Modeling Foreshortening in Stereo Vision using Local Spatial Frequency

Mark W. Maimone Steven A. Shafer

January 1995

CMU-CS-95-104

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

A condensed version of this report has been submitted to IROS'95.

A version of this paper is available online at `http://www.ius.cs.cmu.edu/usr/cil/www/fore/tr.html`

This research is sponsored by the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006, and the NASA Ames Graduate Student Researchers Program NGT 51026. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army or the United States Government.

Keywords: perspective foreshortening, local spatial frequency, Gabor filters, wavelets, scalogram

Abstract

Many aspects of the real world continue to plague stereo matching systems. One of these is perspective foreshortening, an effect that occurs when a surface is viewed at a sharp angle. Because each stereo camera has a slightly different view, the image of the surface is more compressed and occupies a smaller area in one view. These effects cause problems because most stereo methods compare similarly-sized regions (using the same-sized windows in both images), tacitly assuming that objects occupy the same extents in both images. Clearly this condition is violated by perspective foreshortening.

We show how to overcome this problem using a Local Spatial Frequency representation. A simple geometric analysis leads to an elegant solution in the frequency domain which, when applied to a Gabor filter-based stereo system, increases the system's maximum matchable surface angle from 30 degrees to over 75 degrees.

1 Introduction

Stereo matching provides the foundation for many methods in computer vision, and a large number of practical applications: robot navigation, parts inspection, aerial or satellite mapping, and medical imaging to name a few. In all of these, the distance to objects in the scene is computed by comparing several images of the world. The distance thus computed might then be used to plan a robot's path, determine the pose of a CAD model, generate landscaping contours, or position a robot arm. Since these tasks may be of critical importance, the distance estimates must be well-characterized and precisely determined. Although one can try to extract distance from a single image (e.g., using shape from shading with certain assumptions) or many images (e.g., using multibaseline stereo or optical flow), we will study a minimal configuration for model-independent stereo: two cameras with known position and orientation.

The problem of computing distance from two images can be reduced to that of finding which pixels correspond between the two images. Given a pair of corresponding pixels, the distance between the two cameras and their orientations, it is easy to apply triangulation to find the distance to the point in world coordinates represented by those pixels. So our task is to find the vector offsets between corresponding pixels: this vector is called the *disparity* at a given pixel, and is measured in units of pixels in the image plane. Informally, the disparity tells you how far you must shift a pixel in one image to have it line it up with its correspondent in the other image.

Object surfaces are rarely viewed head-on in both images of a stereo pair. Instead, they may appear more compressed in one image, due to perspective foreshortening, as in Figure 1. When a surface has a textured appearance, this effect makes matching its two images very difficult, since its appearance differs so much between the two images. This leads to confusing results from area-based stereo matching techniques, because the visible areas vary so much between the two images.

In this paper we develop a model of perspective foreshortening that enables us to quantitatively predict its effect on stereo image pairs. We present two equivalent forms of a correction factor that allow us to reason about foreshortening effects in both 3D world coordinates and 2D image coordinates. We show how to improve the accuracy of phase-based stereo matching systems using this information, and demonstrate its application to a particular Gabor filter-based stereo system. Applying the correction factor to this system increased

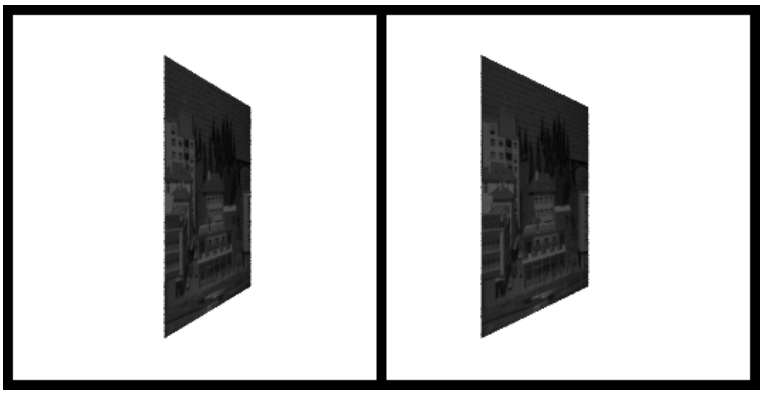

Figure 1: Stereo pair illustrating the effects of foreshortening; image compression, differing spatial extents.

its maximum matchable surface angle from 30 degrees to over 75 degrees.

2 Related Work

Local spatial frequency has already been identified as a valuable tool for modeling surface shape and segmenting multiple textures in a single image [Kru94] [MP89]. These approaches use filter magnitude in the frequency domain as the feature of interest, and require either that the surface textures exhibit specific properties (e.g., periodicity), or that they be viewed directly head-on.

Local spatial frequency representations have also been successfully appled to optical flow problems [BFBB92] [XS94], using phase information as well as magnitude. The stereo problem is more constrainted than optical flow, so we should be able to do better by taking advantage of the additional constraints.

There have been a few attempts at modeling foreshortening in the context of stereo matching. Jones and Malik [JM91] attempted to apply local spatial frequency to the problem, but got unsatisfying results. Belhumeur [Bel93] addressed the problem in the spatial domain, but his method requires an estimate of the disparity derivative, an inherently noisy estimator. The variable window method of Kanade and Okutomi [KO90] implicitly addresses foreshortening in the spatial domain by allowing corresponding windows to have different sizes, but is intended to function as a high-precision refinement technique: without proper guidance from other sources it tends to get stuck in local minima and flatten out sloped

surfaces.

Several phase-based stereo methods have been described in the literature [FJJ91] [San88] [Wen90], and a review of the more popular variations can be found in [JJ94]. Although some of these mention foreshortening as an issue, none has explicitly modeled it or corrected for it.

3 Background

3.1 Stereo Matching

Stereo matching is a useful tool in a variety of applications: robot vision, parts inspection, and aerial mapping to name a few. It is used to acquire knowledge about distance to objects in the world. It is a passive technology, relying on the interpretation of pixel luminance intensities in two or more images to reconstruct 3D information. Active technologies also exist (e.q., laser rangefinders, sonar, and radar), but this paper will limit its scope to the passive imaging method of stereo matching.

The minimum system requirements for stereo vision are a pair of cameras positioned with overlapping fields of view. These cameras could be arranged in any number of ways, but to simplify the forthcoming discussion we will restrict our attention to a simple case: both cameras on a horizontal plane with optical and vertical axes parallel, and known *baseline* (distance between them). Thus we explicitly avoid issues dealing with camera vergence (rotation about the vertical axis), torsion (rotation about the optical axis), unknown baseline separation, image rectification, and camera calibration. Such issues can be addressed by other methods (e.g., weak calibration [RBH94], Fundamental Matrix recovery, explicit hardware control [Ros93]) and thus can reasonably be avoided here.

The primary task in stereo matching is to locate pairs of pixels that are images of the same point in space. Once a correspondence has been established, it is a simple matter to determine the distance to that point using triangulation. We can derive the exact relation by considering the overhead view in Figure 2. Similar triangles give us two equations relating the pixel indices x_{iL} and x_{iR} to depth Z:

$$\frac{\Delta B}{Z} = \frac{x_{iR}}{f} \qquad \frac{B + \Delta B}{Z} = \frac{x_{iL}}{f} \tag{1}$$

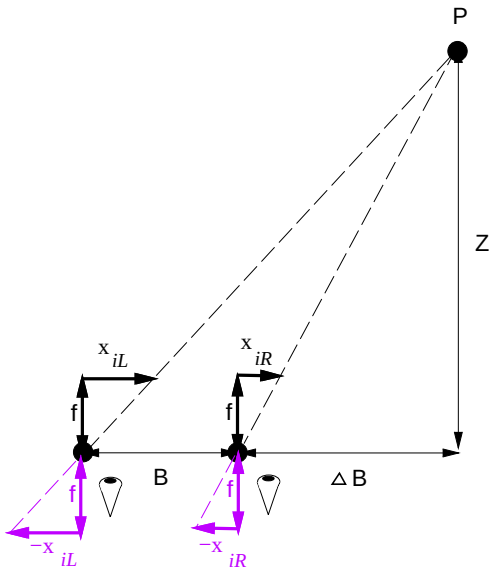

Figure 2: Overhead view of a typical Stereo Vision setup. A pair of cameras with focal length f, separated by baseline distance B, are represented by their focus points on the lower left. Object point P has corresponding image coordinates x_{iL} and x_{iR}, and lies at depth Z from the focus points. Mirror image vectors $-x_{iL}$ and $-x_{iR}$ are shown to make the similar triangles used in Equation 1 more explicit.

Solving both of these for ΔB and setting them equal, we obtain the canonical expression relating horizontal disparity to depth:

$$Disparity \quad = \quad x_{iL} - x_{iR} \quad = \quad \frac{Bf}{Z} \tag{2}$$

Equation 2 gives pointwise disparity only; we will show how to extend this description to surfaces in Section 4.

3.2 Traditional Stereo Algorithms

Existing stereo algorithms in the computer vision literature can be loosely classified under one of three headings: traditional correlation-based stereo, feature-based stereo, or frequency-based (often phase-based) stereo. In correlation-based stereo (henceforth called "traditional" stereo, e.g. the two image case of [OK91]), disparity is computed by fixing a small window around a pixel in the left image, then measuring the correlation or Sum-of-Squared-Difference

error between intensities in that window and those in similar windows placed at different locations in the right image. The placement that yields the lowest error is used to compute the disparity estimate. This procedure is applied to the image at successively higher resolutions in a coarse-to-fine manner, restricting the search based on the disparity estimate from the previous (lower resolution) image. In feature-based stereo (e.g., [Mat89]), the dense image is converted into a spatially sparse set of features (e.g., corners, edges) which are matched. This results in a sparse disparity map which must be interpolated to yield disparities at every pixel. Finally, in frequency-based stereo (e.g., [San88]) the original signal is transformed to Fourier space, and the phase of the transformed signal is used to compute the disparity, in any of several possible ways [JJ94].

