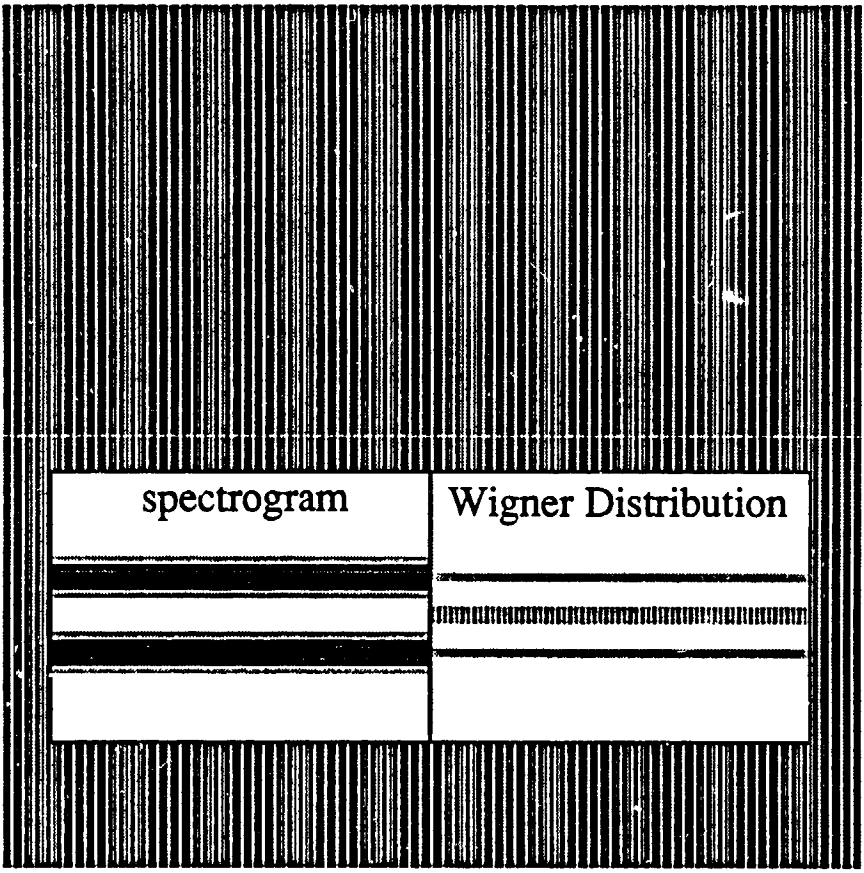

Figure 6: Spectrogram and Wigner distribution of two summed sinusoids

would make it even more difficult to distinguish the true frequency peaks. It is for this reason that we have chosen not to use the Wigner distribution.

The WD is just one member of a more general family of joint representations. Others [CW89][ZAM90], may be able to deal with nonstationarities as well as the WD while still suppressing cross terms. However, there does not exist a definitive method for calculating the space/frequency distribution.

3 Two-Dimensional Texture Segmentation With the Spectrogram

It is often the case that regions in an image can be grouped by their similarities in texture. In segmenting a road image, for example, it may be that the only common feature that the grassy areas share is texture, because the intensity and color of the grass in the image may be very different from shadowed to nonshadowed regions. By *two-dimensional* texture segmentation we mean segmentation on images with textures whose frequency does not change appreciably over the image. The textures must be viewed frontally; this is how almost all texture segmentation algorithms are tested.

The spectrogram of a structured texture shows that the spectrogram gives a clear, easily interpretable representation of the texture and a good idea of the texture's boundaries. In Figures 7 and 8 we present two pairs of textures along with the spectrograms of the rows indicated by the lines across the middle of the images. The smaller, left plate in Figure 7 has a sinusoidal intensity pattern, while the larger plate visible on the right has a square wave pattern. The left half of the spectrogram shows one peak in frequency which is constant with respect to position, as we expect from a sinusoidal intensity pattern. The right half of the spectrogram shows the fundamental frequency of the square wave pattern as the dark line near the bottom of the spectrogram along with fainter overtones at evenly spaced intervals above. The frequency of the square wave's first harmonic happens to be about equal to the frequency of the sinusoid on the left. The sharp transition between the two textures produces a short region in the spectrogram where nearly all frequencies are present. The light, vertical bars on the right half of the spectrogram are due to the interaction of the simulated pixels with the periodic pattern.

Figure 8 shows the same two plates with Brodatz textures superimposed. The complexity of the Brodatz images makes the spectrograms messier, but the representation is still easy to interpret. The white band at the bottom of the spectrogram has been zeroed to eliminate low frequency intensity variations due to lighting. We see that the scan line of the canvas texture on the left is close to sinusoidal since it has only one significant frequency component. The screen texture on the right has a lower fundamental frequency than the canvas as well as some overtones.

There have been many efforts aimed at 2D texture segmentation using windowed Fourier transforms, for instance [Gra73] and [Kir76]. These algorithms usually proceed by picking some set of features from Fourier space and then clustering using traditional pattern recognition techniques. The method has been compared to others both empirically [WDR76][DR76] and theoretically [CH80]. While the Fourier features performed adequately, they were outperformed by other statistical texture measures.

The advantages of Fourier texture measures over other statistical texture measures come from the variety of textures it can manage and the ease with which it can be extended to textures which are viewed obliquely. For structural textures, the Fourier transform approach requires no feature detection. Windowed Fourier transforms can be used for purely statistical textures, because Fourier transforms can bring out statistical coherence. In all textures, the spectra remain coherent over changes in shape, which means that the method can be smoothly extended to non-frontally viewed textures. In addition, the spectrogram is a

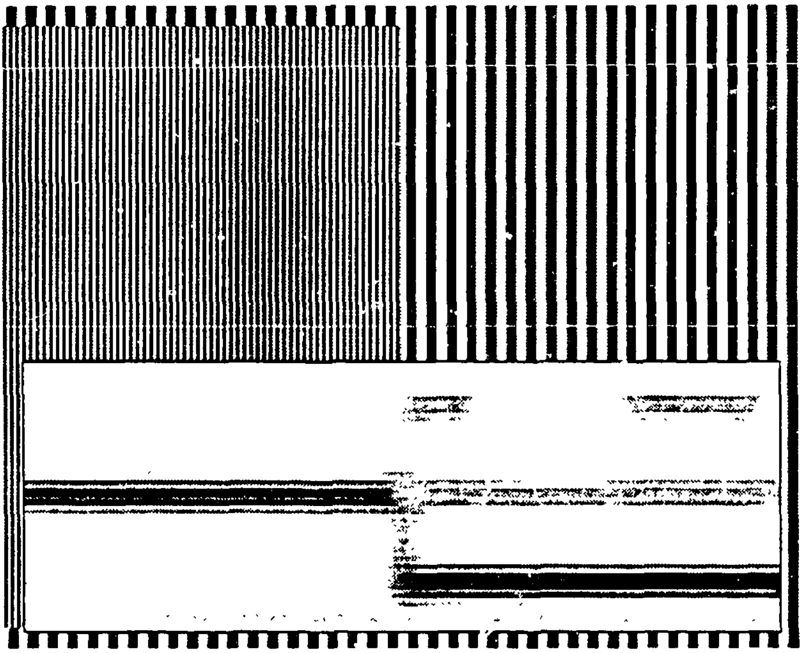

Figure 7: Two plates with sinusoidal and square wave gratings

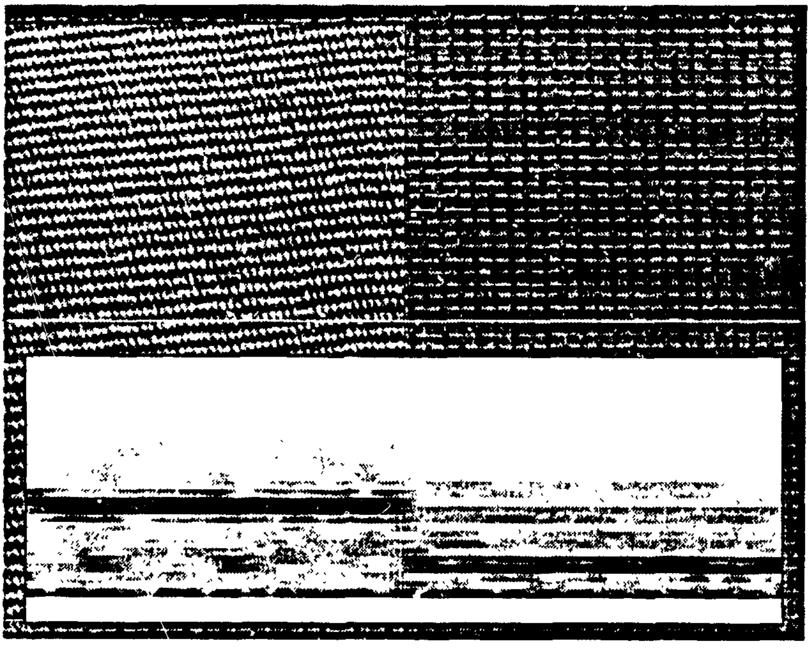

Figure 8: Two plates with Brodatz textures

powerful framework for analyzing many other scene phenomena and can be used to extract intrinsic scene characteristics. These intrinsic parameters provide another, more reliable basis for segmentation.

4 Three-Dimensional Shape and the Spectrogram

Texture is an important indication of 3D shape, and the connection has been studied extensively in computer vision. However, past efforts at exploiting this connection have been based on either the detection of explicit features or the computation of local statistics. The features and statistics are normally conceived in an *ad hoc* manner for the specific task of shape extraction. The spectrogram is a more natural choice for this kind of analysis, because the projected, local spatial frequencies on a textured surface change with the surface's depth and orientation, and because it is simple to account for other phenomena besides shape such as aliasing, defocus, and lens parameters.

In Figure 9 we show a plate receding into the distance with a sinusoidal intensity pattern superimposed. The spectrogram of the center scan line shows that the projected frequency increases as the plate recedes. This scene illustrates the effect of a *vanishing line*. Both the plane (from which the plate is taken) and the spectrogram asymptotically approach a line. The plane's asymptote is its vanishing line in the image. The corresponding frequency rises to infinity as it nears the vanishing line, as shown in the sketch of the ideal spectrogram. Before the plate reaches this point, the frequency has so grown that the actual spectrogram shows aliasing (see Section 5), which is the "fuzz" just to the left of the asymptote. The ideal spectrogram has no upper bound on frequency.

Figure 10 shows two plates meeting at a convex corner, each with a sinusoidal intensity pattern. The spectrogram shows how the projected pattern increases in frequency as the plates recede.

In Figures 11 and 12 we show the plates of Figures 7 and 8 rotated around a vertical axis. Both the fundamental frequencies and the overtones show the same effects of the change in orientation. In the following discussion, we describe how to quantitatively extract shape information from the spectrograms of textured surfaces by calculating the effect of depth and orientation on the spatial frequencies of the texture pattern.

4.1 Mathematical Formulation

The coordinate system and other quantities are defined as in Figure 13. The pinhole of a pinhole camera is placed at the origin of the right-handed (x_{3D}, y_{3D}, z_{3D}) coordinate system, looking along the $-z_{3D}$ axis. Objects are projected onto the image whose axes are (x, y). The pinhole-to-sensor distance is d, meaning that point (x_{3D}, y_{3D}, z_{3D}) will be projected onto the image plane at the point $(x, y) = (\frac{x_{3D}d}{-z_{3D}}, \frac{y_{3D}d}{-z_{3D}})$ under perspective. There is a surface in front of the camera whose depth is given by the function $c(x_{3D}, y_{3D})$. Superimposed on the surface is an intensity pattern given by $g(s, t)$, where (s, t) are coordinates of a coordinate system on the surface. We will ignore the y_{3D}, y, and t coordinates, in effect confining our attention to the x_{3D}-z_{3D} plane ($y_{3D} = 0$) and a 1D image plane in x.

On the x_{3D}-z_{3D} plane, a line runs in front of the camera whose equation is $x_{3D} \sin\theta + z_{3D} \cos\theta = -\rho$. [2] We will suppose that this line has a periodic pattern $g(s)$ superimposed on it. We will find the perspective projection of this pattern onto the image plane, and then calculate the instantaneous frequency of the projection so we

[2] In terms of traditional shape-from-texture notation (*c.f.* [Wit81]), the tilt angle here is always zero because we are working in only two dimensions, while θ is like the slant angle except that the slant angle cannot be negative and θ can be.

