A Taxonomy for Stereo Computer Vision Experiments

Mark W. Maimone
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA USA
mwm@cs.cmu.edu

and

Steven A. Shafer
Microsoft Research
Microsoft Corp.
Redmond, WA USA
stevensh@microsoft.com

http://www.ius.cs.cmu.edu/project/cil/tax/

Abstract

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 [1], 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 paper we outline a framework for choosing a reasonable trade-off of ground truth density v. image realism in the analysis of stereo algorithms.

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 paper 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 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 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 [2] 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 [3], 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."

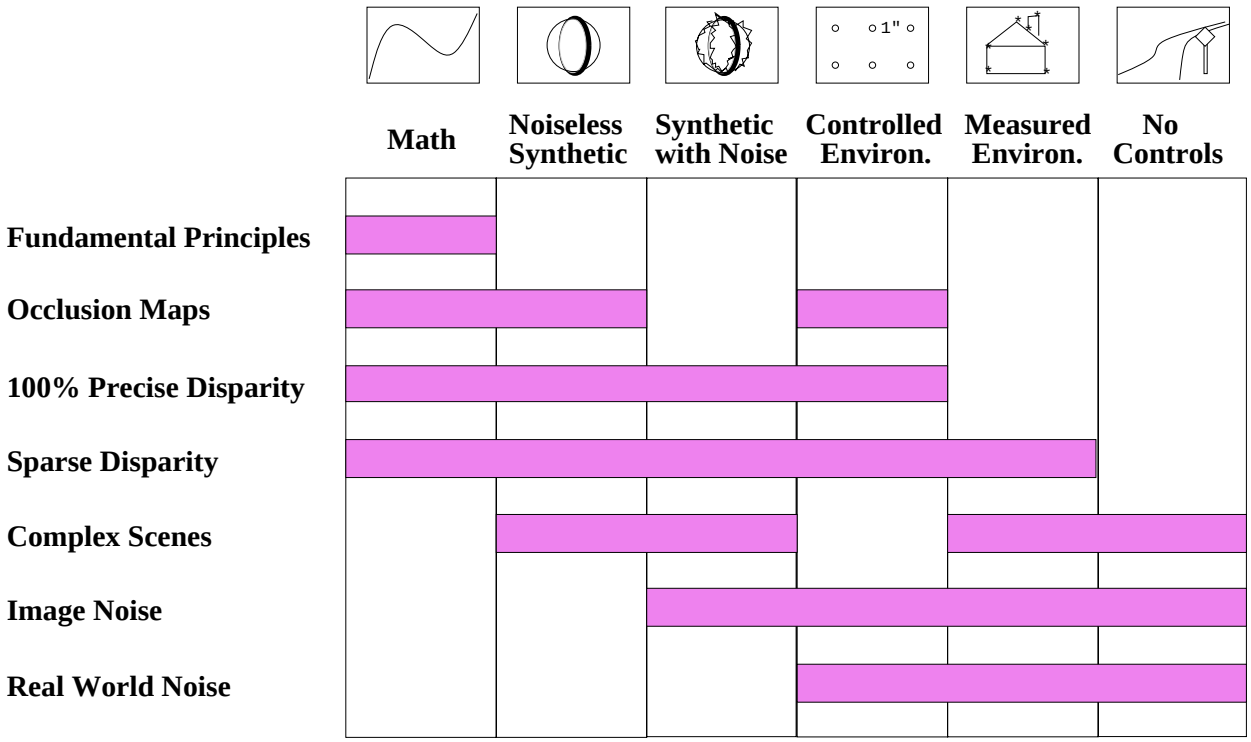


Table 1: Stereo Imagery Characteristics and the types of scenarios for which they are available.

Even those studies that included ground truth have been limited by available technology. An ISPRS study [4] 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 × 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 1), and providing a concrete example for each level. Sections 2 through 7 present the scenarios, ordered by increasing realism and decreasing amount of ground truth, Section 8 describes the tools used to implement these scenarios and their availability, and Section 9 summarizes our contribution.

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 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. This paper 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 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 [5] [6].

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.