3.3 The Scalogram: A Unified View of Scale Space

The scalogram is one of several representations that makes use of the local frequency content of an image.

Why Local Frequency? Many of the problems in traditional stereo arise from its limited image representation. Many imaging phenomena are more succinctly described (and more easily manipulated) in the Fourier domain than in the spatial domain. For example, a sinusoidal pattern at any scale can be fully described by four values in the Fourier domain: amplitude, frequency, phase shift, and a constant (DC) offset. The power of the Fourier transform is its ability to extract this information from any signal in a straightforward and deterministic way. You can think of an image as expressing a function as a sum with delta functions forming the basis, and the Fourier representation as representing the same function but with sinusoidal basis functions. The problem with using the Fourier transform directly is that it extracts frequency information contained *everywhere* in the image; you may find the precise frequencies present in a signal, but you won't know *where* in the signal those frequencies occur. This is unacceptable for image matching.

However, we can compromise by applying the Fourier transform not to the entire image, but rather to a small subset, or *window* of the image. By restricting attention to the parts of the image immediately surrounding a given pixel, we can learn about the *local* frequency context, the patterns present only at that pixel. There is a trade-off, however. By restricting ourselves to a small window on the image, we sacrifice the precision with which we can isolate particular frequencies.

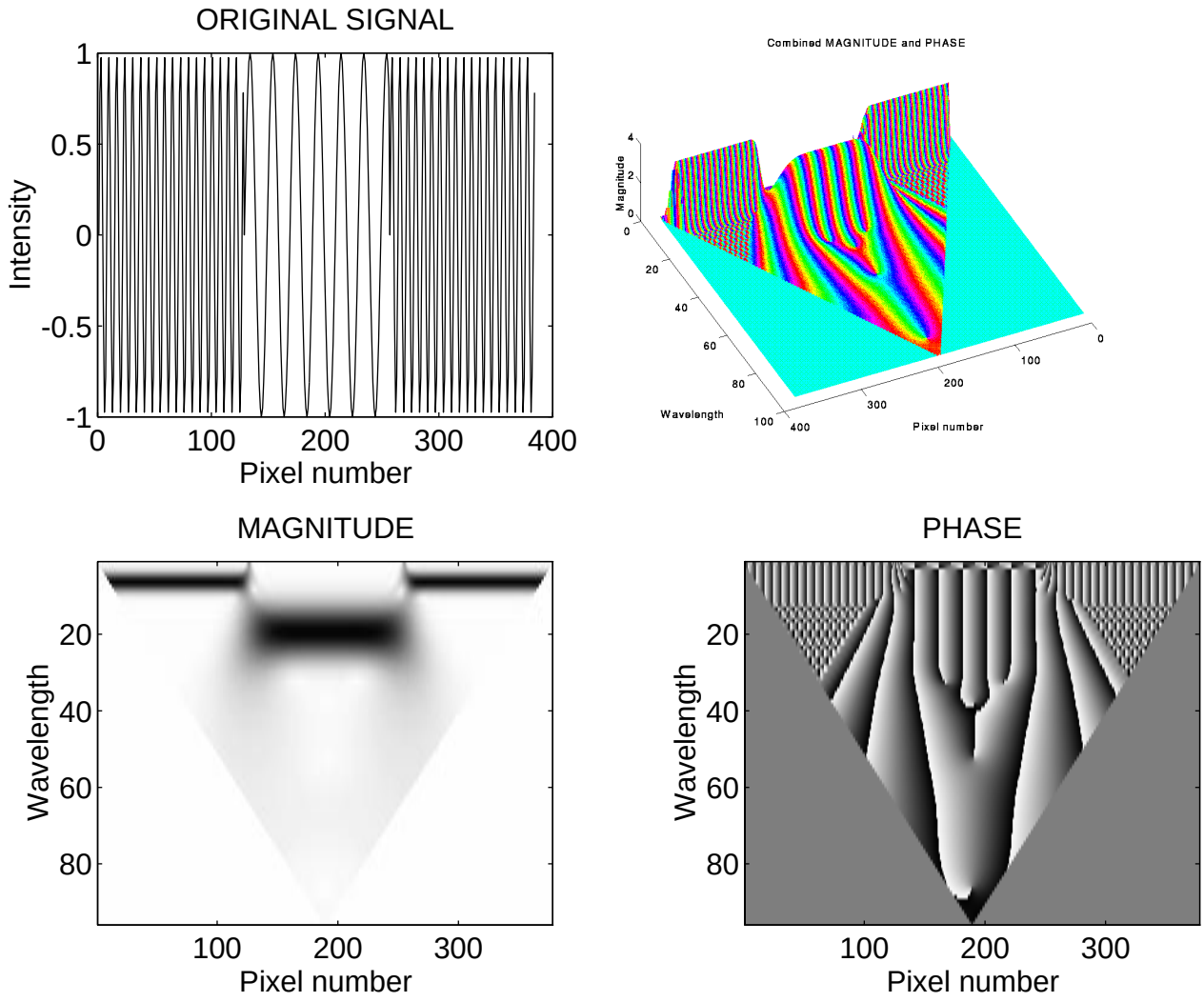

Figure 3: Double sine wave signal and associated scalogram (both magnitude and phase).

There are many local frequency representations: spectrograms (Short Time Fourier Transforms), Wigner-Ville distributions, wavelets and scalograms to name a few. All are similar in effect, but slightly different in structure. The spectrogram uses a fixed window size at all scales and a logarithmic sampling of wavelengths. This can be useful for texture analysis, where you expect to see the same pattern repeated often at small scales, but seems less useful for image matching. The fixed window size means that high frequency results will not be easily localized, and low frequencies may not have enough support. In contrast, the scalogram uses a variable window size, one which is always a constant number of wavelengths long. This makes high frequencies much more localizable, and provides the necessary support for low frequencies. The scalogram is actually a special case of the very

general wavelet functions: the scalogram is a wavelet with a Gabor function as the transfer function. Wigner-Ville is a compromise between spectrograms and scalograms, but often contains many cross terms that complicate automated analysis. A nice overview of all these representations can be found in [RV91].

Figure 3 illustrates a simple signal and its scalogram, computed as follows:

$$Gabor_\lambda = \left[e^{-\left(\frac{x}{m\lambda\sigma}\right)^2} \cdot e^{-i\frac{2\pi}{\lambda}x} \right] \quad \text{for } x \in \left[-\frac{m\lambda}{2}, \frac{m\lambda}{2} \right]$$

$$\text{Scalogram}_R(x, y) = (R * Gabor_y)(x)$$

where R is the one dimensional input row, λ is the filter wavelength, m is the number of wavelengths to fit in the window, σ is the Gaussian parameter expressed as a fraction of the window size $m\lambda$, and $*$ denotes convolution. The signal (upper left) consists of a sine wave with fairly high frequency on the outside, and a sine wave of lower frequency spliced into it. The scalogram plots have a straightforward interpretation: the horizontal axis is the same as in the original signal (Pixel number) and the vertical axis is linear with respect to wavelength (in pixels). Short wavelengths are on top, longer wavelengths are at the bottom. The intensity of the points in the magnitude image (lower left) encodes the strength of the signal at a given location and resolution, or wavelength; darker spots mean stronger response. It's easy to see the relative contributions of each frequency to each point in the image; the frequency information is very well localized, except at the junctions between the signals. The phase plot (lower right) shows interesting structure, but is less easily interpreted because many of the calculated phase values are unreliable: only those phases that have high corresponding magnitudes are of measurable interest. The most reliable phase values are most easy seen on the peaks in the combined plot (upper right), where magnitude is encoded as surface height. These plots have a triangular shape because no data is plotted where the filter window would extend beyond the signal boundary.

This kind of representation is very useful for image matching. In particular, the phase measurements translate directly into disparity measurements: $disp = \frac{(\phi_L - \phi_R)}{2\pi} \cdot \lambda$. This gives us a means of generating disparities to subpixel accuracy without having to explicitly interpolate the original signal.

3.4 Phase Difference Measurement Errors

Instead of matching intensities directly, phase methods fit regions of intensities to continuous filters, then use properties of the filter output to determine disparity. This relation lies at the heart of all phase-difference methods:

$$\textit{disparity} \;=\; \frac{\Delta\phi}{2\pi}\cdot\lambda \;=\; \Delta\phi\cdot\frac{1}{\text{frequency}} \tag{3}$$

This equation can be readily understood by applying it to the continuous domain of simple sine waves. It simply says that disparity is equal to the translation of the sine wave (i.e., phase difference $\Delta\phi$, measured in radians) scaled by its wavelength, or period λ. The same intuition works for *windowed* continuous sine waves as well (e.g., Gabor filters which are Gaussian-modulated sine waves). Problems arise when applying this to the discrete case, however. How is the actual frequency of the sinusoid measured? Phase plays an important role in the disparity computation, but how accurately can it be measured? Also implicit in Equation 3 is the assumption that the left and right phases are measured at the same frequency, i.e. on similar sinusoids; is that assumption reasonable?

To illustrate the sensitivity of these equations consider what happens when the signal being studied is a simple sinusoid.

3.4.1 Measuring sinusoid frequency

When filtering a periodic signal there are two common techniques for determining the frequency of the signal: using the frequency of the pass-band filter with highest magnitude response or measuring the phase derivative of the filtered signal.