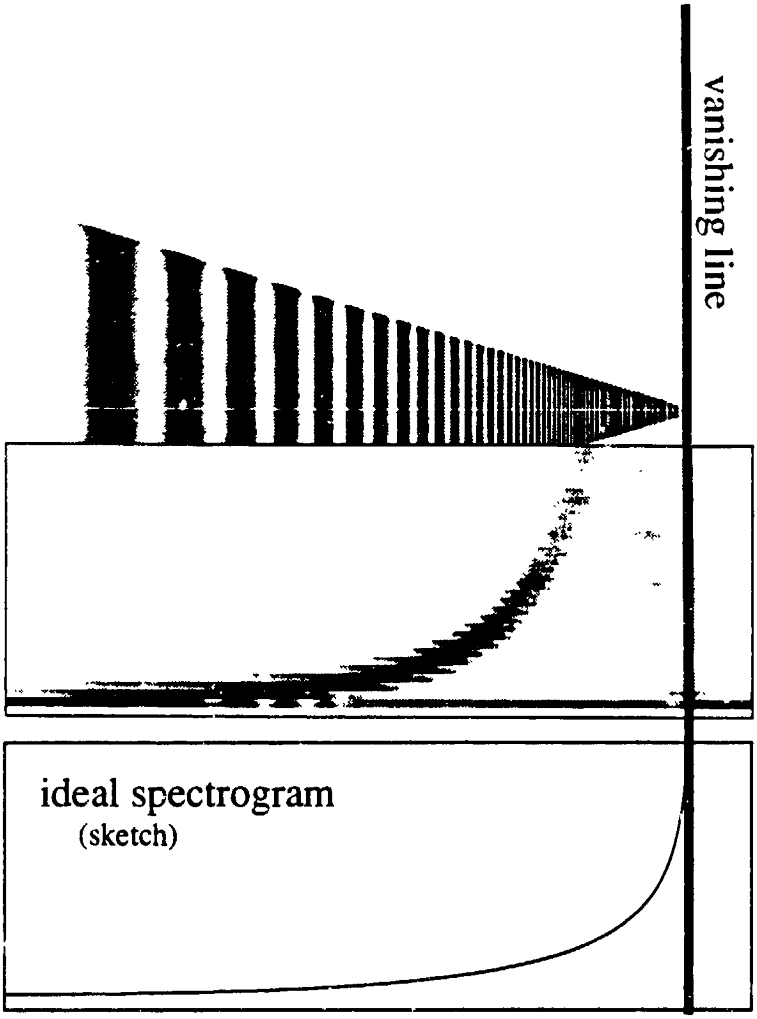

Figure 9: Plate with sinusoid receding to vanishing point

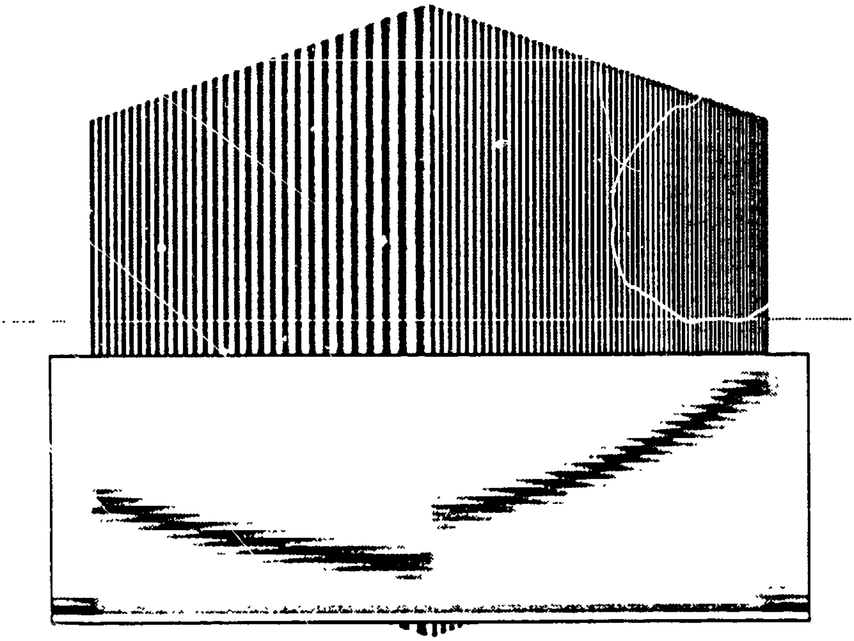

Figure 10: Two plates with sinusoids forming a convex corner

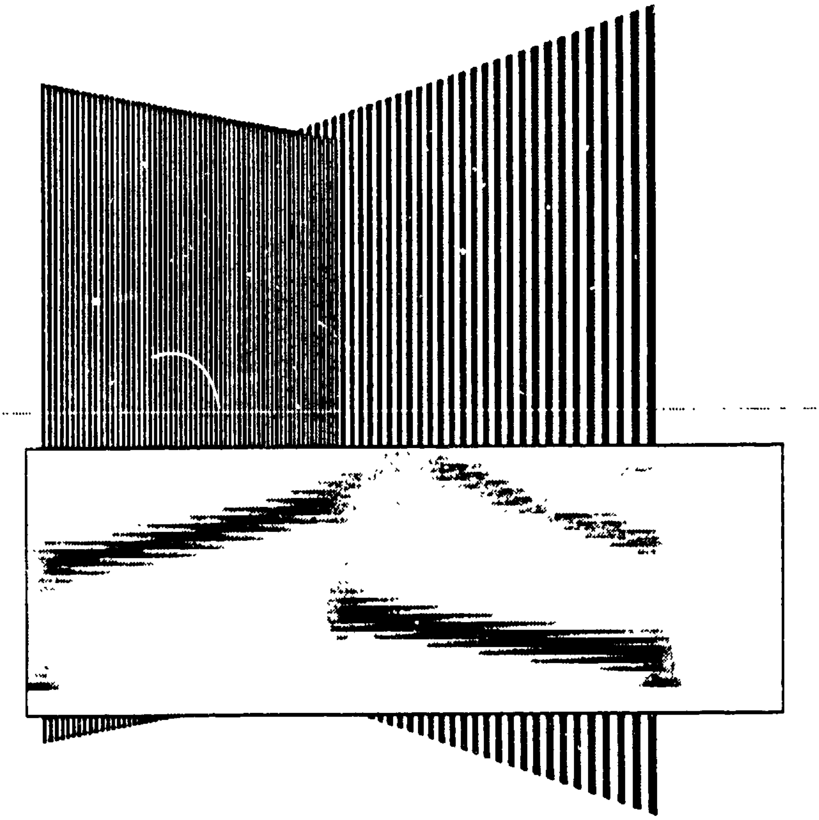

Figure 11: Two rotated plates with sinusoidal and square wave gratings

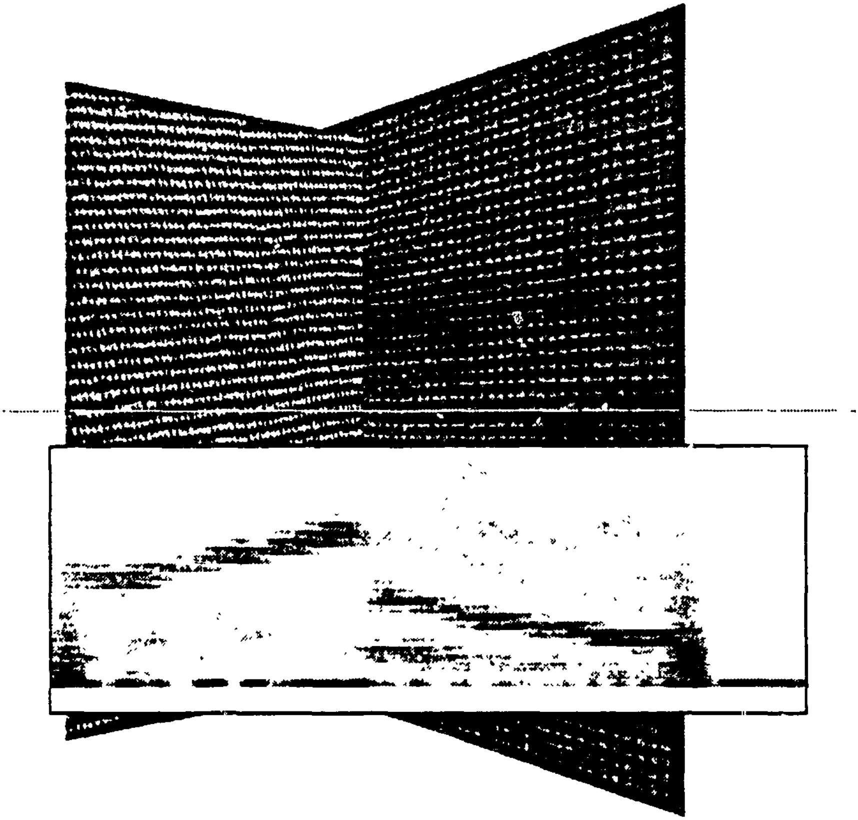

Figure 12: Two rotated plates with Brodatz textures

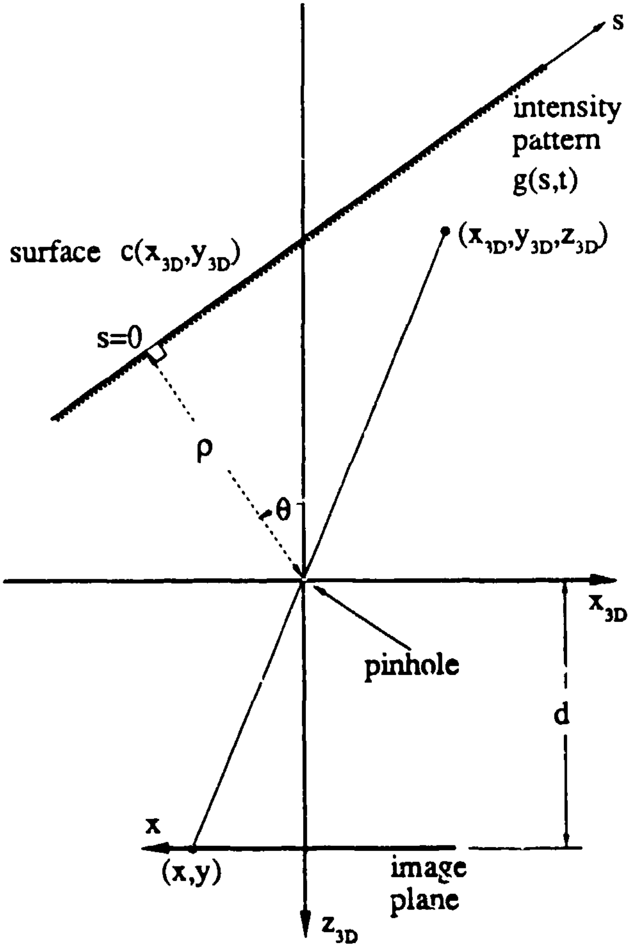

Figure 13: Geometry of 1D image formatucn through pinhole

can apply the spectrogram. We will find that the instantaneous frequency is a function of the orientation of the line, meaning that the spectrogram can be use to determine this parameter. Points on this line are parameterized by s, where $s = 0$ occurs at the intersection of the line and its perpendicular to the origin. Given an s, we have

$$(x_{3D}, z_{3D}) = (-\rho \sin\theta + s\cos\theta, -\rho\cos\theta - s\sin\theta)$$

which projects to

$$x = d\frac{-\rho\sin\theta + s\cos\theta}{\rho\cos\theta + s\sin\theta}.$$

Solving for s, we have have the position along the line for a given x on the image plane:

$$s(x) = \frac{-d\rho\sin\theta - x\rho\cos\theta}{x\sin\theta - d\cos\theta}. \tag{3}$$

Suppose that the line has superimposed on it a periodic reflectance pattern given by $g(s) = \cos(2\pi u_l s)$, such that the frequency of the pattern along the line is u_l. If the pattern is projected onto the image plane, we can write the equation of the projected pattern by replacing the s in $\cos(2\pi u_l s)$ with the equivalent value of s given in terms of x in Equation 3. Thus, the projected pattern on the image plane will be given by

$$\cos[2\pi u_l s(x)] = \cos\left[-2\pi u_l \rho \frac{d\sin\theta + x\cos\theta}{x\sin\theta - d\cos\theta}\right].$$

The instantaneous frequency, $u(x)$, of $\cos[2\pi u_l s(x)]$ is defined in the signal processing literature to be the derivative of the argument with respect to x, which is

$$u(x) = \frac{u_l \rho d}{(x\sin\theta - d\cos\theta)^2}. \tag{4}$$

The peak frequency in the spectrogram of the projected cosine will occur at approximately this frequency. In a computer vision application, the known quantities in Equation 4 are d (the pinhole-to-sensor distance), x (the pixel position), and $u(x)$ (the instantaneous frequency from the spectrogram). The unknowns are u_l (the frequency of the pattern along the line), and ρ and θ (the parameters of the line). Since u_l and ρ occur as a product in Equation 4, they cannot be distinguished from each other. This is a manifestation of a familiar effect: a small object (high frequency) at a small distance is indistinguishable from a large object (low frequency) at a large distance. Thus, we treat the product $u_l\rho$ as a single unknown. With θ as the other unknown, we can solve Equation 4 for θ and $u_l\rho$ if we have two or more measurements of $(x, u(x))$. The result is a space/frequency formulation of the shape-from-texture paradigm.

