Characterizing Stereo Matching Problems using Local Spatial Frequency

Mark W. Maimone

May 1996

CMU-CS-96-125

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891 USA

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Thesis Committee:
Steven A. Shafer, *chair*
Martial Hebert
Paul Heckbert
Michael Jenkin, York University

This research was sponsored in part 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 author 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: Computer Vision, Stereo Matching, Local Spatial Frequency, Gabor Filters, Scalogram, Phase-based Stereo, Scale Space, Perspective Foreshortening, Ambiguous (false) Matches, Ground Truth

Abstract

The model of local spatial frequency provides a powerful analytical tool for image analysis. In this thesis we explore the application of this representation to long-standing problems in stereo vision. As the basis for this analysis, we develop a phase-based algorithm for stereo matching that uses an adaptive scale selection process. Our approach demonstrates a novel solution to the phase-wraparound problem that has limited the applicability of other phase-based methods.

The problem of ambiguous matches, or false targets, can greatly reduce the accuracy of a stereo vision system. A common approach to alleviating the problem is the use of a coarse to fine refinement strategy, but we show that this approach imposes some (perhaps overly) strong requirements on the stereo images. Our phase-based method relaxes those requirements, and is therefore able to handle a wider variety of otherwise ambiguous images. But sometimes ambiguity is inherent in the images, so we propose a generalized disparity model to explicitly represent multiple candidates.

Perspective foreshortening, an effect that occurs when a surface is viewed at a sharp angle, can reduce the precision of stereo methods. Many methods tacitly assume that the projection of an object will have the same area in both images, but 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 our phase-based system, increases the system's maximum matchable surface angle from 30 degrees to over 75 degrees.

The analysis of stereo vision algorithms can be greatly enhanced through the use of datasets with ground truth. We outline a taxonomy of datasets with ground truth that use varying degrees of realism to characterize particular aspects of stereo vision systems, and show that each component of this taxonomy can be effectively realized with current technology. We propose that datasets generated in this way be used as the foundation for a suite of statistical analyses to effectively characterize the performance of stereo vision systems.

Acknowledgements

I am indebted to many people for the completion of this thesis.

My thanks go first and foremost to my advisor Steve Shafer. He helped me jump start my research career in vision and image processing, he encouraged me to develop my mathematical debugging skills, and his vision and ability to present ideas clearly have been inspirational. I am especially grateful for his encouraging comments over the years, and for his support during the final wrapup and writing process.

The other members of my committee have done a great deal to improve the content of this thesis. Thanks to Martial Hebert for providing overall direction, Michael Jenkin for our face-to-phase discussions, and Paul Heckbert for teaching me something new at every meeting. This document has benefited greatly from their excellent suggestions, as well as those of Yalin Xiong.

I would also like to express my sincere appreciation for the support, wisdom and friendship of A. Nico Habermann, founding Dean of the School of Computer Science. With his encouragement I learned to look beyond the present and literally reach for the stars.

For financial support, I thank CMU, the Office of Naval Research, and the folks at the NASA Ames Vision Research Lab led by Beau Watson.

The members of the Calibrated Imaging Lab, including Joyoni Dey, John Krumm, Bruce Maxwell, Reg Willson and Yalin Xiong, have been good friends and outstandingly supportive colleagues throughout my development as a vision researcher. Carlo Tomasi helped me overcome my initial fear of Fourier, and the dozens of members of the Vision and Autonomous Systems Center and its director Takeo Kanade have made this an intellectually challenging and personally satisfying environment. I am especially grateful for the support and friendship of Mei Chen, Jennie Kay, Pete Rander, Henry Rowley, Harry Shum, David Simon, Michael Smith and more.

The CMU Computer Science Department has been an incredible place to work. Students like Darrell Kindred, Corey Kosak, Henry Rowley, Bennet Yee, and the amazing folks of SCS Facilities have been a tremendous resource for all things technical. Our liaisons to the world outside of the department do an impossibly good job of hiding the gory details; I cannot give enough thanks to Sharon Burks, Catherine Copetas, Maggie Muller, Karen Olack and the whole SCS Support Staff for their friendship and all of their hard work (but I am certain Aunt Catherine is going to make me try :-).

My musical colleagues helped keep me sane during this whole process. Many thanks to the members and directors of the CMU Jazz Choir and Concert Choir, CS folks Angela Kennedy, Su Yuen Ling, Cliff Mercer, Scott Neal Reilly, Benli Pierce, John Velonis, and especially Geli Zobel for her dedication to singing and keeping me well fed. Special thanks to Robert Page for his energy, inspirational direction, support and personal encouragement of my musical goals.

Thanks too to the many other friends who have made the trip enjoyable in the good times, and bearable in the bad. Classmates Nevin (and Tess) Heintze, Puneet Kumar, Juan Leon, Francesmary Modugno, Joe Tebelskis, and Amy Moormann Zaremski helped start and keep me going. Officemates over the years put up with me and my strange musical tastes: Doug Reece, Raul Valdes-Perez, John Cheng, Keith Gremban, Ari Rapkin and Doug Rohde. And though I'm running out of space and cannot possibly include everyone, Drew Anderson, Eka Ginting, Chris McGlone, Phil Miller and my first advisor Jeannette Wing have all made special contributions. Warm thanks also to Patty, Mary, Chris, Tom and the rest of the late night crew.

Most of all, I thank all of my family for their encouragement, especially my parents and grandparents, without whose love and support none of this would have been possible.

Thanks to Larry Matthies for the image texture used in Figure 1.3 and Chapter 5. Thanks also to Blue Mountain Arts, Inc. for permission to use the image in Figure 4.1 (details follow).

Notation

Below is a list of notations that are used in this thesis and some of their consequences:

$\lvert x\rvert_b$	**Remainder**	remainder(x,b), or x modulo b, or $x - \lfloor x/b \rfloor b$
$x =_b y$	**Modular Equality**	remainder(x,b) = remainder(y,b). Note that for all $a \in \mathbb{R}$, $x =_b y$ implies $x+a =_b y+a$, and for all $a \neq 0$, $x =_b y$ implies $ax =_{\frac{b}{a}} ay$
f	**Focal Length**	Intrinsic camera (lens) parameter associated with the pinhole lens model
$\omega \left(= \frac{1}{\lambda}\right)$	**Frequency**	Cycles per unit distance. Although this symbol is often reserved for *angular* frequency (in units of radians), we use ω to avoid confusion with focal length f. Frequency is the reciprocal of wavelength.
$\lambda \left(= \frac{1}{\omega}\right)$	**Wavelength**	Sinusoid or filter wavelength, often in units of pixels
$f(x) \Longleftrightarrow_{\mathcal{F}} F(\omega)$	**Fourier Transform**	Function $F(\omega)$ is the frequency-domain equivalent of spatial domain function $f(x)$. In this work $f(x)$ will be real-valued, thus $F(\omega)$ will be complex and odd-symmetric about the origin, so only the positive frequencies will be plotted.
ϕ	**Phase**	Phase of a complex number (often a Fourier or Gabor coefficient $\rho e^{j\phi}$), or phase of a sinusoid $\sin(x-\phi)$
ρ	**Magnitude** or **Amplitude**	Magnitude of a complex number (often a Fourier or Gabor coefficient $\rho e^{j\phi}$)

continued

$[low, high]$	**Range (inclusive)**	The set of numbers from *low* to *high* including the endpoints.
$\lfloor x \rfloor$	**Floor**	Floor of x, i.e. the greatest integer $i \leq x$.
$*$	**Convolution**	
$\{x : property\}$	**Set Constructor**	The set of all x such that *property* is true.
$X \cap Y$	**Set Intersection**	If X and Y are sets, the set of elements common to both.
$\|S\|$	**Cardinality / Absolute Value**	If S is a set, the number of elements in S. Else if S is a scalar, the absolute value of S.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is a foolish thing to make a long prologue, and to be short in the story itself.

— Second Maccabees 2:32

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 passively 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) 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, observing a scene comprising diffuse objects.

The best way to measure the success of a stereo method is to compare its results against the *ground truth*, or range information measured using means other than stereo. Acquisition of total and accurate ground truth is very expensive, often prohibitively so. But just as there are trade-offs in modeling complexity v. precision in the design of vision algorithms, so too there are trade-offs in the analysis of their performance. Synthetically generated imagery gives total ground truth knowledge, but fails to model the complexities of real-world imaging. Real imagery provides better test data, but greatly reduces the amount and density of ground truth available for analysis. If computer vision is to become more of an engineering discipline than craftwork (Shaw, 1990), engineers must be able to predict and experimentally characterize the behavior of their systems. Such characterization is only possible when ground truth is available. In Chapter 2 we outline a framework for choosing a reasonable trade-off of ground truth density v. image realism in the analysis of stereo algorithms.

Our approach to the stereo problem will explore the benefits of a *local spatial frequency* representation. There are two fundamental extremes in the study of 2D imagery: the detail-oriented *spatial* view and the wholistic *frequency* view. In the spatial view each individual pixel is paramount; an image is represented by the concatenation of independent pixel values, and each pixel is considered unique and important. In the frequency view only the entire image matters; although the image is broken down mathematically into several frequency components, information in each component relates to the image as a whole. Each view has its benefits: the spatial view can represent discontinuous textures and local (pixel-level) segmentations directly, while the frequency view is a mathematically elegant representation that enables many useful analyses over large regions. The trouble is, the benefits of one view are difficult to attain in the other. Fortunately, a compromise can be reached, one which preserves the localizability of the spatial approach and the analytical benefits of the frequency approach: this combined approach is called *local spatial frequency,* and forms the foundation for the discussion of our stereo method in Chapter 3.

The evaluation of any experimental method requires a characterization of the accuracy and precision of its results. By *precision* we mean the number of significant digits in the numeric result, and by *accuracy* we mean the agreement between the experimentally-derived result and the ground truth. The claim that for all stereo images, all pixels have a disparity of 3.14159 pixel units is very precise, but probably not very accurate. In contrast, the claim that for all stereo images, all pixels have a disparity of 0 ± 512 pixel units is accurate, but not very precise. Chapter 4 will show how to improve accuracy using better models of ambiguous matches and Chapter 5 will show how to improve the precision of stereo results in the presence of perspective foreshortening.

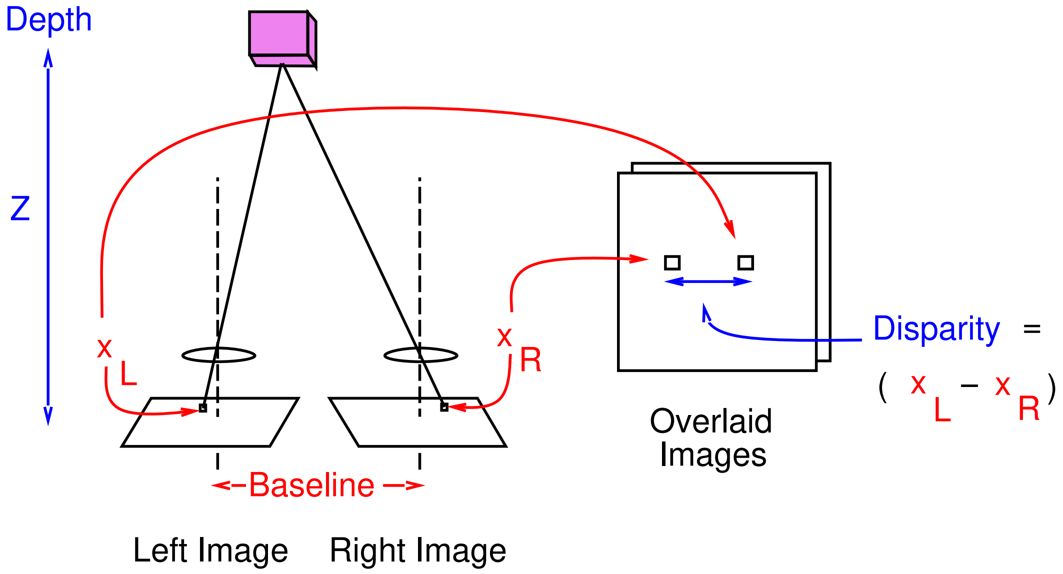


Figure 1.1: Illustration of the geometry of a typical stereo vision system.

1.1 Terminology

Some of the terminology used in describing the behavior of stereo algorithms can be described as follows:

Local Spatial Frequency Two common approaches to image interpretation are frequency analysis (e.g., using Fourier transforms) and direct pixel-based analysis (spatially dividing a picture into a grid of pixels). For years only one of these approaches was used at a time: frequency analysis for global effects, spatial analysis for local effects. But now *local spatial frequency* representations allow both techniques to be used simultaneously, at the cost of greatly increasing the amount of data used to represent the image. A 1D scanline and its Fourier transform can each be expressed in one dimension, but a local spatial frequency version of the same scanline will span two dimensions. Some examples of local spatial frequency representations are wavelets, Wiener filters, scalograms (which are comprised of Gabor filters), and spectrograms (short time Fourier transforms).

Disparity 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. The stereo 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.

Disparity Image The goal of all stereo methods is the creation of a *disparity image* (also known as a *disparity map*) from a pair of intensity images. Pixels in a disparity image represent not the intensity of incoming light, but rather the amount of disparity assigned to a pixel in the original image pair. Unfortunately the range of disparities can sometimes be very small, so for purposes of visualization all the disparty images in this thesis have been scaled in intensity to bring out as much structure as possible. Smaller disparities indicate objects that are far from the camera, and are rendered with dark grey intensities. Pixels that are occluded (or for which no disparity is known) will be rendered in black. Objects that are nearer have light grey intensities; there will be no purely white or black pixels representing a numeric disparity. Unfortunately our intensity scaling occurs independently in each disparity image, so it is not possible in general to compare absolute intensities between disparity maps. When such a comparison is important, the disparities from a single scanline will be plotted separately so their numeric values can be seen.

Disparity Error Image When the ground truth is known, it is possible to display not only the raw disparities returned by a stereo method, but also the difference between the computed and actual disparities. This is illustrated in a *disparity error image*, in which pixel intensity indicates the absolute difference between the ground truth and measured values. Lighter intensities indicate better matches, and black pixels indicate an absolute error of two or more pixels. Unlike the disparity maps, pixel intensities here span the whole range from white (no error) to black (two or more pixel error). Areas in which ground truth is either not available or not appropriate are rendered in white.

Evaluation Function The function that measures the difference between two small areas from a pair of images is called the *evaluation function.* It gives the pair of areas a

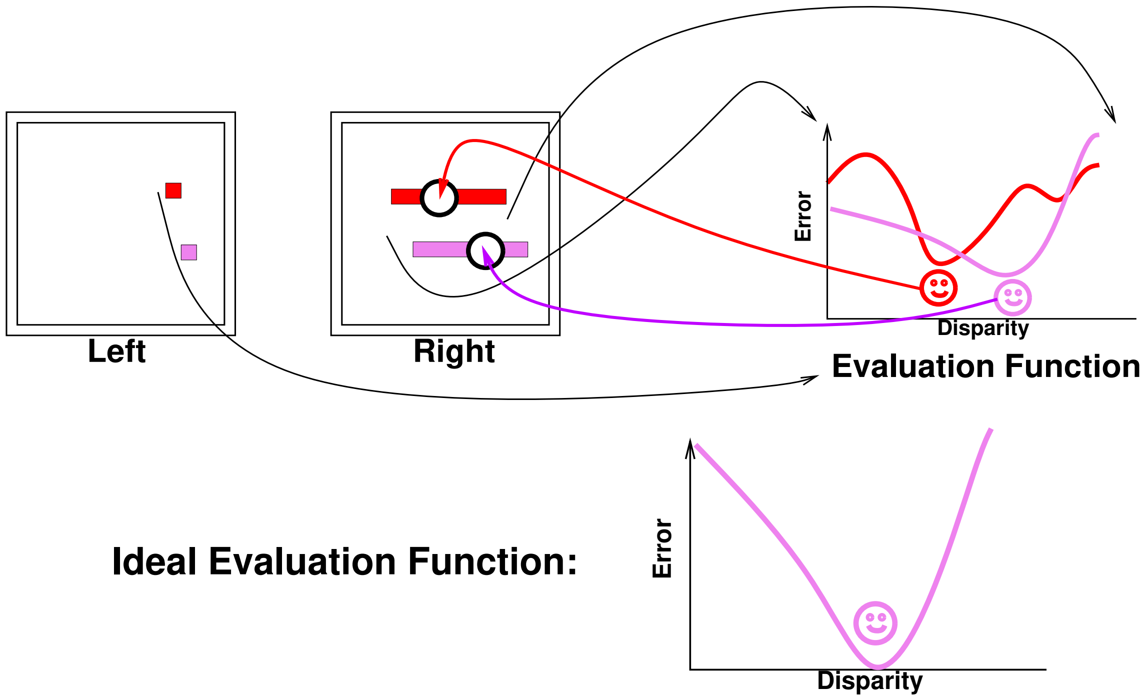


Figure 1.2: Evaluation functions are the tools used to find corresponding pixels. Given a pixel in the left image, a set of pixels in the right image is evaluated. The right pixel that exhibits the smallest error in the evaluation function is selected as the correspondent.

numeric value that indicates the likelihood that the two areas represent different images of the same 3D object. Some common evaluation functions include cross correlation, sum of squared differences (SSD), and sum of absolute differences (SAD). The best match between areas is indicated by larger values for cross correlation, but by smaller error values for SSD and SAD.

Evaluation function profile Since each pixel in one image will be compared with a number of pixels in the other, a visualization of the results of these comparisons can be helpful; see Figure 1.2. The *evaluation function profile* provides this information for a given pixel. The x-axis is Disparity, and the y-axis is the result of applying an evaluation function to the pair of pixel areas (i.e., this pixel and the one that corresponds with disparity x in the other image). The evaluation function profiles that appear in this thesis all use a SAD-like evaluation function, in which smaller values indicate a better match. Thus the y-axis label will be "Error" to reflect the fact that larger values

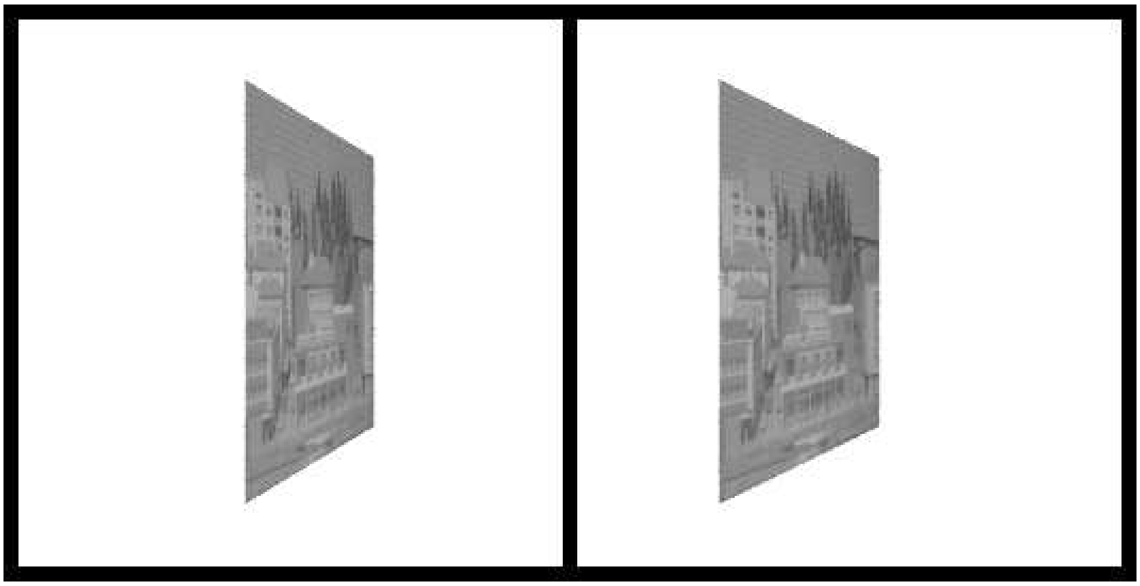


Figure 1.3: Stereo pair illustrating the effects of foreshortening; image compression, differing spatial extents.

indicate worse matches, i.e., matches having greater error.

Window Size The core calculation in all stereo methods is the comparison of a small area in one image with a similar area in the other. The width of this area is known as the *window size* because the comparison occurs only through this small "window" on the original image. While this region might be as small as a pixel or as large as the entire image, often it is a rectangular region several pixels wide. Although windows can have height as well as width, for much of this work we will be concerned with windows that are only 1 pixel tall.

Several factors encourage designers of area-based stereo algorithms to use small search windows. Smaller windows not only reduce the computational cost of a method, but also improve its accuracy by reducing the number of pixels affected by the blurring that inevitably occurs near depth discontinuities, since the search windows are typically run over entire images without regard to an initial segmentation. They cannot be *too* small, however; trying to match single pixels between images is notoriously error prone, since lighting effects and measurement errors often cause corresponding pixels to have different intensities in the stereo images. Even when the corresponding pixels have similar intensities, very often *many* nearby pixels share similar intensities.

Foreshortening

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 1.3. 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 Chapter 5 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 its maximum matchable surface angle from 30 degrees to over 75 degrees.

1.2 The Scalogram: A Unified View of Scale Space

We will use a model of Local Spatial Frequency to address long–standing problems in stereo vision. 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. Certain 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.

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 general overview of all these representations can be found in (Rioul & Vetterli, 1991), with a nice overview of their application to computer vision in (Krumm, 1993).

1.3 Stereo Vision Algorithms

1.3.1 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.g., laser rangefinders, sonar, and radar), but we will limit our scope to the passive imaging method of stereo matching.

The minimum system requirements for stereo vision are a pair of cameras positioned with

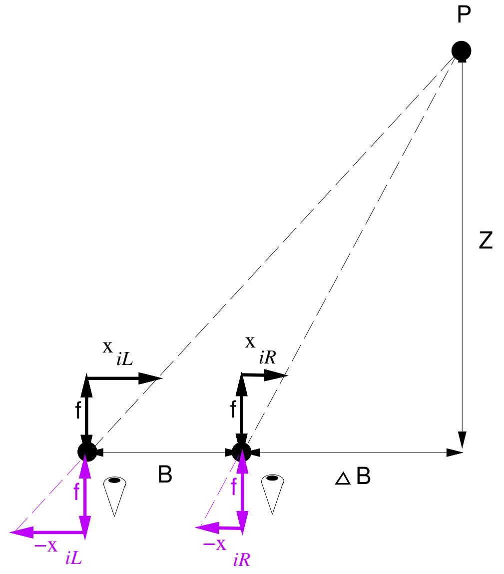


Figure 1.4: Overhead view of a typical Stereo Vision setup. A pair of cameras with focal length f, separated by baseline distance B, are represented by their focus points on the lower left. Object point P has corresponding image coordinates x_{iL} and x_{iR}, and lies at depth Z from the focus points. Mirror image vectors $-x_{iL}$ and $-x_{iR}$ are shown to make the similar triangles used in Equation 1.1 more explicit.

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 (Robert et al., 1994), Fundamental Matrix recovery, explicit hardware control (Ross, 1993)) and thus can reasonably be avoided here.

A constraint that is often applied to stereo systems is the *epipolar constraint*. The main idea is that given a pixel in the left image, one need not search through the entire right image looking for a correspondent. Instead, attention may be limited to a straight line, the so-called epipolar line. Why is it a line instead of some 2D region? The object represented by a pixel in the left image must lie on a ray that extends in the world from the left focal point through that pixel on the left image plane. The epipolar line is the image on the right image plane of that ray in the world; the projection of a straight line is still a straight line. A nice introduction to epipolar anaysis can be found in (Boufama & Mohr, 1995), and a very detailed description in (Faugeras, 1993). When the optical axes are parallel, as we have assumed here, we enjoy the further property that epipolar lines are guaranteed to be horizontal, and therefore all correspondents will lie on the same-numbered scanline in the other image.