Using the filter frequency directly is troublesome because any discrete filter will have a blurred response; it combines the responses to all of its passed frequencies. Some systems use the approximation that the filter's peak tuning frequency is a good enough guess[San88] [Wen90], but this assumption may reduce the precision of the results. One "advantage" to using this is that its variance is a fixed quantity; it does not depend on the actual signal content.

The phase derivative (aka *instantaneous frequency*) provides a more accurate measure of a sinusoid's frequency in theory, but in practise its accuracy depends on the amplitude of the input signal. This has led several researchers to develop constraints that filter out

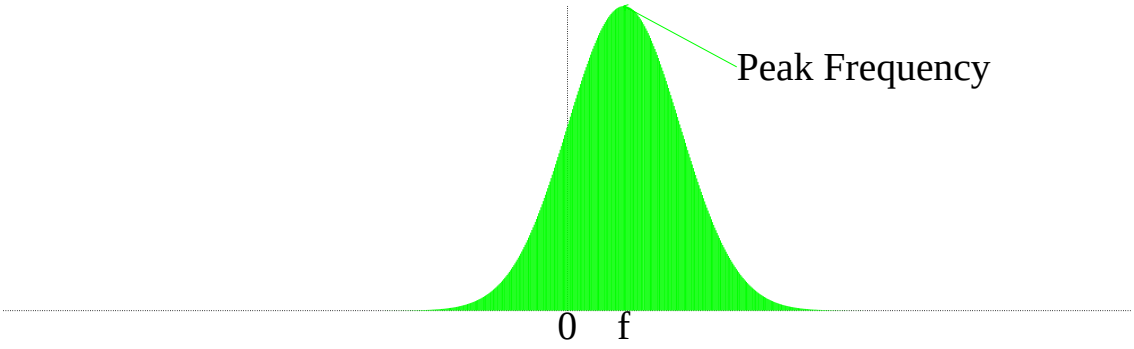

Figure 4: The transfer function of a Gabor filter.

unreasonable phase-derived frequencies[FJJ91] [XS94]. The constraint developed by Fleet et al filters out those responses whose phase derivatives predict frequencies that lie outside the range of the filter. While this constraint provides a useful "sanity check" on the validity of the computed frequency, it cannot be used to filter out the effects of aliasing or compensate for perspective foreshortening.

3.4.2 Measuring phase

The output of a Gabor filter is a complex number. We are primarily concerned with measuring the phase component of this number, but how accurate is a given phase value? Figure 5 gives us some intuition. The vectors represent complex numbers; the length of a vector is its magnitude, and the angle between the vector and the horizontal axis is its phase. Adding an error vector $\vec{\epsilon}$ with length ϵ has little effect on the phase of the longer vector, but results in a completely random phase value for the smaller vector. The precision of a particular vector's phase can be determined exactly by comparing its length to that of the error vector $\vec{\epsilon}$.

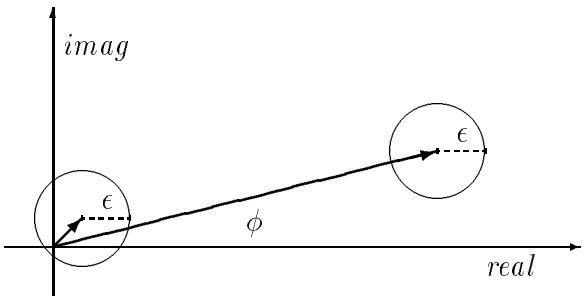

Figure 5: The effect of measurement error vector $\vec{\epsilon}$ on phase angle ϕ.

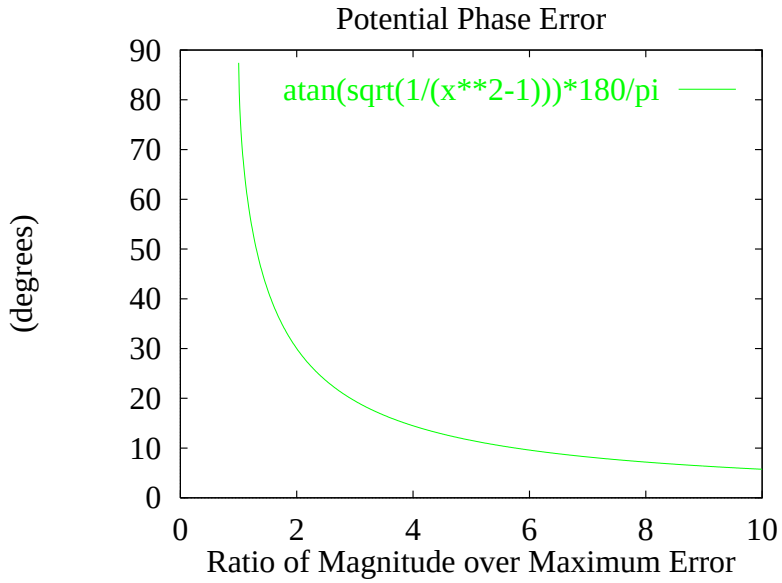

Figure 6: Maximum phase angle error as a function of the length ratio.

Given a vector $\vec{v}$ and an error vector $\vec{\epsilon}$, we can compute the potential error in its phase $\angle\vec{v}$ exactly using Equation 4 below, derived as follows. Place the tail of $\vec{v}$ on the origin and rotate $\vec{v}$ so that it points right on the horizontal (i.e., with phase angle zero). Draw a circle of radius ϵ centered at its head. To find the maximum phase error, draw the line tangent to the error circle that also intersects the origin (actually there are two symmetric such lines; draw the one on top). The slope m of that tangent line gives us the maximum error angle: $m = \tan error$. To find the slope, plug the equation for the line ($y = mx$) into the equation for the circle ($(x - |\vec{v}|)^2 + y^2 = |\vec{\epsilon}|^2$) and solve the resulting quadratic equation for x. At the point of intersection this equation will have two real and equal roots, so we set the discriminant equal to zero and solve for m:

$$\tan error = m \quad = \quad \sqrt{\frac{1}{\left(\frac{|\vec{v}|}{|\vec{\epsilon}|}\right)^2 - 1}} \tag{4}$$

Equation 4 tells us that the maximum phase error depends solely on the ratio between the lengths of vectors $\vec{v}$ and $\vec{\epsilon}$. Some phase angle errors are plotted as a function of this ratio in Figure 6.

There are many factors that contribute to the error vector $\vec{\epsilon}$ and therefore reduce the precision of the measured phase value: The precision used in the floating point arithmetic, the blurring of the filter response due to the presense of information at nearby frequencies,

the discretization of the filters themselves (especially at high frequencies) to name a few. A characterization of the error may be found in [FJ93].

3.4.3 Comparing Frequencies

In order for phase difference to yield a precise disparity estimate, it must reflect measurements taken at the same frequency. But what if the instantaneous frequencies measured by the same filter on a pair of images differ, as can occur with perspective foreshortening? Others have simply used the filter tuning frequency, or the average of the two instantaneous frequencies, but these are very coarse approximations not based on physical reality. The main contribution of this paper is the development of a theory for finding the proper frequencies, based on the physical geometry of the scene, presented in Section 4.

3.5 Our phase method

The foreshortening analysis to be presented in Section 4 applies equally well to many phase methods. We will apply it to a particular phase method we have developed, similar in spirit to those in [San88] and [Wen90], to demonstrate the concrete results in Section 5. In this section we outline the uncorrected (for foreshortening) algorithm to use as a framework for later discussion.

Our algorithm computes the disparity at each pixel using phase differences. Phase values with low corresponding magnitude will be filtered out using a heuristic peak-finding process. The remaining phase differences (which correspond to disparity guesses at many scales) are then fitted to an "ideal" phase-difference curve. A given disparity corresponds to a single, unique phase-difference curve; the curve that best fits the measured phase differences will give our disparity estimate.

We will illustrate the algorithm by describing the steps needed to generate a disparity map for the left image.

Let a pair of images, I_L and I_R be given. We assume that these images have the same dimensions, and were taken with a pair of cameras whose optical axes are parallel (so that the epipolar constraint holds). If the cameras are properly calibrated, this constraint reduces the matching problem from two dimensions to one, because like-numbered scanlines in the two images are guaranteed to correspond to the same plane in the world. We can thus limit the following discussion to the problem of matching a single row in the image pair.

Pick a pair of corresponding rows R_L and R_R, each containing n pixel intensities. Compute the scalograms of these rows, using $m = 4$ and $\sigma = \frac{1}{6}$. The scalogram for a given row is a two dimensional matrix of complex numbers: for convenience in discussion, we split it into two matrices, magnitude ρ and phase ϕ. [1] So $\rho_L(n, \lambda)$ is the magnitude of the left row's scalogram entry at pixel n with wavelength λ, $\phi_L(n, \lambda)$ is the phase of the left row's entry at that point, and similarly for the right row with ρ_R and ϕ_R.

To compute the disparity for a given pixel n, we look at the corresponding columns in the magnitude plots $\rho_L(n)$ and $\rho_R(n)$. These columns represent the strength of the signal at many wavelengths. Since regions with low magnitude will yield unstable phase values, we will restrict our attention to those areas with the largest magnitude by finding peaks in the magnitude plot. While this should (in general) be a function of both ρ_L and ρ_R, in the present implementation only ρ_L is used.