4.2 Extracting Shape from the Spectrogram

To demonstrate the use of Equation 4, we will determine parameters of the two plates in Figure 11 based on the spectrogram of the center row. We simplify the spectrogram to $u(x)$, the dominant frequency, determined

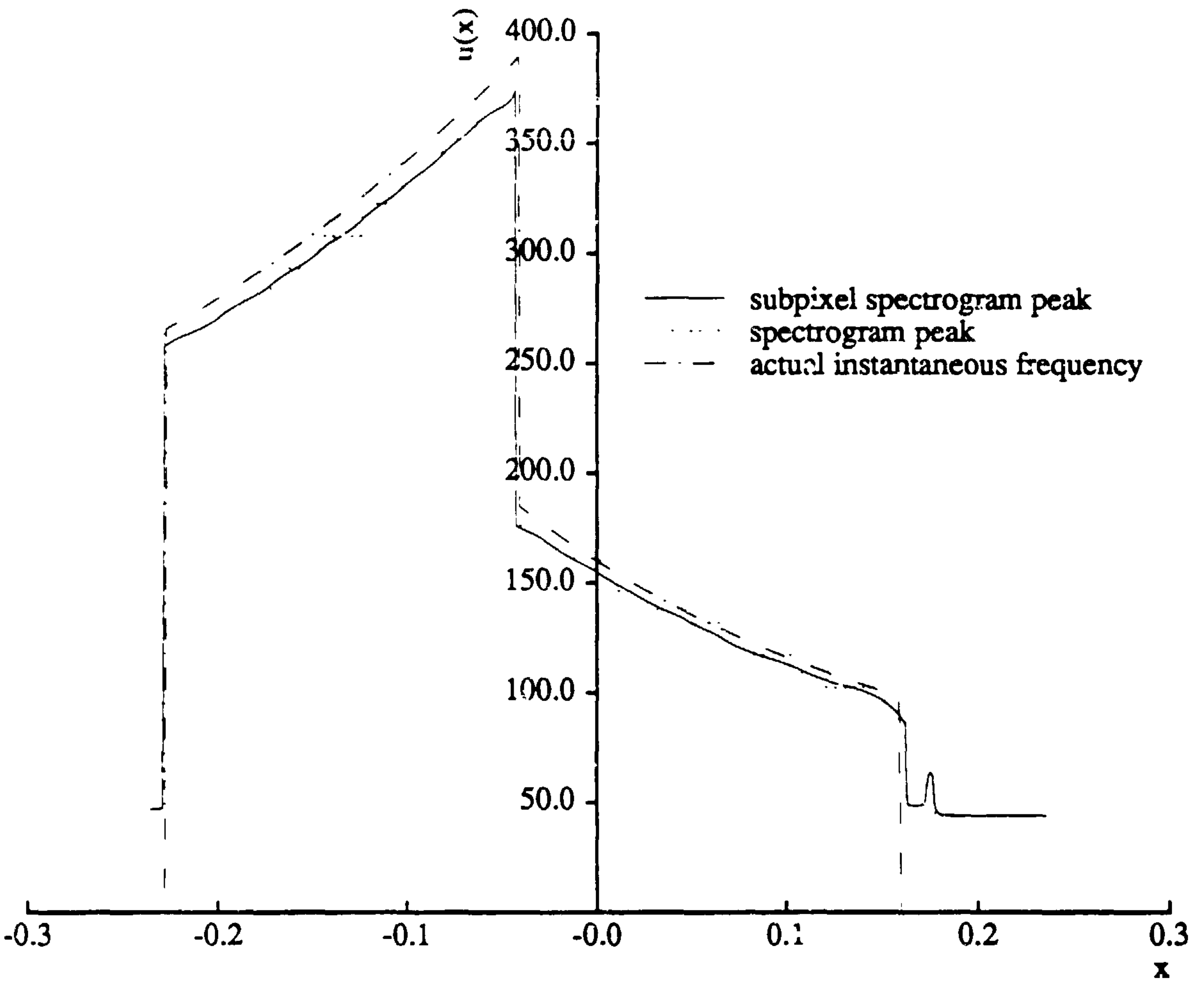

Figure 14: Peak frequencies from spectrogram of Figure 11

by finding the maximum value in each column of the spectrogram. These values are shown in Figure 14 as the dotted, stairstep-like line. The stairstep effect is due to the limited resolution of the DFT, which is in turn due to the limited size of the window used to calculate the spectrogram. This low resolution means that many adjacent points will appear to have equal instantaneous frequencies. If the instantaneous frequency of two adjacent points is equal, it implies that the surface is perpendicular to the line of sight, which is usually not the case. Thus, we calculate a "subpixel" value of the instantaneous frequency which gives better resolution than the raw DFT. We calculate the subpixel estimate by fitting a quadratic to the peak value and its two vertical neighbors and then finding the maximum of the quadratic. This is done for each column in the spectrogram. The higher resolution estimate is shown as the solid line in Figure 14. As a point of reference, we show the actual instantaneous frequencies (calculated from Equation 4) as the dash-dot line in the same figure. The estimate based on the spectrogram seems to consistently underestimate the actual frequency, and we are currently investigating the reason.

Each pair of $(x.\ u(x))$ values from the high-resolution spectrogram estimates can be used to calculate a value of $(u_{l}\rho.\ \theta)$. In order to reduce the effects of the wavering in the instantaneous frequencies, we calculate each $(u_{l}\rho.\ \theta)$ using five pairs of $(x.\ u(x))$'s placed symmetrically around the point of interest. We then segment the regions by histograming the $(u_{l}\rho.\ \theta)$'s, manually picking the peaks, and classifying each $(u_{l}\rho.\ \theta)$ pair by finding which peak it is closest to.

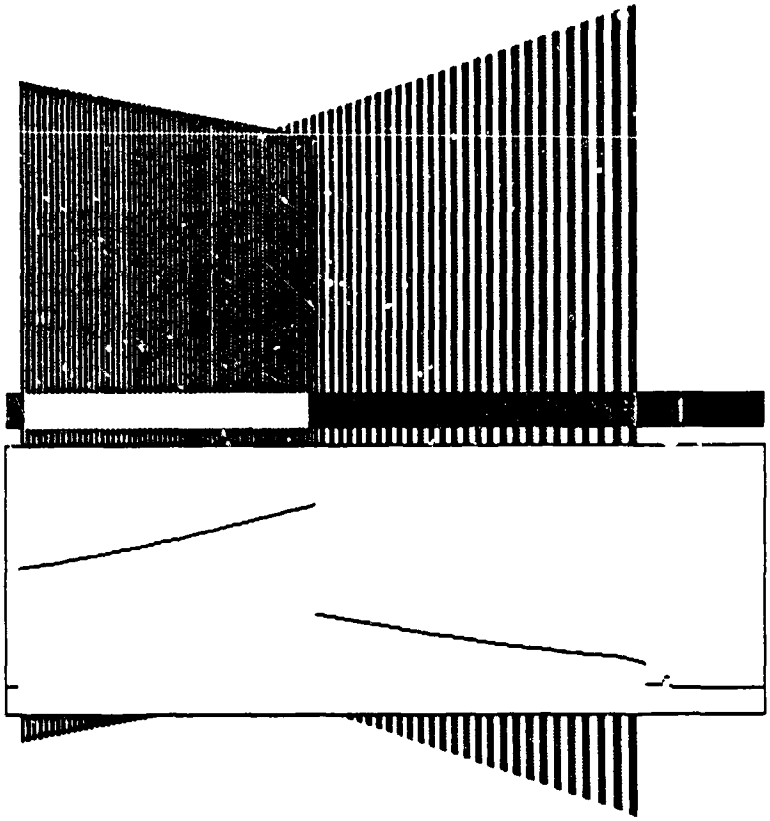

Figure 15: Segmentation of center row of rotated, patterned plates

The resulting segmentation is shown in Figure 15. The bar across the middle of the image indicates the regions, and we show the dominant instantaneous frequencies below. This segmentation works not only in spite of the changing frequencies across similar regions, but *because* of the changing frequencies as dictated by the mathematical projection of a single 3D plane onto a 2D image. In contrast to traditional region-grouping methods, note that this segmentation is based on reasoning about the uniformity of intrinsic properties of the scene, not merely the uniformity of a property in the image. In this sense, it is based on the "model coherence" approach developed for color image segmentation [SKKN90].

With the regions segmented, we calculate the best fit (u_f, ρ, θ) from Equation 4 based on the region's $(x, u(x))$'s using a gradient descent, minimization routine. The results are shown in Table 1. We know the actual values of the parameters from the graphics routine used to generate the images. In this example the errors are quite small.

We performed the same analysis for the textured plates in Figure 12. The results of the segmentation are shown in Figure 16. This segmentation is not as good as for the other set of plates. Much of the error occurs near the boundaries of the plates where the Fourier transform window contains only part of one of the textures or some of both. The other misclassified areas occur in regions where the instantaneous frequency value has unusual dips or wiggles. Possible solutions to this problem are using a spectral estimator which accounts for noise, or averaging the dominant frequencies from the spectrograms of neighboring points. Also, using a variable-sized window as described in Section 7.1 may help alleviate the problem. The performance figures in Table 1 are based on a manual (perfect) segmentation of the instantaneous frequencies for the rotated, textured plates of Figure 12. The line parameters were calculated with the same gradient descent method used for the plates in Figure 11.

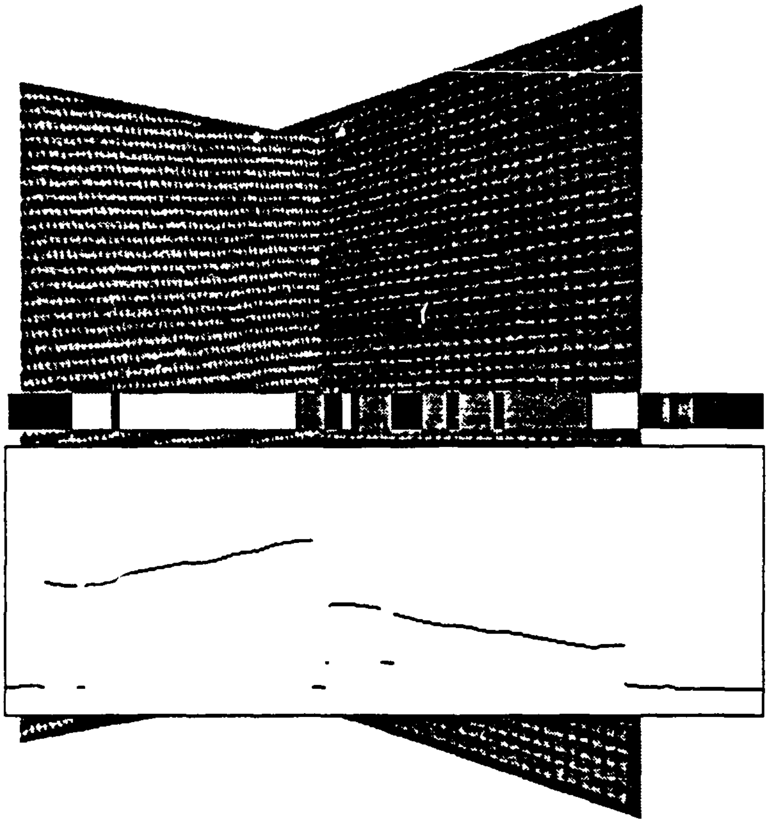

Figure 16: Segmentation of center row of rotated, textured plates

	From Figure 11 Periodic Pattern semi-automatic segmentation				From Figure 12 Brodatz Textures manual segmentation			
	Left Plate		Right Plate		Left Plate		Right Plate	
	$u_l\rho$	θ	$u_l\rho$	θ	$u_l\rho$	θ	$u_l\rho$	θ
actual	177.25	50.00^o	40.00	-60.00^o	152.1	50.00^o	47.0	-60.00^o
calculated	172.92	49.75^o	39.31	-59.72^o	141.37	50.82^o	48.27	-58.85^o
error	-2.4%	-0.25^o	-2.4%	0.28^o	-7.1%	0.82^o	2.7%	1.15^o

Table 1: Actual and calculated line parameters

4.3 Other Shapes

This method could be extended to other shapes in two different ways. Above we presented a method in which the instantaneous frequencies are fit to a known class of shapes (lines) in order to derive the parameters of the shape. The parameters were those which best fit Equation 4, which describes the instantaneous frequencies on a line. Other equations could be derived which relate instantaneous frequencies to any parameterized shape. Given some *a priori* knowledge of the shapes in the scene, the spectrogram peaks (as well as overtones) could be used to instantiate the shapes' parameters. Alternatively, a program could calculate local surface normals by using the instantaneous frequencies from a small neighborhood along with an equation which relates frequency and surface normal.

Although this method and results are meant to be only illustrative, they show the power of the spectrogram for reasoning about the effects of 3D shape in images. The spectrogram is a simple, natural method of quantifying the relationship between texture and shape, and it requires no feature detection except for finding frequency peaks.