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 1.4. Similar triangles give us two equations relating the pixel indices x_{iL} and x_{iR} to depth Z:

$$\frac{\Delta B}{Z} = \frac{x_{iR}}{f} \qquad\qquad \frac{B + \Delta B}{Z} = \frac{x_{iL}}{f} \tag{1.1}$$

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

$$Disparity = x_{iL} - x_{iR} = \frac{Bf}{Z} \tag{1.2}$$

Equation 1.2 gives pointwise disparity only; we will show how to extend this description to surfaces at arbitrary angles in Chapter 5.

1.3.2 Problems with Traditional Methods

In spite of the strong physical constraints available to stereo methods, many problems remain in their implementation. While the mapping from disparity (i.e., pixel correspondence) to depth is well understood, the automatic extraction of disparity is still subject to error. A complete review of all stereo matching problems is beyond the scope of this work, but we will review some of the basic advantages and disadvantages of the standard approaches here.

In feature-based stereo, it is assumed that the objects in the world have distinguishing features that can be easily extracted by a 2D image processing operator. Features might be objects of a particular color, areas with sharp intensity contrast in one direction (edges) or in four directions (the Moravec operator), or any other easily detectable property. One advantage to the use of features in stereo is that the properties of the 3D object represented by the 2D features are usually very well understood, and can therefore be located with high (subpixel) precision. Another advantage is a computational savings; instead of searching every pixel in the other image, there will generally be only a small number of features detected in the corresponding epipolar line. Thus fewer comparisons are required, and only those image areas that exhibit interesting structure will be checked.

There are many disadvantages to feature-based stereo, however. Since correspondence is only establishing at a small number of pixels, the resulting depth map will be very sparse. Thus any underlying shape may be difficult to extract. There may be large areas in the images that exhibit only small changes in visible texture, and therefore will not have obvious features; no disparity will be computed in such areas. And features visible in one image might not be found in the other; this can cause nonrobust methods to match features incorrectly, resulting in highly inaccurate disparities. Thus while feature-based matching may be useful in applications where the features are known to be visible and only sparse disparities are required, it is not so useful for extracting dense depth maps.

Correlation-based stereo, on the other hand, can provide completely dense depth maps. Subpixel precision may also be obtained, by interpolating the evaluation function's error values in the neighborhood around the minimum disparity. Some of the feature-based methods' problems due to the incorrect matching of features can be alleviated by increasing the size of the correlation window, to provide a disambiguating context around the features. And while this type of stereo requires much computation, there are already many real-time implementations of correlation-based stereo, so computation time is not a bottleneck.

Unfortunately, although correlation-based stereo provides dense disparity maps, there is no guarantee that the results will be accurate. Some of the issues that arise in picking a window size have been discussed in Section 1.1, but the main problem is that the results at depth discontinuities and occlusion boundaries are generally quite unreliable; the extent of an object tends to "spill over" into the background, or vice versa. Also, use of the same-sized matching window in both images means that objects are assumed to be basically planar and viewed from the front. Finally, traditional correlation-based stereo performs matching based on a single view of the world; more robust methods use multiple views, either acquired using additional cameras or generated from the image pair using filters.

1.4 Related Work

Many types of stereo algorithms have been published in the literature. Overviews of the then-strongest techniques can be found in (Barnard & Fischler, 1982) and (Dhond & Aggarwal, 1989), while more recent results on image registration are described in (Brown, 1992). Existing stereo algorithms from 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., (Mori et al., 1973) and the two image case of (Okutomi & Kanade, 1991)), 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 gives the disparity estimate. This procedure is applied to the image at successively higher resolutions in a coarse-to-fine manner, restricting the search according to the disparity estimate from the previous (lower resolution) image.

In feature-based stereo (e.g., (Matthies, 1989)), a dense image is converted into a spatially sparse set of features (e.g., corners, edges) which are then 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., (Sanger, 1988)) the original signal is transformed to Fourier space, and some part of the transformed signal is used to compute the disparity. Often the phase of the transformed signal is used, in any of several possible ways (Jenkin & Jepson, 1994). Since our method evolved from those methods which use the phase component of frequency, we present in Section 1.4.1 a short history of phase-based stereo methods and discuss the improvements each method has made over its predecessor.

Several frequency-based methods do not use phase as the primary matching feature, however. Jones and Malik (Jones, 1991) applied 2D oriented derivative-of-Gaussian filters to a stereo pair, and used the magnitude of the filter responses at each pixel as matching features. Their iterative framework allowed them to address scale space selection, half occlusion, and perspective foreshortening, but their solution requires an initial disparity map that is computed without the benefit of models for these problems. Thus their solution is more of a constraint-based post-processing refinement technique. The cepstral method first presented in (Yeshurun & Schwartz, 1989) and improved upon in (Smith & Nandhakumar, 1996) also works in the frequency domain, but uses cepstral magnitude as the matching feature.

Another post-processing refinement technique is the Kanade/Okutomi variable-window method (Kanade & Okutomi, 1991). This method addresses the occlusion and foreshortening problems by dynamically adjusting the size of the matching window according to constraints on the local variation of both intensity and disparity. In this way they attempt to avoid the boundary problems that arise when the correlation window encompasses two objects at different depths. While the method seems to work very well at depth discontinuities, we will see in Chapter 5 that its ability to handle foreshortening problems is limited (its depth estimates are approximately correct, but imprecise).

Perhaps the most promising spatial domain approach for two image stereo is that of (Belhumeur, 1993). The method uses dynamic programming to segment each scanline into a number of surfaces, and evaluates the surface matches using a Bayesian probability functional; the segmentation with greatest probability yields the final result, which is a completely dense disparity map. Although computationally expensive, the real power of this method is its use of the hypothesized segmentation; it is much easier to compare two objects, whatever their shapes, if you know their boundaries in advance. His examples show excellent results even in the presence of occlusion, surface creases, and foreshortening, but only involve a small number of separate surfaces.

Another successful spatial domain approach is the use of multiple cameras, sometimes known as multibaseline stereo. First described in (Kanade et al., 1992), the approach advocates using a relatively simple SAD stereo matching algorithm over several image pairs. By incorporating multiple views of the world using known camera calibration, many of the shortcomings of the direct yet simple SAD method are eliminated: e.g., specular highlights are ignored, noisy disparity maps become smoother, and some occluded surfaces become visible. Recent extensions have addressed the issues of occlusion (Nakamura et al., 1996) and real-time stereo (Kanade et al., 1995) more directly, but the current approach still suffers from window effects at depth discontinuities and fails to explicitly address foreshortening.

1.4.1 Evolution of Phase-based stereo

One of the first applications to exploit phase as translation was the Kuglin-Hines Phase Correlation Image Alignment method (Kuglin & Hines, 1975). The method assumes that two images — related by pure translation — overlap in some way, and it is conceptually very

```
function y = kuglin (l, r)
% Compute translation between 2D matrices l and r

pdiff = angle(fft2(l))-angle(fft2(r));
kuglin = ifft2 (exp (i*pdiff));

[ rowmax, rowind ] = max(kuglin);
[ allmax, colind ] = max(rowmax);
y = [ rowind(colind) colind ] - 1;

[ row col ] = size (l);
y = y - (y(1) > row/2) .* [row 0] - ...
        (y(2) > col/2) .* [0 col];
end;
```

Figure 1.5: Matlab implementation of the Kuglin-Hines Phase Correlation Image Alignment method (Kuglin & Hines, 1975). Actual computation of the global translation vector occurs in the first two lines of the function, the final six lines extract the indices of the maximum element in a 2D array. The method actually achieves subpixel precision, but this unrefined index-processing code only returns integer results.

simple: take the image FFTs, subtract phases, and apply the inverse FFT (see Figure 1.5). The result is a matrix whose element with maximum magnitude corresponds to the best-fit global translation. In that sense the technique is similar to image correlation, but phase correlation results in a matrix that has better properties: fewer maxima, and a sharper peak around the principal maximum (Horner & Gianino, 1984). Although it returns a single translation vector, it can still be used in the context of stereo vision by comparing subwindows within a larger image. Such a strategy was adopted by Szeliski in (Szeliski, 1994), in which this method provides initial estimates for his image mosaicing technique.

Although the Kuglin-Hines method succeeds at demonstrating the potential of phase-based methods, it has several important problems when considered as a stereo vision solution. Chief among these is the fact that it fails to account for Fourier coefficients whose magnitudes are so small that the phase is unreliable (see Section 3.4.3). Although the paper mentions that the phase differences can be weighted by magnitude, they provide no details or insight on how to accomplish this. The following method provides one solution to that problem.

The Windowed Fourier Phase (WFP) method by Weng (Weng, 1993) is similar to the multiple-window use of the Kuglin-Hines method, but uses variable-sized windows, and linear interpolation of phases in regions where the magnitude is too small. The justification for interpolation comes from the idea of instantaneous frequency: if the phase derivative is constant, then a linear interpolation between known phases is appropriate. This method uses far fewer filters within each region ($O(\log n)$ where n is the width of the largest window) than Kuglin-Hines ($O(n)$), and allows the window size to vary so that the measurements made by high-frequency filters can be better localized. The filters are assumed to be band pass, and are applied in a coarse-to-fine manner to handle large disparities. Truly ideal filters would respond at a single frequency, but would require input signals of infinite extent, so in practise band pass filters are the best we can expect.

Unfortunately, this method has several problems. The core of the algorithm assumes that phase is linear, or as he calls it "quasi-linear;" that is, in regions where the magnitude is low, the phase is simply interpolated across the scanline to preserve its supposed linearity. In many real images, phase measurements are simply too nonlinear, and this fact cannot be detected reliably simply by using the fixed magnitude threshold he proposes. Also, the use of a pure box filter as the windowing function results in filters that are *not* band pass, but rather have multiple peaks in the frequency domain. Mention is made of the possible use of Gaussian windows (the so-called WFG), but all the results and further theory rely on the box filter. Results obtained with the WFP have been found to be less reliable than those from other phase-based techniques (Jenkin & Jepson, 1994).

To better approximate the band pass behavior desired in the filter set, Sanger proposed a method using Gabor filters (Sanger, 1988).[1] Unlike the coarse-to-fine method of Weng, Sanger's method combined independent disparity estimates from each filter simultaneously, with a weighting function designed to eliminate outliers and filter out phases with dissimilar magnitudes.

Although this work predates Weng's, it uses a better filter set and handles filter magnitude

[1]The seminal work of Gabor (Gabor, 1946) showed that there are fundamental limits on the resolution attainable by any local-spatial-frequency filter, and the class of filters he described which achieves the maximum possible space/frequency resolution are now known as *Gabor filters*. The real and imaginary components of these filters are sinusoids modulated (multiplied) by a Gaussian window.

more effectively. Still, it still suffers from a naive integration of filter estimates. For example, while Weng's range of candidate disparities is limited to half the width of the largest filter, Sanger's is bounded by half the width of the *smallest* filter. The similar-magnitude constraint is also ill-founded; it assumes that each filter's magnitude will be similar over the entire extent of each image of the same world object. Filter magnitudes rarely appear equal between two images, and are in any case a less important feature than phase (Oppenheim & Lim, 1981) in this context. Finally, this work assumes that the response of a given filter corresponds exactly to the filter's peak frequency. But since the filter is band pass, it may in fact be measuring responses from any frequency (or many) within a range.

The work of Fleet, Jepson and Jenkin (Fleet et al., 1991) improved this model by allowing the filter's frequency to vary, providing additional phase measurement constraints, and employing a coarse-to-fine strategy (similar to that later used by Weng) thus eliminating the smallest-filter restriction of Sanger's method. Instead of assuming that measured frequencies correspond to the filter's peak frequency, the local phase derivatives from the two images were combined, the assumption being that each filter measured a single frequency at some point in its band pass region. The additional phase constraints also did a better job of filtering out "unstable" phase measurements, thus improving the resulting disparity estimates.

However, this work failed to address the control-strategy problems produced when phase estimates at coarse scales are unavailable. Presumably a simple linear interpolation of the surrounding disparities was used to fill in regions where all the phase information from the $O(\log n)$ filters was not available (i.e., the Gabor filter outputs failed to satisfy Equation 3.10). And while the instantaneous frequency improved the resulting precision somewhat, it was used in a fairly *ad hoc* manner, simply averaging frequency estimates from the two images.

A much more sophisticated solution was developed by Xiong (Xiong, 1995). Instead of assuming that each filter output represents a response at a *single* frequency, he models the response of each Gabor-like filter as a polynomial. Various properties of his so-called *hypergeometric* and *moment* filters make them a much better representation for high-precision matching, since they are able to model directly the effects of finite-width windows and small amounts of perspective foreshortening. Xiong shows how the constraints developed by Fleet *et al* represent a constant approximation to his more general representation. Use of these filters that account for the window effect and local foreshortening yields results that can be orders of magnitude more precise than those of other methods.

However, like the other methods mentioned above this too is limited to a small range of disparities, being bounded by the filter width. This framework also only allows for a small amount of perspective foreshortening, since it compares phases using the same filters from each image. And finally, like most other stereo methods, this method uses a limited model of disparity that is unable to represent the possibility of an ambiguous match. These topics will all be addressed in the forthcoming chapters.

1.4.2 Foreshortening

Local spatial frequency has already been identified as a valuable tool for modeling surface shape and segmenting multiple textures in a single image (Krumm, 1993; Malik & Perona, 1989). 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 applied to optical flow problems (Barron et al., 1994; Xiong, 1995), using phase information as well as magnitude. The stereo problem is more constrained 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 (Jones & Malik, 1991) applied local spatial frequency to the problem, but used an affine transformation matrix in the *spatial* domain without providing a description of its effect in the frequency domain. Belhumeur (Belhumeur, 1993) 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 (Kanade & Okutomi, 1990) 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. The cepstral method of (Smith & Nandhakumar, 1996) is claimed to be robust in the presence of foreshortening, but only slants up to 43 degrees have been demonstrated.

Several phase-based stereo methods have been described in the literature (Fleet et al., 1991; Sanger, 1988; Weng, 1993), and a review of the more popular variations can be found in (Jenkin & Jepson, 1994). Although some of these mention foreshortening as an issue, none has explicitly modeled it or corrected for it.

1.5 Relationship to Prior Work

Some of the results in this thesis have appeared (often in compressed form) in other papers. Chapter 2 is a revised version of (Maimone & Shafer, 1996) and includes more information about the CMU Calibrated Imaging Lab (CIL) Stereo Datasets with Ground Truth that have been made available on the Internet.[2] The specific procedures for using CMU CIL facilities to acquire such data are presented in some detail in (Maimone, 1995). Earlier versions of Chapter 5 and portions of Chapter 3 appeared in (Maimone & Shafer, 1995a) and (Maimone & Shafer, 1995b), but the sample image pair was unreadable, the Internet URL has changed, and the pseudocode in Table 2 of the former reference was missing an important term. Additional information is available on the Internet via the *Calibrated Imaging Lab home page*,[3] *Mark Maimone's index page*,[4] and the *Computer Vision home page*.[5]

1.6 Thesis Organization

Chapter 1 has introduced the problems that inspired this thesis, given background to provide the context for understanding them, and discussed the related work. In Chapter 2 we describe a general framework for the acquisition of high-precision datasets and independent validation of stereo results, and discuss the particular datasets used in this thesis. Chapter 3 presents the concept of local spatial frequency and outlines the phase-based stereo method that provides the foundation for later discussion and analysis. Chapter 4 addresses the problem of ambiguous matches, and Chapter 5 that of perspective foreshortening. Finally, Chapter 6 lists our contributions and summarizes their impact.

[2] `http://www.cs.cmu.edu/~cil/cil-ster.html`

[3] `http://www.cs.cmu.edu/~cil/cil.html`

[4] `http://www.cs.cmu.edu/~mwm/`

[5] `http://www.cs.cmu.edu/~cil/vision.html`

Chapter 2

A Taxonomy for Stereo Computer Vision Experiments

The beauty of truth, as of a picture, is not acknowledged but at a distance.

— Joseph Glanvill, *The Vanity of Dogmatizing XV*

Much of computer vision research is an attempt to solve the impossible: to acquire full three-dimensional knowledge given limited two-dimensional data. The state of the art has advanced to a point where there now exists a plethora of partial solutions to computer vision problems. We're getting lots of answers, but just how accurate are they? A few methods provide an estimate of uncertainty with each answer, but those uncertainties do not tell us what we *really* need to know: by how much does the estimated answer differ from the truth?

If computer vision is to become more of an engineering discipline than craftwork (Shaw, 1990), engineers must be able to predict and experimentally characterize the behavior of their systems. Such characterization is only possible when ground truth is available. Synthetically generated imagery gives total ground truth knowledge, but fails to model the complexities of real-world imaging. Real imagery provides better test data, but greatly reduces the amount and density of ground truth available for analysis. In this chapter we outline a framework for choosing a reasonable trade-off of ground truth density v. image realism in the analysis of stereo algorithms.

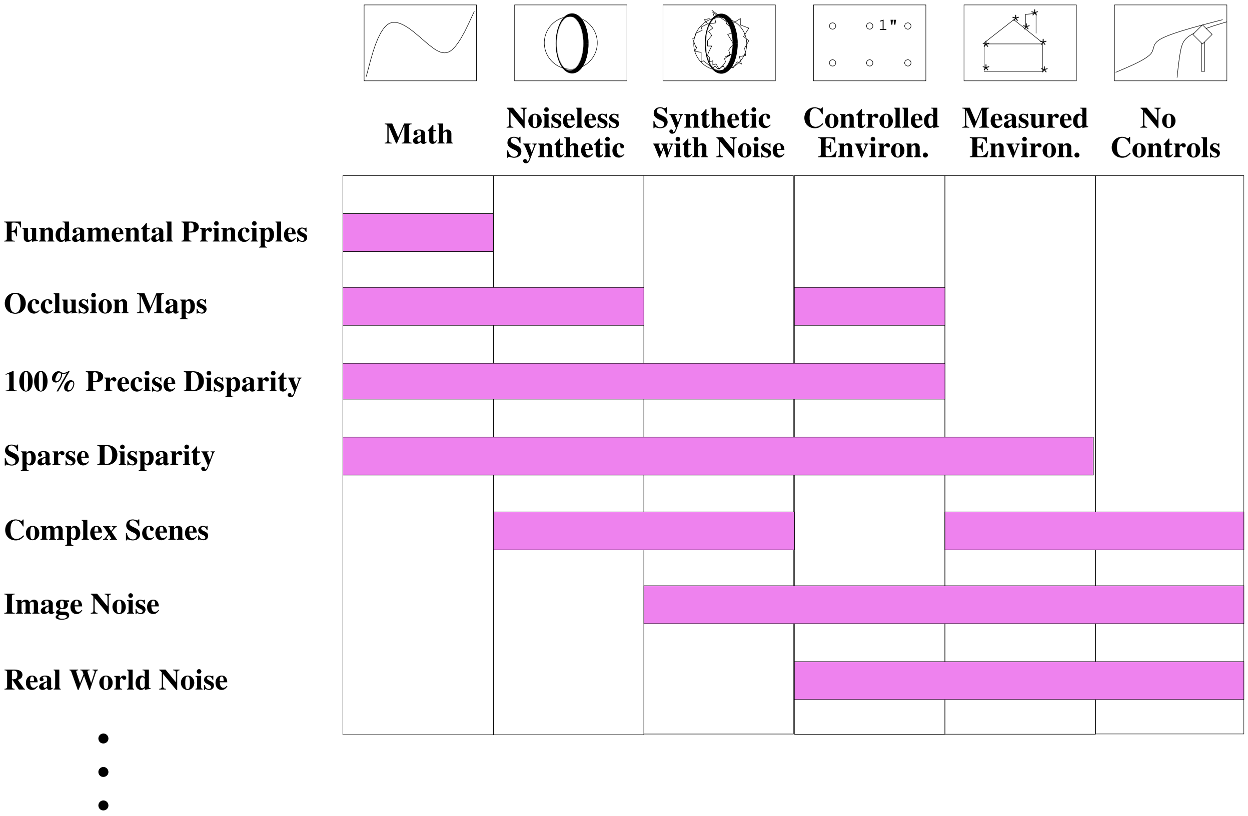


Table 2.1: Stereo Imagery Characteristics and the types of scenarios for which they are available.

In performing the research described in this thesis, especially in Chapter 5, it was determined that the only way to effectively judge its success would be to use stereo imagery with precise ground truth. Unfortunately, very little such data was publicly available, so it became necessary to create our own datasets. We found that there were several distinct scenarios through which ground truth could be acquired, each serving an important function in the evolution of the stereo algorithm. What follows is a description of these various scenarios and the types of situations in which each is most appropriate.

2.1 Overview

The characterization of a vision algorithm's performance is a tedious and difficult task. This is primarily due to the lack of appropriate performance measurement specifications. This chapter will enumerate the scenarios and types of data that can be used, give examples of interesting properties of stereo vision systems, and explain how best to measure them, in the sense of achieving the most accurate and representative ground truth.

Table 2.1 illustrates some interesting properties of stereo vision data, and matches them with the scenarios in which they can be measured. Because many properties can be measured using any of several scenarios, only a few are likely to be needed for a particular analysis set. The list of rows in Table 2.1 is merely suggestive; other properties can be easily matched with the appropriate scenarios.

A complete statistical framework to take advantage of this data is beyond the scope of this work (see (Matthies, 1989) for an approach assuming unimodal disparity error). We will instead discuss the representation of the data and ways to acquire it, with the understanding that its most effective use will be in the context of a complete analysis package. Such a package would measure interesting properties over several subimages, e.g., the whole image, occluded regions, non-occluded regions, pixels above a known ground plane, etc.

Although comparisons of stereo algorithms have been done before, many have suffered from a lack of available ground truth. One such study, the ARPA JISCT stereo evaluation (Bolles et al., 1993), compared the results of four stereo methods. However, since ground truth was not available, most of their statistics dealt with agreement between the results; not "method A is 80% accurate", but "methods A and B agree on 80% of the image". Thus they could neither evaluate stereo methods independently nor quantitatively characterize their performance. The study's conclusion states in part that "Ground truth is expensive, but there is no substitute for assessing quantitative issues."

Even those studies that included ground truth have been limited by available technology. An ISPRS study (Gülch, 1991) compared the results of several stereo algorithms using ground truth, but in all but one image pair each true pixel disparity was computed *manually,* without interpolation. Their manual collection of 23,000 total pixel correspondences from eleven 240 $\times$ 240 image pairs was a noble, not to say Herculean, effort, but we argue that the technology for the construction of ground truth images of complex scenes (with much greater density and less required manual intervention) not only exists, but is easy to use.

We demonstrate this claim by outlining a taxonomy of ground truth scenarios (abbreviated in the column headings of Table 2.1), and providing a concrete example for each level. Sections 2.2 through 2.7 present the scenarios, ordered by increasing realism and decreasing amount of ground truth, Section 2.8 describes the tools used to implement these scenarios and their availability, and Section 2.9 summarizes our contribution.

2.1.1 Representation of Results

Stereo vision is a powerful approach for computing 3D information, but unlike active rangefinding systems stereo works indirectly, by converting pixel correspondences into depth measurements. The quality and density of the resulting depth map depends directly on the character of these correspondences.

Unfortunately, even exact pixel correspondences alone rarely give a complete picture of the range map. This is due to several effects, e.g., nonoverlapping fields of view in parallel and outward-verging cameras, self and half occlusion of objects in the scene, and a lack of intensity variation in areas with bland texture. It is unreasonable to expect a correspondence-based stereo method to calculate completely dense depth maps, since even perfect pixel correspondences can leave gaps in their implied depth maps.

How then should stereo data be represented? There are several possibilities, principal among them disparity maps, depth maps, and object models. Researchers have reported stereo results using all of these representations, but each has its drawbacks. We would prefer to use a representation that allows different stereo methods to be compared on an equal basis. Object models are quite useful, but can only be computed from raw stereo data by making many model-based assumptions. Depth maps would be ideal, but require exact knowledge of the camera geometries (which may not be available), and cannot be completely computed from correspondences alone. To compare results computed from two images without requiring camera calibration information then, disparity maps with occlusion masks are the most general representation.

One need not eliminate metric information completely when providing disparity maps, however. By simply including the parameters of the actual camera system in the dataset (e.g., baseline and camera focal lengths), disparity can be converted easily to depth when needed. Since depth resolution of a stereo system can be increased simply by adjusting the camera separation and/or focal length, the precision of a stereo method is often measured in pixels (units of disparity) rather than Euclidean length (units of depth). But most applications will express requirements in depth units, so this metric information should still be kept available.