The heuristic peak finding scheme works as follows. We attempt to find those regions of the function that exhibit explicit peaks by repeatedly fitting and subtracting polynomial curves to the peak regions of the current function. The extent of the peak region is determined by a simple heuristic: start with a maximum value, move outward until the second derivative is no longer negative or the end of the function is reached, then back up one or two pixels. The indices of the current peak region are stored at each step, a third degree polynomial is fitted to that region and subtracted from the function, and the process repeats, terminating when the current (subtracted) function's maximum is less than some fraction of the original maximum (this is an arbitrary cutoff; for this paper a low threshold of 0.02 was used).

Having isolated regions with high magnitude, and hence having ignored potentially unreliable phase values, we proceed to fit the remaining phase differences. First we enumerate a list of possible disparities and compute the "ideal" phase difference curve for each disparity. From our original equation for disparity, we have: $d = \lambda \frac{\Delta\phi_{ideal}}{2\pi}$. Multiplying through gives us $\Delta\phi_{ideal} = 2\pi \frac{d}{\lambda}$. Unfortunately, we cannot match directly against this ideal curve, since we can only measure the phase difference modulo 2π (phase unwrapping techniques, such as [Tri77] require high magnitudes at nearly all wavelengths, an unreasonable assumption in this context). So we assume that no measured phase difference ($\phi_{measured}(n, \lambda) = \phi_L(n, \lambda) - \phi_R(n, \lambda)$)

[1] A complex number (C) can be represented by a pair of real numbers in several ways; real and imaginary ($C = a + bi$) or magnitude and phase ($C = \rho e^{i\phi}$) are the most common.

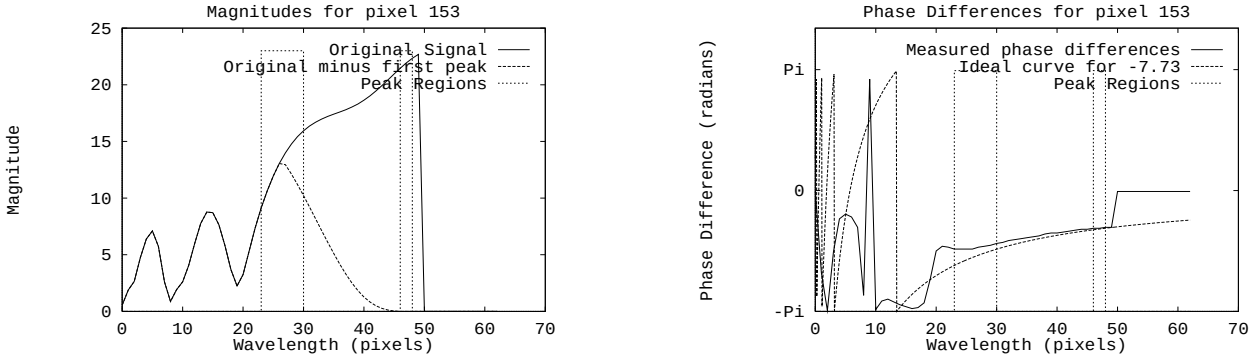

Figure 7: Sample column from the scalogram: (a) Magnitude column, illustrating the peak finding procedure. (b) Phase column, illustrating the fit to an "ideal" phase difference curve; only those values inside the dotted rectangles are used in the fit.

will be greater than π, and compute errors based on $(\Delta\phi_{ideal} - \Delta\phi_{measured}) \bmod 2\pi$. If we define W to be the set of wavelengths extracted by the peak-finding process, then the error for a given disparity d at pixel number n is defined as:

$$Error = \sum_{\lambda \in W} \rho(n, \lambda) |(\Delta\phi_{ideal} - \Delta\phi_{measured}) \bmod 2\pi|$$

Finally, we select the disparity with minimum error as the result for this pixel.

This method will be our base case; we will compare this phase method against a similar one that includes foreshortening correction in Section 5.

4 Analysis

In this section we show how perspective foreshortening is manifest in the local spatial frequency representation of stereo images. Ours will be a forward-reasoning analysis, beginning with complete knowledge of the three-dimensional geometry of the scene and ending with its two-dimensional projection in the image plane. The primary result is the presentation (in Equation 14) of the Frequency Shift scale factor that allows us to compensate for arbitrary foreshortening effects without explicitly warping the images. This result makes no restrictions on the surface texture, and will not require the use of disparity derivatives. The complementary technique (starting with the projections to determine three-dimensional geometry) will be presented in Section 5.

To simplify the analysis, we assume the only object in the world is a textured flat plate that is either parallel to the image plane, or rotated about the vertical axis by some angle θ. We further assume that the stereo cameras have parallel optical (depth) and vertical (height) axes. Note that we can restrict our attention to the effects of foreshortening in one-dimensional image *scanlines*, rather than complete two-dimensional images, since all disparities will be horizontal under this assumption. Our world model will likewise be a two-dimensional slice through the three-dimensional scene. Figure 8 shows an overhead schematic of a horizontal slice through the world. We adopt the convention that parameters measuring distances in the world will be capitalized (e.g., X_S, Z_L), and those measuring pixel or camera distances will be lower case (e.g., x_{iL}, f).

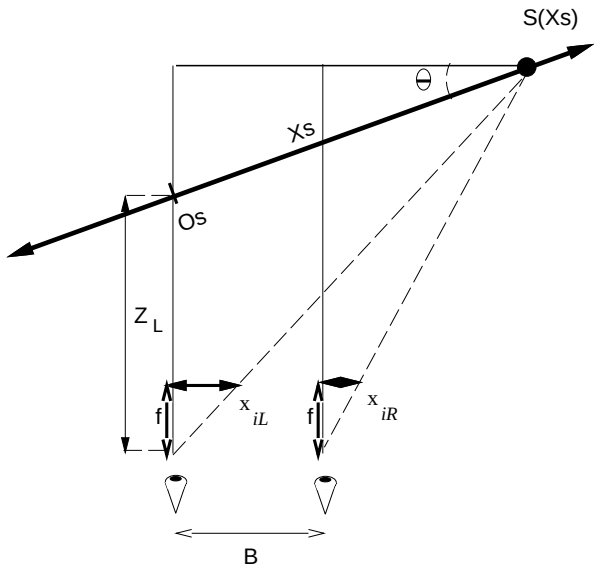

Figure 8: Overhead view of the foreshortening model. X_S is the distance from the point exactly in front of the left camera (the origin O_S at distance Z_L) to the point (S) on the plate being studied; x_{iL} and x_{iR} are the left and right pixel indices of the image of surface point S; the cameras are separated by baseline B and the surface tilts away from the cameras at angle θ.

Although our ultimate goal is to find the disparity between two stereo images, we must first determine how the appearance of the object's surface texture will differ between them. Specifically, we want to know how the sampling rate varies between the two images. This is a geometric formulation; what matters is how much of the surface is being mapped to each pixel, not the actual surface texture (i.e., color intensity). So for each location X_S on the

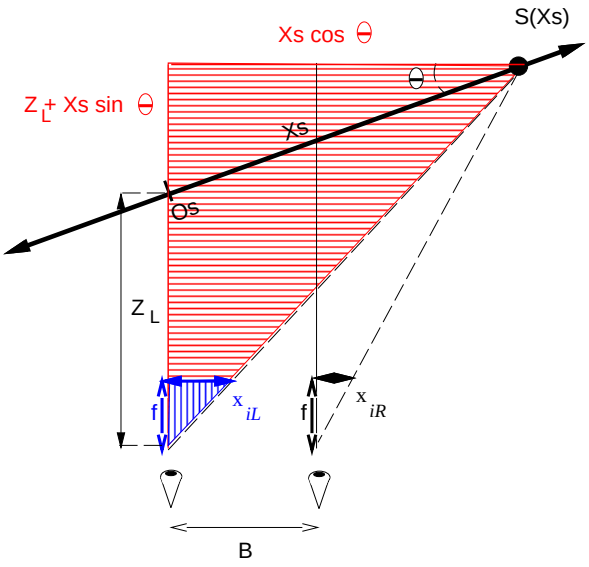

Figure 9: Overhead view of the foreshortening model. Similar triangles for the left camera geometry are highlighted (see Equation 7).

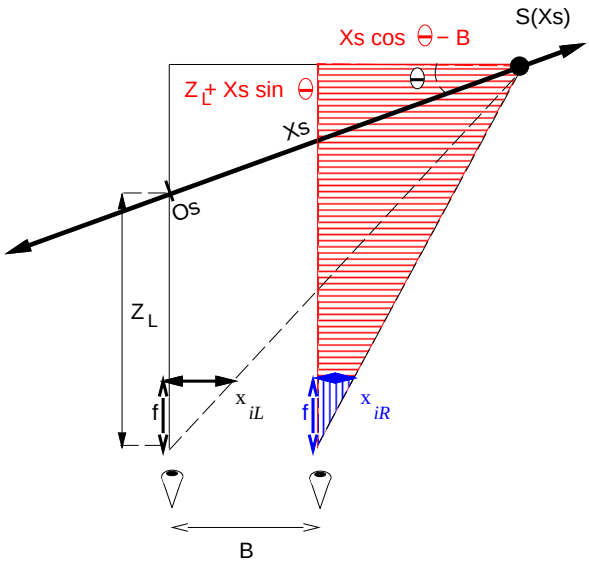

Figure 10: Overhead view of the foreshortening model. Similar triangles for the right camera geometry are highlighted (see Equation 8).

surface, we want to compare the pixel areas in the left and right images. Mathematically, we want to compare the left sampling rate $\frac{\delta X_S}{\delta x_{iL}}$ to the right sampling rate $\frac{\delta X_S}{\delta x_{iR}}$:

$$\text{Sampling ratio} \quad = \quad \frac{\frac{\delta X_S}{\delta x_{iL}}}{\frac{\delta X_S}{\delta x_{iR}}} = \frac{\delta x_{iR}}{\delta x_{iL}} \tag{5}$$

Simplifying the ratio in this way proves most useful. The resulting formula tells us we can compute the sampling ratio (which will be called *frequency shift* later) in *image space*, without having to explicitly model the distance X_S along the object. Unfortunately, it also implies that we need the disparity derivative (recall δdisparity is simply $\delta(x_{iL} - x_{iR}) = 1 -$ Sampling Ratio). Since our ultimate goal is to estimate disparity, it would be best if we could avoid using both disparity and its derivative in our calculations (the derivative of a noisy signal will be even noisier). The remainder of this section will show how we can express this ratio with terms that do *not* require disparity derivatives.