5 Aliasing

Aliasing occurs when a signal is sampled at a rate less than twice its maximum frequency, causing lower-frequency artifacts to appear in the sampled signal. This phenomenon can often be seen on television in images of periodic patterns like striped clothes, automobile grills, or tall buildings. In two dimensional imaging, these artifacts are called *moire patterns*, and they can lead to insidious problems in machine vision, *e.g.* stereo matching errors [Mat89](p. 117). This is because the patterns cannot be detected in single images without detailed *a priori* knowledge of the scene, meaning that in most situations there is no hope of recovering the true signal.

The DFT of such a signal does not give a true indication of the original signal's frequency content. The DFT can only show frequencies up to and including the Nyquist frequency (one half of the sampling frequency). Frequencies higher than the Nyquist frequency are "aliased down" into lower frequencies of the DFT.

This is illustrated in Figure 17, which shows a plate with a sinusoidal intensity pattern rotated to the right. Beginning at the left of the plate, the spectrogram shows that the instantaneous frequency is rising as the plate recedes into the distance. At a little less than halfway across the spectrogram, the peak frequency has risen to the top of the spectrogram, which corresponds to the Nyquist frequency. Although the actual frequency on the image plane continues to rise, it appears to decrease after the Nyquist rate has been exceeded. In this region of the image, moire patterns begin to appear as lower-frequency variations caused by the beating of the signal frequency against the sampling frequency. There is another "bounce" on the spectrogram after the apparent peak frequency has fallen to zero. This bouncing would continue if the plate were longer. If the signal had overtone frequencies, these will bounce also, although not at the same places as the fundamental or other overtones. This is shown in Figure 18, which is a plate whose intensity pattern is the sum of two sinusoids. Below we examine the mathematics of the bouncing frequencies and show how the spectrogram provides an elegant basis for analyzing these artifacts.

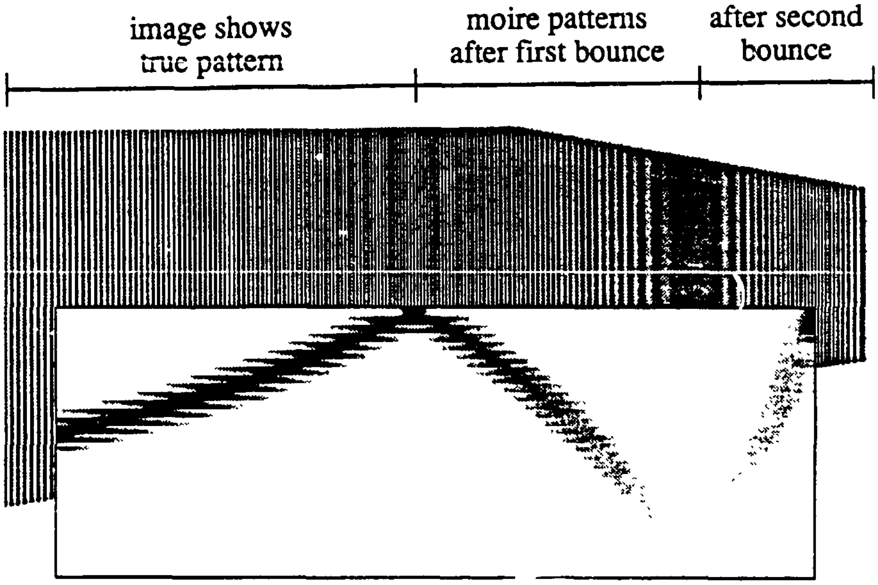

Figure 17: Plate with sinusoid showing aliasing

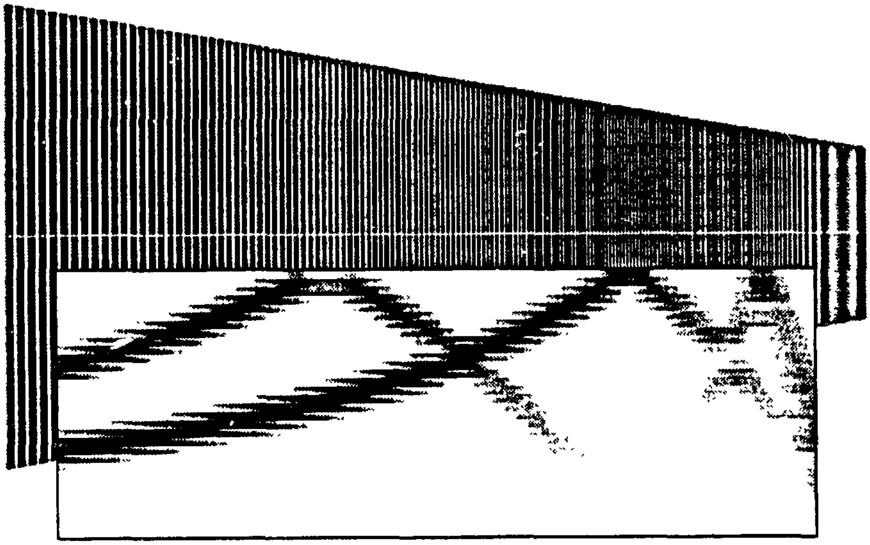

Figure 18: Plate with sum of two sinusoids showing aliasing

5.1 Bouncing Frequencies

In this section we discuss the mathematics of aliasing and how it produces bouncing in the spectrogram. We will demonstrate the effect using a simple cosine wave, although the ideas are generally applicable. The effect is most easily visualized in the Fourier domain, so we will develop the equations in the spatial and spatial frequency domains in parallel.

Suppose the original, continuous signal is a cosine of frequency u_o cycles per unit distance.

$$f(x) = \cos(2\pi u_o x)$$

Its Fourier transform is two delta functions placed symmetrically around the frequency origin.

$$F(u) = \frac{1}{2}\left[\delta(u + u_o) + \delta(u - u_o)\right]$$

Sampling at a frequency of u_s is modeled as multiplication by a series of δ's spaced at intervals of $1/u_s$. The sampled signal, f_s, is

$$\begin{aligned} f_s(x) &= f(x) \sum_{i=-\infty}^{\infty} \delta(x - \frac{i}{u_s}) \\ &= \cos(2\pi u_o x) \sum_{i=-\infty}^{\infty} \delta(x - \frac{i}{u_s}) \end{aligned}$$

The corresponding operation in the Fourier domain is convolution with the Fourier transform of the space-domain δ's.

$$\begin{aligned} F_s(u) &= F(u) * u_s \sum_{i=-\infty}^{\infty} \delta(u - iu_s) \\ &= \frac{1}{2}\left[\delta(u + u_o) + \delta(u - u_o)\right] * u_s \sum_{i=-\infty}^{\infty} \delta(u - iu_s) \\ &= \frac{u_s}{2}\left[\sum_{i=-\infty}^{\infty} \delta(u + u_o - iu_s) + \sum_{i=-\infty}^{\infty} \delta(u - u_o - iu_s)\right]. \end{aligned}$$

$F_s(u)$, the Fourier domain version of the sampled cosine wave, is illustrated in Figure 19-a. It consists of the Fourier transform of the cosine repeated at intervals of u_s, the sampling frequency. These repeated Fourier transforms are called *spectral orders*. Spectral order $o_s = \{\ldots -2. -1.0.1.2\ldots\}$ is centered at frequency $o_s u_s$.

In order to recover an estimate of the original signal from the samples, the Fourier domain representation is multiplied by a rectangle function to extract one repetition of the repeated transforms. (It is also scaled by $\frac{1}{u_s}$ to recover the original amplitude.) The rectangle function, also shown in Figure 19-a, is cut off at the positive and negative Nyquist frequencies. This corresponds to interpolation with a sinc function in the spatial domain. Thus, the reconstructed signal becomes

$$\begin{aligned} f_r(x) &= f_s(x) * \mathrm{sinc}(u_s x) \\ &= \left[\cos(2\pi u_o x) \sum_{i=-\infty}^{\infty} \delta\left(x - \frac{i}{u_s}\right) \right] * \mathrm{sinc}(u_s x) \\ &= \sum_{i=-\infty}^{\infty} \cos(2\pi u_o i/u_s) \mathrm{sinc}\left[u_s(x - i/u_s)\right] . \end{aligned}$$

where $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

In the Fourier domain,

$$\begin{aligned} F_r(u) &= \frac{1}{u_s} \mathrm{rect}\left(\frac{u}{u_s}\right) F_s(u) \\ &= \frac{1}{2} \mathrm{rect}\left(\frac{u}{u_s}\right) \left[\sum_{i=-\infty}^{\infty} \delta(u + u_o - iu_s) + \sum_{i=-\infty}^{\infty} \delta(u - u_o - iu_s) \right] . \end{aligned}$$

where

$$\mathrm{rect}\left(\frac{x}{b}\right) = \begin{cases} 0 & \text{if } |\frac{x}{b}| > \frac{1}{2} \\ \frac{1}{2} & \text{if } |\frac{x}{b}| = \frac{1}{2} \\ 1 & \text{if } |\frac{x}{b}| < \frac{1}{2} \end{cases}$$

is a rectangle with support length b.

As shown in the top graph of Figure 19, if $|u_o| < \frac{u_s}{2}$, the original cosine can be recovered exactly. We illustrate in both Figure 19 and 20 what happens as the frequency of the original signal rises past the Nyquist frequency. Figures 19a-d show "side views" of the situation for various, increasing values of u_o from the top down. The horizontal arrows indicate which direction the δ's will move with increasing u_o. Figure 20 shows a "top view" as u_o increases linearly from left to right. The spectrogram has been shaded. The four vertical cuts in this figure correspond to the four situations shown in Figure 19.

In Figure 19-b, the cosine's frequency has exceeded the Nyquist rate, and δ's from neighboring spectral orders have moved into the the interpolation rectangle. We show how the various δ's correspond with the dashed lines drawn from graph to graph. The apparent effect of a rise in u_o is a bounce in frequency, which is more apparent in Figure 20. Just as the outgoing δ's leave the interpolation rectangle, incoming δ's enter, moving toward the frequency origin. These two incoming δ's continue past each other, producing another bounce in apparent frequency, as shown in Figure 19-c. When these δ's leave, they are replaced by two more, as in Figure 19-d, and the process continues on and on. This process causes the apparent bouncing in the spectrogram illustrated in Figure 20.

In Table 2 we illustrate with equations what is happening in each of the four subfigures of Figure 19. We label each situation with o_s, the spectral order which contributes the δ in the positive half of the interpolation