2.1 Example 1: Phase difference as disparity

The use of phase difference as disparity lies at the heart of many phase-based stereo algorithms [6] [7] [5] [8]. But unless one is already quite familiar with the frequency domain, 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:

$$L(x_L) = sin\left(\frac{2\pi}{8} x_L\right) \qquad\qquad R(x_R) = sin\left(\frac{2\pi}{8} x_R - \phi\right) \tag{1}$$

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

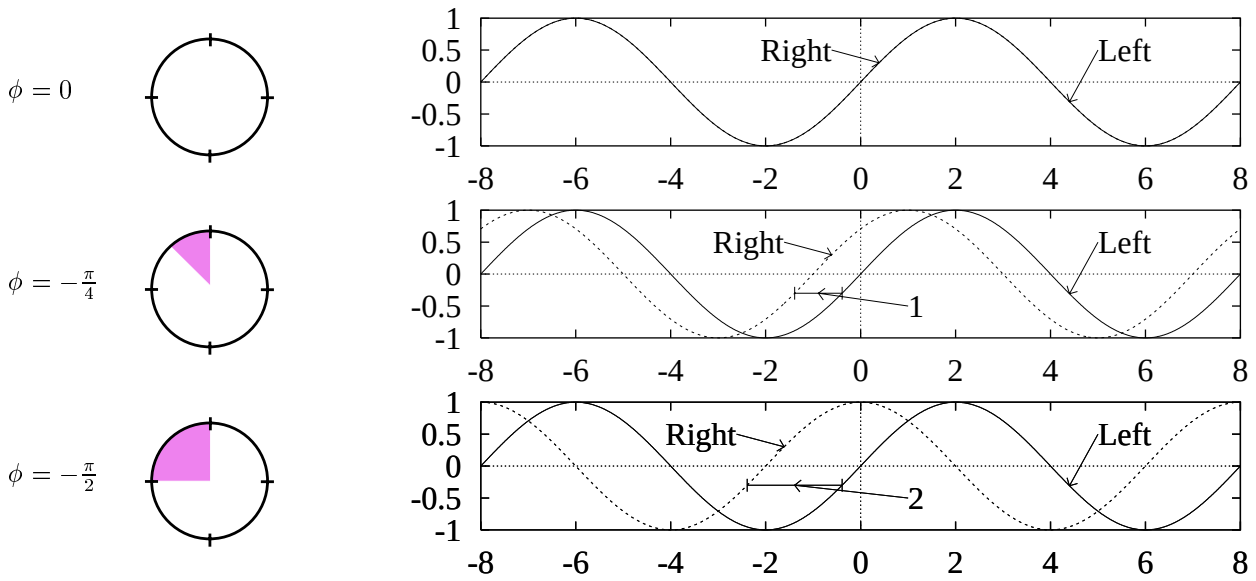


Figure 1: Disparity as a function of Phase Difference. Disparity is the horizontal separation between the two signals, and is indicated by the labeled bar just to the left of 0.

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 Eq. 1 equal and solving for this difference ($=_b$ denotes equality modulo b):

$$\begin{aligned} \sin\left(\frac{2\pi}{8}x_L\right) &= \sin\left(\frac{2\pi}{8}x_R - \phi\right) \\ \frac{2\pi}{8}x_L &=_{2\pi} \frac{2\pi}{8}x_R - \phi \\ \textit{Disparity} := x_L - x_R &=_8 -\frac{8}{2\pi}\phi \end{aligned} \tag{2}$$

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 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 Eq. 2 and comparing the answer with the amount of shift visible in the graphs of Figure 1.

This example also illustrates an important issue in the design of phase-based stereo techniques: phase-wraparound. In interpreting Figure 1, we knew the phase difference was relatively small (i.e., less than 2π), so the disparity could be computed directly. But the disparity formula in Eq. 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 [7] [5], others choose to ignore it [6] 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) [9].

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.

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 [10].

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 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.