4.1 Relating Disparity to Surface Angle

How is disparity related to the surface angle? Equation 2 gives the disparity for an individual point, but we will now show how it varies across a surface. We will focus our attention on the distance from the left camera to the surface point immediately in front of it, expressing other depths in terms of this value Z_L.

Recall that disparity is the difference of the left and right pixel indices. So let's see how each of the left and right indices (x_{iL} and x_{iR}) relates to the surface angle θ. A quick look at Figure 8 shows us the general answer using similar triangles:

$$\frac{\textit{pixel index}}{\textit{focal length}} = \frac{\textit{X World Coordinate}}{\textit{Z World Coordinate}} \tag{6}$$

Figures 9 and 10 highlight the similar triangles for the left and right scene geometries. Applying Equation 6 to those figures we obtain expressions for x_{iL} and x_{iR}:

$$\frac{x_{iL}}{f} = \frac{X_S \cos\theta}{Z_L + X_S \sin\theta} \tag{7}$$

$$\frac{x_{iR}}{f} = \frac{X_S \cos\theta - B}{Z_L + X_S \sin\theta} \tag{8}$$

Equations 7 and 8 give us expressions for x_{iL} and x_{iR} in terms of the focal length f, baseline B, distance in front of the left camera Z_L, surface angle θ, and location on the surface X_S. These equations represent projections of the same surface point X_S into two image planes, and we can find the relationship between them by solving Equations 7 and 8 for X_S and setting them equal.

$$\frac{x_{iL} Z_L}{f\cos\theta - x_{iL}\sin\theta} = \frac{x_{iR} Z_L + Bf}{f\cos\theta - x_{iR}\sin\theta} \tag{9}$$

Solving Equation 9 for the right pixel index gives us:

$$x_{iR} = x_{iL}\left(1 + \frac{B}{Z_L}\tan\theta\right) - \frac{Bf}{Z_L} \tag{10}$$

And finally, recalling that disparity is the difference of the two indices:

$$\textit{disparity} = x_{iL} - x_{iR} = \frac{Bf}{Z_L} - x_{iL}\frac{B}{Z_L}\tan\theta \tag{11}$$

Equation 11 is nearly the answer we want. It relates disparity to the scene parameters, and does not depend on knowing the actual surface location. It does require knowledge of Z_L (distance to the surface point in front of the left camera), unfortunately, but we will eliminate this restriction below.

Equation 11 has some interesting interpretations. When the surface is frontoplanar (i.e., $\theta = 0$ and thus $\tan\theta = 0$) it reduces to the familiar expression relating disparity to depth from Equation 2; this is correct since all surface points would lie at the same depth Z_L. And for an arbitrary fixed angle θ the disparity *derivative* is constant, i.e., the disparity varies linearly with respect to the image location x_{iL}. While we won't take advantage of this property of the derivative, it could prove useful to shape-recovery techniques.

4.2 Expressing the Sampling Ratio using Image Parameters

Now that we know how the disparity and pixel locations relate to surface angle, let us return to the Sampling ratio (Equation 5) and eliminate the derivative by substituting for x_{iR}:

$$\begin{aligned} \text{Sampling ratio} &= \frac{\delta x_{iR}}{\delta x_{iL}} = \frac{\delta\left(x_{iL}\left(1+\frac{B}{Z_L}\tan\theta\right)-\frac{Bf}{Z_L}\right)}{\delta x_{iL}} && \textit{from Equation 10} \\ \textit{Geometric Form} &= 1+\frac{B}{Z_L}\tan\theta && (12) \end{aligned}$$

This expression is very interesting. It tells us that for a given flat surface, the sampling ratio is *constant* over both images of the surface. In other words, the local spatial frequencies of the left and right images are related by a simple constant scale factor. You can get a feel for this by visually tracking the low magnitude phase singularities (white spots) between the two image scalograms in Figure 11.

The fact that foreshortening causes frequency shifts has been noted in the literature [FJ91], but no explicit model was given to explain it. Instead, the instantaneous frequency was recovered using a heuristic averaging technique. This technique yielded somewhat better results than the use of direct frequency, but did not take advantage of the scene geometry to compute the precise shift. This averaging technique also failed whenever the frequency shift caused the instantaneous frequency to fall outside the range of the filter in either of the images. Our model overcomes these problems by making use of all available frequency bands, rather than limiting attention to a small number.

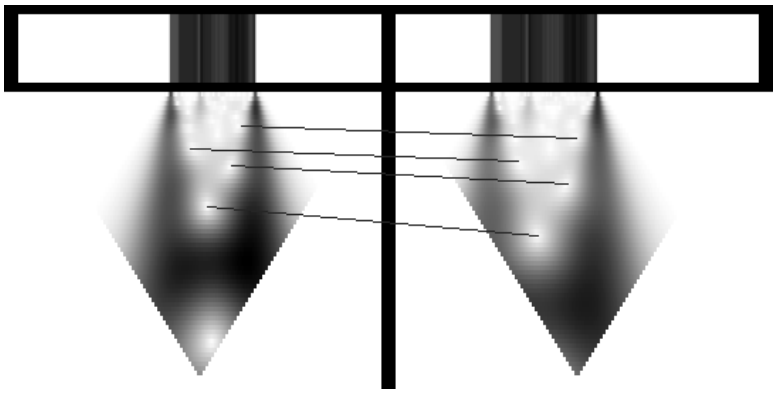

Figure 11: Left and right views of a surface tilted 65 degrees. Upper images are the central scanlines, lower images are their corresponding scalograms. You can see similar features in both scalograms: those in the left image are present at higher spatial frequencies because the left image is subject to greater foreshortening effects than the right image.

The result in Equation 12 is useful for describing the form of the foreshortening effect (that of a constant scale factor), but it would be useless in a stereo matcher since it requires knowledge of the depth Z_L. A program that computed depth given depth would not be very impressive. So how can we eliminate the need to know Z_L? Consider the ratio $\frac{B}{Z_L}$. We can rewrite Equation 11 as:

$$\frac{B}{Z_L} = \frac{disparity}{f - x_{iL}\tan\theta} \tag{13}$$

and replace that in Equation 12, giving us this final expression for Frequency Shift (aka Sampling Ratio):

$$Projected\ form = 1 + \frac{disparity\tan\theta}{f - x_{iL}\tan\theta} \tag{14}$$

This is what we want! Equation 14 relates parameters in the image plane to the surface slope θ, but does not require prior knowledge of the distance to the object or an estimate of the disparity derivative. It does require use of some known parameters (focal length f, image location x_{iL}) and variables being estimated (disparity, surface angle θ), but we will see how to manage these algorithmically in Section 5.

In this section we described the effect of perspective foreshortening in terms of local spatial frequency. We developed this theory in steps to demonstrate several properties: the

frequency shift between images of an oriented flat surface is constant, it is independent of the surface texture, and it can be expressed using only disparity and surface angle (without disparity derivatives). Section 5 will show how these results can be applied to a stereo matching system.

4.2.1 Verifying the Scale Factor

Before continuing, we will verify the geometric form of this scale factor using a simple example: a flat surface with a sinusoidal texture. If the model is correct, the surface's apparent spatial frequencies will be shifted between the two images by the amount given in Equation 12. Note that we're not solving the stereo problem yet, in fact this demonstration will use the *known* disparity to compare the left and right *image* frequencies at the same *surface* locations. What this will show is that Equation 12 accurately predicts the frequency shift of a simple signal. We will use synthetic data so that our ground truth can be as precise as possible.

Recall the geometric form of the Scale Factor from Equation 12:

$$\text{Scale Factor} \quad = \quad 1 + \frac{B}{Z_L} \tan\theta$$

Just what is this scale factor? It describes the relationship between the spatial frequencies at two image pixels representing the *same* surface point. How can we measure such frequencies, and how do we know they correspond to the same surface point?

Finding the frequency is easy, but imprecise; we will use an artificial surface texture that contains a single peak in the positive frequency domain, i.e., a sine wave. Its apparent frequency can be found simply by locating the filter output with highest magnitude.[2] As a further refinement, we will use the instantaneous frequency (phase derivative) of that filter output as our frequency estimate. Under the scalogram representation this corresponds to picking the maximum magnitude value in each column.

The procedure for finding corresponding points is somewhat complex, but simply stated involves using knowledge of the ground truth to give the disparity at each pixel (disparity is inversely related to depth, which is known from the 3D model). Remember, we are not

[2] In practise our windowing scheme provides only high frequency info at the image borders, so our computed scale factor will become inaccurate at the ends of the graph since the actual spatial frequency is lower than the lowest measured by the filters at that pixel.