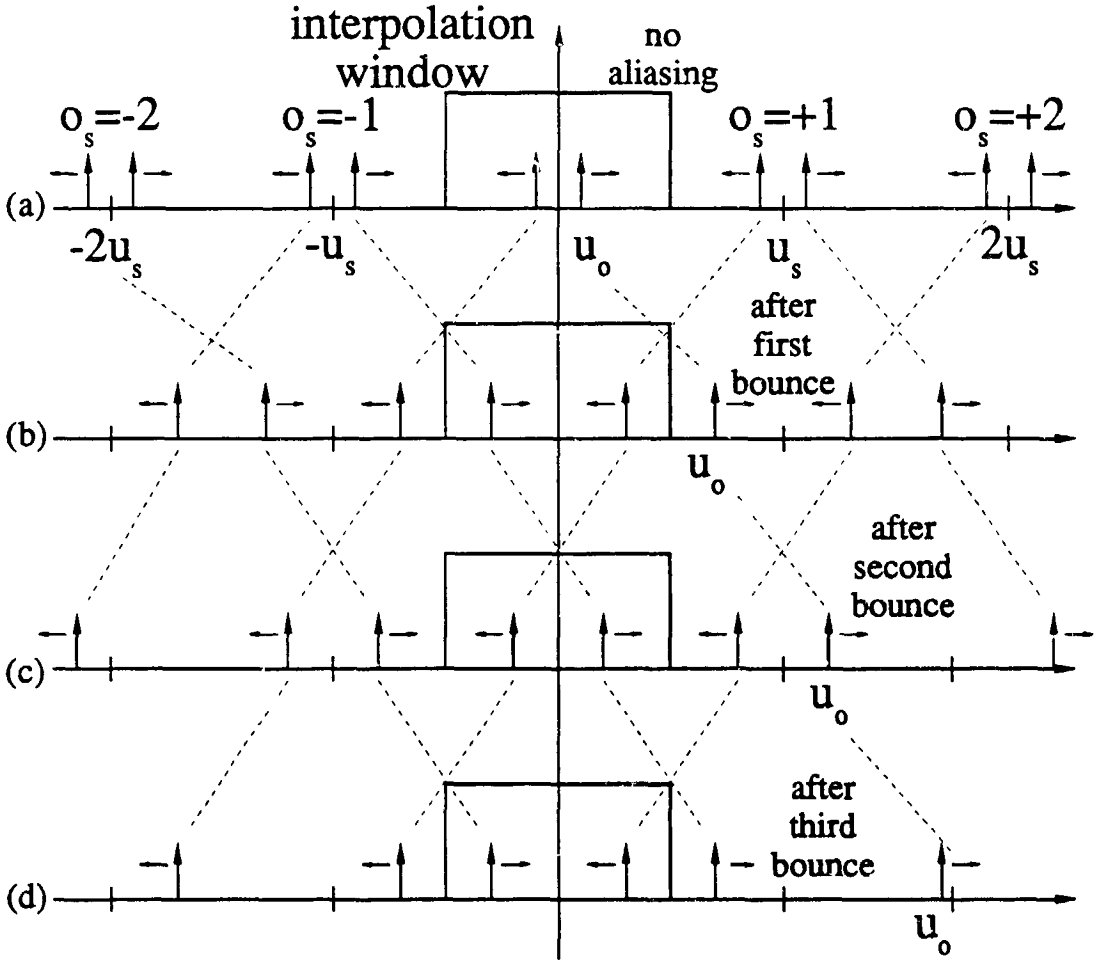

Figure 19: Aliasing causing bouncing, u_o is increasing from the top graph down

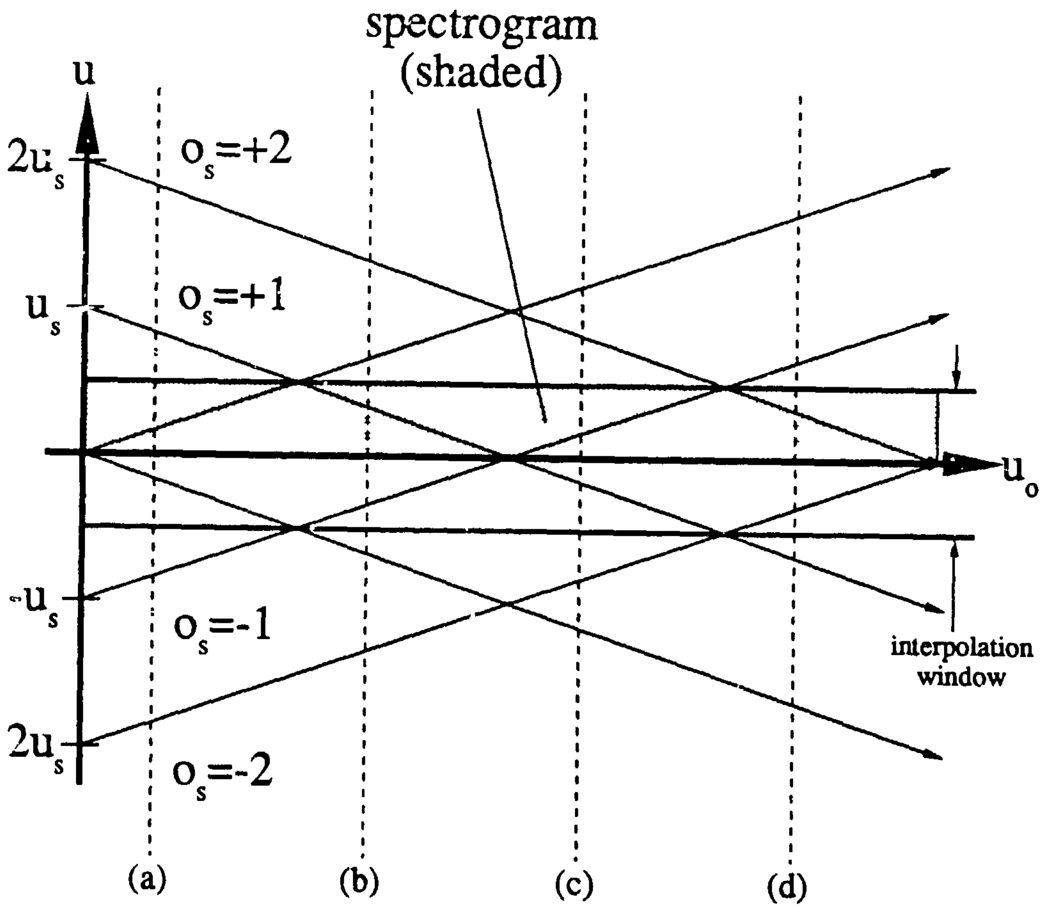

Figure 20: Aliasing causing bouncing, u_o is increasing from left to right

u_o	o_s	frequency domain reconstruction	space domain reconstruction
(a) $0 \le u_o \le u_s/2$	0	$\frac{1}{2}\left[\delta(u+u_o)+\delta(u-u_o)\right]$	$\cos[2\pi u_o x]$
(b) $u_s/2 \le u_o \le u_s$	+1	$\frac{1}{2}\left[\delta(u+u_s-u_o)+\delta(u-u_s+u_o)\right]$	$\cos[2\pi(u_s-u_o)x]$
(c) $u_s \le u_o \le 3u_s/2$	-1	$\frac{1}{2}\left[\delta(u-u_s+u_o)+\delta(u+u_s-u_o)\right]$	same as above
(d) $3u_s/2 \le u_o \le 2u_s$	+2	$\frac{1}{2}\left[\delta(u+2u_s-u_o)+\delta(u-2u_s+u_o)\right]$	$\cos[2\pi(2u_s-u_o)x]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(o_s-\frac{1}{2})u_s \le u_o \le o_s u_s$	$o_s > 0$	$\frac{1}{2}\left[\delta(u+o_s u_s-u_o)+\delta(u-o_s u_s+u_o)\right]$	$\cos[2\pi(o_s u_s-u_o)x]$
$-o_s u_s \le u_o \le (-o_s+\frac{1}{2})u_s$	$o_s < 0$	$\frac{1}{2}\left[\delta(u-o_s u_s+u_o)+\delta(u+o_s u_s-u_o)\right]$	same as above

Table 2: Analytic expressions of Figure 19

window in frequency space. In (a), $o_s = 0$, and the cosine's frequency is below the Nyquist frequency, so the reconstruction is true to the original signal. In (b) the reconstruction is based on one δ from each of the two closest neighboring spectral orders, and $o_s = +1$. The reconstructed signal is $\cos[2\pi(u_s - u_o)x]$. Since $u_o \le u_s$ in this case, an increase in u_o (the original signal's frequency) will cause a *decrease* in the frequency of the reconstructed signal. In (c) no new δ's are introduced, but the two δ's pass each other. Thus, in (c) $o_s = -1$. The reconstructed signal is $\cos[2\pi(-u_s + u_o)x]$, which is the same as case (b) (because $\cos(-t) = \cos(t)$). However, in (c) $u_o \ge u_s$, so an increase in u_o causes an *increase* in the frequency of the reconstructed signal. The transition from (c) to (d) is like the transition from (a) to (b), thus the frequency of the reconstructed signal decreases again with increasing u_o. In general, the frequency of the reconstructed cosine is given by

$$u = \begin{cases} u_o & \text{if } o_s = 0 \\ o_s u_s - \operatorname{sgn}(o_s) u_o & \text{otherwise.} \end{cases} \tag{5}$$

where o_s is the spectral order contributing a δ to the positive half of the interpolation function, u_s is the sampling frequency, and

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ +1 & \text{if } x > 0. \end{cases}$$

5.2 Unfolding the Spectrogram

Of course, it would be better to have no aliasing in the spectrogram. We could then get an accurate idea of the true signal at every point. We can think of the spectrogram as a distorted, windowed version of an ideal, space/frequency representation – the ideal spectrogram. The ideal spectrogram's frequency axis extends from zero to infinity, and it does not suffer from aliasing. We can see from the analysis in the previous subsection that the actual spectrogram of a simple sinusoid whose frequency is changing is a folded version of the ideal spectrogram. This is illustrated in Figure 21. The folds occur at positive, integer multiples of the Nyquist frequency, $u_s/2$. In the ideal spectrogram, the frequency peak continues to grow with the frequency of the underlying signal, while in the actual spectrogram aliasing causes the apparent frequency to bounce between zero and the Nyquist frequency.

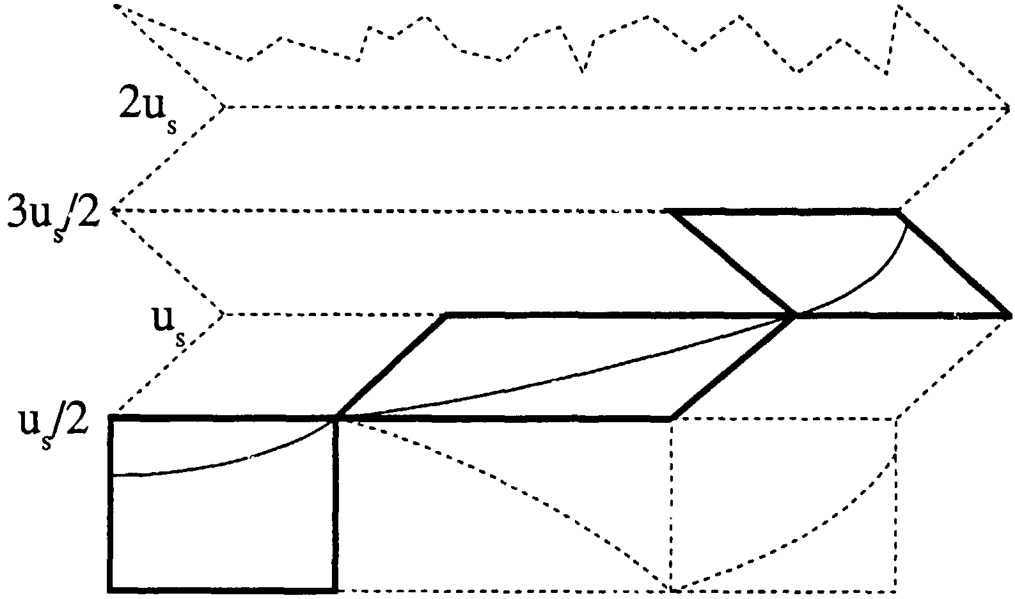

Figure 21: Folding the ideal sp͏ectrogram to show aliasing

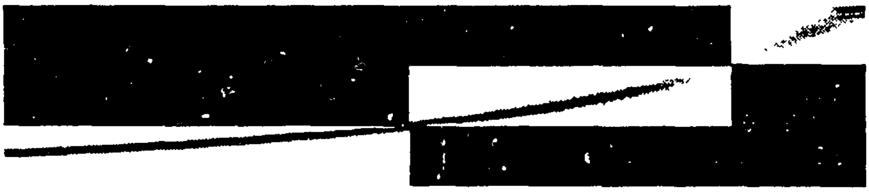

Figure 22: Unfolded version of spectrogram in Figure 17

In Figure 22 we show an unfolded version of the spectrogram in Figure 17. The unfolded spectrogram gives a true indication of the signal's frequency, even beyond the Nyquist limit. Unfolding the spectrogram of a signal with overtones, like that in Figure 18, would not be as simple. Multiple peaks in the same column may come from different folds of the ideal spectrogram. The key is to determine which fold a given peak came from. In the next section, we propose an algorith for this based on computer-controlled zooming of the lens.

6 Lens Parameters and the Spectrogram

Much research in "active vision" concerns the control of the three lens parameters: zoom, focus, and aperture. We show in this section how these parameters affect the spectrogram, which in turn provides new insights into how they affect the image. This point of view leads to algorithms which let us deduce intrinsic scene parameters by purposefully altering the lens settings.

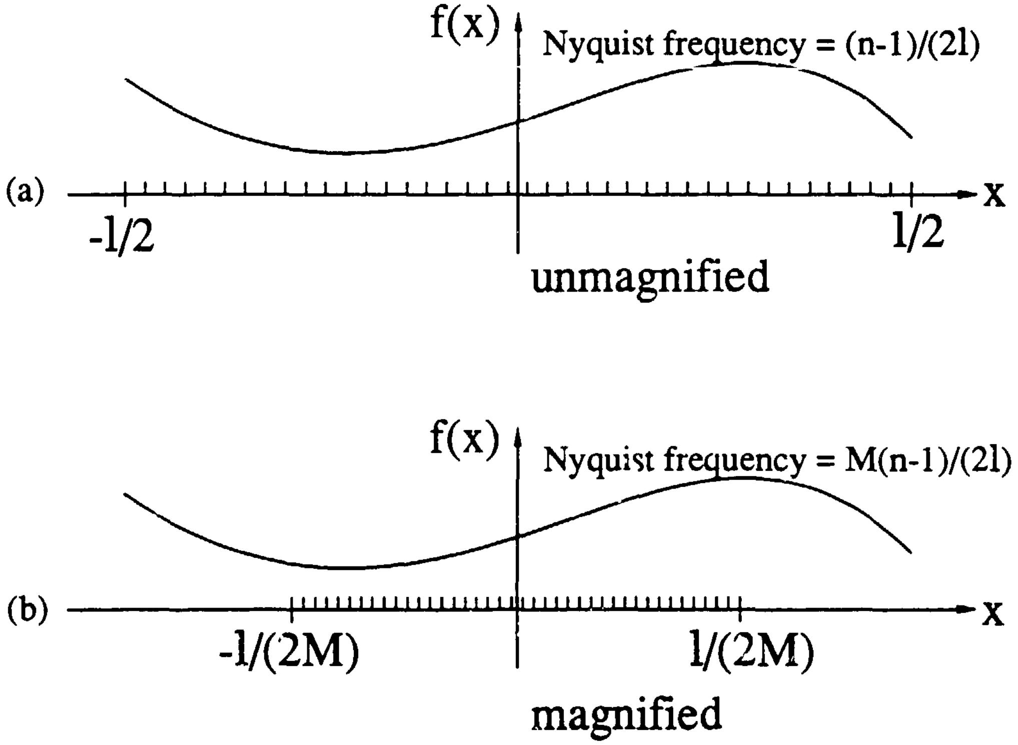
f(x)
Nyquist frequency = (n-1)/(2l)
(a)
-1/2
1/2
x
unmagnified
f(x)
Nyquist frequency = M(n-1)/(2l)
(b)
-1/(2M)
1/(2M)
x
magnified

Figure 23: Effect of zooming on imaged signal

6.1 Zoom

6.1.1 How Zoom Affects the Spectrogram

In *equifocal* camera lenses (such as most one-touch zoom lenses) a change in zoom can be modeled as simply a change in magnification. We can imagine the situation in Figure 23-a where the section of the signal which falls on the center window of the spectrogram extends from $\frac{-l}{2}$ to $\frac{l}{2}$. We will arbitrarily call the magnification here one, and we will say that the entire portion of the signal seen by the camera is of length L. Both l and L are measured on the image plane. If there are n pixels in the spectrogram window, the sampling frequency is $\frac{n-1}{l}$ pixels per unit distance, making the Nyquist frequency $\frac{n-1}{2l}$. Since the spectrogram extends in frequency from zero to the Nyquist frequency, the spectrogram resulting from this signal will cover the region indicated by the short, wide box in Figure 24.