The ground truth must also be expressed as a disparity map. This is accomplished by running the 3D ground truth information through the appropriate camera model, to generate a depth map in the same coordinate system as that of the image being matched. From here it is a simple matter to generate a disparity map, and we will see in Section 2.3.1 how to compute occlusion masks from disparity maps derived in this way.

The model for stereo experimentation is thus to run the stereo algorithm, and compare the computed disparity map with that of the ground truth known from other methods. The rest of this chapter presents several scenarios and examples of test data with ground truth, explains the benefits and limitations of each, and discusses some implementation issues that arose in the course of generating the sample data. Thus we demonstrate that the technology for creating datasets with interesting properties and dense ground truth already exists, and is easy to use.

2.2 The Basics: Mathematical Foundations

The first scenario is that of continuous functions. When describing an algorithm, the first step should be to demonstrate the principle in the simplest possible domain. In stereo vision, the simplest case is typically the comparison of two 1-D functions that represent scanlines. There is often much insight to be gained by focusing attention to a level of detail in which all quantities can be interpreted directly.

This is the level at which the general principles of an algorithm can be demonstrated. At this point it is not necessary for the inputs to have precisely the same qualities as those present in actual discrete imagery. Indeed, using continuous functions as input can often simplify the presentation by allowing the solution to be expressed analytically (in closed form) rather than operationally (e.g., "the result after 10 iterations") as in (Fleet et al., 1991; Sanger, 1988).

This level of description is also useful for discerning and describing any theoretical limitations of the method, e.g., the points at which its assumptions break down.

2.2.1 Example 1: Phase difference as disparity

The use of phase difference as disparity lies at the heart of many phase-based stereo algorithms (Sanger, 1988; Weng, 1993; Fleet et al., 1991). But unless one is already quite familiar with the frequency domain (and Equation 3.6 in particular), the name itself inspires fear. Suppose one remained uncertain of the underlying principles; how could you convince yourself that the technique really works? By considering the simplest possible examples according to the representation, and following the processing step by step.

In this example we are interested in studying how the phase of a sine wave relates to stereo disparity. So consider the simple case in which the left and right image scanlines are both sinusoids. A one-dimensional sinusoid is in general completely determined by three parameters: amplitude (A), frequency (ω), and phase (ϕ).

$$\text{Sinusoid: } \boxed{\text{A}}\sin\left(2\pi\boxed{\omega}x - \boxed{\phi}\right)$$

For this demonstration we will fix the amplitude A at 1, frequency ω at $\frac{1}{8}$, phase of the left image at 0, and allow the right phase ϕ to vary freely:

$$L(x_L) = sin\left(\frac{2\pi}{8}x_L\right) \qquad\qquad R(x_R) = sin\left(\frac{2\pi}{8}x_R - \phi\right) \tag{2.1}$$

Stereo disparity is the amount of shift required to make the left and right images appear equal. While in general the disparity in an image will vary at every pixel, in our example all pixel disparities will be equal (this actually happens in a real image whenever a planar surface is viewed head-on, so it is a realistic assumption for this demonstration). Mathematically, disparity is the difference between the left pixel index (x_L) and the right pixel index (x_R). So we find disparity by setting the formulas in Equation 2.1 equal and solving for this difference:

$$\begin{aligned}
\sin\left(\frac{2\pi}{8}x_L\right) &= \sin\left(\frac{2\pi}{8}x_R - \phi\right) \\
\frac{2\pi}{8}x_L &=_{2\pi} \frac{2\pi}{8}x_R - \phi \\
Disparity := x_L - x_R &=_{8} -\frac{8}{2\pi}\phi
\end{aligned} \tag{2.2}$$

Thus we see that disparity is indeed related to the difference of the left and right phases (remember the left phase is zero in this example). Figure 2.1 graphically shows the disparities that result from particular values of the right function's phase. You can convince yourself that the mapping from phase to disparity works by plugging the values of ϕ into Equation 2.2 and comparing the answer with the amount of shift visible in the graphs of Figure 2.1.

This example also illustrates an important issue in the design of phase-based stereo techniques: phase-wraparound. In interpreting Figure 2.1, we knew the phase difference was relatively small (i.e., less than 2π), so the disparity could be computed directly. But the

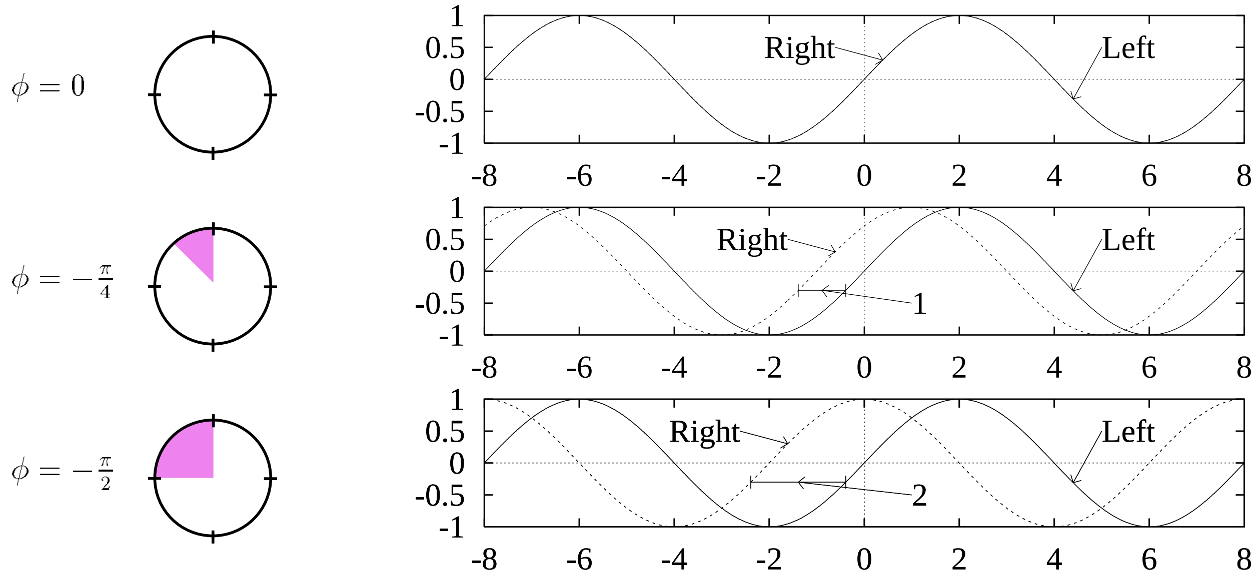


Figure 2.1: Disparity as a function of Phase Difference. Disparity is the horizontal separation between the two signals, and is indicated by the labeled bar just to the left of 0.

disparity formula in Equation 2.2 is only defined *modulo the wavelength of the sine wave.* This means we cannot compute a unique disparity from a single phase value; at this frequency, disparities of 1, 9, 17, ... all appear equivalent. This is an important observation: when evaluating any phase-based stereo method, be sure to consider how it addresses the problem of phase-wraparound. Some authors use a coarse-to-fine approach to alleviate it (Weng, 1993; Fleet et al., 1991), others choose to ignore it (Sanger, 1988) thus restricting themselves to finding only small disparities. Our solution combines phases from several filters at arbitrary frequencies simultaneously, so instead of wrapping around at the wavelength of the smallest or largest filter, our estimates wrap at the least common multiple of all wavelengths that comprise the signal (this is typically larger than the size of the image, and thus tends to yield a unique result). More detail can be found in Section 3.6.1.

This simple, direct analysis has resulted in several important insights: an understanding of the basic technique, and an appreciation for an important property of all phase-based stereo methods.

Limitations: While this level of demonstration is important in communicating the intuition behind an algorithm, it has many limitations. The most obvious is that when images are processed only discrete samples are measured, since images have a fixed resolution. So this general description must be restated in more concrete terms that take the limited resolution into account. Also, while a purely mathematical scene description is easy to reason about, complex scenes would require such detailed modeling that constructing a continuous version would be too cumbersome.

2.3 Noiseless Synthetic Imagery

The next scenario is that of discretely sampled synthetic imagery. Having established basic principles using continuous functions, the generalization of the method to complete images must be characterized. The broadening of attention from minute pixel-level details to those encompassing entire objects can also yield important insights. For these purposes synthetic data prove most useful. Such data may also be used to verify the implementation of an algorithm on full-sized images.

Synthetic images can be generated by any means, but should initially be created according to a model of the imaging environment in which an algorithm will be deployed. This model should come as close as possible to approximating the real world, though for the moment a noiseless environment should be assumed. In particular, objects being "imaged" should have 3D structure, and should be rendered using the same model as that assumed by the algorithm. For stereo vision this model will typically include full perspective projection, e.g. the pinhole lens model. Use of computationally simpler imaging models such as orthogonal projection or linear (affine) warping, should be avoided except in providing data for debugging an algorithm that makes that assumption. This task is not as difficult as it might seem: the Computer Graphics community has developed many realistic renderers, some of which are freely available and easily modified (Haines, 1993).

The camera model used by the synthetic image generator should be as similar as possible to that used in the actual laboratory camera calibration. Using the same model makes it possible to close the loop; 3D info computed from real imagery can be re-rendered from the same (now virtual) camera position. Figure 2.5 shows how useful this can be: the 3D locations of dots in a real image of a calibration grid are rendered according to the computed camera model, and overlaid on the real image. Such tools can make inspection and validation of 3D reconstructions much easier.

Perfect ground truth is also quite helpful in debugging. While this may seem a trite truism, the difficulty of developing image processing software and the current lack of integrated matrix debugging environments have discouraged vision software developers from adopting this approach of using full-sized image ground truth. Yet freely available software can provide arbitrarily complex test cases that can help the debugging process tremendously. The following example demonstrates how complete ground truth pointed out the flaws in a common technique for generating occlusion masks.

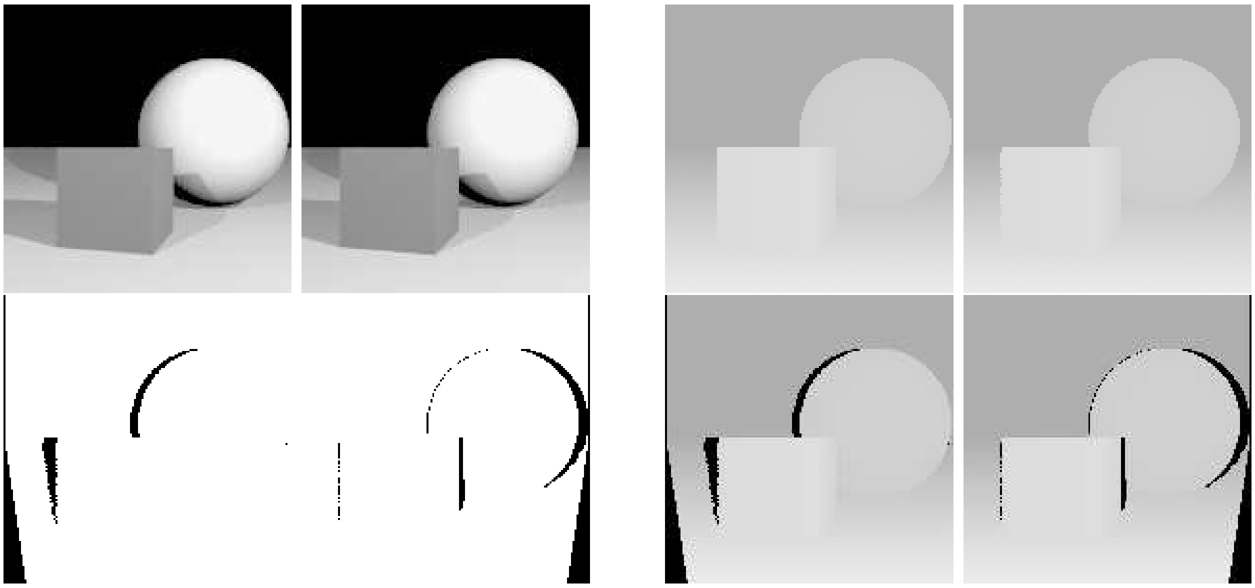
2.3.1 Example 2: Occlusion Masks

Depth maps in complex scenes are typically discontinuous, and therefore difficult to reason about analytically. Also, when opaque objects are imaged from different viewpoints, portions of those objects will be visible only in one image. For these reasons it becomes important to provide occlusion masks with stereo ground truth: correspondence-based stereo algorithms cannot be expected to predict accurate disparity in areas where no correspondence exists.

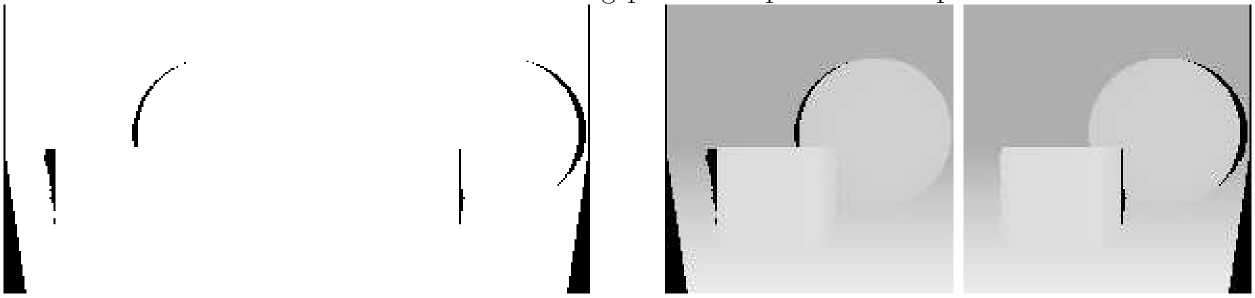
Occlusion masks are binary images that indicate which pixels are visible in both images of a stereo pair. They are defined relative to a pair of viewpoints: screen pixels in one viewpoint are occluded if the 3D point they represent is not visible from the other viewpoint. How can occlusion masks be generated? One method that works nicely with synthetic data is to simply place a point light source at the focal point of the second viewpoint and re-render the scene (Williams, 1978), taking care to turn off interreflections, translucency, and other light sources. Pixels in shadow are then marked as occluded. However, if the pixel subsampling should differ between the image and occlusion mask renderings, border pixels may be labeled incorrectly.

Another method works directly with the depth map used in the construction of the synthetic image. The top row of Figure 2.2 shows a sample stereo pair of images and their disparity maps. A popular technique in stereo algorithms computes occlusion masks by performing a pointwise comparison between the disparity maps for the left and right images. Any corresponding pixels whose disparities are not equal in magnitude and opposite in sign are marked occluded by this method (Jones, 1991). While this is a useful approximation, it often fails at object boundaries because of steeply-sloped surfaces, as illustrated by the results in the middle row of Figure 2.2. Since this algorithm produces noisy results even with absolutely correct disparity maps, a more robust approach is clearly needed.

There are two main problems with the pointwise method: sharply sloped surfaces may cause the corresponding pixels to point to the same surface but at very different depths, and



Resolution used for floating point comparison: 1.0 pixels



Resolution used for floating point comparison: 0.1 pixels

Figure 2.2: Occlusion Mask Generation: *Top Row:* Stereo pair of images from the Left and Right cameras; actual disparity maps for those images. *Middle Row:* Pointwise occlusion masks for left and right images (note especially the noise in the right mask around the object borders); disparity maps with pointwise occlusion masks overlaid. *Bottom Row:* Plane-fitting occlusion masks for left and right images; disparity maps with plane-fitting occlusion masks overlaid.

the pixel subsampling strategy may cause border pixels to point to the wrong object entirely. We can address these problems by fitting a plane to each pixel in the disparity map. If each scene object has an extent of at least two pixels, then it is reasonable to assume that for a given pixel, the 2×2 surrounding window with the least variation in depth will be the appropriate surface patch (see Figure 2.3). We compute each pixel's best plane by finding that 2×2 window which has the least variation in depth (i.e., for which $\max_{2\times 2}(\text{disparities}) -$

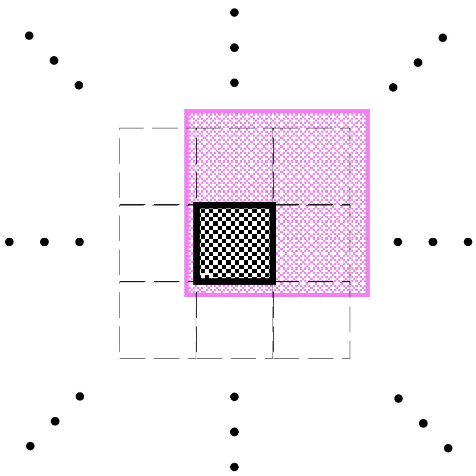


Figure 2.3: Finding the best-fit plane in the disparity map. There are four 2x2 windows that contain the center (dark) pixel; the upper right window is highlighted.

$\min_{2\times 2}$(disparities) is minimized). This window selection therefore helps avoid the pointwise method's error of pointing from a slanted surface to the wrong object. (Grant, 1992) provides a good overview of shading techniques in rendering applications. Our approach is most similar to his P2-buffer algorithm, which also fits a plane to adjacent depth values.

The problem of varying depth on a surface is addressed by using the *range* of disparities found in the 2×2 surrounding window selected above. Instead of comparing disparity values directly, the disparity in one image is compared to the range of values contained in the 2×2 window surrounding its correspondent. In practise, we found that extending the measured range by 100% on either side allowed us to increase the floating point resolution of the comparison from 1.0 pixels in the pointwise method to 0.1 pixels; some results are shown in Figure 2.2. That is, two values are considered equal if they differ by at most this amount.

It should be possible to extend this notion to arbitrary viewpoints. The key point is to perform the range check using *normalized inverse depth*. When the optical axes are parallel, this simply means using the disparity. Given two arbitrary viewpoints, the same term $Bf/depth$ can be used, though it will no longer represent the magnitude of the disparity. Converting the depth values from one viewpoint to another will require the application of the complete camera models from both views (not just the linear translation/rotation matrix), but this should pose no problem with synthetic data since both camera models are known exactly and an analytic inverse of the perspective projection is easily coded. This use of arbitrary viewpoints is important for many extensions to the standard stereo model of parallel optical axes: e.g., verged cameras, multibaseline stereo, and arbitrary motion between frames.

In summary, the use of synthetic data enabled the elucidation of noise-causing effects in a popular stereo method of generating occlusion masks. We also showed how those effects could be mitigated by using a new method of fitting planes locally, and how this method can be extended to a completely general 3D case with arbitrary viewpoints.

This problem is also important to the Computer Graphics community, where occlusion masks are used as shadow maps. However, their goal is to find the amount of incident light over a large area, not to find individual point matches. Most shadow map methods simply smooth over pointwise matches to even out the shading effects (Williams, 1978; Reeves et al., 1987).

Limitations: Though useful for validating software integrity, noiseless synthetic imagery cannot be used to measure robustness to noise, an inevitable problem with real imagery.

2.4 Synthetic Imagery with Noise

A good way to characterize the robustness of a method is to take known data and introduce noise along independent dimensions of the imaging model (e.g., gaussian or focus distance blurring, white noise, lighting intensity, camera misalignment). Using synthetic imagery, the amount of error introduced along each dimension can be quantified precisely, and the degradation of an algorithm according to a particular type of noise model can be determined.

This type of data can be quite useful in experimentally characterizing an algorithm's performance, providing engineers with a quantitative measure of robustness. The ability to track performance loss as a function of noise is an important parameter in engineering design.

2.4.1 Example 3: Virtual Checkerboard

Figure 2.4 illustrates some types of noise that can be easily modeled using synthetic data. The top row contains perfect data, computed as described at the beginning of Section 2.3. As before, we have a perfect disparity map with the occlusion mask in black (in this figure it is computed for the rightmost image). The middle rows illustrate how typical noises affect image formation; here the left image is shown modified according to each particular type

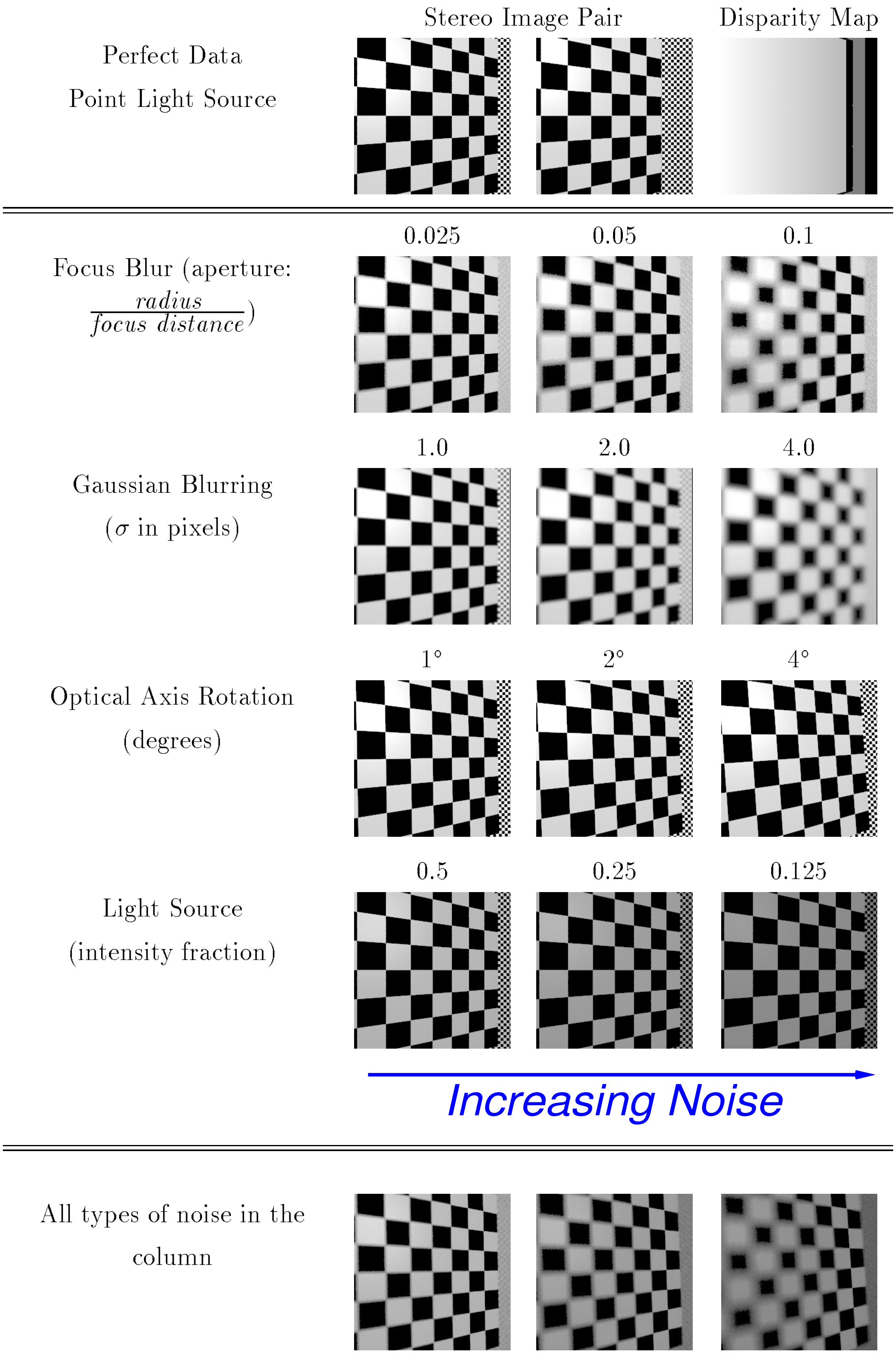


Figure 2.4: Examples of noise easily modeled with synthetic data.

of noise. When actually performing experiments both images will be modified, and the original disparity map used as ground truth. The effect of each type of noise is quantified by comparing the disparities computed from each noisy image pair against the known (constant) ground truth and measuring residual error. In this way the robustness of the algorithm to various types of noise can be estimated, and statistics derived by applying the same noise models to many synthetic datasets and tracking overall error. These noise effects can be combined arbitrarily; the bottom row of Figure 2.4 shows some examples.