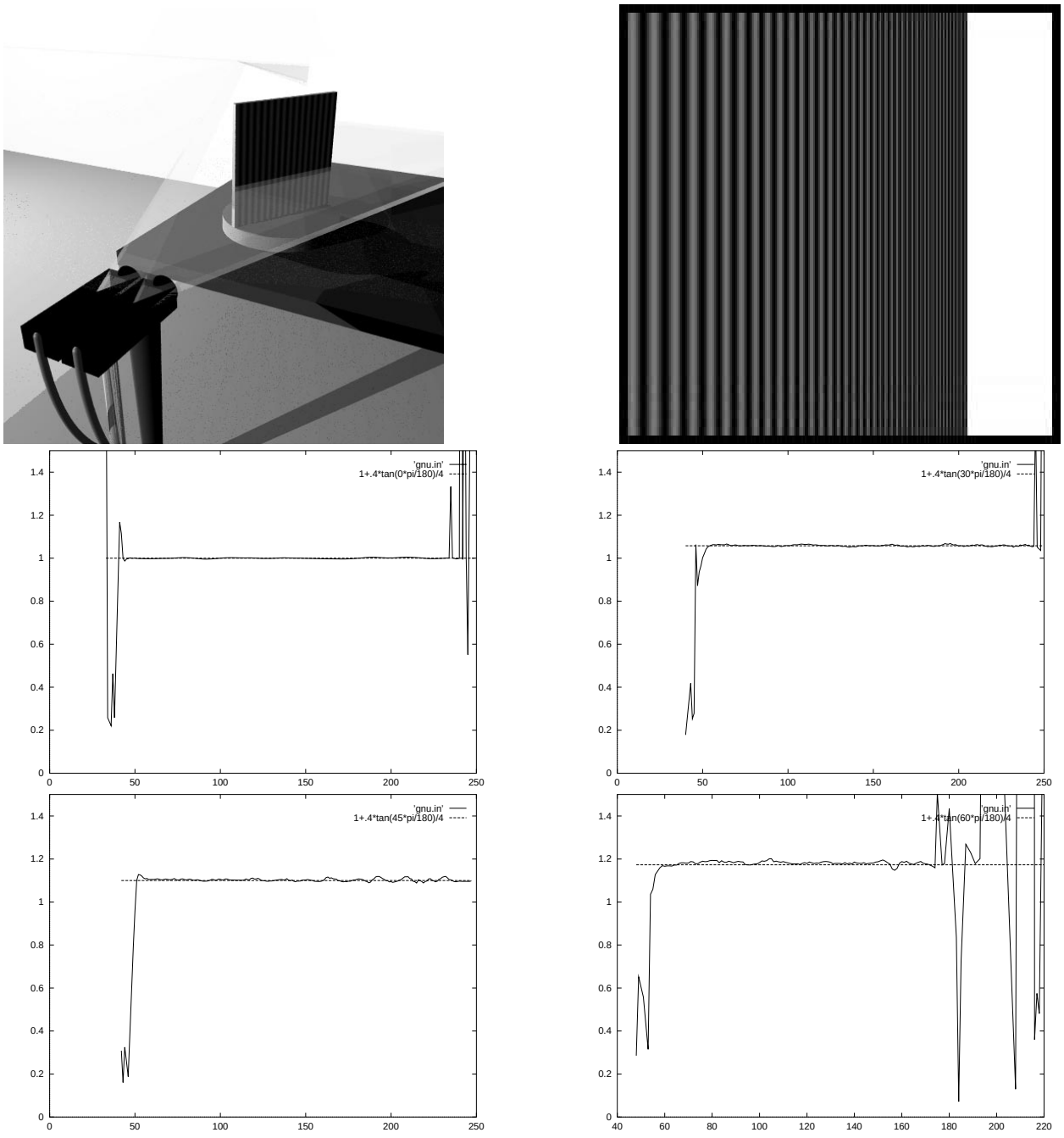

Figure 12: Verifying the Scale Factor - These graphs compare the predicted scale factor (solid line) against that computed using only image information (dashed line). The virtual lab setup (top left) and an example input image with surface angle of 60° (top right) are shown first. Next we have the results derived from a surface angled at 0° (middle left), 30° (middle right), 45° (bottom left), and 60° (bottom right). The virtual surface is 4.0 units from the left camera, both cameras have a field of view of 45° and are separated by a baseline of 0.4 (the surface in the actual images is larger than that shown in the top left rendering).

trying to solve the stereo problem at this point, we are simply trying to verify a property of corresponding pixels.

Having established the correspondence in the 2D images, we extract the apparent frequency at each pixel using the method described above, linearly interpolating the instantaneous frequency measurements from the right image. Finally, we graph the ratio of the computed image frequencies values against the predicted ratio in Figure 12, for several surface angles. The computed ratio is quite accurate but gets progressively less precise as the angle increases. The loss of precision occurs from several factors, e.g., our use of simple linear interpolation to compute the frequencies, and our filter set which only samples the highest frequencies very sparsely.

4.3 Applicability

How important is this foreshortening analysis? More specifically, how often do situations arise in which the assumption that a surface is frontoplanar can cause problems for stereo systems? Intuitively the analysis would seem to be needed any time a surface is slanted at a sharp angle; but what if the surface is so far away the slant can't be measured? One might also think it only necessary for surfaces at the sharpest angles; but close up images can exaggerate even small angles. We will use the Scale Factor to quantify these effects in the spatial domain.

Since we want to consider the scenery being imaged rather than the images themselves, we will use the geometric formulation of the Scale Factor from Equation 12. Although this Scale Factor is a function of three variables, we can reduce it to two if we consider the ratio of depth over baseline $\frac{Z_L}{B}$ to be a single variable. In the rest of this section the word *depth* will denote this unitless version of depth, expressed relative to the camera baseline. For example, the distance between a person's eyes would be 1, the distance to their computer monitor 4-6, and the distance to the far wall in a typically small three-person graduate student office about 100. Figure 13 plots the near-complete Scale Factor space for a person looking at objects in such an office.

Figure 13 shows the scale factor computed from many combinations of depth and orientation (except for the extreme values near the point at which it approaches infinity). The graph makes it clear that the Scale Factor has its greatest impact when objects are sharply slanted and/or located near the cameras. We can quantify its influence using the contour lines that

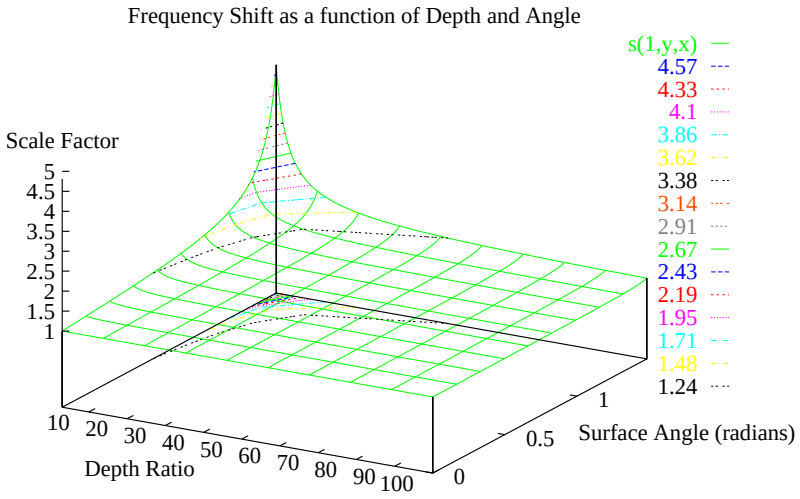

Figure 13: Frequency Shift as a function of Depth and Angle. Depth is unitless relative to the baseline, and varies from 3 to 100. Angle varies from zero to 85°.

separate regions of large and smaller foreshortening effects. Suppose we assume that surface depth and orientation are uniformly distributed throughout a scene. Then we can compute the probability that a surface will require at least a 10% correction term by finding the area under the 1.1 Scale Factor contour curve. The derivation follows in Appendix A, but the result is that given a uniform distribution of angles from 0° to 90° and depths from 0 to 100, the probability that a surface will require at least a 10% correction is 0.210355. Try it out; if you're sitting in an office, see if you can find one sharply foreshortened surface for each set of four nearly head-on surfaces in your immediate vicinity.

Of course the probability of finding foreshortened surfaces depends very much on the domain being studied. Robot vehicles like Carnegie Mellon's NAVLAB often use a very wide baseline, on the order of one meter. With the nearest visible ground point being about five meters away, depth ratios of 5 to 20 are common in this domain. In that range, under the same assumptions of uniform distribution, the probability of finding a foreshortened surface jumps to better than one in three (see Table 1). Inspection robots typically use much smaller baselines, with corresponding depth ratios from 30 to 100. Even in that range, the probability of finding a 10% foreshortened surface is significant (nearly one in twelve). These results suggest that a wide variety of stereo vision systems could benefit from an analysis that considers the effects of foreshortening.

Depth Range	P(10% effect)	Example Domain
0-100	0.210355	*Human in office*
5-20	0.354404	*Robot Vehicle*
30-100	0.0808227	*Inspection Robot*

Table 1: Probability that a surface exhibits 10% variation between images due to perspective foreshortening. The distribution of surfaces is assumed to be uniform within the range of orientation angles from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, and depth ratios (distance divided by baseline) are as specified.

5 Application

The mathematics developed in Section 4 is not only theoretically interesting, it can also improve the performance of real stereo algorithms. Phase-based methods such as [FJJ91], [San88] and [Wen90] as well as our method can benefit from this analysis. In this section we explain how to apply the Frequency Shift to these phase-based stereo matching algorithms and demonstrate how its application to our system increased the maximum matchable surface angle from 30 degrees to over 75 degrees.