If the magnification M is changed, a larger or smaller portion of the original signal will be contained by each window. In Figure 23-b we have indicated the effect of an increase in magnification, showing how a smaller part of the signal is now imaged. The section of the signal which falls on the central window now extends from $\frac{-l}{2M}$ to $\frac{l}{2M}$, and the entire signal seen by the camera covers $\frac{-L}{2M}$ to $\frac{L}{2M}$. The magnified window is spread out over the same number of pixels as before, so the Nyquist frequency is now $\frac{M(n-1)}{2l}$ pixels per unit distance.

The spectrogram after the magnification change is shown in Figure 24. For an increase in magnification, the spectrogram covers more in frequency but less in space. The "area" of the spectrogram (actually a unitless quantity, "spatial dynamic range") is $\frac{L(n-1)}{2l}$ and is independent of the magnification. Thus for changes in

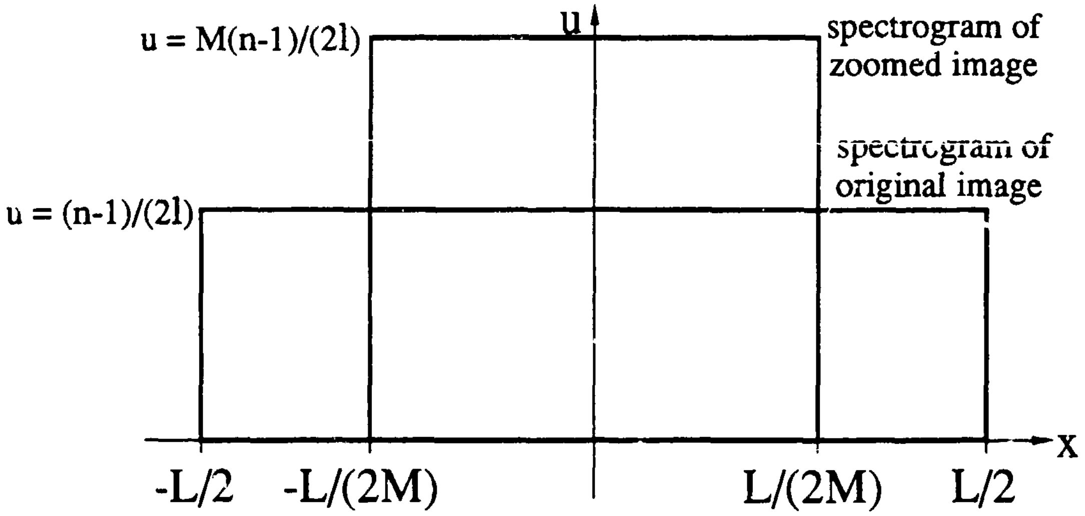

Figure 24: Effect of zooming on spectrogram dimensions

zoom, there is a direct tradeoff between coverage in spa e and spatial frequency. These arguments also apply to the four-dimensional hypervolume of the spectrogram of a two-dimensional signal.

6.1.2 Dealiasing With Zoom Changes

A slight change in zoom can be used to find the true, unaliased frequency of a sinusoid, because aliased frequencies from different spectral orders respond differently to changes in magnification. Since image textures can be decomposed into simple sinusoids, we could use two images taken at slightly different zoom settings to dealias texture images.

Suppose as above that we have a 1-D image of a cosine of frequency u_o cycles/pixel sampled at a rate of u_s cycles/pixel. The cosine may be sampled above or below the Nyquist rate. Referring to Figure 20, we can see there will be only one spectral order contributing a δ to the spectrogram (because the spectrogram only shows positive frequencies up to $u_s/2$). The apparent frequency of the unmagnified ($M = 1$) signal, u_1, is given by Equation 5, *i.e.*

$$u_1 = \begin{cases} u_o & \text{if } o_s = 0 \\ o_s u_s - \operatorname{sgn}(o_s) u_o & \text{otherwise.} \end{cases} \tag{6}$$

If the lens is zoomed slightly such that the magnification is changed to M, the sampling frequency (measured in cycles/pixel of the *unmagnified* image) will be Mu_s cycles/pixel, where u_s is the sampling frequency on the unmagnified image. The apparent frequency of the cosine will then be

$$u_2 = \begin{cases} u_o & \text{if } o_s = 0 \\ o_s M u_s - \operatorname{sgn}(o_s) u_o & \text{otherwise.} \end{cases} \tag{7}$$

We can eliminate u_o from Equations 6 and 7 by subtracting. Solving this difference for o_s gives

$$o_s = \frac{u_2 - u_1}{u_s(M-1)}.$$

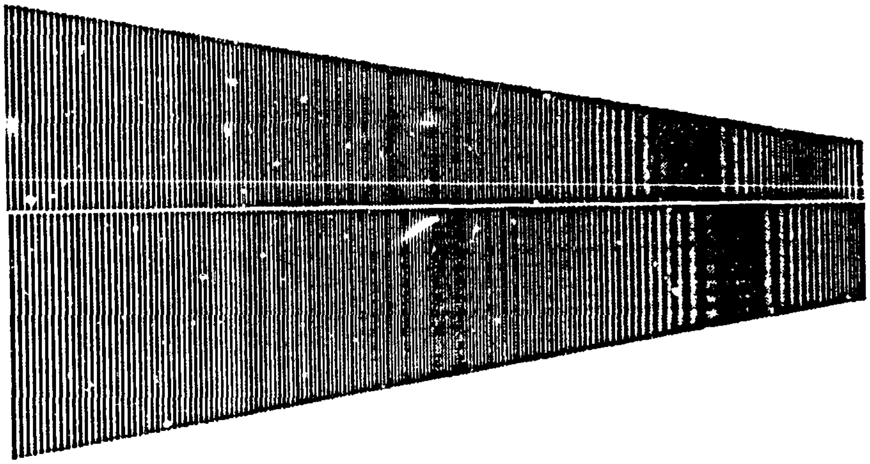

Figure 25: Horizontally split image of aliased plate and magnified aliased plate

We note that this equation applies for both $o_s = 0$ and $o_s \neq 0$. Thus, the difference in apparent frequency between the two images is proportional to the spectral order o_s. After solving for o_s, we can use Equation 6 or 7 to solve for u_o, which is the true frequency of the signal. The dealiasing does not require the solution of a correspondence problem, since the two signals are related by a simple difference in magnification.

An implicit assumption here is that o_s remains the same in both images. This will be true for small changes in magnification unless the $\hat{o}$ is very close to either extreme of the interpolation window and the zoom change causes it to be replaced by another $\hat{o}$.

We have applied this technique to the image of the receding plate in Figure 17. We show a split version of the image in Figure 25. On the top is the unmagnified image, and on the bottom is the same image magnified by $M = 1.075$. It is easily seen how the moire patterns shift. Figure 26 shows the "subpixel" frequency peaks from the spectrograms of the center rows of the two images. The frequency data from the magnified image has been adjusted so it is shown in terms of the space and frequency units of the unmagnified image. The dotted line shows the dealiased frequency based on the technique outlined above. Except for the glitches at the frequency extremes, the figure shows correctly the dealiased frequency. Thus, the spectrogram has been dealiased without detailed *a priori* knowledge of the scene.

6.2 Focus and Aperture

Changes in the lens' focus and aperture combine to change the point spread function (psf) of the lens, which can be easily visualized with the spectrogram. (The psf is a function which can be convoluted with an ideal, sharp signal to model the effects of blur.) In general, points in sharper focus will show more high frequencies than if they are blurred. A smaller aperture tends to have the same general effect as sharper focus. In fact, in the pinhole model we have been using (Figure 13), the aperture is infinitesimally small, meaning that every point in the scene is in perfect, sharp focus.

We will generalize the pinhole model by introducing a single, thin lens with a variable aperture as shown in Figure 27. The aperture of the lens is a, the focal length of the lens is b, and the distance to the image plane remains d. We can approximate the effects of focus and aperture with geometric optics. Each point in the scene with a different value of z_{3D} will be in sharp focus at only one point behind the lens. This point, z, is given by the Gaussian Lens Law: $\frac{1}{z} + \frac{1}{-z_{3D}} = \frac{1}{b}$. If the image plane is not at the proper distance behind