Limitations: While these types of experiments are useful for calculating robustness along particular dimensions, they will nevertheless fail to address all the possible effects of real-world imaging. Multiple extended light sources, complex interreflections, nonlambertian objects, complex shapes, and errors introduced in the imaging process such as sensor pixel noise are extremely hard to model precisely, yet they affect every image. Synthetic data are useful for simple validation and characterization, but for algorithm performance verification there is no substitute for actual imagery.

2.5 Controlled Environment

The most useful stereo datasets are those with real imagery and 100% ground truth. Having used synthetic data to establish correctness and characterize robustness along particular model dimensions, one can move on to real images of controlled scenery. This introduces many unmodeled errors in the lighting, camera and optics, but allows them to be characterized by highlighting differences between the disparity map known from the ground truth and that computed by the algorithm.

Several types of noise are introduced here that are typically not modeled in synthetic imagery. The CCD array is subject to preamplifier noise, dark current, shading effects, and photon noise (Photometrics, 1990). Optical effects also become apparent: radial and tangential lens distortions, poor overall focus due to lens manufacturing errors and dust, and misalignment of the lens with the CCD array. Perhaps most importantly, effects such as the interaction of complex light sources with the objects being imaged and the sheer complexity of actual scenes introduce artifacts into real imagery that must be treated as noise by systems that fail to model them.

Some measurement errors can be compensated for in preprocessing. For example, one common problem with stereo imagery is a difference in gain between the two cameras. The problem is manifest as very different brightness, or distributions of pixel values between the two images. This effect can arise from many causes, e.g., differing apertures on the two lenses and specular highlights on the objects. Yet a simple histogram equalization can bring these distributions closer and make matching easier.

To acquire this type of data the shapes of the objects being imaged must be precisely known. This information ideally will be acquired using methods other than vision, since our objective is to evaluate the quality of a vision-based reconstruction. This can be accomplished by machining an object to precise specifications (as is often done with calibration targets), or by measuring the dimensions of existing objects with known shape (e.g., the sphere in (Jenkin & Jepson, 1994)).

Camera calibration is also a requirement for collecting accurate ground truth. Even if the shapes of objects in the scene are known, their absolute distance and orientation will be unknown. Camera calibration information should be computed even if the stereo algorithm being evaluated does not make use of it, to aid in the computation of dense ground truth. Euclidean camera calibration will enable quantitative analysis of the algorithm's precision.

2.5.1 Example 4: Calibration Targets

A common application of totally structured environments is the acquisition of camera calibration parameters, as will be described in Section 2.5.2. Most camera calibration techniques depend on the reliable extraction of 2D feature points from images, and many require precise 3D localization of features in the world as well. Since industrial applications often allow the scene environment to be manipulated during calibration, a common technique is to construct (and position) a target using these constraints:

1. The target must contain many feature points,
2. Those points will be spread throughout most of the camera's field of view when imaged,
3. The points can be easily located by simple image processing techniques, and
4. The 3D locations of the feature points are known to a high degree of precision.

Figure 2.5 shows a sample calibration image used in the Tsai-derived calibration procedure developed by Willson (Willson, 1994), and used in this work as well. The target is a grid

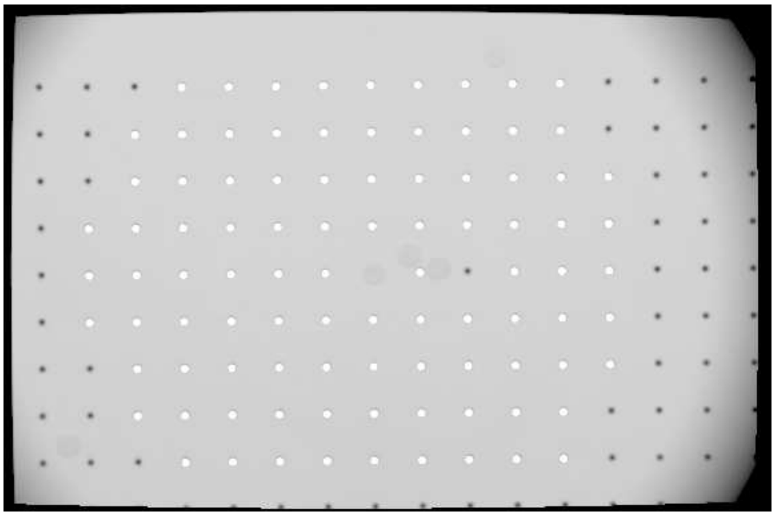


Figure 2.5: Actual image of the Calibrated Imaging Laboratory (CIL) calibration target with virtual rendering overlaid. The grey background and grid of black dots are part of the original picture, the white dots are rendered dots located at the 3D grid point locations. The dots were rendered as spheres using a virtual camera with the same parameters as those computed from the real image.

of black dots on a flat white background, each dot one inch from its horizontal and vertical neighbors. The center dot has been whited out to provide a reference spot for interimage registration. This flat target is mounted on an accurate translation stage, so by imaging it at several locations many 3D calibration points may be acquired. Because the 3D structure is known, the 2D extracted features are easily mapped into a 3D representation. To demonstrate how well the 3D representation fits with the original data, some of the recovered 3D feature points have been rendered as white spheres using the computed camera model (via a computer graphics ray tracer), and overlaid onto the original image in Figure 2.5. This projection back into 2D agrees nicely with the original data.

Figure 2.6 shows some more calibration targets. While CMU's Calibrated Imaging Laboratory (CIL) target from Figure 2.5 must be imaged at several locations to sweep out a 3D volume, these other targets have inherent 3D structure. Shum's cube (Shum et al., 1995) (leftmost in Figure 2.6) is useful because its regular structure means it can be imaged from any angle and still provide a number of features. The disadvantage is that this regularity implies there will always be some ambiguity when matching feature points from different cameras, especially if the camera separation is large and the optical axes all intersect within

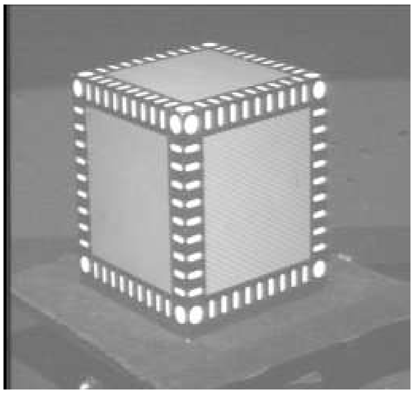


Figure 2.6: More sample calibration targets. The left image is the calibration cube from (Shum et al., 1995), right is an image of the MOVI (INRIA) "inverted cube" calibration pattern from (Boyer & Berger, 1995).

the target. The MOVI "inverted cube" target (Boyer & Berger, 1995) on the right suffers no such ambiguity, because its faces are joined at different angles (the vertical faces are joined at approximately 120 degrees). It is an "inverted" cube because it resembles an office corner: the angles between its visible faces are closer to 90 degrees than to the 270 degree angles found on a cube. A useful property of this target and the CIL target is that either may be left in place during later experiments and can thus provide useful ground truth for the background pixels.

Construction of calibration targets can be a difficult and expensive task. The MOVI target in Figure 2.6 is a set of precisely-machined metal plates, and the CIL target in Figure 2.5 is a PostScript file printed in high resolution on laminated and self-sticking paper, mounted on posterboard and attached to a metal frame on an automated translation stage.

Having seen examples of real objects built to precise specifications, we now outline the procedure required to put them to good use.

2.5.2 Camera Calibration

The task of determining in general how 2D image points correspond to 3D scene points is accomplished by assuming a camera model and performing camera calibration. First the correspondence between a representative set of 2D and 3D points is determined, then the parameters of the chosen camera model that best fit those data are found.

Typical camera models comprise external (or *extrinsic*) and internal (or *intrinsic*) parameters. Some common external parameters are X, Y and Z axis translations and rotations, and common internal parameters include image center, thick/thin/pinhole lens focal lengths, lens distortion coefficients, and aspect ratio. In astronomical image processing, lens models must also compensate for specific manufacturing defects by imposing a dense grid of coefficients over the lens surface, to model the point-spread function at each pixel or small group of pixels. Computer vision lens models tend to be much simpler, requiring only a few parameters. This is largely due to the fact that in computer vision, objects can be moved close enough to the camera to compensate for the small-scale lens defects that astronomers are forced to model.

Some calibration techniques require 3D information (Tsai, 1987; Willson, 1994; Bani-Hashemi, 1993), some use correspondences between images to compute the stereo epipolar geometry without complete Euclidean knowledge (Zhang et al., 1995; Faugeras & Toscani, 1986), and others control or restrict camera motion instead (Hartley, 1994; Stevenson & Fleck, 1995; Stein, 1995). We will summarize here the requirements for the first type of procedure (e.g., the one described in (Willson, 1994, Chapter 5)) for acquiring Euclidean geometry without restricting camera motion, and then extend it to include the acquisition of arbitrary ground truth in Section 2.6.

Define and Register Coordinate Frames

The first step is to define and register all coordinate frames. The eventual goal is the acquisition of imagery and co-registered ground truth of well-understood objects. Thus the mapping between images and the world must be modeled to perform the calibration and ground truth registration. This is purely a representational issue, and is used to determine the types (i.e., units) of measurements that will be used in both the calibration and data acquisition steps below.

To relate imagery to the real world, there are four reference frames of interest: 3D object coordinates, 3D world coordinates, 3D viewing coordinates, and 2D screen coordinates. These are illustrated in Figure 2.7 along with a range frame (explained in Section 2.6). The shape of every object in the scene is described in its own coordinate system, each of which is related to a single world coordinate frame. World coordinates are related to viewing coordinates by the external parameters of the camera model (translation and rotation), and viewing coordinates are related to screen coordinates by the remaining internal parameters.

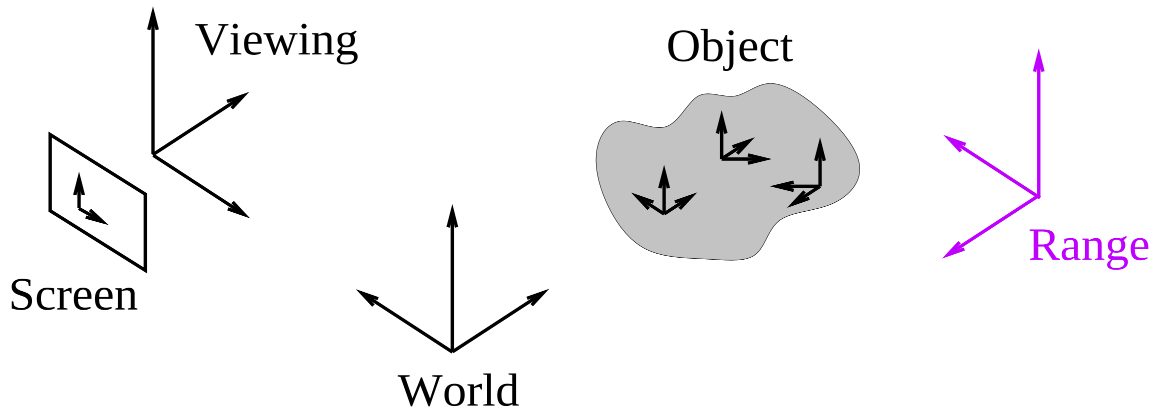


Figure 2.7: Coordinate frames used in dataset acquisition.

In many cases the dataset collection process can be streamlined somewhat by equating the world reference frame with the calibration target's object coordinate frame.

Specification of a coordinate system comprises the directions of the axes, the unit of length along each axis, and the location of the origin. Choosing 2D screen coordinate axes is relatively straightforward: the axes point horizontally and vertically along the CCD array, the units are typically pixels, and the origin is either a corner of the image, the center pixel of the CCD, or is specified by the internal calibration parameters. Specification of the remaining 3D frames will depend upon the application, but the units of the viewing coordinate Z axis will typically be the same as the depth estimates computed by the stereo algorithm, and each set of coordinate axes will usually be orthogonal.

The parameters that relate these coordinate frames are calculated during the camera calibration process.

Establish Calibration Target and Scene Objects

The 3D structure of the calibration target and scene objects must be determined to the best resolution possible, typically on the order of a millimeter or some fraction thereof. This can be accomplished by measuring existing objects or by manufacturing new ones. The calibration target should satisfy the constraints given in Section 2.5.1, but most importantly will ideally sweep out the entire volume of the area to be imaged: not just the volume of the target, but also that of any textured background areas. An easy way to do this is to use a flat or concave calibration target that occupies the entire image, and leave it in the background while imaging the scene objects.

Software that extracts features and builds an internal CAD model of everything in the scene must be developed concurrently with the physical construction of the calibration target and scene objects. These pieces are complementary; e.g., the specifications for the target should be driven by the capabilities of the software. Feature points on the calibration target should be constructed so that their image locations can be accurately measured to subpixel resolution. For example, the targets shown in Section 2.5.1 all use circular feature points for which the centroid can be determined to subpixel precision (Willson, 1994, Appendix D). The software must be able to robustly fit extracted features to the target's known 3D structure, allowing for those distortions that will inevitably occur due to the as-yet unmodeled camera parameters. Although this process is often performed manually (and should always be manually verified), by automating it a substantial bottleneck in dataset acquisition can be avoided.

The costs of establishing a good calibration target and creating recognition software will be easily recovered, in terms of both resources and time, in the form of more accurate measurements and faster dataset acquisition. However, the construction of scenes with complete ground truth is not always possible. For outdoor scenes in particular, the cost of constructing a large enough calibration target and acquiring completely dense ground truth is often prohibitive. The best way to approximate this is to construct models in a laboratory and adjust the lens parameters to simulate outdoor imagery, or compromise the results by using a calibration target smaller than the scene being imaged and acquiring only sparse ground truth such as will be described in Section 2.6.

Acquire Calibration Imagery and Calibrate

An integral part of the calibration process is the acquisition of several images of the calibration target. Before acquiring the calibration imagery, however, the layout must be checked to ensure that the scene images will accurately reflect the desired properties of the scene. The lighting levels must be set, the camera parameters adjusted (e.g., lens aperture, exposure time, focal length), the objects positioned, and some sample imagery taken (both with and without the scene objects). Performing this run-through prior to acquiring the calibration imagery will help ensure that the data collected will measure the properties actually found in the dataset scene imagery.

The calibration data should be acquired under the same conditions as the later scene imagery. Therefore the camera parameters should be established with both the scene objects and calibration target in mind. For example, when using a target with a white background the aperture must be adjusted so that the calibration imagery is not overexposed.

No matter how many images in the dataset, some calibration target feature points must be visible from all viewpoints. The target should be fixed in one place if possible, to facilitate the inter-image registration that must follow. If the target is moved to accommodate multiple viewpoints, then this transformation must also be known to high precision.

Although the 3D structure of the calibration target and any scene objects are assumed to be known, an independent measurement of the objects (or feature points) can provide a useful sanity check. Tools such as those described in Section 2.6, if available, could be put to good use in such redundant measurements.

Finally, images of the calibration target and the corresponding camera locations (if known) are recorded. These images, the inter-camera transformations, and known structure of the calibration target are all input to the camera calibration procedure.

Camera calibration itself occurs in two steps. First, calibration features are located in the images and mapped to their corresponding 3D world coordinates. This is best accomplished with calibration target-recognition software as described in the previous section. In the second step, these data points (image coordinate and world coordinate vector pairs) are fed into the calibration routine, which uses them to determine the parameters that best fit the camera model.

Once the camera calibration parameters have been computed to a reasonable level of precision, images of the actual scene objects may be taken.

2.5.3 Acquire Dataset Imagery

Finally, the actual stereo datasets can be acquired. The cameras are positioned in the same locations and orientations as were used for the calibration, and with the same settings (e.g., aperture, focal length, and gain). This can be achieved (with difficulty) by building a stable multi-camera head (e.g., as in (Ross, 1993)) or a single camera platform with highly reliable positioning capabilities (e.g., (Willson, 1994, Appendix A). In addition to imagery, completely dense ground truth will also be acquired in each camera's reference frame.

The best tools for performing the ground truth measurement will vary with the application. One possibility is to use image-based fitting just as was done with the calibration target. But this technique limits the types of objects that can be imaged to only those with software robust and precise enough to do the fitting. Another is to use measurement tools such as those described in Section 2.6 below to locate key features in 3D world coordinates, and apply the known object shape to fill in the details. In this case the ground truth measurements need only be acquired once since they are expressed in the world coordinate frame; the final result for each camera is computed by simply applying the external parameters for that camera to the world coordinate measurements.

Finally, these completely dense depth maps can be processed as described in Section 2.3.1 to yield occlusion masks for the resulting disparity maps.

Limitations: By controlling the environment, completely dense ground truth may be acquired along with real imagery. However, this approach is very restrictive. The hard work of constructing scenes so that precise ground truth is available over an entire image limits the approach to relatively simple scenes. If the eventual application will include complex scenes as well, imagery of more complex scenes with ground truth must be developed.

2.6 Measured Environment

Adding a range sensor to the laboratory allows images of complex static scenes to be acquired with ground truth. But by relaxing constraints on the objects to be imaged, the acquisition of complete ground truth becomes untenable. Some information is available, e.g., piecewise-planar patches can be measured, but most of the imagery will have unknown ground truth. Even so, this is an important scenario, because it comes the closest to the application of the method outside the laboratory, while still providing some measure of confidence in the results since at least part of the disparity map can be computed precisely. It is generally no longer possible to compute the complete occlusion mask however, because often the missing disparities are exactly those at occlusion boundaries. But at least some of the disparities computed by the stereo method can be verified.

Object shape knowledge and range sensors are used to acquire the ground truth. The locations of feature points in the scene can be measured using pointwise devices such as surveyor's theodolites. Theodolites measure angles rather than distance, but with two theodolites and a simple calibration step, pairs of measured angles can be directly converted to 3D coordinates. By choosing feature points effectively, dense depth maps can be computed using knowledge of the shape of the objects in the scene. For example, when imaging polyhedral objects, corner point locations can be interpolated to yield dense depth maps over the objects' surfaces.

Why not just use an imaging rangefinder? Rangefinders are indeed useful tools, and could be used to provide some ground truth information, but there are limits (Besl, 1988). Even a perfectly accurate rangefinder would not provide an exact depth map unless it were co-located with the imaging elements. The scientific use of rangefinders is still being studied, and even some popular LIDAR laser rangefinders are subject to outright errors in their measurements, particularly at occlusion boundaries (Hebert & Krotkov, 1992), which makes them unreliable sources of information for evaluating the effect of occlusion on stereo data. Image-based rangefinders that use controlled lighting together with the same CCD array could be very useful in some laboratory situations (see (Tada et al., 1993; Nayar et al., 1995) for two image-based range sensors), but would likely depend on the same camera calibration methods used in the stereo method, resulting in "ground truth" measurements that have some of the same biases as the stereo method.

2.6.1 Camera and Range Sensor Calibration

Data acquisition at this level requires calibration of both the range sensor and the camera. Camera calibration was described in Section 2.5, but the additional requirements and some example datasets are described below.

Calibrate Range Sensor

Ground truth measurements are acquired by introducing a range sensor into the laboratory. Placement and use of the range sensor is complicated by the fact that its use must not interfere with the acquisition of stereo data. Completely dense ground truth measurement will be impossible, for unless the range sensor uses the same CCD elements as the stereo cameras, it cannot be co-located with them, and will therefore be unable to view all the same points.

One choice for range sensor is a pair of surveyor's theodolites. A theodolite is an optical measurement tool consisting of a lens system, stable platform with levels, and instrumentation that measures the angle between an initial direction and that of a point in the world. Angles are converted to distance measurements by combining angles from two theodolites with the known baseline between the instruments and then triangulating. This system has the desirable properties that it provides range measurements independent of the stereo camera equipment, and makes no restriction on the type of objects that can be measured (except that some features must be visible to each theodolite). The resolution attainable by this system is discussed in Appendix A.

Theodolite calibration occurs in several steps. First the vertical axes are aligned by checking the levels. The horizontal coordinate frame is initialized by aiming each instrument at the other to set a preliminary zero angle, then rotating each by 90 degrees and resetting to zero again. Once both instruments' vertical and horizontal axes have been made parallel, the baseline separating them can be calculated by placing a ruler somewhere in the shared field of view, measuring the angles to two points on the ruler and plugging those angles and the interpoint distance into a known formula.

Determining the best separation for the theodolites is difficult. The best resolution in depth measurements will be obtained with a wide baseline separation, but when the theodolites are too far apart there will be many points near occlusion boundaries that are not visible to both theodolites. This will limit the number of points for which ground truth can be measured, thus reducing the density of the final depth map. Therefore it is important to coordinate the placement of objects, including the camera calibration target, with the positioning of the theodolites.

Define and Register Coordinate Frames

As in Section 2.5.2, the four types of coordinate frames (screen, viewing, world, and object) must be defined and registered. In addition, the coordinate frame of the range sensor must also be determined (see Figure 2.7). This is typically done by locating features on the calibration object using the range sensor, and computing the transformation between the range frame and the calibration object frame.

2.6.2 Acquire Imagery and Range Data Concurrently

Finally the actual datasets can be acquired. Because the recommended range sensor requires a long time to gather measurements, only static scenes may be imaged.

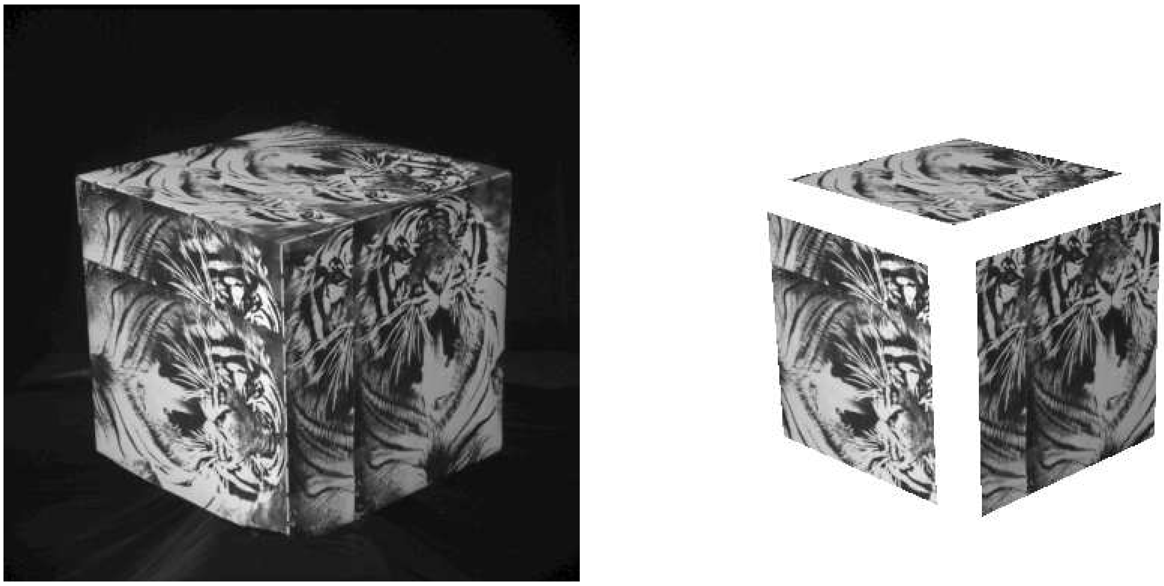


Figure 2.8: Textured cube image and the piecewise-planar patches used by Xiong for error analysis (Xiong, 1995, Figure 3.20). Used with permission.

2.6.3 Example 5: Textured Cube

An example of a dataset from the literature that benefits greatly from even sparse ground truth is the textured cube from (Xiong, 1995). This dataset consists of one stereo image pair, from which an image is reproduced in Figure 2.8. Since each face is known to be planar, only a few points need to be measured to acquire reasonably dense ground truth.

Xiong does in fact use the known planarity to characterize the shape of this object, but instead of taking independent distance measurements he fits planes to the computed disparities on each face. Thus he is able to quantify his algorithm's ability to recover shape information from stereo disparity, using the residuals from the planar fit as the error measure.