5.1 Extending Phase-based Stereo Algorithms

Some have argued that a small number of Gabor filters are sufficient for stereo matching. [FJ91] [Wen90] The main motivation for this has been the claim that phase information is relatively stable over nearby frequencies. The idea is that although the phase may vary slightly across nearby frequencies, the amount of variation is small enough that the error introduced in measuring it at what might be the wrong frequency will be insignificant. But the assumption is made that the same filters can be applied to both images, i.e., that both images can be sparsely sampled at the same set of spatial frequencies. As was shown in the preceding section, that assumption is not true when perspective foreshortening occurs in the images. Instead of introducing error by sampling at the wrong frequency, we would like to turn these perturbations to our advantage by using them to confirm hypotheses of surface tilt.

We will need a dense sampling of the phase space to get the most accurate results. We

Given: A pair of images containing greyscale intensities.

Lists of potential disparities and surface angles.

Camera focal length f.

For each row

Compute Left and Right Scalograms L and R

For each column c

For each disparity d

For each angle a

$$correction = 1 + \frac{d \tan a}{c \tan a - f}$$

$$error = \sum_{\lambda:\rho(\lambda)>threshold} \rho_L(c,\lambda)\,|\phi_L(c,\lambda) - \phi_R(c+d, \lambda \cdot correction)|$$

Return d (and a) that yield minimum *error*

Table 2: Pseudocode for the foreshortening-corrected algorithm. For purposes of computing the correction factor the column index c must be zero in the center of the image.

will also interpolate phase values between adjacent frequencies when possible. The image scalogram provides a useful framework for such computations, and will be used as the basis for our foreshortening-corrected stereo algorithm.

The method outlined in Section 3.5 uses a global minimization strategy to find the best disparity from a list of candidates. This framework makes it easy to include a foreshortening correction term: in addition to searching disparity space, we also search over surface angle. Pseudocode for this revised algorithm is given in Table 2. The only difference between this and the earlier algorithm is the presence of the correction term on the right image phase measurements. This simple presentation of the algorithm is only made feasible because of the large number of filters used in the scalogram. The large filter set gives us a dense set of phases at many scales from which to compute the appropriate subsampled phases.

There are several implementation details that arise from this simple correction factor. It depends on three variables: the currently hypothesized disparity, surface angle, and the current location within the image. Because these values vary at each pixel on the image scanline, it must be recomputed for each hypothesis. And as was mentioned above, the

corrected frequency will almost certainly not be one of those already present in the scalogram; some method of interpolation will be required. These are not serious problems, but imply that their implementation will be very compute-intensive.

5.2 Results

We added the correction term to the algorithm presented in Section 3.5 using linear interpolation between adjacent phases. In this section we present the results of our method on real images that have been synthetically mapped onto planar surfaces. The use of synthetic data allows us to quantify its precision using perfect knowledge of the ground truth.

Consider the stereo pair in Figure 1. It shows a synthetic stereo image pair of a flat plate rotated 65 degrees from the image planes, with the image of a city scene texture-mapped onto the plate. The actual disparity map (computed from the 3D world model) and results computed from the image pair by three stereo methods are presented in Figure 14. The figure shows disparity maps rendered as perspective surfaces; only the area known to have texture is shown since the plain white background makes depth recovery impossible in those areas. Figure 14 also includes images of the difference between the computed disparity and that which is known from the world model.

For this demonstration of the foreshortening-corrected algorithm, a set of 501 potential disparities were considered (0 to 50 in steps of 0.1), and the angle was fixed at 65 degrees. The RMS error of this result is 0.38 pixels over the entire plate, with $\sigma = 0.63$. The bulk of this error can be attributed to two causes: the sharp spikes and a subtle but systematic error over the surface. The spikes most likely arise from an artifact of the rendering process which caused a few nearby pixels in one image to map to the same intensity. The more subtle effect is that the disparity error, which is within measurement bounds at the ends and center of the plate, tends to uniformly vary as much as 0.5 pixels between the center and end of the plate (see Figure 14, upper right).

Contrast our foreshortening-corrected results with those of the Kanade-Okutomi variable-window refinement method [KO90]. This method uses correlation to match windows around pixels, but uses a statistical analysis to grow the window from 3x3 to some maximum, stopping when an error criterion (based on local changes in intensity and disparity) is exceeded. For this test we let disparity vary between 0 and 50 pixels (as in our method), let the window size vary from 3 to 21 pixels, and ran the method for 10 iterations. It did an admirable job

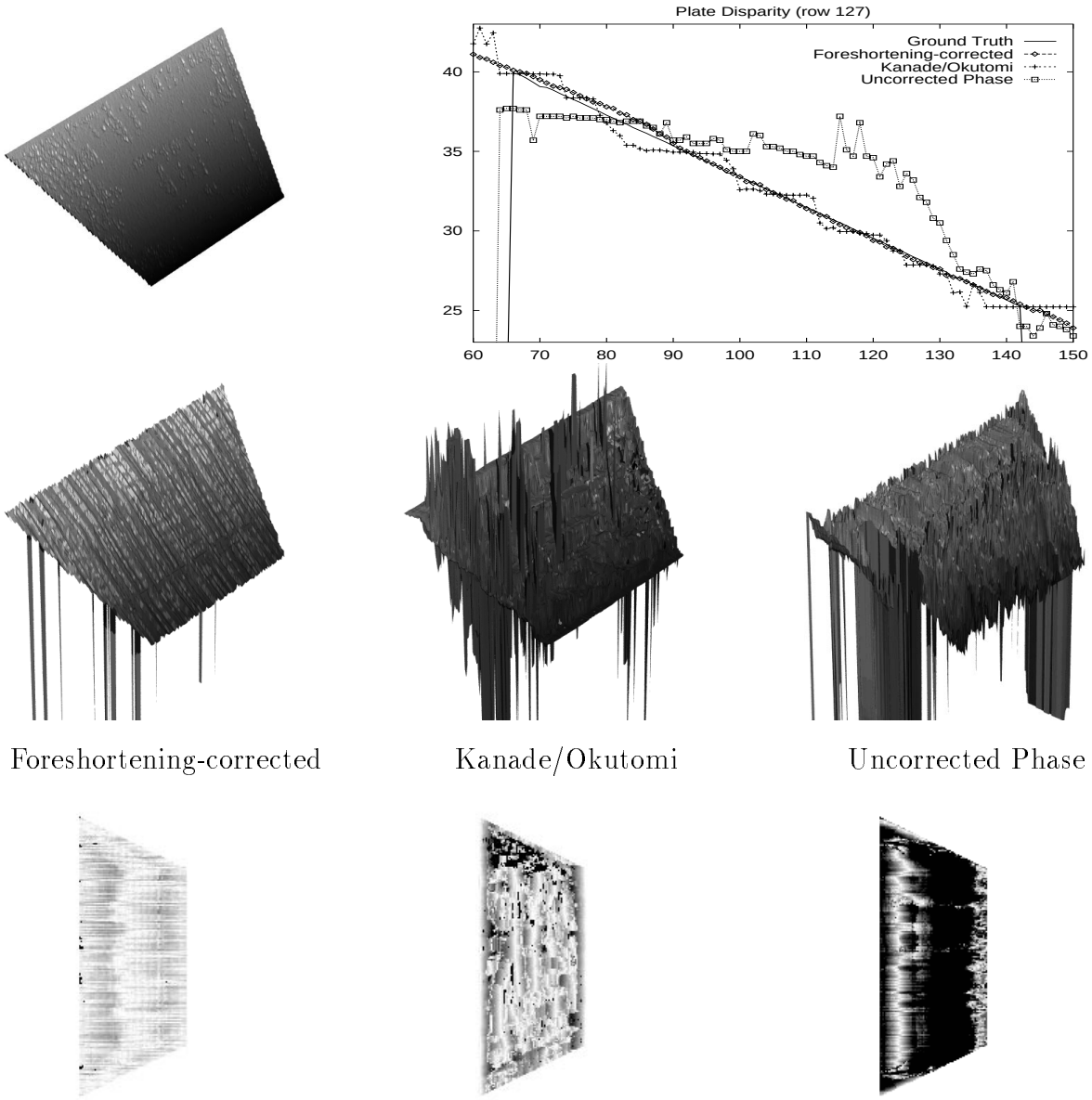

Figure 14: Ground Truth and computed disparity maps for a surface angled at 65°. The top row shows ground truth on the left, a graph of a representative scanline from all methods on the right. The middle row shows perspective views of the disparity maps computed by the foreshortening-corrected method, Kanade/Okutomi and the uncorrected phase method. The bottom row shows differences between computed and actual disparities for pixels that image the plate; darker values denote larger errors. Only differences between 0 and 2 pixels are shown, errors larger than 2 pixels appear as a 2 pixel error. Actual plate disparities range from 25.3 to 39.9 pixels.

of approximating the surface shape, but produced many more outliers and quantized the flat tilted surface into several stair-step frontoplanar patches (see Figure 14, upper right). The RMS error of this method is 0.99 pixels over the entire plate, with $\sigma = 2.36$.

For completeness the uncorrected phase method results are also shown in Figure 14. The same 501 potential disparities were considered, but foreshortening correction was not applied. The RMS error of this result is 3.77 pixels over the plate, with $\sigma = 6.23$. The main source of error is a general flattening trend over the entire plate, most likely due to the larger windows used at lower frequencies. Like most traditional stereo matchers, the uncorrected method has a strong bias toward frontoplanar surfaces, but unlike Kanade/Okutomi this uncorrected phase method is unable to restrict its attention to the smallest-sized windows.

Other Rotation Angles A cross-section of results for different angles of rotation is presented in Figure 15. For these results only a representative scanline is shown, to demonstrate how closely the computed disparity matches the actual ground truth. Only the disparities on the plate itself are correct because the region behind it is a plain white background, and there is no way to distinguish the correct disparity of a featureless surface.