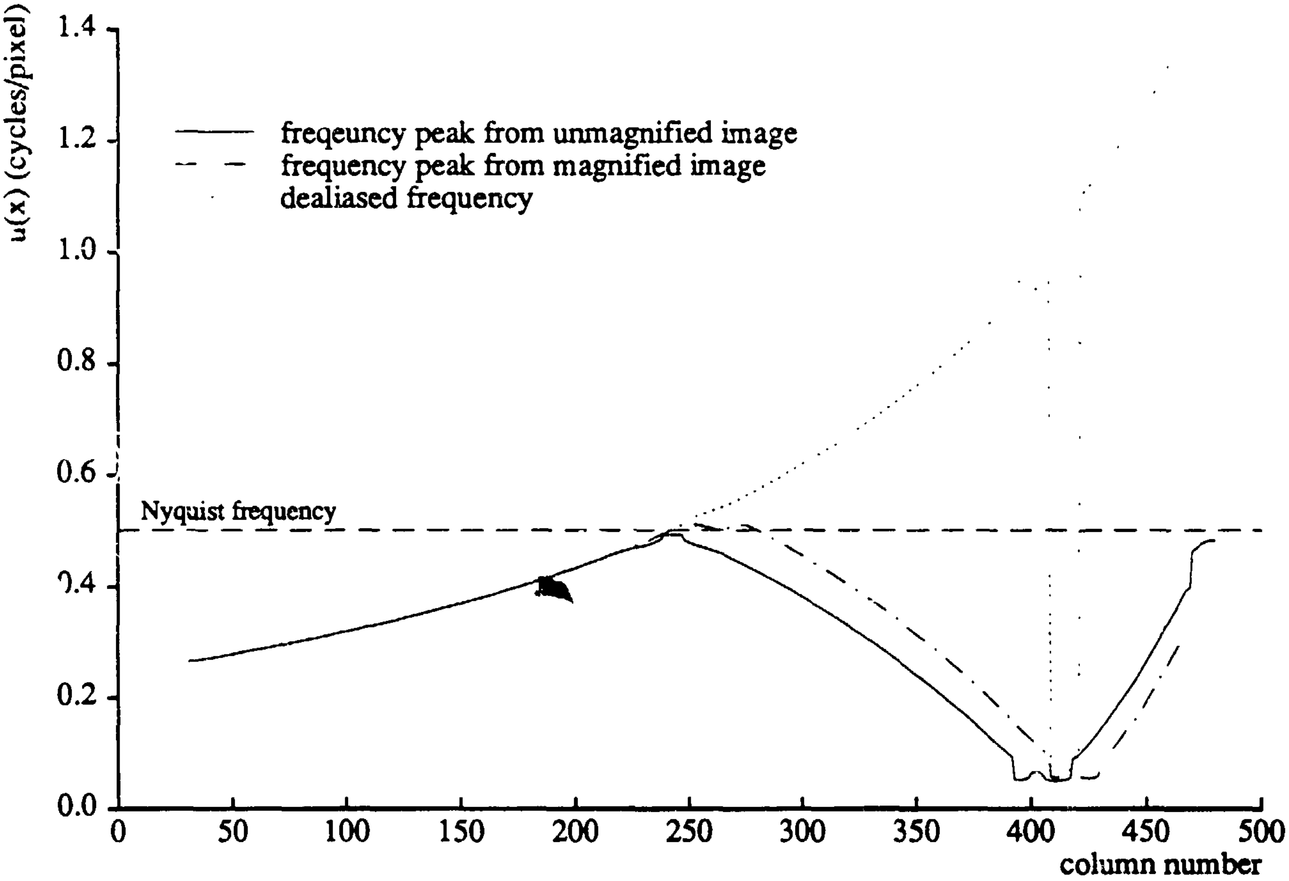

Figure 26: Dealiasing with magnification change

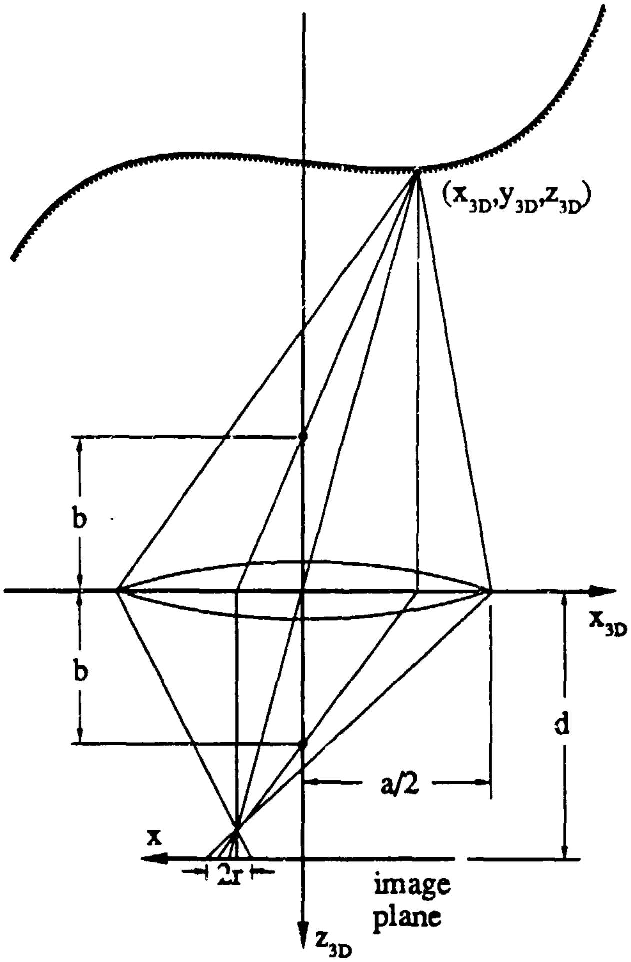

Figure 27: Geometry of 1D image formation through thin lens

the lens, *i.e.* $d \neq z$, the point will be spread into a blur circle. Using geometric optics, the radius of the blur circle is given by

$$r(z_{3D}) = \frac{ad}{2}\left|\frac{1}{b} - \frac{1}{d} + \frac{1}{z_{3D}}\right|.$$

A point can be out of focus by having the image plane in front of or behind the point of best focus. The equation above applies to both cases. $r(z_{3D})$ goes to zero when $\frac{1}{d} + \frac{1}{-z_{3D}} = \frac{1}{b}$, which is a restatement of the Gaussian Lens Law above. In the one-dimensional imaging case illustrated here, the shape of the blur "circle" is actually a rectangle of width $2r(z_{3D})$. Thus, the point spread function of the 1D camera system is

$$h(x, z_{3D}) = \frac{1}{2r(z_{3D})}\mathrm{rect}\left[\frac{x}{2r(z_{3D})}\right].$$

where we have normalized so the area under the rect() is one. [3] The corresponding transfer function, H, is the Fourier transform of h:

$$H(u, z_{3D}) = \mathrm{sinc}\left[2ur(z_{3D})\right].$$

In order to calculate the effect of $h(x, z_{3D})$ on the spectrogram, we suppose there exists a function $f(x)$ which is an unblurred, pinhole projection of the scene. The new image, $f_h(x)$, taking into account the point spread function, is a convolution of the unblurred image with h. Thus,

$$f_h(x) = \int_{-\infty}^{\infty} f(\zeta)h(x - \zeta, z_{3D})d\zeta.$$

where the z_{3D} is the one corresponding to ζ on the image plane. This equation holds for changes in the camera's aperture. It does not apply for change in the focus distance d, because this causes a change in magnification as well as a change in the point spread function.

The point spread function h is *not* space-invariant, because it depends on the depth of the surface. This means that its effect cannot be described accurately by multiplication in the frequency domain. If h were space-invariant, *e.g.* due to integrating over the surface of the pixels, then the effect on the spectrogram would be simple to describe: each windowed Fourier transform would be multiplied by the Fourier transform of the point spread function. This is also approximately true for the space-variant point spread function if the surface depth varies slowly and/or the window used for the spectrogram is small. Then we have

$$S_{f_h}(x, u) \approx \left|\left[e^{-j2\pi xu}W_l(u)\right] * F(u) * H(u, \overline{z_{3D}})\right|^2.$$

where $\overline{z_{3D}}$ is a representative depth value for the region centered at x, and $F(u)$ is the Fourier transform of the unblurred image. Each windowed Fourier transform has associated with it its own transfer function which depends on the approximate depth of the region within the window.

[3] This psf ignores three optical effects. One is diffraction, whose magnitude is much smaller than defocus effects in typical TV images. The second is the fact that points which are occluded in the pinhole image can actually be seen by parts of the lens in an image with a finite aperture. The third is that, by normalizing the area of the psf to one, we are ignoring the most obvious effect of a change in aperture: a change in the overall brightness of the image.

This is the approximation used for most depth from focus and depth from defocus algorithms in computer vision. Following Krotkov's [Kro87] depth from focus algorithm, the spectrogram can be used as a criterion function to calculate the point of best focus over several images taken at different focus settings. The setting closest to perfect focus is the one which gives the most high frequency energy in the spectrogram at that point. Knowing this setting along with a precalibrated table of focus distances, the depth to all points in the scene can be calculated. Pentland [Pen85] uses a spectrogram, essentially, to calculate depth from defocus based on only two focus settings. He uses the two spectrograms to calculate directly the depth to each scene point by calculating the width of the psf.

Formulating the effects of the psf in terms of the spectrogram is a natural way to reason about the space-variant nature of the transfer function. For example, it reveals how precisely each point can be focused. Points in the scene with no high frequencies will never show high frequencies no matter how well they are focused, meaning that a focusing criterion function based on frequency would not be sensitive to such points. Another issue is the separation of the space-invariant part of the psf (due to, say, pixel averaging and the camera electronics) from the space variant part. It may be that the space-invariant psf is so large that depth effects are insignificant.

7 Other Issues

7.1 Variable Window Size

A constant window size for the spectrogram means that the Fourier transforms cover a different number of wavelengths of each constituent frequency. That is, a window size l over a signal of frequency u covers lu wavelengths or periods of the signal. In detecting repetitions at different frequencies, it makes intuitive sense that the detector window should cover a predetermined number of wavelengths rather than a predetermined length or area. This intuition is based on the feeling that a texture pattern is one comprised of some minimum number of similar elements rather than some minimum sized region. The conventionally defined spectrogram uses a constant window size, which means that for higher frequency signals, more wavelengths of the signal will be included in the window than for lower frequency signals. Thus the localization (spatial resolution) of the *constant-window spectrogram* is effectively reduced at higher frequencies, because the window is spread out over more wavelengths.

We propose adding another dimension to the spectrogram which indicates the window size l. We define the *3D spectrogram* given by

$$S_f(x,u,l) = \left| \int_{-\infty}^{\infty} w(\alpha - x,l) f(\alpha) e^{-j2\pi u\alpha} d\alpha \right|^2 .$$

which covers all possible (positive) window sizes.

The 3D spectrogram is a great deal of data which is highly redundant. The constant-window spectrogram, $S_f(x,u)$, is a slice of $S_f(x,u,l)$ with l = constant. The problem with a constant l is that, as we mentioned above, the number of wavelengths included in the window varies with frequency. A more reasonable slice through the 3D spectrogram is to have $l \propto 1/u$, which means that the window width will shrink with decreasing wavelength. This tends to make the spectrogram scale-invariant, in that the detector window will cover a constant number of elements of a given wavelength independent of their spacing frequency. We call this the *variable-window spectrogram*.

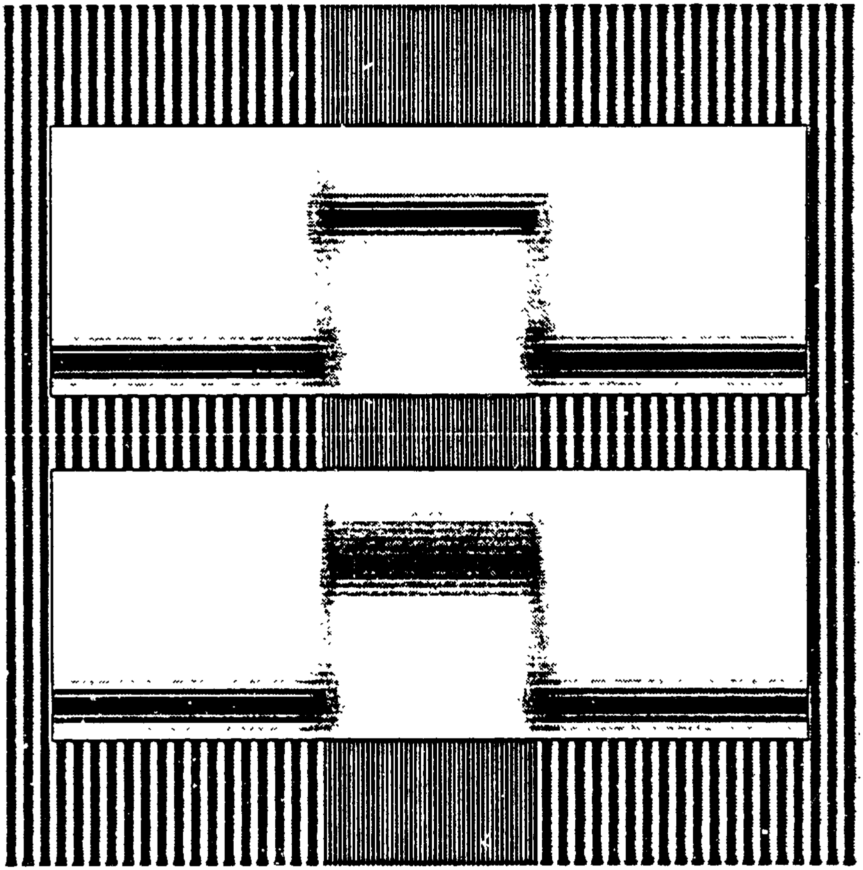

Figure 28: Constant wir^dow (top) vs. variable window (bottom) spectrogram

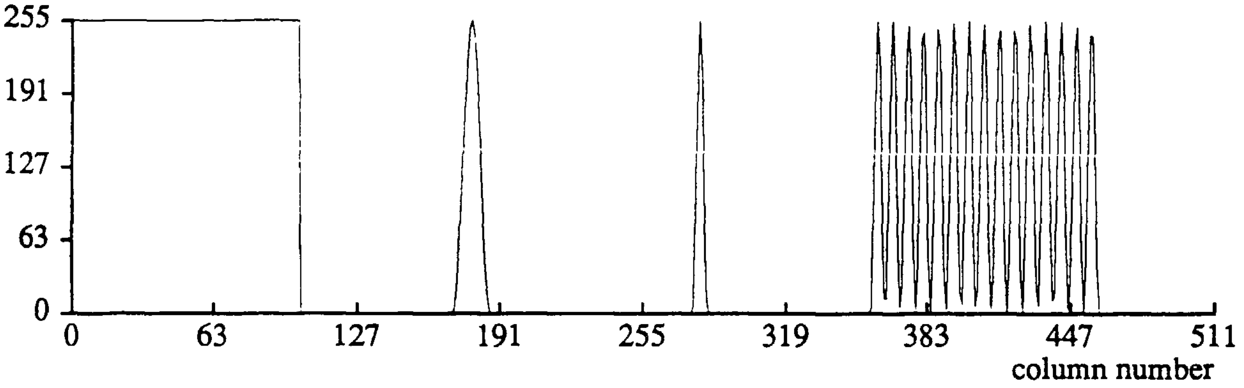

Figure 29: Intensity profile to be matched

We show an example of the the variable-window spectrogram in the bottom half of Figure 28, which can be compared to the traditional, constant-window spectrogram in the top half of the same figure. The variable-window spectrogram has window size $l = 10/u$. One notable aspect of the variable-window spectrogram is the large spreading of the higher frequencies. This is due to the familiar effect in Fourier analysis of a smaller spatial domain window giving more spread in the frequency domain. Thus, the variable-window spectrogram provides greater spatial resolution at a cost of frequency resolution. The spreading of the high frequencies leads us to a conjecture that a nonlinear sampling in frequency may be appropriate for the variable-window spectrogram. In the case of $l \propto 1/u$, the frequency sampling interval should get larger as the frequency increases.