2.6.4 Example 6: Model Train Set

Figure 2.9 illustrates a sample image of a more complex scene with sparse ground truth, taken from the publicly accessible **CIL-0001** dataset.[1] The scene includes many complex surfaces, few of which are absolutely planar, yet some reasonable approximations can be made. For instance, planes can be fit to those roofs where corner locations are available, and to the front faces of the houses and castle towers. The background pixels can also be filled

[1] `http://www.cs.cmu.edu/~cil/cil-ster.html`

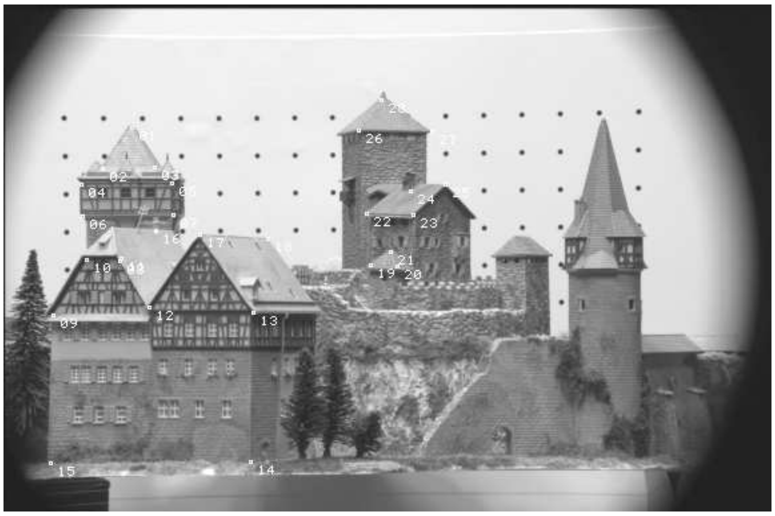


Figure 2.9: Image from CIL-0001 dataset with the locations of ground truth measurements.

in completely, since the shape of the background grid is known from the camera calibration.

The precision of these ground truth measurements is derived in Appendix A.

2.7 Unconstrained Imagery

The final and heretofore most common type of stereo data is that for which no ground truth is made available. The difficulty with such data is that the disparity maps computed by stereo algorithms cannot be assessed metrically, only by human inspection or by ground truth expressed in pixel units.

A common application for this is autonomous navigation. In a typical scenario the acquisition of ground truth is impractical, due to the sheer volume of data being processed. But at least one group has attempted to address this, by simulating road conditions using a static outdoor scene with explicit fiducial marks.[2]

2.8 Implementation

Our work has benefited greatly from the use of stereo datasets with ground truth. Data from each level of this taxonomy has been used in the research presented throughout this

[2] The Linköping University Division of Computer Vision provides a small calibrated outdoor dataset at `ftp://isy.liu.se/images/calib.ic/`

thesis. Both synthetic and real imagery were used to develop algorithms, debug their implementations, and characterize their performance. Several examples are given throughout this chapter.

Rayshade, a freeware ray tracing program, provided the foundation for the synthetic datasets used throughout this thesis. Several enhancements were made to adapt this tool to the task of providing stereo datasets with ground truth: automatic depth map extraction, extension of the camera model to include configurable image center and radial lens distortion, addition of back-end tools for depth map manipulation, disparity and occlusion processing, and image format interchange. These extensions and their documentation have been made freely available to the research community.[3]

Software developed for the Calibrated Imaging Laboratory (CIL) has been successfully used in the collection of stereo datasets with ground truth by several researchers. Camera calibration tools originally developed by Willson (Willson, 1994) have been improved and made more robust, with the result that the time required for dataset acquisition has been reduced from days to minutes. These tools, though somewhat specific to the CIL, are readily available to laboratory members and visitors.

The datasets collected in this laboratory are among the first of their kind to be made available to the general research community: high-quality images with piecewise-dense ground truth.[0] Publications by other researchers using these datasets have already appeared in peer-reviewed publications, e.g., (Wang & Jepson, 1994).

2.9 Summary

The analysis of stereo vision algorithms can be greatly enhanced through the use of datasets with ground truth. We have outlined a taxonomy of datasets with ground truth that use varying degrees of realism to characterize particular aspects of stereo vision systems, and shown that each component of this taxonomy can be effectively realized with current technology. We proposed that datasets generated in this way be used as the foundation for a suite of statistical analyses to effectively characterize the performance of stereo vision systems.

---

[3] See the Computer Vision Source Code web page at `http://www.cs.cmu.edu/~cil/v-source.html`.

Chapter 3

Phase-based Stereo

In theory, a tomographic imaging scanner capable of multiphasic resolution would be able to penetrate this much interference.

— Lt. Commander Data, helping to save the day in the final Star Trek:TNG episode *All Good Things*

Section 1.4 described several types of stereo vision algorithms. In this chapter we restrict our attention to a particular class of filter-based methods, the so-called *phase-based stereo methods.* We will give a short overview of local spatial frequency, develop our phase-based method, contrasting it with existing ones, and point out issues that must be considered when using any type of phase-based method. The method developed here will be analyzed and extended in later chapters to address issues of foreshortening and ambiguous matches.

3.1 Introduction

There has been a resurgence in interest in filter-based stereo methods. Unlike methods that work by tracking particular features in an image (thus yielding sparse results), filter-based methods allow disparity to be densely computed over entire images. The exponential growth of computer processing power continues to enable new, more powerful and compute-intensive techniques such as multi-camera stereo. Those methods derive their power from the ability to process more data quickly and effectively. Another approach whose appeal is growing is the class of phase-based stereo methods.

In this chapter we will present a phase-based stereo method that requires only a pair of images, but takes advantage of extra processing power to transform its data into representations most appropriate for the task of stereo matching. This presentation of our method takes advantage of the epipolar constraint, and the discussion is therefore restricted to operations on 1D image scanlines. The method can be summarized in one equation, for the disparity at a given pixel:

$$\text{For all } \lambda, \quad \textit{Disparity} \;=\; \frac{\Delta\phi(\lambda)}{2\pi} \cdot \lambda_{\text{ref}} \tag{3.1}$$

The parameters of Equation 3.1 will be discussed as follows. The overall framework for the mechanics of the computation is a model of *local spatial frequency* using Gabor filters. *Disparity* will be incorporated as a search parameter in the spatial domain. The *phase* ϕ will be computed by the Gabor filters, and the *phase difference* $\Delta\phi$ will be the principal measured value extracted from the filtered input. The selection of filter wavelengths *for all* λ will be discussed, and the process of wavelength (nee frequency) refinement λ_{ref} will be described.

This gives us the structure for this chapter. We will first review the concept of local spatial frequency, and discuss why Gabor filters seem to be the most appropriate representation for the problem of stereo matching. Next we give some intuition behind the use of the phase of Gabor filter outputs as a measurement of disparity, and the use of instantaneous frequency as a means of refining the value of the signal's wavelength. Having thus outlined the use of a single filter, we discuss the use of multiple filters at different frequencies and the issues that arise in combining their estimates. We then combine all of these into a description of our initial phase-based stereo method, giving several example image pairs and disparity maps. This method will be refined still further in Chapter 5.

3.2 Local Spatial Frequency

There are two fundamental extremes in the study of 2D imagery: the detail-oriented *spatial* view and the wholistic *frequency* view. In the spatial view each individual pixel is paramount; an image is represented by the concatenation of independent pixel values, and each pixel is considered unique and important. In the frequency view only the entire image matters;

although the image is broken down mathematically into several frequency components, information in each component relates to the image as a whole. Each view has its benefits: the spatial view can represent discontinuous textures and local (pixel-level) segmentations directly, while the frequency view is a mathematically elegant representation that enables many useful analyses over large regions. The trouble is, the benefits of one view are difficult to attain in the other. Fortunately, a compromise can be reached, one which preserves the localizability of the spatial approach and the analytical benefits of the frequency approach: this combined approach is called *local spatial frequency.*

In the hope of appealing to those already familiar with Fourier transforms, we will demonstrate this concept using a *spectrogram,* also known as the Short Time Fourier Transform (STFT). If you are already familiar with spectrograms, or do not care to become more familiar, feel free to skip this paragraph. The 2D spectrogram is made up of a sequence of Fourier transforms applied to small, overlapping segments of the signal: each column is the 1D Fourier transform of a window on the signal centered at that same column (i.e., pixel number). The size of this window is fixed (e.g., at 100 pixels in Figure 3.1), but as the window grows adjacent columns will become more and more similar, since most of their windowed input will be the same. It is a highly redundant representation of a signal, an example of which can be found in Figure 3.1. At the limit, if the window size equals the length of the signal, the spectrogram will be only one column wide, and its contents will be the same as the 1D Fourier transform. Now back to the topic at hand.

The general concept of local spatial frequency is illustrated in Figure 3.1. For the moment we consider magnitude only; phase will be introduced later. We begin with a 1D Original Signal, which in this example is a sinusoid with wavelength $\lambda = 20$ pixels (frequency $\omega = 1/20 = 0.05$) embedded within a sinusoid having $\lambda = 7$ pixels ($\omega = 1/7 = 0.14$). This signal is then sampled into 128 pixels from each sinusoid fragment, for a total of 384 data points (we will also call them *image samples*). The frequencies of the original signal can be calculated directly from these samples using the 1D Discrete Fourier Transform (or DFT, see figure), but cannot be localized; we see there are two strong frequencies in the DFT, but we cannot determine where they occur in the original signal. What we *want* to see is a 2D *Ideal* Local Spatial Frequency plot, also shown in the figure. In such a plot, not only would the particular frequencies be known, but also the areas of the original signal in which they occur. In fact such a plot can be approximated, as is demonstrated by the spectrogram in

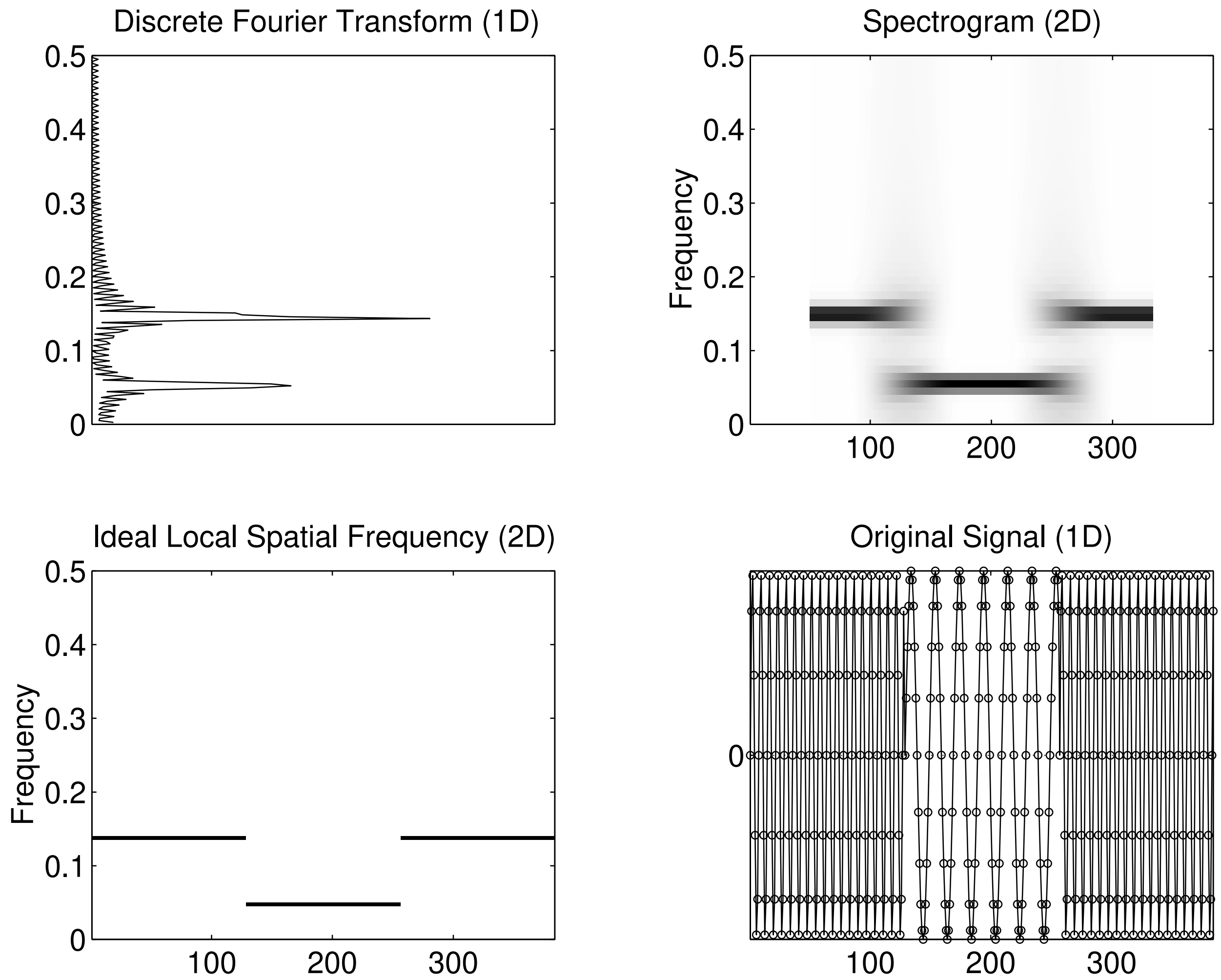


Figure 3.1: Local Spatial Frequency illustration (magnitude only). The Original Signal (lower right) is a low frequency sine wave embedded in a high frequency wave. Its Fourier Transform (upper left) has two peaks, one for each frequency. The Ideal Local Spatial Frequency plot (lower left) associates each sample point with its proper frequency. The Spectrogram (upper right) only approximates this ideal plot, but is computed directly from the sample points *without* explicit knowledge of the original signal's analytical form. Peaks in the 1D DFT plot correspond to the dark areas in the 2D images.

Figure 3.1, which was computed directly from the image samples *without* prior knowledge of the original continuous signal.

But this representation has some practical and fundamental limitations. On the practical side, the Nyquist interval associated with the Sampling Theorem tells us the analysis window must be at least twice as wide as the longest wavelength being analyzed; but a wide window will tend to blur the measurements around discontinuities in the original signal (witness the horizontal blurring around the dark lines in the spectrogram of Figure 3.1). More fundamentally, there is a limit to the amount of frequency and localization information that can be extracted. The greater the number of independent frequencies being studied (i.e., the finer the frequency resolution), the coarser the spatial resolution that can be attained, and vice versa. The first mathematical statement of this phenomenon was made by Gabor in (Gabor, 1946), by analogy to the Heisenberg Uncertainty Principle. We can state this "Law of Conservation of Misery" informally as:

$$\Delta x \cdot \Delta\omega \geq \frac{1}{2} \tag{3.2}$$

where Δx and $\Delta\omega$ are the spatial and frequency resolution, respectively. Without going into details (these can be found in (Gabor, 1946, eqs 1.21 – 1.26)), Equation 3.2 says that there is a limit to the amount of local spatial frequency information that can be automatically extracted from sampled data: if you want more frequency information, you have to give up spatial resolution.

3.2.1 Gabor Filters

Having established these fundamental bounds, Gabor went on to describe a family of analysis filters that achieves this optimal resolution. These are now referred to as *Gabor filters*, and they consist of a complex exponential multiplied by a Gaussian window:

$$\mathrm{Gabor}(x,\omega) = u \cdot e^{-i2\pi\omega x} \cdot e^{-0.5\left(\frac{x\omega}{m\sigma_f}\right)^2} \quad \text{for } x \in \left[-\frac{m}{2\omega}, \frac{m}{2\omega}\right] \tag{3.3}$$

$$\text{with Magnitude } \rho(x,\omega) = u \cdot e^{-0.5\left(\frac{x\omega}{m\sigma_f}\right)^2} \tag{3.4}$$

$$\text{and Phase } \phi(x,\omega) = 2\pi\omega x \tag{3.5}$$

where ω is the *tuning frequency* of the filter, $u = \omega/(\sqrt{2\pi} m\sigma_f)$ is a scaling term and m and σ_f are constant tuning parameters; these are discussed below. Gabor filters are closely related to the Fourier transform. In fact, the complex exponential component of the filter is actually *identical* to the kernel of the Fourier transform. Where the Gabor filter differs is in its view of the input data. Whereas the Fourier transform uses the entire image to compute its results, the Gaussian window in the Gabor filter limits attention to a small

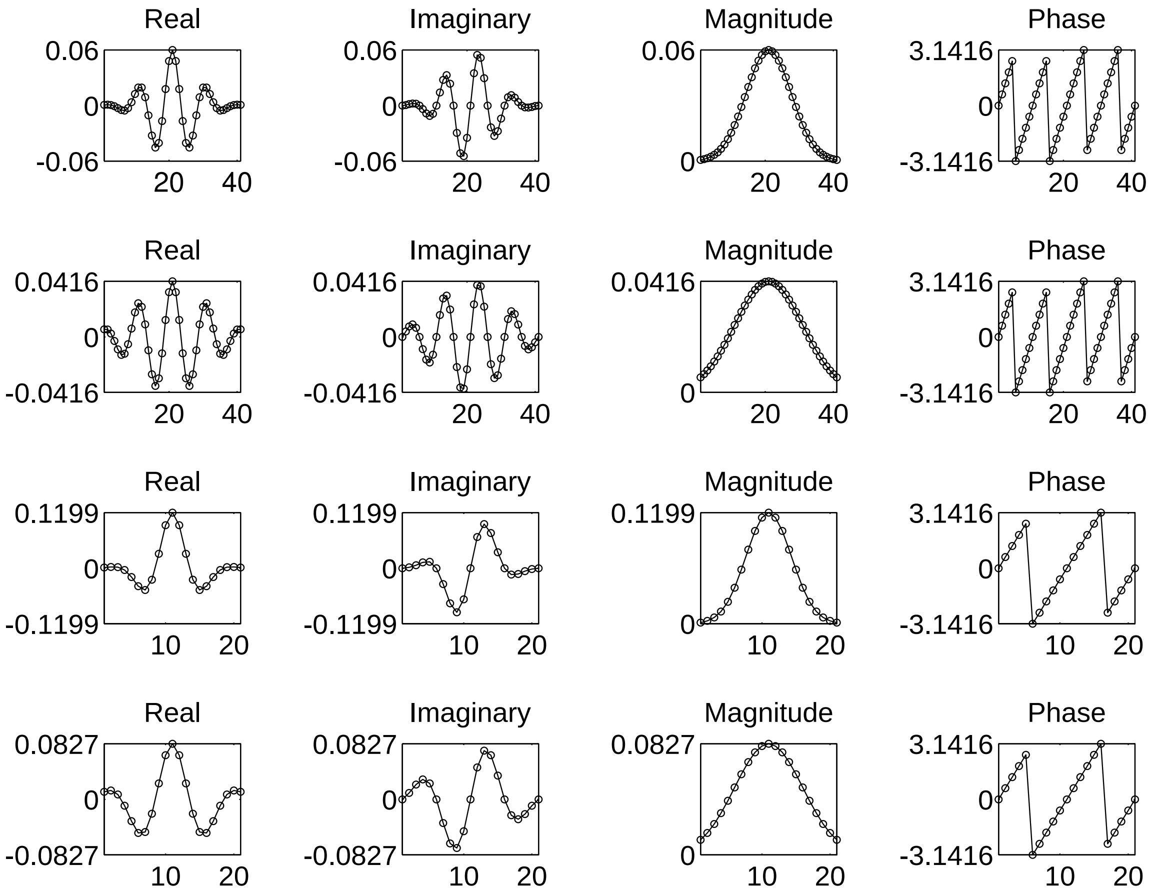


Figure 3.2: Gabor filter examples with $\omega = 1/10$ but different m, σ_f on each row. The horizontal axis indicates the number of samples in the filter.

region in the input image. Also, unlike the Fourier transform which uses a *fixed* window size at each frequency, the Gabor filter uses a window that shrinks and grows as the tuning frequency changes (witness the ω term in the Gaussian component of Equation 3.3). At high frequencies the effective window width will be quite small, thus ensuring that only the nearest pixels will be used to compute filter outputs. At low frequencies the window will be much wider, and will therefore require more data from the original signal.

In addition to the tuning frequency, two additional parameters must be specified to completely define a Gabor filter. The first is the number of wavelengths (m) to include in the window. The Sampling Theorem tells us that for an input signal whose lowest frequency component is ω_L (with corresponding wavelength $\lambda_L = \frac{1}{\omega_L}$), we must include $2\lambda_L$ samples of the signal to reliably extract information about that frequency. More data would be even better, but this number of wavelengths is the minimum. Thus we can choose any value as long as $m \geq 2$.

The second parameter is how much of the Gaussian to include in the window (σ_f). Parameter σ_f is the fraction of the window size that corresponds to one standard deviation (σ) of the Gaussian, typically 1/6. Although theoretically the Gaussian has infinite extent, 99.7% of its area is located within three standard deviations from the origin. Thus it is reasonable in practise to limit the extent of a Gaussian window to six standard deviations (three on either side). Since even extreme values of the Gaussian never really equal zero, this causes the tails of the Gaussian to be truncated arbitrarily, but the numerical insignificance of these extreme values makes this a reasonable compromise for digital implementation.

Figure 3.2 illustrates four Gabor filters. Each horizontal row of the figure corresponds to a single filter. Since Gabor filters are complex-valued, two plots are required to completely illustrate their components. The figure presents each Gabor filter in two ways: as a real-imaginary pair, and as a magnitude-phase pair. All of the filters in the figure have constant tuning frequency $\omega = 1/10$, but the extra parameters (m,σ_f) vary from top to bottom: $(4, 1/6)$ on the top, $(4, 1/4)$, $(2, 1/6)$, and $(2, 1/4)$ on the bottom. The plots have been scaled in both directions to make the filters appear as similar as possible, so that the variations in shape and number of samples that result from changes in these parameters can be emphasized.

Parameter Tuning

Tuning of these two additional 'magic' parameters is still somewhat of an art. The resolution of the frequency information is greatest when both m and $1/\sigma_f$ are large, allowing lots of data and many wavelengths to be used in the computation. This is fine when the input image consists of entirely one type of signal. Unfortunately, images often contain many objects, with only a small number of pixels representing each one. And in that case we would prefer to use as *little* data as possible (just the relevant pixels) to perform computations; otherwise our results will be corrupted with what is effectively noise from images of the other objects. This adds a new wrinkle to Gabor's uncertainty principle: not only is the resolution limited, but to accommodate the analysis of multiple objects we should restrict the number of spatial data points even further.

The effects of tweaking these parameters are illustrated in Table 3.1. Ignoring for the moment the details of the stereo algorithm, we can still see that simply changing these filter

$\sigma_f \setminus m$	4	6	8	10	12	14	16
0.2	0.372339	0.204195	0.185642	0.192559	0.187348	0.196636	0.191339
0.25	0.342084	0.192126	0.180964	0.187143	0.166857	0.182314	0.178613
0.3	0.394752	0.232775	0.211052	0.214945	$\boxed{0.166194}$	0.18913	0.183344

$\sigma_f \setminus m$	18	20	22	24
0.2	0.910356	0.668425	1.18127	1.64487
0.25	0.20824	0.230053	1.10322	1.68861
0.3	0.211346	0.223421	1.09542	$\boxed{1.78614}$

Table 3.1: Effects of Tuning Parameters on stereo disparity accuracy. Filters with given values for σ_f and m were applied to the scanline stereo pair from Figure 3.3 using the method in Section 3.7. Here the table shows the RMS error between the computed disparity and that known over the portion of the image with visible texture (153 pixels). 401 disparities from 0 to 10 were tested, and the smallest and largest errors are highlighted.