The uncorrected method does reasonably well with small angles, but at slants greater than 30° its performance degrades by several pixels. In contrast, the foreshortening-corrected method performs well even at 75°, though at 80° the systematic error becomes more obvious.

6 Conclusion

We have presented a theory for modeling the physical effects of perspective foreshortening in stereo vision systems. The crux of the theory was the development of the dual Scale Factors that allow us to reason about foreshortening both in the geometric domain of the world model, and the frequency domain of the stereo images. We presented experimental results validating both forms of the scale factor, and showed how it can be applied to phase-based stereo matching systems. Applying it to our Gabor filter-based system increased the system's maximum matchable angle from 30 degrees to over 75 degrees.

7 Acknowledgements

Thanks to Larry Matthies for providing the city scene used in the synthetic examples.

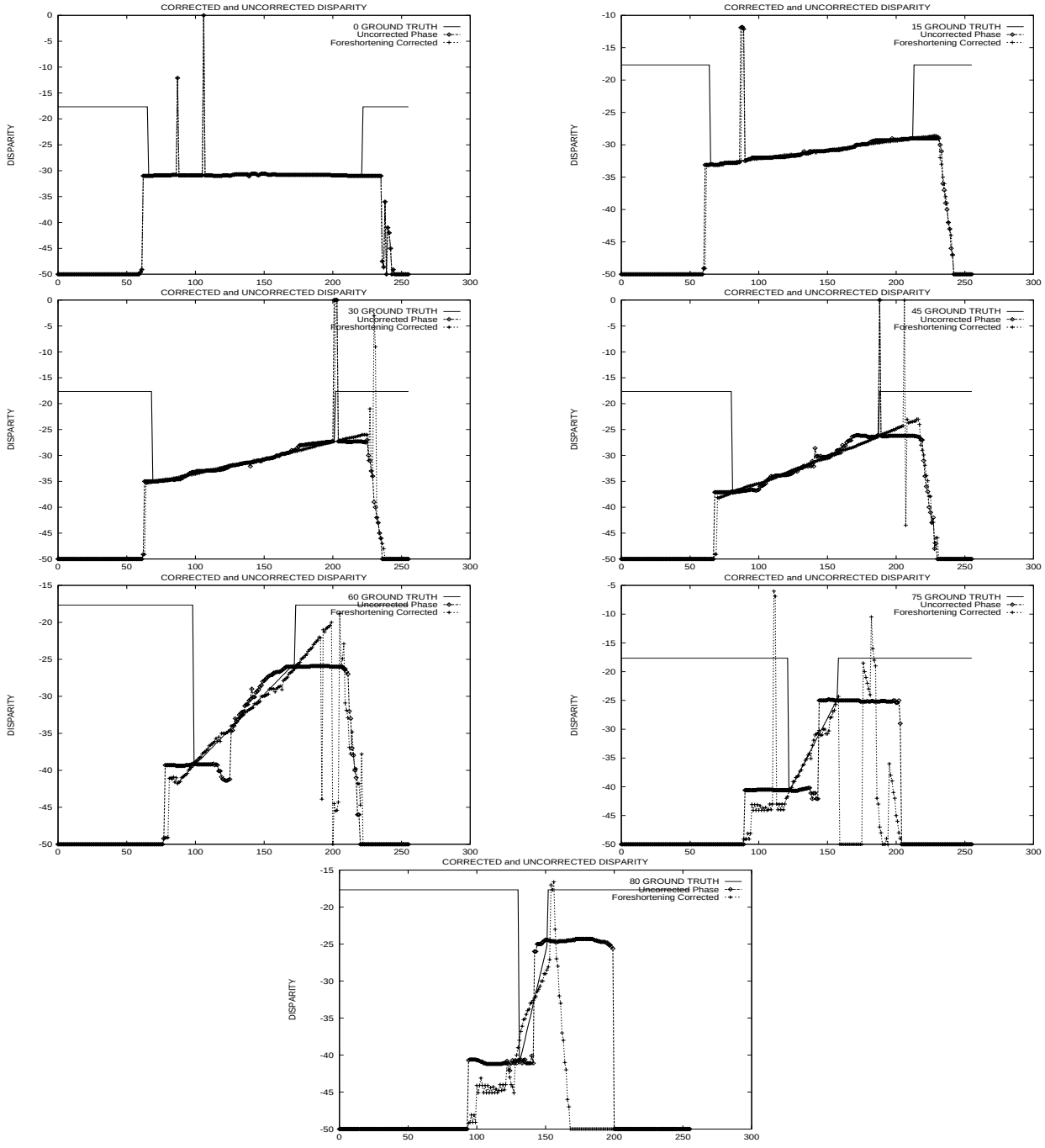

Figure 15: Ground truth and disparity (computed by both the uncorrected and foreshortening-corrected phase methods) for the center scanline of the city scene at various rotations. From left to right (and top to bottom): 0, 15, 30, 45, 60, 75, and 80 degrees.

A Derivation

We present here the derivation of one of the probabilities from Table 1. This derivation assumes the depth range begins at zero (the more general results require a little more work).

We want to find:

$$P(\text{Scale Factor} \geq 1.1 \text{ or } \leq 0.9) = P\left(\left|\frac{\tan\theta}{d}\right| \geq 0.1\right)$$

for $d = \frac{Z_L}{B} \in [0:100]$ and $\theta \in [-\frac{\pi}{2} : \frac{\pi}{2}]$. Since tan is symmetric we can eliminate the absolute value by restricting the angle θ to $[0 : \frac{\pi}{2}]$. Continuing:

$$\begin{aligned} P\left(\left|\frac{\tan\theta}{d}\right| \geq 0.1\right) &= P\left(\frac{\tan\theta}{d} \geq 0.1\right) = \frac{\int_0^{\frac{\pi}{2}} \min\left(\frac{\tan\theta}{\text{Scale Factor}-1}, 100\right) d\theta}{\int_0^{\frac{\pi}{2}} \int_0^{100} dd\, d\theta} \\ &= \frac{\int_0^{\frac{\pi}{2}} \min\left(\frac{\tan\theta}{0.1}, 100\right) d\theta}{50\pi} \end{aligned}$$

To eliminate the min from the integral we must find the minimum angle requiring a 10% correction at distance 100:

$$\theta_{\min} = \arctan 100(\text{Scale Factor} - 1) = \arctan 10 = 84.2894^\circ$$

Now we can split up the integral into two parts and evaluate it:

$$\begin{aligned} \int_0^{\frac{\pi}{2}} \min\left(\frac{\tan\theta}{0.1}, 100\right) d\theta &= \int_0^{\theta_{\min}} \frac{\tan\theta}{0.1} d\theta + \int_{\theta_{\min}}^{\frac{\pi}{2}} 100 d\theta \\ &= \left[\frac{\ln\sec\theta}{0.1}\right]_0^{\theta_{\min}} + 9.96688 \\ &= 23.0756 + 9.96688 \end{aligned}$$

This brings us to the final result:

$$P\left(\left|\frac{\tan\theta}{d}\right| \geq 0.1\right) = \frac{33.0425}{50\pi} = 0.210355$$

So under the assumption of uniform distribution on depth ratio from 0 to 100 and angle from -90° to 90°, the probability of a surface exhibiting at least a 10% foreshortening effect is 0.210355.

References

[Bel93] Peter N. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken surfaces in the presence of half-occlusion. In *International Conference on Computer Vision*, pages 431–438, 1993.

[BFBB92] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 236–242, June 1992. revised in July 1993 Queens U tech report, RPL-TR-9107.

[FJ91] David Fleet and Allan Jepson. Stability of phase information. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 52–60. IEEE Computer Society Press, October 1991.

[FJ93] David Fleet and Allan Jepson. Stability of phase information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1253–1268, December 1993.

[FJJ91] David J Fleet, Allan D Jepson, and Michael R M Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, March 1991.

[JJ94] Michael R. M. Jenkin and Allan D. Jepson. Recovering local surface structure through local phase difference measurements. *CVGIP: Image Understanding*, 59(1):72–93, January 1994.

[JM91] David G. Jones and Jitendra Malik. Determining three-dimensional shape from orientation and spatial frequency disparities part ii - using corresponding image patches. Technical Report UCB/CSD 91/657, University of California, Berkeley, Computer Science, October 1991.

[KO90] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. In *DARPA Image Understanding Workshop Proceedings*, pages 383–398, September 1990.

[Kru94] John Krumm. Shape from texture and segmentation using local spatial frequency. Technical Report CMU-RI-TR-93-32, Carnegie Mellon Univiersity Robotics Institute, May 1994.

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

[MP89] Jitendra Malik and Pietro Perona. A computational model of texture segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–332, June 1989.

[OK91] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 63–69, 1991.

[RBH94] Luc Robert, Michel Buffa, and Martial Hebert. Weakly-calibrated stereo perception for rover navigation. In *ARPA Image Understanding Workshop*, 1994.

[Ros93] Bill Ross. A practical stereo vision system. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 148–153, 1993.

[RV91] Olivier Rioul and Martin Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–38, October 1991.

[San88] Terence D Sanger. Stereo disparity computation using gabor filters. *Biological Cybernetics*, 59:405–418, 1988.

[Tri77] José Tribolet. A new phase unwrapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(2):170–177, April 1977.

[Wen90] Juyang Weng. A theory of image matching. In *Third International Conference on Computer Vision*, pages 200–209, Los Alamitos, California USA, 1990. IEEE Computer Society, IEEE Computer Society Press.

[XS94] Yalin Xiong and Steven A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. Technical Report CMU-RI-TR-94-28, Carnegie Mellon Univiersity Robotics Institute, September 1994.