The 3D *Gabor energy spectrum* (*c.f.* [JW88]) of a 1D signal is just the 3D spectrogram with a Gaussian window. Gabor functions are Gaussian modulated sinusoids and are maximally compact in both space and frequency. Since Gaussians have infinite support, the window length l in the 3D spectrogram is replaced by σ, the standard deviation of the Gaussian window. Although Gabor functions have proven popular in computer vision applications, we have chosen not to use the Gabor energy spectrum because other, finite-support windows give better resolution in the frequency domain.

7.2 Repetition and Image Matching

Image matching is important for 3D stereo and motion sequence analysis. In these tasks, matches are found by shifting one image to match the other; the amount of shift needed at each point reveals the 3D structure of the scene. If a portion of the image is uniform with no features, then matching is impossible; if features are present, a match can be obtained. In the ideal case of a step intensity edge, a match can be made with infinite precision. Usually, heuristic measures of potential precision are used, such as finding "feature points". But here, as in other spatial vision tasks, the spectrogram is useful to quantify this effect. The match precision available at any point in the image is limited by the highest spatial frequency present at that point. This is illustrated in Figure 29: a narrow bump or step edge can be matched with greater precision than the shallow, broad bump in the signal. This is reflected in the higher spatial frequency content for the more precise features, as shown in Figure 30. The figure shows an image whose scanlines are all identical to the intensity profile shown in Figure 29. On top is the variable-window spectrogram of one scanline, which shows that the step edge and narrow bump have higher spatial frequencies than the broad bump, and would therefore give higher precision matches. This spectrogram has window size $l = 5/u$.

The spectrogram also provides insight about another aspect of image matching: False matches. One of the

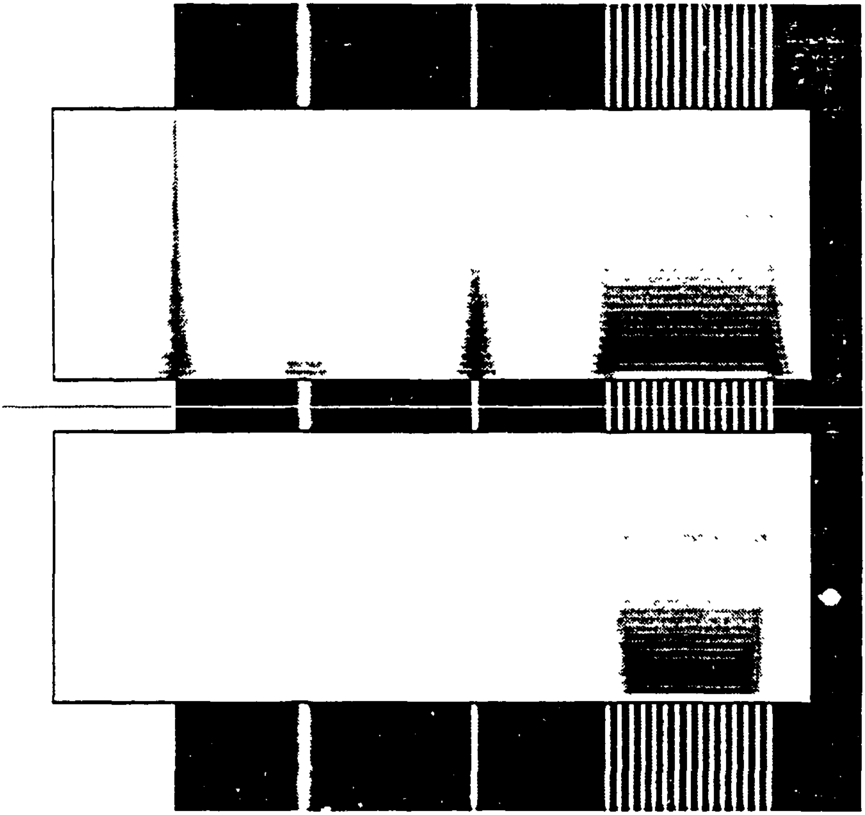

Figure 30: Spectrogram and repeatogram for image matching

hardest problems in motion or stereo vision is to know whether a potential feature match is a real, dependable correspondence, or whether it is a false match with a different feature in the other image. For example, in Figure 29 above, the right side shows several bumps closely spaced – if there were a large uncertainty in the displacement between images, the wrong bumps might be matched with each other.

There is a clear relationship between false match potential and frequency content, for a false match must be characterized by a repetition in the image signal at the corresponding scale. Yet, each bump in the group on the right of the figure has the same profile as the isolated bump just to their left. So, the distinction must be more complex than just examination of the spectrogram at each point. The key is that false match potential implies not just high frequency content, but a real repetition of the image data, which means frequency content that *persists* over more than one wavelength of the underlying sinusoid. Thus, to detect false matches (or image structure repetition in general), one must search the spectrogram for frequency content that persists over long intervals in the spatial dimension.

To represent this, we propose a new transform we call the *repeatogram*, which is derived from the spectrogram as follows: At each point x and frequency u, the repeatogram $R(x.u)$ is the minimum magnitude of the spectrogram over an interval centered at x and extending for $k/2$ wavelengths of the underlying sinusoid on either side of x, *i.e.*

$$R(x.u) = \min\left[S(x'.u)\right] \quad \text{for } x' \in [x-(k/2u).x+(k/2u)].$$

We call this the k-repeatogram, and note that for $k \geq 1.5$, there must be at least two relative maxima or two relative minima of the underlying sinusoid within the interval of examination; for $k \geq 2$, there must be at least two of each. In general, where $R(x.u)$ is high, a real repetitive structure exists in the image, with period $1/u$ pixels wide.

For a spectrogram with a nonzero window size, these considerations must be modified slightly, because a window can contain part of a repetition before it is actually centered on the repetition. Specifically, for a window of length l and a repetition over the range $[x_1.x_2]$, the spectrogram will show a reaction to the repetition over the range $[x_1-l/2.x_2+l/2]$. The matter is further complicated by the fact that most windows, including our window in Equation 2, drop off toward zero at their ends, meaning that the spectrogram will be fairly insensitive to the repetition until the window is almost centered over the repetition. The effect of these complicating factors is that the choice of k for the k-repeatogram is dependent on the window size and shape.

Figure 30 shows, in the bottom half, the 4-repeatogram for the profile in Figure 29. As seen in this figure, the repeatogram makes it quite clear that the features on the right of the signal exhibit real repetition, while the isolated bump of the same shape does not.

With the repeatogram and the spectrogram, we therefore have a powerful pair of tools for predicting the accuracy and precision of matching displaced images. At any point, the highest significant frequency content in the spectrogram tells how precise a match can possibly be obtained; the highest frequency with significant content in the repeatogram tells the maximum displacement search window size that can be tolerated before there is a danger of obtaining a false match.

8 Conclusion

For now, we have developed several useful theories for computer vision based on space/frequency representations rather than to bringing any one of the techniques described above to completion. Our work thus far has shown the versatile power of the image spectrogram rather than demonstrating any end-to-end analysis. Our goal has been to assess the potential for this kind of approach to vision rather than try to build specific programs to analyze this or that particular phenomena. We have presented a few experimental results, but they are meant to be suggestive rather than definitive algorithms. Instead, we wish to point out the breadth of this approach to low-level spatial vision, and in particular its potential contribution for:

General Vision: As an alternative to traditional edge-finding and region-grouping methods, which are known to be very brittle and noisy. The spectrogram also captures the 3D shape and 2D texture characteristics of surfaces.

Matching Problems: As a way of showing specifically what displacement of stereo or motion can be tolerated for reliable matching at each point in the image.

Active Lens Control: As a way of formulating the constraints and goals for purposeful zoom, focus and aperture.

This line of investigation, obviously, is far from complete. In particular, we see challenges in the analysis of complex textures such as the Brodatz patterns rather than simple sinusoid and square waves; expressing the relationship between 3D surface texture, radiometry (lighting and reflection), and 2D image texture; and the development of effective algorithms to compute and analyze the spectrogram. It may also turn out that the spectrogram is primarily useful not as a representation to use in the vision system itself, but rather as a way of understanding the theory behind an implementation that uses, for example, a small set of Gabor functions instead. In any event, we believe that the power of the space/frequency distribution will make it possible to develop far more comprehensive methods for low-level spatial vision than the current, limited, techniques allow.

9 Acknowledgements

Thank you to Sheryl Young for suggesting the method for detecting aliasing by changing zoom as described in Section 6.1.2.

References

[BCG90] Alan Conrad Bovik, Marianna Clark, and Wilson S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.

[BL76] Ruzena Bajcsy and Lawrence Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5:52–67, 1976.

[Boa88] Boualem Boashash. Note on the use of the wigner distribution for time-frequency signal analysis. *IEEE Transactons on Acoustics, Speech, and Signal Processing*, 36(9):1518–1521, September 1988.

[Bro66] Phil Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, 1966.

[CH80] Richard W. Conners and Charles A. Harlow. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(3):204–222, May 1980.

[CM80a] T.A.C.M. Claasen and W.F.G. Mecklenbrauker. The wigner distribution - a tool for time-frequency signal analysis, part i: Continuous-time signals. *Phillips Journal of Research*, 35(3):217–250, 1980.

[CM80b] T.A.C.M. Claasen and W.F.G. Mecklenbrauker. The wigner distribution - a tool for time-frequency signal analysis, part ii: Discrete-time signals. *Phillips Journal of Research*, 35(4/5):276–300, 1980.

[CM80c] T.A.C.M. Claasen and W.F.G. Mecklenbrauker. The wigner distribution - a tool for time-frequency signal analysis, part iii: Relations with other time-frequency signal transformations. *Phillips Journal of Research*, 35(6):372–389, 1980.

[CW89] Hyung-Ill Choi and William J. Williams. Improved time-frequency representation of multicomponent signals using exponential kernels. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(6):862–871, June 1989.

[Dau85] John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, July 1985.

[Dau88] John G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, July 1988.

[DR76] Charles A. Dyer and Azriel Rosenfeld. Fourier texture features: Suppression of aperture effects. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(10):703–705, October 1976.

[FS89] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, June 1989.

[Gab46] D. Gabor. Theory of communication. *The Journal of the Institution of Electrical Engineers, Part III*, 93(21):429–457, January 1946.

[Gra73] Nicholas Gramenopoulos. Terrain type recognition using erts-1 mss images. In *Symposium on Significant Results Obtained from the Earth Resources Technology Satellite*, pages 1229–1241. NASA Scientific and Technical Information Office, March 1973.

[Har78] Fredric J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, January 1978.

[Har79] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[Hee88] David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, January 1988.

[Hor68] Berthold Horn. Focusing. Artificial Intelligence Memo 160, MIT, May 1968.

[JC88] Y.C. Jau and Roland T. Chin. Shape from texture using the wigner distribution. In *Computer Vision and Pattern Recognition*, pages 515–523. Computer Society Press, June 1988.

[JW88] Lowell D. Jacobson and Harry Wechsler. Joint spatial/spatial-frequency representation. *Signal Processing*, 14(1):37–68, January 1988.

[Kir76] Leonard Kirvida. Texture measurment for the automatic classification of imagery. *IEEE Transactions on Electromagnetic Compatibility*, EMC-18(1):38–41, February 1976.

[Kro87] Eric Krotkov. Focusing. *International Journal of Compter Vision*, 1(3):223–237, May 1987.

[Mal89] Stephane G. Mallat. A theory for multuresolution signal decomposistion: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[Mar80] S. Marcelja. Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America*, 70(11):1297–1300, November 1980.

[Mat89] Larry Matthies. Dynamic stereo vision. Technical Report CMU-CS-89-195, Carnegie Mellon University, October 1989.

[MMN83] Takashi Matsuyama, Shu-Ichi Miura, and Makoto Nagao. Structural analysis of natural textures by fourier transformation. *CVGIP*, 24:347–362, 1983.

[Pen85] Alex P. Pentland. A new sense for depth of field. In Aravind Joshi, editor, *Proceedings of he Ninth International Joint Conference on Artificial Intelligence*, pages 988–994. Morgan Kaufmann Publishers, Inc., August 1985.

[Pen88] Alex Pentland. The transform method for shape from shading. Media Lab Vision Sciences Technical Report 106, MIT, July 1988.

[RW90] Todd R. Reed and Harry Wechsler. Segmentaion of textured images and gestalt organization using spatial/spatial frequency representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):1–12, January 1990.

[SKKN90] Steven A. Shafer, Takeo Kanade, Gudrun J. Klinker, and Carol L. Novak. Physics-based models for early vision by machine. In *SPIE Conference on Perceiving, Meausring, and Using Color, #1250*. SPIE, February 1990.

[Tur86] M.R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55(1):71–82, October 1986.

[VGDO85] L. Van Gool, P. Dewaele, and A. Oosterlinck. Texture analysys anno 1983. *Computer Vision, Graphics, and Image Processing*, 29:336–357, 1985.

[WDR76] Joan S. Weszka, Charles R. Dyer, and Azriel Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):269–285, April 1976.

[Wec80] Harry Wechsler. Texture analysis – a survey. *Signal Processing*, 2:271–282, 1980.

[Wit81] Andrew P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, June 1981.

[ZAM90] Y. Zhao, L. Atlas, and R. Marks. The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, (to appear) June 1990.