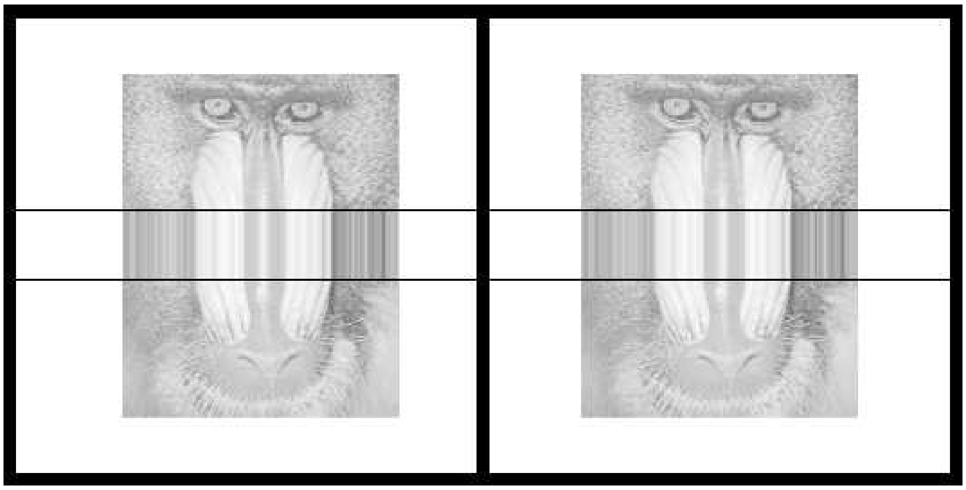


Figure 3.3: Scanline Pair used in Table 3.1, highighted in the image from which it was extracted.

parameters has a significant effect on the apparent precision of a method using them. In this case, the overall precision varies by an order of magnitude. But predicting such deviations is quite difficult; for example, the results in the table are not strictly unimodal in either dimension, so no general trends can be inferred.

A final note about the magnitude of the Gabor filter envelope as used in this thesis.

Although Gabor filters can be used as wavelets, where shape is preserved across scales, our filters will instead preserve constant area across scales. The is accomplished by using a Gaussian envelope with unit area (scaling with u in Equation 3.3) to normalize the magnitude of the filter outputs across all frequencies. Low frequency filter outputs would otherwise disproportionately dominate the those at high frequencies because the magnitude would grow in proportion to the window size.

3.3 Using Phase as Disparity

The notion that phase corresponds to translation is a familiar one to the signal processing community. The best example of this is the Fourier Shift Theorem, which states that a translation a in space corresponds to a phase shift $2\pi a\omega$ in the frequency domain:

$$\begin{aligned} \text{If} \quad f(x) &\Longleftrightarrow_{\mathcal{F}} \quad F(\omega), \quad \text{Then} \\ f(x-a) &\Longleftrightarrow_{\mathcal{F}} \quad e^{-j2\pi a\omega}F(\omega) \end{aligned} \tag{3.6}$$

The problem with this theorem is that it is strictly only true for functions with infinite support. Since we want to work with finite images, we have several options: ignore the problem; extend images to infinity using zero-padding, repetition or reflection; or modify the theory to account for the finiteness of function f. Work that extends the theory to handle the finiteness of images (aka the "window effect") can be found in (Xiong, 1995). Although that work doesn't completely eliminate the window effect, it reduces it to numerical insignificance through the use of polynomial approximation. But the intuition from this theorem is still good for finite images, as we will see, so our work will call upon the first two options instead.

An illustration of this concept can be found in Section 2.2.1.

3.3.1 Relating Phase to Feature-based Methods

Convolution with a Gabor filter is a procedure similar to finding the local shape of a smoothed version of a signal. More precisely, it is like fitting a sinusoid with varying amplitude and phase to a smoothed input signal. Unlike methods that extract binary features such as edges from their input, by using phase one gets the advantage of a detector of *continuously* varying features. You don't just get edges, you get "edginess" too. Think of phase as saying what the local signal shape is, and amplitude as the confidence of that interpretation. This view, that phase values are features to be matched, provides the intuition behind our stereo method which will be presented in Section 3.7.

But you get more than just features from phase measurements. Because the shape of the analysis filter is known (it has a sinusoid as its kernel), you can use the numeric value of the phase measured at one point to predict the phase at another. This is the basis of the so-called *direct phase* methods, which work by assuming that phase will change linearly, so by measuring the local difference you can jump from one pixel directly to its correspondent.

There are problems in interpreting Gabor filter outputs directly, however. The next section discusses the reliability of measurements made using Gabor filters.

3.4 Phase Difference Measurement Characterization

Instead of matching intensities directly, phase methods convolve intensities with continuous filters, then use properties of the filter outputs to determine disparity. This idea was illustrated in Section 2.2.1, and can be derived algebraically using the Fourier Shift Theorem (stated above in Equation 3.6). We will now derive this basic result, already presented in Equation 3.1, but to simplify the derivation we will only consider the effect of translation on a 1D image scanline.

Let image scanline $l(x)$ be given; this will be the left image. Now let an arbitrary disparity d be given; this is a global disparity that will apply to every pixel in the image. The right image is the same as the left, shifted by the amount d:

$$\begin{aligned} l(x) \quad & \textit{is the left image} \\ r(x) = l(x-d) \quad & \textit{is the right image} \end{aligned}$$

We can compute the Fourier transform of each scanline independently:

$$\begin{aligned} l(x) \quad & \Longleftrightarrow_{\mathcal{F}} \quad L(\omega) \\ r(x) \quad & \Longleftrightarrow_{\mathcal{F}} \quad R(\omega) \end{aligned}$$

The Fourier Shift Theorem tells us how $L(\omega)$ and $R(\omega)$ are related:

$$R(\omega) \quad = \quad e^{-j2\pi d\omega}\, L(\omega) \tag{3.7}$$

and note that Equation 3.7 holds for any value of ω. So all of the complex Fourier coefficients from the right image are exactly the same as those from the left image, except that their phases are shifted by some amount. Since the magnitude of the coefficients does not change, let us restrict our attention to just the phase components of Equation 3.7. We will use the notation that $\phi_L = \arg[L(\omega)]$.

$$\begin{aligned}
\arg[R(\omega)] &= \arg[e^{-j2\pi d\omega}\,L(\omega)] \\
\phi_R &= -2\pi d\omega + \phi_L \\
d &= \frac{\phi_L - \phi_R}{2\pi\omega}
\end{aligned}$$

Solving for the disparity d results in the formula that was presented back in Equation 3.1, the relation that lies at the heart of all phase-difference or direct phase methods. For convenience we restate it here:

$$\mathit{disparity} = \frac{\Delta\phi}{2\pi\,\mathit{frequency}} = \frac{\Delta\phi}{2\pi}\cdot\lambda_{\mathrm{ref}} \tag{3.8}$$

This equation can be readily understood by applying it to an example of a continuous sine wave. It simply says that disparity (or shift in x needed to align the signals) is equal to the translation of the sine wave (i.e., phase difference $\Delta\phi$, measured in radians) scaled by a refined estimate of its wavelength, or period λ_{ref}. The same intuition works for *windowed* continuous sine waves as well (e.g., Gabor filters which are sine waves multiplied by Gaussians). Problems arise when applying this to discrete signals, 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 3.8 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 components consider what happens when the signal being studied is a simple sinusoid.

3.4.1 Background on Instantaneous Frequency

The concept of *instantaneous frequency* is useful in many signal processing applications, and to phase-based stereo in particular. The term was introduced in (Carson & Fry, 1937) as part of a description of the mathematics behind frequency modulation transmission (i.e., FM radio). In that application, a low frequency signal (the audio signal) is transmitted on a very high frequency carrier wave (at the radio station's tuning frequency) by adjusting the frequency of the carrier wave over a wide range. Those adjustments, the changes in the carrier frequency, encode the lower frequency signal of interest, and *instantaneous frequency* is the framework that extracts the original signal from those changes.

A more physical way to think of it is like the measurement of velocity on a car's speedometer. Assuming a 1D signal comprises a possibly changing sinusoid at each moment in time (like a car being driven at different speeds), the frequency of that sinusoid (i.e., the car's velocity) at each moment is the *instantaneous frequency* of the signal. There are several ways to compute it, each of which is applied to a small window of the signal surrounding the point of interest.

One bad way to estimate the instantaneous frequency is to convolve the window with a bank of band pass filters tuned to different frequencies, find the filter with maximum magnitude response, and treat the tuning frequency of that filter as an estimate of the signal's instantaneous frequency. The problem is that if only a small number of filters is used it will not be very accurate. Band pass filters respond to any frequency within their pass band, and since a common width is an octave (Fleet & Jepson, 1993) the resulting estimate may be off by a factor of $\sqrt{2}$. The estimate can be improved somewhat by fitting a curve to the responses of adjacent filters and finding the frequency that corresponds to the peak, but the number of filters required for this approach to be effective makes it less attractive than the following option.

If the input signal is known to have come from a single sinusoid, its instantaneous frequency can be computed directly from the *phase derivative* of a *single* band pass filter. The intuition for this notion does not come easily: frequency and phase are often presented as *independent* components of a sinusoid (for example in Section 2.2.1), so how can they be related?

The relationship is illustrated in Figures 3.4 and 3.5, which demonstrate the same analysis on two different sinusoids. Plot a) shows a sinusoid (with fixed frequency in Figure 3.4 and varying frequency in Figure 3.5) and the magnitude envelope of the Gabor filter that is used to analyze it in Plots d) and e). Plot b) shows the phase of the signal, both as absolute phase (zero at the origin and monotonically increasing) and phase modulo 2π, as it will be estimated below. These plots were computed analytically; the phase is known to change linearly over time for fixed frequencies, and it wraps around at 2π exactly where one period of the sinusoid repeats. Plot c) shows the derivative of this phase, i.e., the slope of the plot above. As you can see, the value of this phase derivative is everywhere equal to the frequency

$$\text{Wavelength} = 15 \qquad \text{Frequency} = \frac{1}{\text{Wavelength}} = 0.06667$$

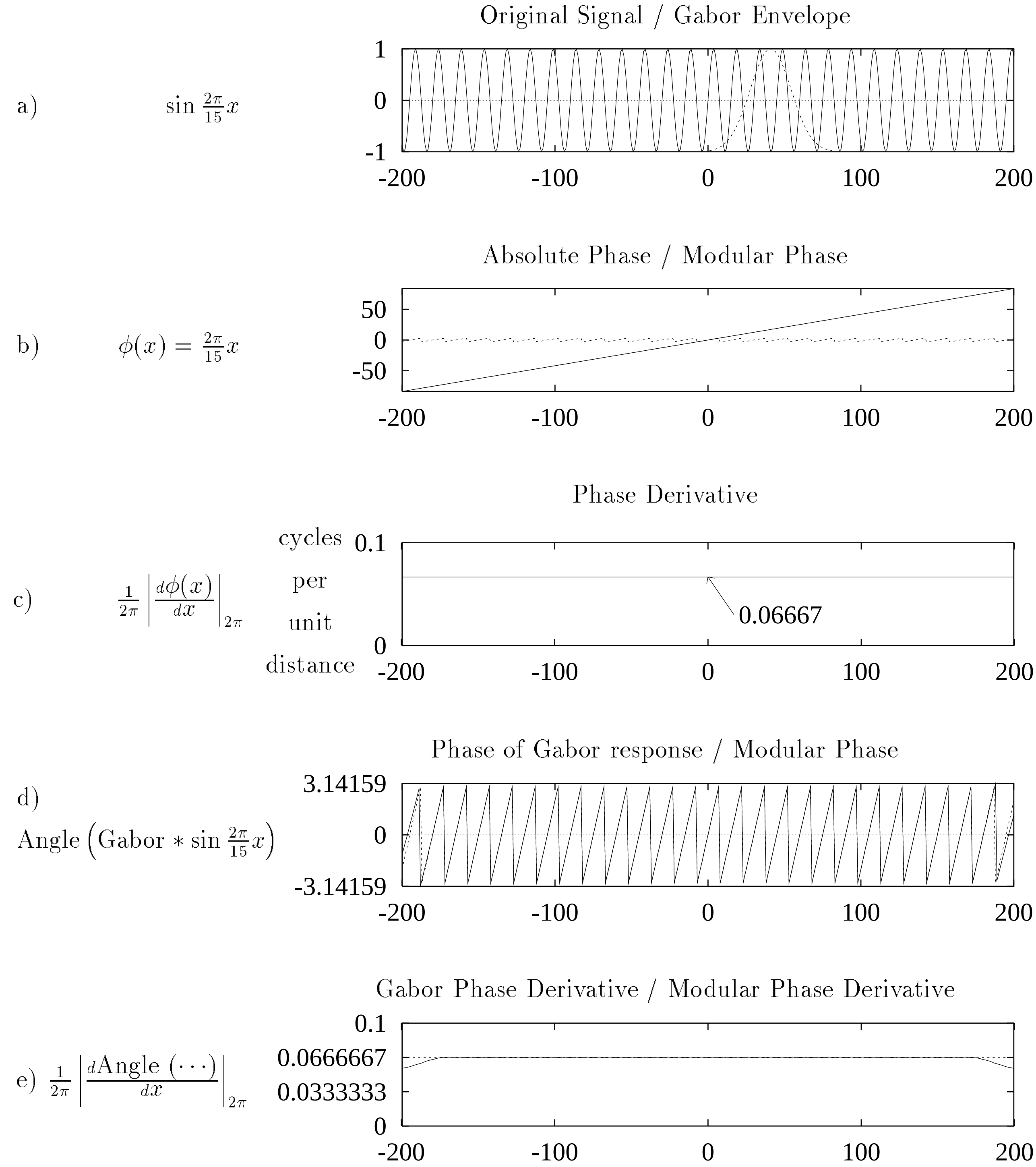


Figure 3.4: Instantaneous Frequency Example: $\omega = 0.06667$. The figure is explained in Section 3.4.1

Wavelength goes from 30 to 15 $\quad$ Frequency $= \frac{1}{\text{Wavelength}} = 0.03333$ to 0.06667 in linear steps

a) $\sin 2\pi x \left(\frac{1}{20} + \frac{x}{24000}\right)$

b) $\phi(x) = 2\pi x \left(\frac{1}{20} + \frac{x}{24000}\right)$

c) $\frac{1}{2\pi}\left|\frac{d\phi(x)}{dx}\right|_{2\pi}$

d) Angle (Gabor $*$ $\sin \cdots$)

e) $\frac{1}{2\pi}\left|\frac{d\text{Angle }(\cdots)}{dx}\right|_{2\pi}$

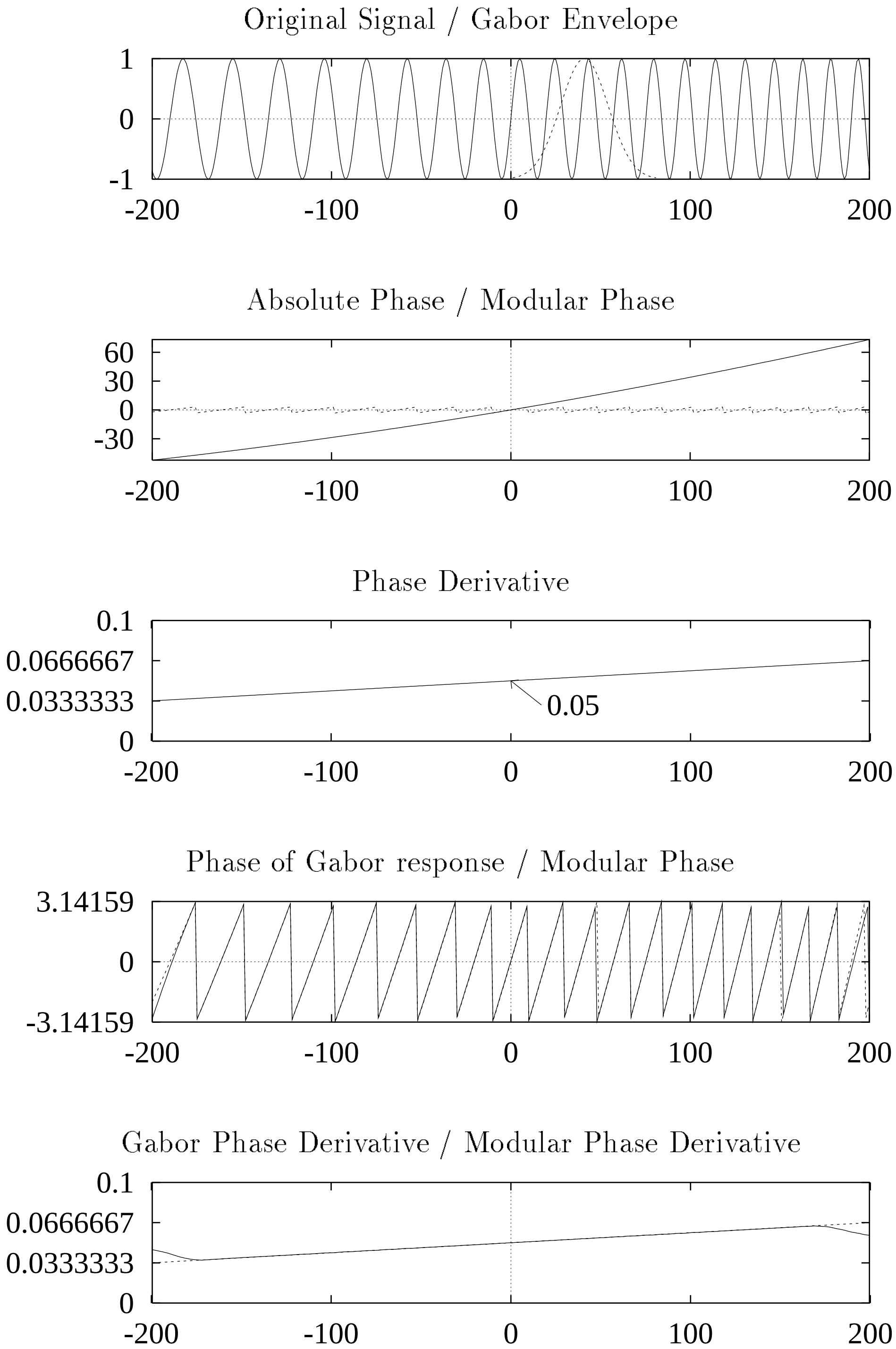


Figure 3.5: Instantaneous Frequency Example: $\omega = 0.03333$ to 0.06667 in linear steps, i.e. a *chirp* signal. The figure is explained in Section 3.4.1

of the original sinusoid. Amazing! To acquire some intuition as to why this must be so, study the figures and compare the original signal's appearance to the X-axis. A sinusoid has a fixed shape, but may be stretched or compressed along the X-axis by changing its frequency. In exactly the same way, the phase of a sinusoid will always change linearly, and its rate of change (i.e., slope) will be related to the original signal's period, or frequency.

The linear change in phase of a sinusoid is not only theoretically interesting, it can also be measured in an arbitrary signal using a band pass filter. The bottom two plots of Figures 3.4 and 3.5 demonstrate this by convolving the original sampled signal with a single Gabor filter whose tuning frequency is $\omega = 1/20$. Even ignoring the details of the computation, one can see in Plot d) that the phase computed by the Gabor filter is approximately equal to the analytically known phase, the dashed line in both Plots b) and d). Finally, the derivative of the Gabor phase is shown in Plot e), again with the analytically known value plotted as a dashed line. Except around the discontinuities at the ends of the signal, the empirically-derived Gabor estimates are extremely close to the analytically known values, even in the chirp signal (Figure 3.5) where the frequency keeps changing. The most interesting property to notice here is that this method of computing instantaneous frequency is reasonably independent of the analysis filter's tuning frequency, since the *same* Gabor filter was used to compute the phase in both figures, and the frequencies in the chirp signal span an entire octave.

As an aside, when the tuning frequency of the filter differs significantly from the frequency of the sinusoid, an *aliased* response may result. In that case the sampled phase derivative may not reflect the instantaneous frequency at all. For a nice explanation and illustration of the effects of aliasing, see (Krumm, 1993).

Now that we have sufficient background in the frequency domain, we must consider the operational issues involved in turning the theory of phase-based stereo matching into a working algorithm. The following sections consider each component of Equation 3.1 in turn, discussing issues of reliability and measurement precision for each parameter in the equation.

3.4.2 Measuring Sinusoid Frequency ($\lambda_{\mathbf{ref}}$)

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 tuning 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 (Sanger, 1988; Weng, 1993), 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*) was shown in Section 3.4.1 to be a good predictor for the example signal. While it provides a more accurate measure of a sinusoid's frequency in theory, in practise its accuracy depends on the amplitude of the input signal. This has led several researchers to develop constraints that filter out unreasonable phase-derived frequencies (Fleet et al., 1991; Xiong, 1995). 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 completely eliminate the effects of aliasing or compensate for perspective foreshortening. Still, it will be useful in improving the accuracy of our results, so we discuss it further in the next section.

3.4.3 Measuring Phase ($\Delta\phi$)

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? Assuming the standard model of broadband white noise (Horn, 1986, p. 129), each complex coefficient in the frequency domain will have a complex error added to it (we represent the error as a 2D vector $\vec{\epsilon}$). $\vec{\epsilon}$ may differ at each frequency, but will always be a perturbation with magnitude of at most ϵ and a random phase orientation. Figure 3.6 gives us some intuition. The vectors represent the Gabor coefficients which are 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 $\vec{\epsilon}$.

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 3.9 below, derived as follows. Place the tail of $\vec{v}$ on the origin

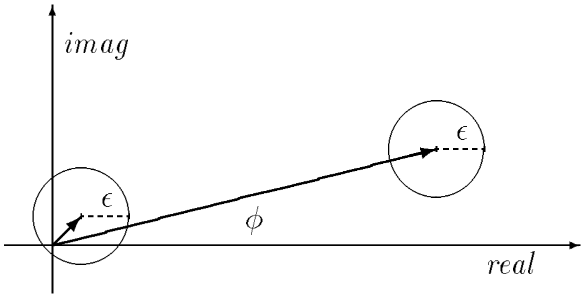


Figure 3.6: The effect of measurement error vector $\vec{\epsilon}$ on phase angle ϕ.

and rotate $\vec{v}$ so that it points to the right on the horizontal axis (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:

$$\tan error = m = \sqrt{\frac{1}{\left(\frac{|\vec{v}|}{|\vec{\epsilon}|}\right)^2 - 1}} \tag{3.9}$$

Equation 3.9 tells us that the maximum phase error depends solely on the ratio between the lengths of vectors $\vec{v}$ and $\vec{\epsilon}$. Sample phase angle errors are plotted as a function of this ratio in Figure 3.7.

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 presence of information at nearby frequencies, the discretization of the filters themselves (especially at high frequencies), and aliasing due to image textures with wavelength less than two pixels to name a few.

A technique for filtering out unreliable phase measurements can be found in (Fleet & Jepson, 1993). They identify two sources of unreliability, and provide a constraint that identifies those regions of space-frequency where measurements should not be trusted. Using the phase and magnitude notation for a Gabor filter with tuning wavelength λ, their constraint

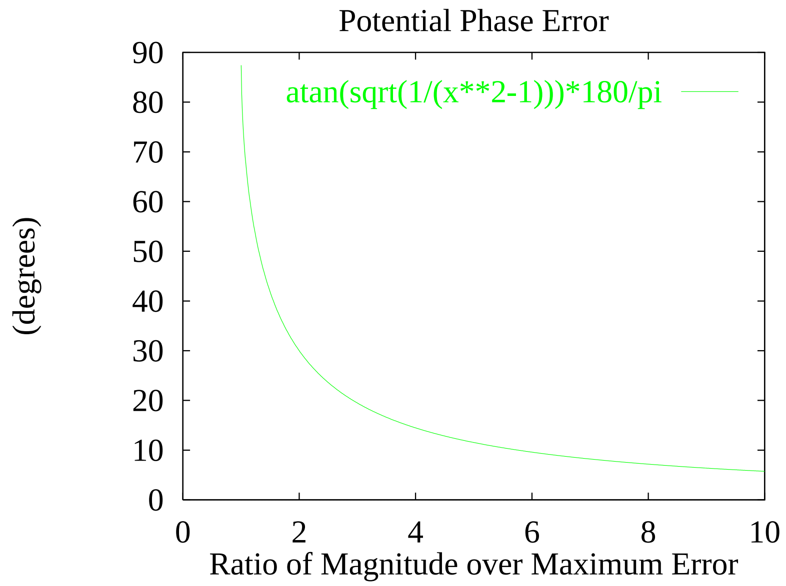


Figure 3.7: Maximum phase angle error as a function of the length ratio.

for finding reasonable measurements can be expressed as:

$$\left| \frac{\delta\phi(x,\lambda)}{\delta x} - \frac{2\pi}{\lambda} + \frac{1}{\rho(x,\lambda)} \frac{\delta\rho(x,\lambda)}{\delta x} \right| < \mathit{Threshold} \tag{3.10}$$

This formula for the constraint was derived from the following two observations:

- The phase derivative (aka instantaneous frequency) of a Gabor filter should be roughly equal to its tuning frequency.
- The amplitude derivative should be small.

These observations (and the constraint in Equation 3.10) are helpful in eliminating unreliable measurements, but they are somewhat simplistic. For example, Chapter 5 will demonstrate that not only *can* the instantaneous frequency differ from the tuning frequency, it *will* differ when the image pair exhibits perspective foreshortening. Also, Xiong has refined this constraint to an arbitrarily higher degree using Moment Filters (Xiong, 1995). But since we will adapt this constraint to handle foreshortening, and since (like Fleet *et al*) we also use Gabor filters, Equation 3.10 will be useful for us as well.

A Peak-finding Heuristic

Another approach at eliminating unreliable estimates that was found to be useful on some images involved a more heuristic process applied to the output of several filters (i.e., one

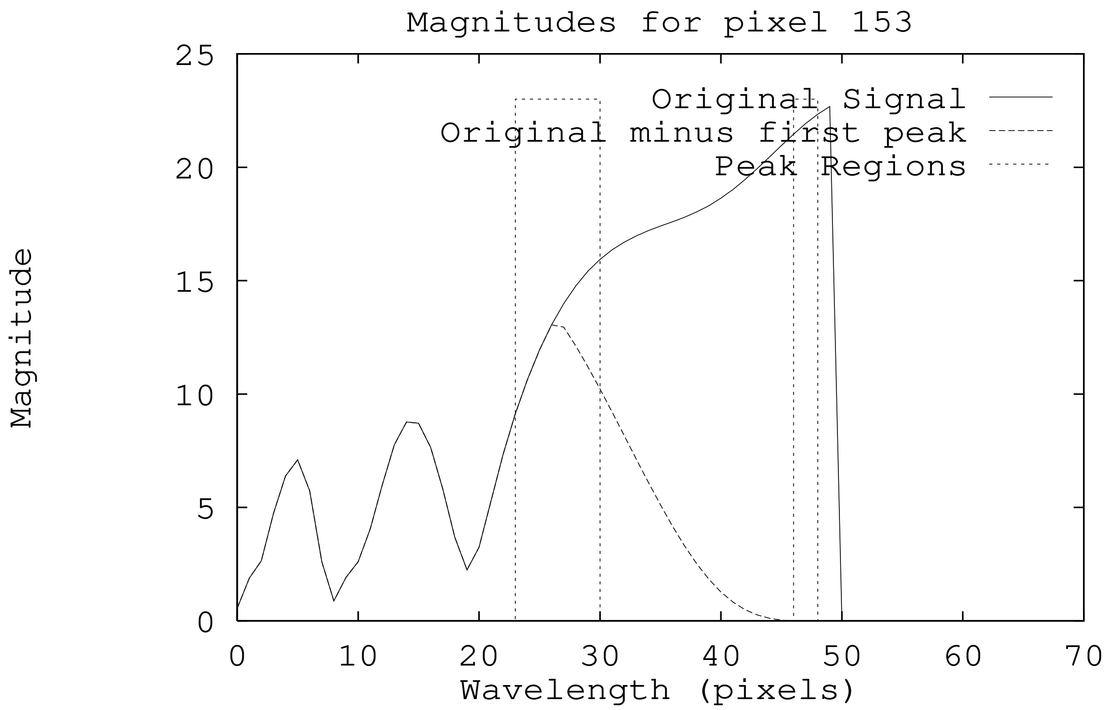


Figure 3.8: Sample Magnitude profile of several Gabor filters, illustrating the peak finding heuristic.

column of the *scalogram*, discussed below). Inspired by the intuition that the most reliable results could be found by concentrating the search on those filters whose magnitude was highest, it isolates those filters whose magnitudes exhibit a peak (or spike) in the overall frequency profile. This kind of scheme works best when the input consists of a sum of band-limited signals, and makes an effort to deal with interference between nearby filters.

The heuristic peak finding scheme works as follows. We attempt to find those regions of the magnitude profile that exhibit explicit peaks by iteratively fitting and subtracting Gaussian curves to the peak regions of the current profile (illustrated in Figure 3.8). 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 Gaussian curve 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.

3.4.4 Comparing Frequencies (λ)

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 corresponding image regions differ, as can occur with perspective foreshortening? Others have smoothed over the effects by using the filter tuning frequency, or the average of the two instantaneous frequencies (Fleet & Jepson, 1993), but these are very coarse approximations not based on physical reality. One contribution of this thesis is the development of a theory for finding the proper frequencies based on the physical geometry of the scene, presented in Chapter 5. However, there are difficulties in combining estimates from different filters even when the frequency contents of the image pair are directly comparable, e.g., when there is no foreshortening.

3.5 Choosing Frequencies (*for all* λ)

So far we have discussed primarily the properties of a single filter, applied to signals with a single frequency component. However, unless the input is known to be of a certain form (e.g., an image consisting only of sine waves with restricted frequencies), any useful analysis will require many filters. A nice overview of local spatial frequency models that use multiple frequencies can be found in (Rioul & Vetterli, 1991), with a longer enumeration of specific models in (Hlawatsch & Boudreaux-Bartels, 1992).

3.5.1 The Image Scalogram

For most of this thesis the *image scalogram's* filter set will be used. The image scalogram is the result of applying a collection of Gabor filters to a 1D image scanline, and therefore consists of a 2D matrix with complex-valued elements; an example is presented in Figure 3.9. Its seemingly unusual triangular shape is due to the interaction between the varying filter window size and the border of the scanline. The high frequency filter outputs found at the top of the scalogram are computed using only a few pixels, so their values can be computed over most of the width of the image. At lower filter frequencies, the number of points that must be sampled increases (as required by the Sampling Theorem), reducing the number of outputs that can be computed from the fixed number of scanline pixel values. Finally, the

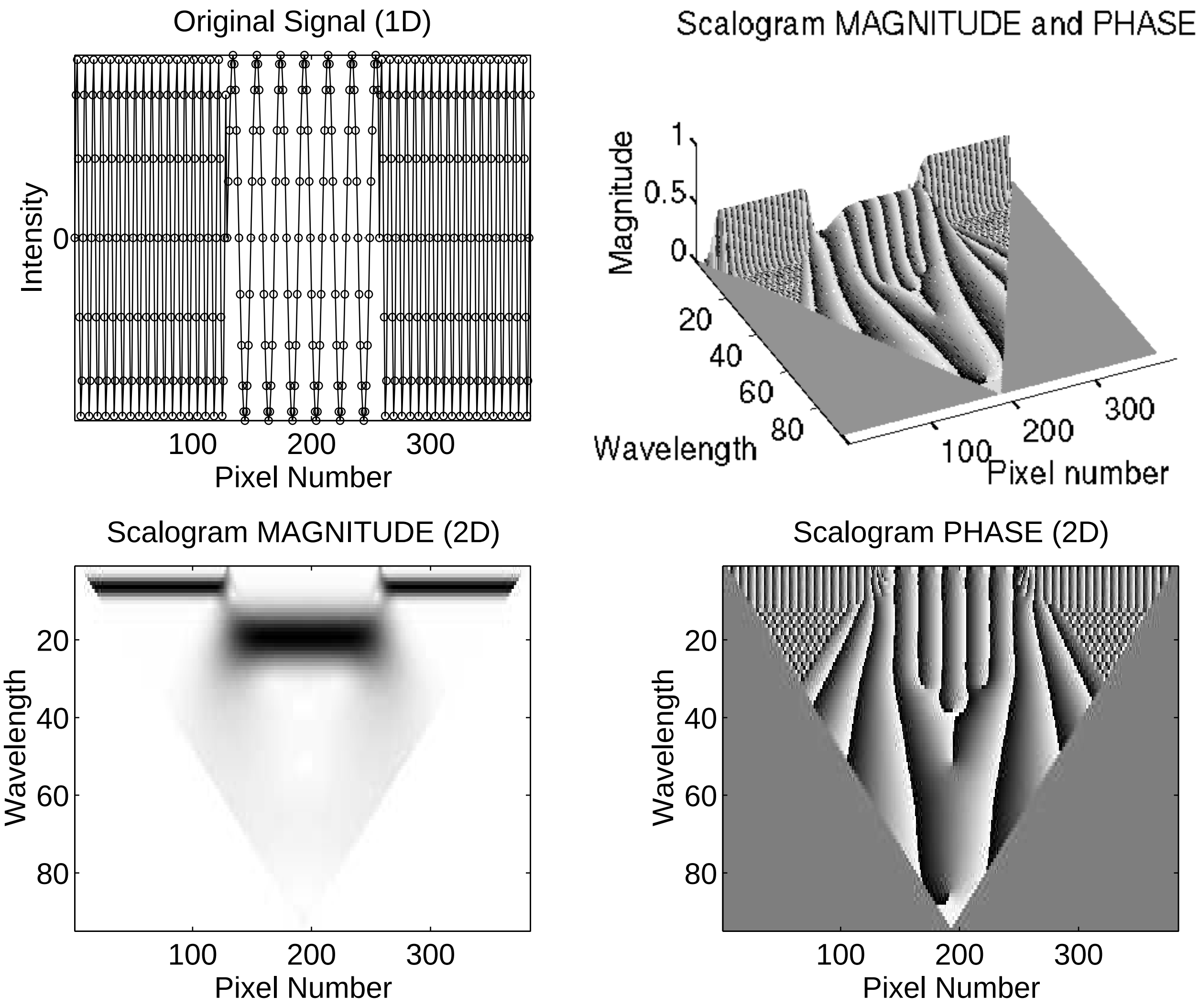


Figure 3.9: Double sine wave signal and associated scalogram (both magnitude and phase).

filter with lowest frequency (located at the bottom of the scalogram) can only be computed at a single point, since it requires the entire scanline as input. Although techniques for filling out the "missing" areas exist, they require either knowledge of the signal beyond the bounds of the input image, or heuristic assumptions about the pattern at the image border. We have chosen to avoid the use of heuristics by considering only those filter outputs for which all the data is available; hence the triangular shape.

The sampling along the (vertical) frequency axis is one of the principal differences between the image scalogram and other local spatial frequency representations. The sampling used in the spectrogram (e.g., Figure 3.1) is the same as that computed by the FFT: a linear sampling of n frequencies from $\omega = 2/n$ to $1/2$ (where n is the width of the sampling window). In

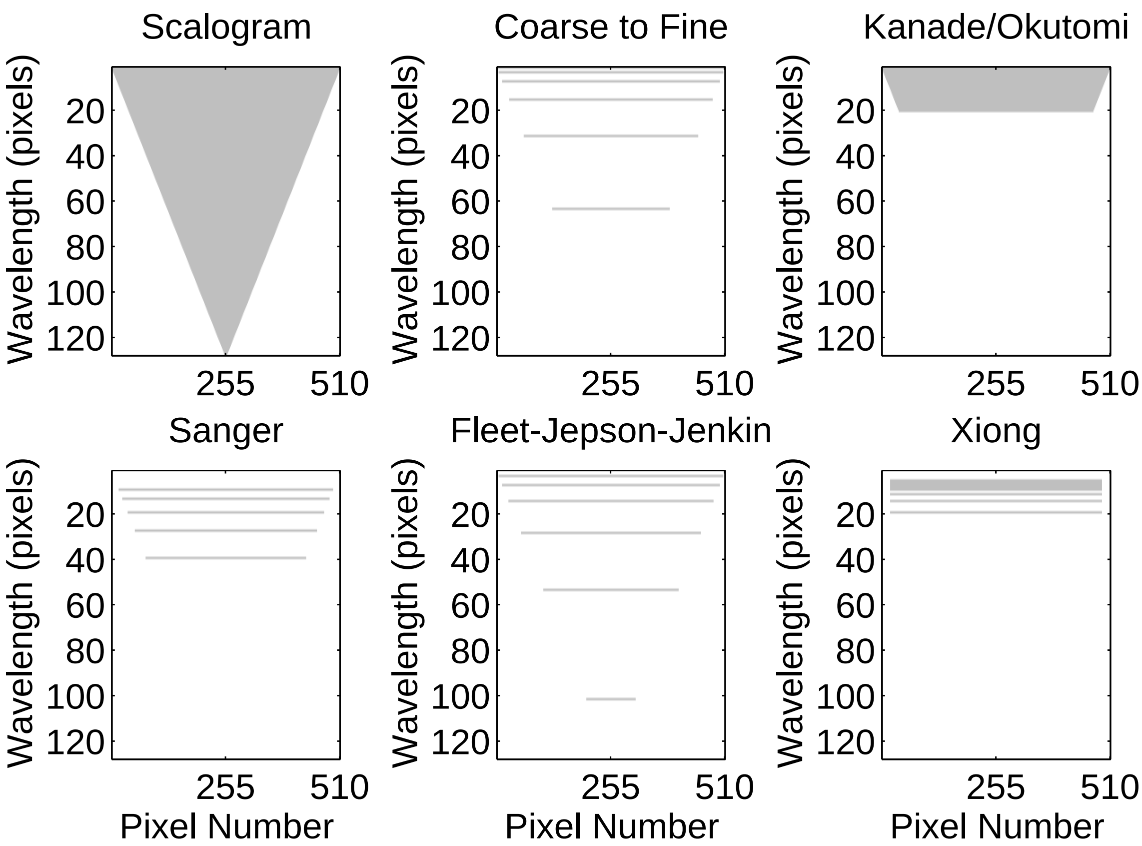


Figure 3.10: Frequency Sampling in the Scalogram and other methods. Grey areas show the central tuning frequencies that are used by each method. Filter widths vary, but typically cover most of the regions between tuning frequencies (see Figure 3.13 for an example).

contrast, the extreme values of the image *scalogram's* frequency axis are the same, but the sampling is linear in *wavelength*, which is the reciprocal of frequency. Why do we prefer the latter? In part because it was found in practise that weighting the contributions of the lower frequencies more strongly (e.g., by using more low frequency filters) often improved the accuracy of the phase-based stereo method. This dense sampling will also be helpful in compensating for perspective foreshortening effects, as will be explained in Chapter 5.

The scalogram also provides a convenient framework for discussion of other samplings reported in the literature. Because its frequency sampling is based on window width, it is easy to see how the selection of filters in this domain compares with spatial domain approaches like coarse-to-fine (aka image pyramid), and variable window size. Since many of those approaches use far fewer filters than the scalogram, their effectiveness at capturing the essence of the useful frequency information from the image can be seen by considering how completely they sample the structures evident in the dense scalogram. Some comparisons are illustrated in Figure 3.10 by highlighting the regions of the scalogram that correspond to the filter sets of those other methods.

Some problems with the often-used coarse-to-fine stereo methods become apparent in the context of the image scalogram. A common approach in many spatial domain stereo methods (Matthies, 1992; Nishihara, 1984), coarse-to-fine has also been used in phase-based stereo methods (Weng, 1993; Fleet et al., 1991). In these approaches, information from the lowest frequency (lowest horizontal stripe in each part of Figure 3.10) is used to provide initial coarse disparity estimates, which will be refined by later processing at subsequent higher frequencies. The difficulty is that there are likely to be regions of the scanline over which the filter outputs will be unreliable,[1] and therefore the initial disparity estimates will be unreliable as well. Even when the higher frequencies theoretically provide enough information to compensate for this problem, the final disparities computed by these methods can be highly inaccurate in such regions. This subject will be addressed in detail in Chapter 4, but it points out the need to consider *all* the filter outputs concurrently, rather than in an arbitrary fixed order.

3.6 Combining Filter Estimates

Section 2.2.1 showed how the phase of a sinusoid, and by extension the measurement of phase using a Gabor filter, could be used to compute stereo disparity. In this section we show how estimates from several filters may be combined to yield more accurate results. This technique is mainly useful in conjunction with methods that compare phases between images using the same-numbered pixel (the so-called *direct phase* method), such as those in (Sanger, 1988; Fleet et al., 1991; Weng, 1993), but it will also be useful in our search-based method.

[1] The prevalence of unreliable phase measurements is discussed at length in (Fleet & Jepson, 1993). That it is a common occurrence can be inferred from the number of "measles" (white spots) in the scalograms shown throughout this thesis; they represent regions whose Gabor filter outputs exhibit low magnitude, and are found in most scalograms of real imagery.

3.6.1 The Phase Wraparound Problem

Direct phase methods make explicit use of phase difference as disparity, so all such methods have to contend with the problem of phase wraparound. The difficulty lies in the fact that phase can only be measured modulo 2π. Therefore, unless special steps are taken, a given filter will only be able to estimate disparities less than the wavelength specified by its tuning frequency. Mathematically, the problem is that the measured phase difference $\Delta\phi$ might not equal the ideal phase difference $\Delta\phi_{ideal}$:

$$\begin{aligned}
\Delta\phi_{ideal} &= 2\pi\frac{Disparity}{\lambda} && (3.11)\\
\text{But}\quad \Delta\phi &= \left|\Delta\phi_{ideal}\right|_{2\pi} && (3.12)\\
\text{So}\quad \Delta\phi_{ideal} &= \Delta\phi + k\,2\pi && (3.13)\\
&\quad \text{where}\quad k = \left\lfloor\frac{\Delta\phi_{ideal}}{2\pi}\right\rfloor = \left\lfloor\frac{Disparity}{\lambda}\right\rfloor
\end{aligned}$$

So $\Delta\phi$ is only part of the answer: the additional factor k must be known if disparity is to be computed exactly from a single filter. Unfortunately, there is no way to measure k without knowing the ideal disparity. While there are some techniques that recover these k values from a set of filter outputs (the so-called *phase unwrapping* techniques of (Tribolet, 1977; Ghiglia & Romero, 1994)), they require accurate phase measurements (and therefore high magnitude response) from *all* filters, a condition rarely met in real images. Because recovery of these k values is so difficult, many methods either assume $k = 0$ or arrange the processing so that k is assumed to be known.

How likely is it that the wraparound issue will arise? Figure 3.11 shows several plots of ideal phase difference as a function of disparity. As the disparity increases, so does the number of times the phase value is expected to wrap around. Methods that rely on many high frequency filters are especially likely to suffer wraparound effects as the disparity grows larger.

This wraparound problem weighs heavily on Sanger's method, which is actually limited to disparities less than the *smallest* wavelength of his filter set. The restriction comes from his technique for combining estimates: averaging the disparity predicted at each filter, assuming $k = 0$ for all phase differences. His method is therefore unable to take advantage of the fine detail present in the highest frequencies of the input images, since the wavelength of

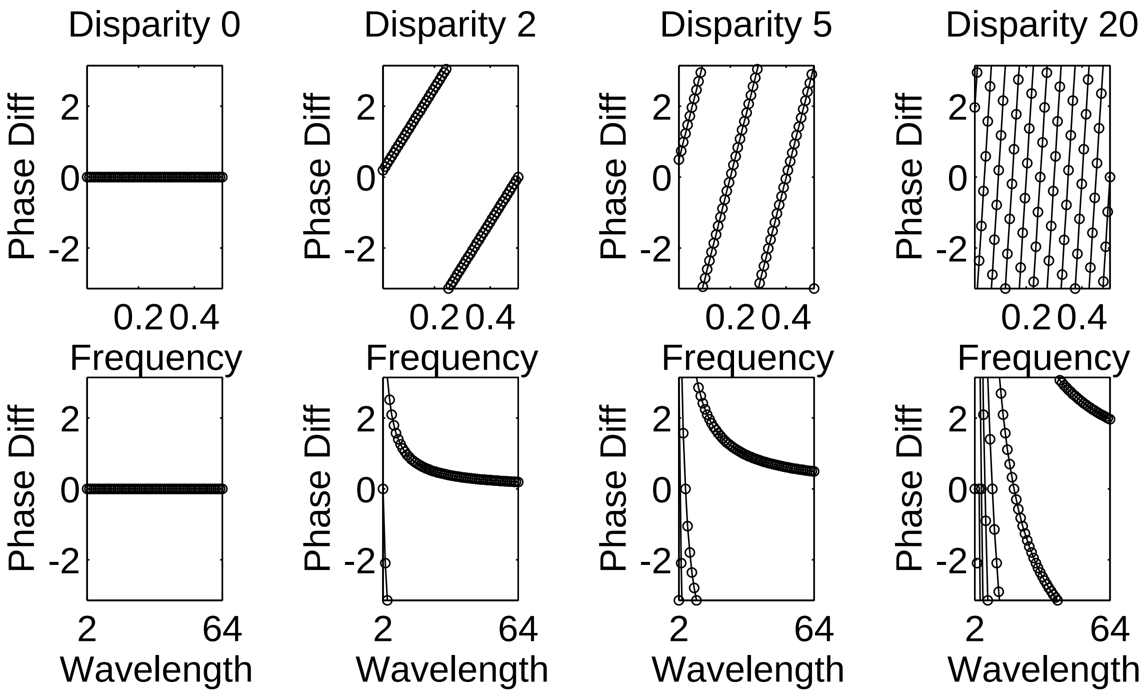


Figure 3.11: Ideal phase difference as a function of Disparity, illustrating the phase wraparound problem. Note that coarser scales are on the *left* in the Frequency plots, but on the *right* in the Wavelength plots. Each graph has 63 sample points spaced linearly along the X axis. Note that unlike the 2D scalogram plots, here the wavelength axis is *horizontal*.

the highest frequency filter must meet or exceed the maximum expected disparity. You can see how his filter set differs from other methods (by lacking the highest frequencies) in Figure 3.10.

Other direct phase methods (Fleet et al., 1991; Weng, 1993) address the phase wraparound problem using a coarse-to-fine approach. Low frequency filters provide initial coarse disparity estimates, which are used only to fix the value of k for the next-higher frequency. This results in two advantages over Sanger's method: disparities as large as the *longest* wavelength in the filter set can be computed, and information from much higher frequencies can also be incorporated. Unfortunately, these approaches make the unreasonable assumption that all filters will have accurate measurements at every pixel. Since there are actually many regions in the scalogram where phase measurements are unreliable (Fleet & Jepson, 1993), the value of k used in the filter with next-higher frequency could be completely wrong, resulting in wildly inaccurate disparity outliers. Weng attempts to address this problem by linearly interpolating phase in the unreliable areas, but his method has still been found to be the least precise of several phase-based methods (Jenkin & Jepson, 1994). How then can we combine estimates from many filters without knowing in advance which of them will have reliable results?

3.6.2 Our Filter Combination Method

The usual approach in direct phase methods is to compute disparity directly from the phase difference measurements. But we can turn the computation around too, and address the complementary problem: *given* a particular disparity, what are the expected phase differences from all filters? The solution to this problem is straightforward; simply apply Equations 3.11 and 3.12 to the known disparity and filter wavelengths to find the expected or "ideal" phase differences. Figure 3.11 shows the ideal values associated with several disparities. The stereo task then becomes one of finding that disparity whose ideal phase differences best match what was measured. Why bother with this restructuring? By stating the problem in this way, we eliminate the need to compute the appropriate value for k at each filter.

The problem now is how to define the "best" match between measured phase differences and the ideal phase differences associated with a particular disparity. We need an evaluation function that computes a quantitative agreement between these two sets of phase differences. It should combine estimates from any number of filters with arbitrary tuning frequencies, should deemphasize the unreliable measurements, and must account for the potential wraparound effects of comparing phases (phase differences). Empirical tests of various filter combinations (sums of unnormalized v. normalized magnitude), weighting functions (magnitude, squared magnitude, square root of magnitude), and phase comparisons (raw difference, modular difference, with and without absolute value) were performed on several image pairs, and this formula:

$$EvalDirect\ (disp, c) = \frac{1}{|W|} \sum_{\lambda \in W} \rho(c, \lambda)\, \mathrm{AbsDiffMod}\left(\Delta\phi_{ideal}(disp, \lambda), \Delta\phi(c, \lambda)\right) \quad (3.14)$$

Where $disp$ is the candidate disparity,

c is the current pixel number,

ρ is the magnitude normalized by window size,

$$\Delta\phi(c, \lambda) = |\phi_L(c, \lambda) - \phi_R(c, \lambda)|_{2\pi}, \quad (3.15)$$

and AbsDiffMod is the smallest difference between phases, i.e.

$$\text{AbsDiffMod}(a,b) = \min_{k\in\{-1,0,1\}} \{||a|_{2\pi} - |b|_{2\pi} + k2\pi|\} \qquad (3.16)$$

was found to provide the most accurate results using the direct phase approach. As mentioned previously, the filters used are typically those of the image scalogram, i.e., linear samples of wavelength from 2 pixels to the one quarter the width of the complete image. Smaller values of Equation 3.14 indicate better matches with the predicted phase differences, so the disparity that yields the minimum value is the best estimate. As an aside, one might be able to replace the AbsDiffMod(a,b) function with $\cos(a-b)$, then use the *maximum* value to indicate the best match. The benefit would be in using a simpler function, but the effect of its new shape (a curve that is smoothly varying rather than being sharply peaked at zero) on the precision of the method would need to be studied.

Figure 3.12 illustrates how each Gabor filter and each candidate disparity contributes to the final error estimate. Each column in the figure is labeled with a disparity, and represents the goodness-of-fit of all of the measured phase differences to the ideal phase differences predicted by that disparity. Each column is summed over all appropriate wavelengths (witness the $\sum$ in Equation 3.14), and forms the overall estimate for the likelihood that this disparity is correct for this pixel.[2]

This evaluation function satisfies the requirements given above. It does not depend on a particular set of tuning frequencies; any set of wavelengths W can be used, and they need not be related according to a fixed hierarchical scheme (although nothing prohibits such a sampling). For example, one might use the scalogram's linear wavelength sampling $W = \{2 \ldots N/m\}$ pixels, where N is the width of the scanline and m is the Gabor filter tuning parameter for number of wavelengths per window. Or one could use just the sparse sampling of the coarse to fine methods; Figure 3.13 shows the frequency response for such a set of hierarchical Gabor filters). Whatever the sampling, unreliable measurements are deemphasized in Equation 3.14 by eliminating the worst of them from W and by weighting most strongly those outputs from filters with high magnitude (Section 3.4.3 explained the relationship between magnitude and reliability). The wraparound problem is addressed by finding the smallest absolute modular difference between $\Delta\phi$ and $\Delta\phi_{ideal}$.

This method has several advantages. It allows regions with unreliable phase measurements to be ignored safely, without compromising multi-scale processing. Since the mea-

[2]Figure 3.12 was actually generated using the *error* function from Table 3.2 instead of Equation 3.14, but the summation method is the same for both error functions.

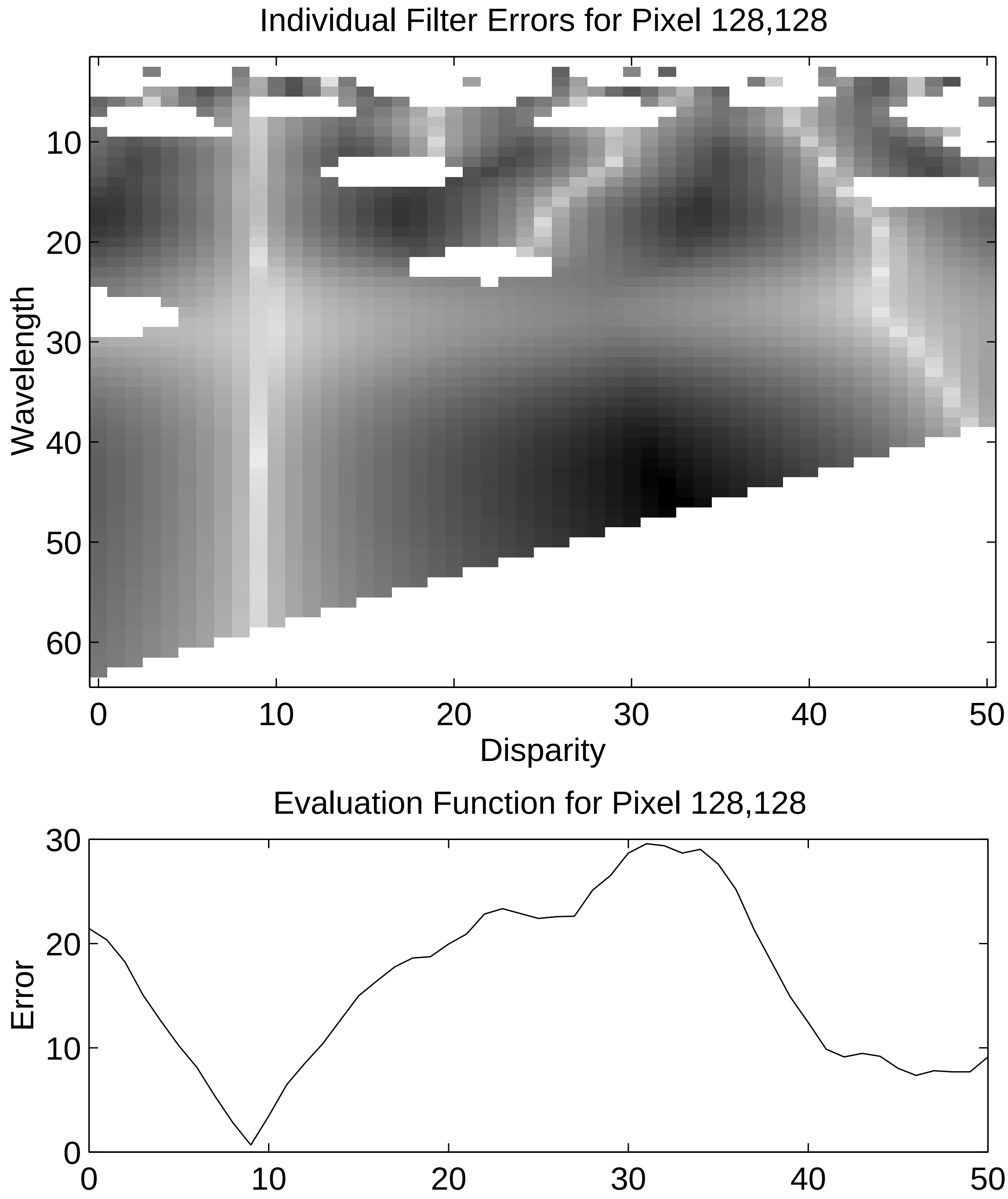


Figure 3.12: Filter contributions to the disparity estimate. Pixel intensity represents the value of the error computed by each filter's phase difference comparison, scaled by the magnitude. Darker pixels indicate larger error, and white spots represent areas where the scalograms contained no useful information. The evaluation function (below) is constructed by summing up the values in the columns and dividing by the number of filters used. This plot represents the disparity computation at pixel (128,128) of the stereo pair in Figure 3.14.

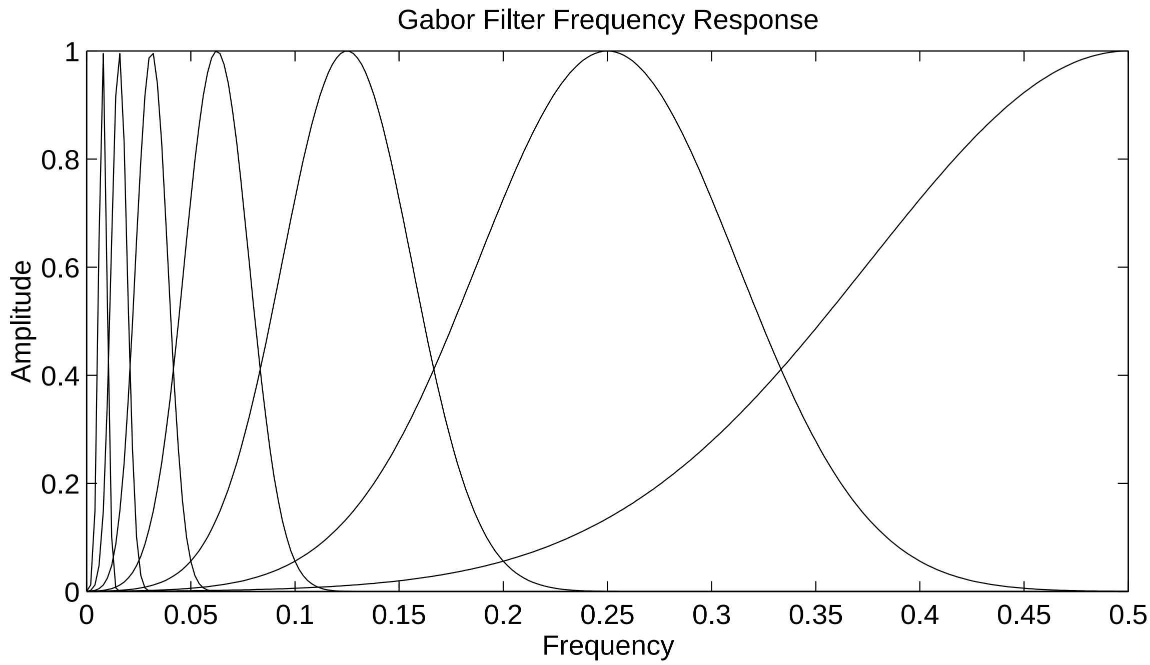


Figure 3.13: Gabor Filter Frequency Response. The Gabor filter magnitude in the frequency domain for filters with wavelength 2, 4, 8, 16, 32, 64, and 128 is presented. The tuning parameters are $\sigma_f = \frac{1}{6}$ and $m = 4$.

surement from each filter (equivalently, each scale) makes a contribution independent of the others, those with unreliable measurements can be safely omitted from W. The method also has the interesting property that it can be used to evaluate the likelihood of disparities much larger than the size of the filters used, when the images being studied have non-interfering periodic texture (i.e., they exhibit specific magnitude peaks in the frequency domain). This is because the evaluation function combines the filter outputs simultaneously, and combines estimates across frequencies based on the content of the signal; the signal determines the scales that are used. Coarse to fine methods, in contrast, use scales that are related in a strictly hierarchical manner and are fixed in advance. For example, given an input signal that is the sum of n sinusoids with wavelengths $\lambda_1, \ldots, \lambda_n$, coarse to fine methods can only compute disparities half as large as the largest Gabor filter in the analysis set, whereas our method can evaluate disparities as large as $\mathrm{LCM}(\lambda_1, \ldots, \lambda_n)$, a potentially much larger quantity.

However, this version of the direct phase approach has a problem. Because this is a direct phase method, $\Delta\phi$ is calculated by subtracting Gabor phases from the two scanlines at the *same-numbered pixel* (witness the c in Equation 3.15). From Section 3.2.1 we know that the window size of a Gabor filter varies as a function of wavelength, so filters with small wavelengths (high frequencies) will use relatively small windows for their input. In the presence of a disparity larger than the smallest wavelengths in W, this means that the phases measured by the corresponding filters will be from parts of the image that not only do not overlap, but may not even represent the same object's texture. This problem is faced by all direct phase methods, and provides one motivation for the coarse to fine approach used by others (Westelius, 1995; Weng, 1993; Fleet et al., 1991) and their assumption that the true disparity is less than the filter wavelength. But one cannot depend on responses at the lower frequencies, since the phase measurements there might be unreliable; Chapter 4 will show this in more detail. Fortunately, there is an alternative.

3.7 Our Stereo Method

Our stereo method is constructed out of the pieces described in the earlier sections of this chapter. We compute disparity using many Gabor filters, caching the outputs in 2D image scalogram matrices. Those measurements assumed to be unreliable are flagged and do not enter into the computations. A search over candidate disparities is made at each pixel, and the ideal phase differences associated with each disparity are compared against those measured in the images. 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.

Pseudocode for this algorithm can be found in Table 3.2. Matlab code for performing the computation is available on the web from *Mark Maimone's Index Page*.[3] This method will be extended in Chapter 5 to address the perspective foreshortening issue, but we outline the basic method here. Let a pair of images 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, each containing n pixel intensities. Compute the

[3] `http://www.cs.cmu.edu/~mwm/`

Given: A pair of greyscale images, and a list of candidate disparities.

For each row
  Compute Left and Right Scalograms L and R
  For each column c
    $W_L(c) = \{\lambda : \rho_L(c, \lambda)$ exists and $\phi_L(c, \lambda)$ is reliable$\}$
    $W_R(c) = \{\lambda : \rho_R(c, \lambda)$ exists and $\phi_R(c, \lambda)$ is reliable$\}$
  For each column c
    For each disparity d such that pixel $c + d$ exists
      $W = W_L(c) \cap W_R(c + d)$
      $error = \frac{1}{|W|} \sum_{\lambda \in W} \rho_L(c, \lambda) \text{AbsDiffMod} \left(\Delta\phi_{ideal}(|d|_1, \lambda), (\phi_L(c, \lambda) - \phi_R(c + d, \lambda)) \right)$
    Return d that yields minimum $error$

Table 3.2: Pseudocode for the basic phase-based stereo algorithm.

scalograms of these rows, using particular Gabor filter parameters (e.g., $m = 4$ 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. So $\rho_L(c, \lambda)$ is the magnitude of the left row's scalogram entry at pixel c with wavelength λ, $\phi_L(c, \lambda)$ 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 number c, we first mark unreliable estimates using Equation 3.10. Having thus ignored potentially unreliable phase values, we use the remaining left and right phases to fit phase differences. First we enumerate a list of possible disparities and compute the "ideal" phase difference curve for each disparity using Equation 3.11. If we let W be the set of wavelengths whose outputs are considered reliable, then the error for a particular disparity is given by the version of Equation 3.14 found in Table 3.2. Finally, we select the disparity that exhibits minimum error as the result for this pixel.

Note that by enumerating candidate disparities, we greatly reduce the phase wraparound problem mentioned in Section 3.6.1. Instead of comparing phase values at the same-numbered column, we use columns that are actually expected to correspond (i.e., we include disparity

d in the phase difference $(\phi_L(c,\lambda) - \phi_R(c+d,\lambda))$ part of Table 3.2's *error* computation). Thus the ideal phase differences need only reflect the remaining *subpixel* disparity (see the $|d|_1$ term in Table 3.2). Since the maximum remaining disparity will be one pixel, and the highest filter frequency will never have a wavelength of less than two pixels, the predicted phase differences will never wrap around. There is no such restriction on the *measured* phase differences, however, so care must still be taken to use the AbsDiffMod function from Equation 3.16 for the comparison.

Some form of phase interpolation is required to achieve the highest precision. However, none is explicitly specified in the implementation given in Table 3.2. The reason is that this interpolation has been encoded into the computation of the ideal phase difference. Rather than interpolate between the measured phases to find the perfect match (with zero disparity and hence uniformly zero ideal phase differences), we tune the ideal difference to compensate for this subpixel shift. So we are using a form of the direct phase method here, but since it is only applied to subpixel disparities we will not incur the wraparound problem.

3.7.1 Sample Imagery

Some example images with stereo results and ground truth are presented in Figures 3.14 and 3.15. The results illustrate some of the limitations of using 1D filters: in particular, errors at depth discontinuities are more jagged than would be expected from a method using 2D filters. And although the average RMS errors seems high (being greater than 1 pixel), it is difficult to compare with other methods since few others have reported ground truth results over complete images. Visual inspection of the area in the center of the images reveals that the error there is usually much less than 1 pixel.

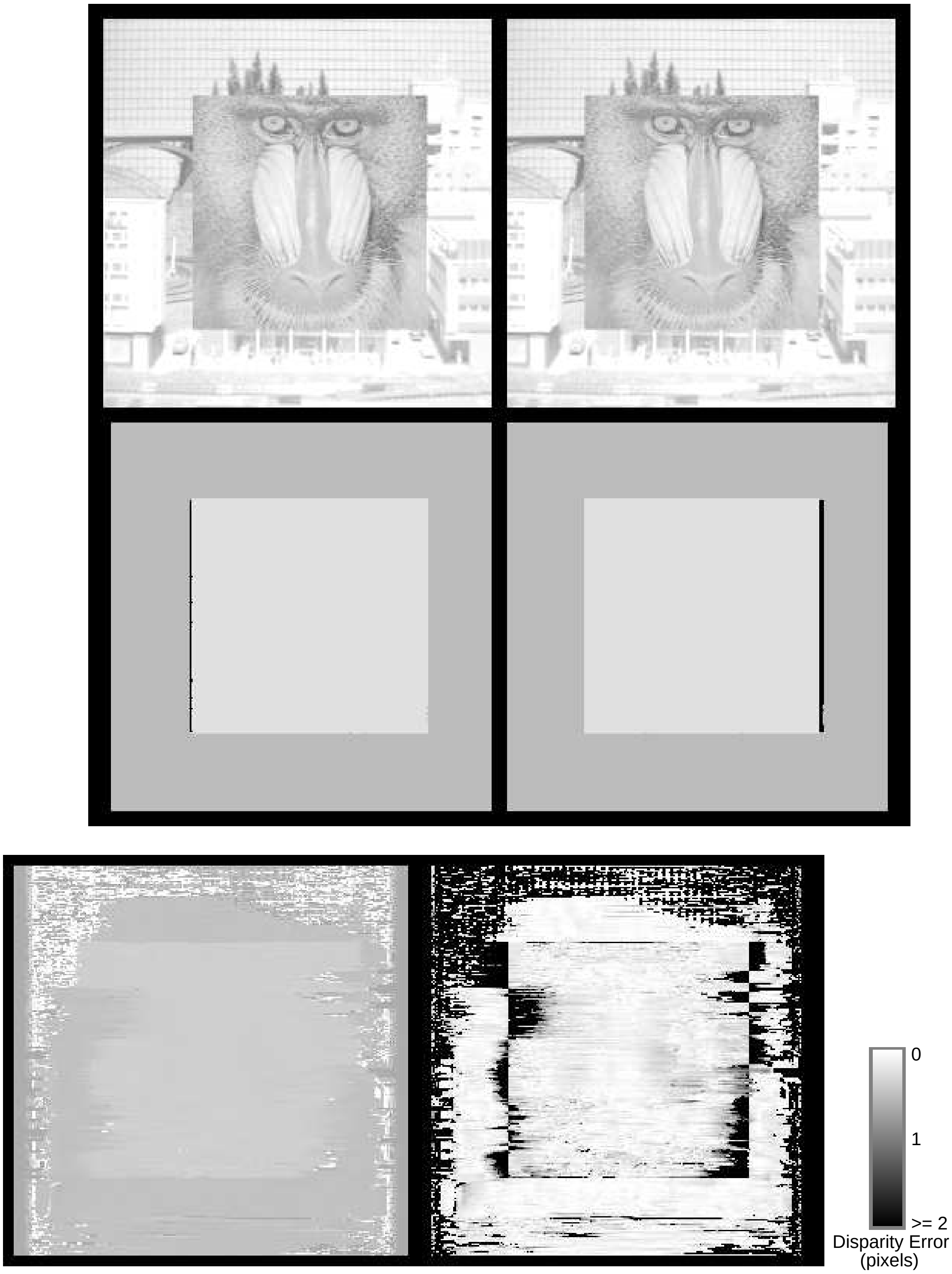


Figure 3.14: Synthetic Stereo Pair with Ground Truth. The top two images form the stereo image pair of two frontoplanar surfaces, the middle images are the associated disparity maps with occluded areas marked in black. The lower left image is the disparity map computed by this method, evaluating 301 disparities from 0 to 30. The lower right image is the difference between the stereo disparity and the ground truth. Mean error over the entire figure is 2.41 pixels (variance 23.09); all errors ≥ 2 are in black. (intensities not to scale)

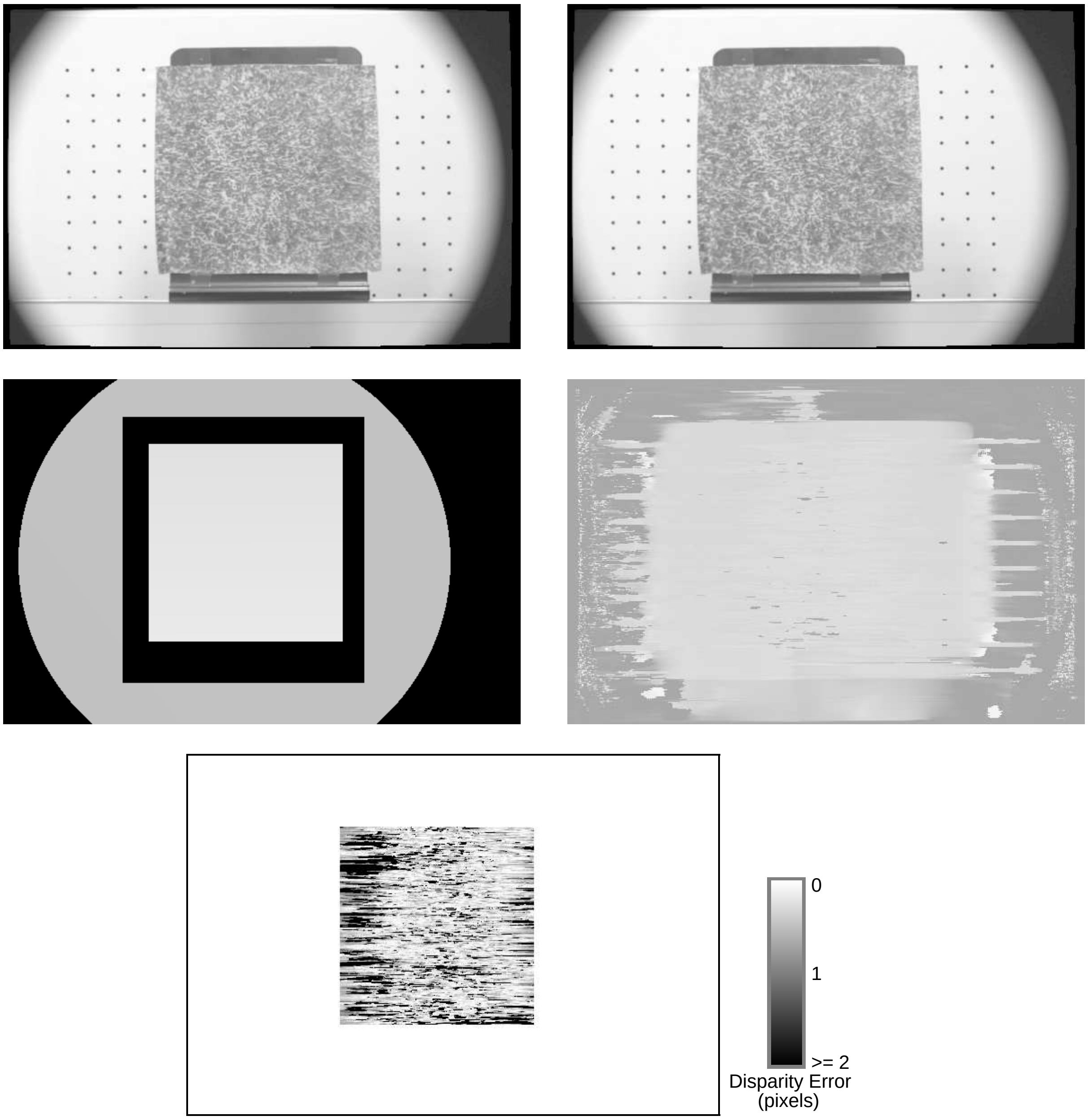


Figure 3.15: Actual Stereo Pair with Ground Truth. The top two images form the stereo image pair of a flat sheet tilted back slightly, and have been corrected for lens distortion. The middle images are the ground truth data on the left (black areas have unknown ground truth), and disparities computed by our method on the right (161 disparities, 0-40). The bottom image shows the differences between our estimates and the ground truth over the central target, scaled from 0 to 2 (all errors ≥ 2 appear black). Mean error in that region in 1.27701 pixels with $\sigma = 2.8446$. (intensities not to scale)