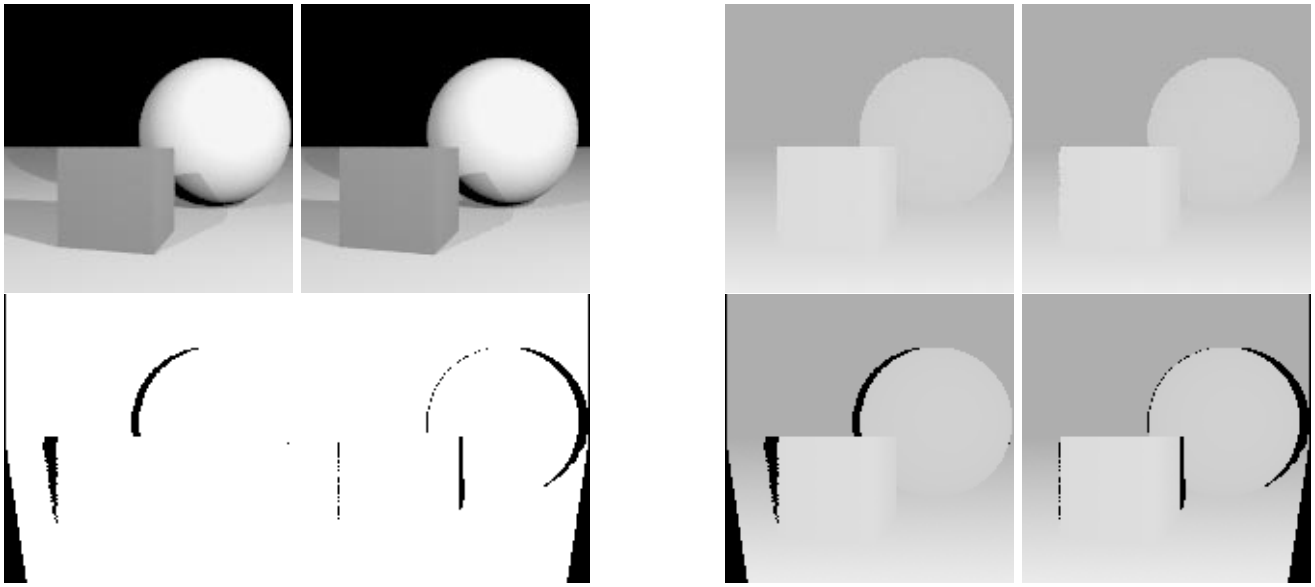
Perfect ground truth is also quite helpful in debugging. While this may seem a trite truism, the difficulty of developing image processing software and the current lack of integrated matrix debugging environments have discouraged vision software developers from adopting this approach of using full-sized 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.

3.1 Example 2: Occlusion Masks

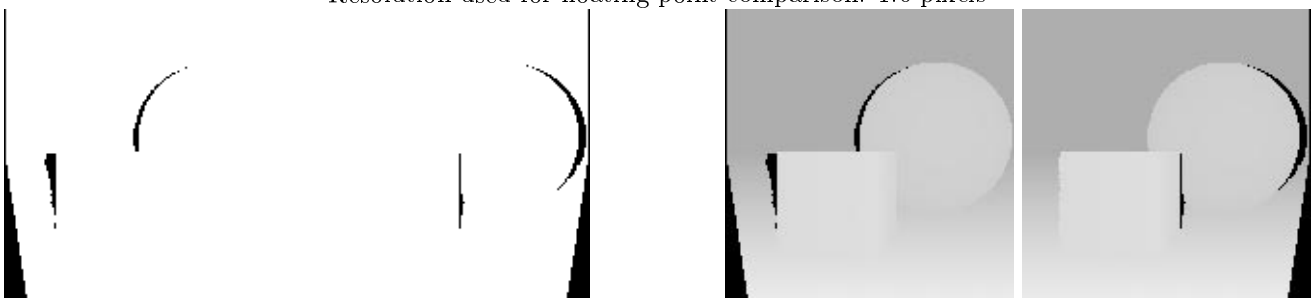
Depth maps in complex scenes are typically discontinuous, and therefore difficult to reason about analytically. Also, when opaque objects are imaged from different viewpoints, portions of those objects will be visible only in one image. For these reasons it becomes important to provide occlusion masks 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 [11], taking care to turn off interreflections and translucence. Pixels with zero intensity 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 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 [12]. 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. Since this algorithm produces noisy results even with absolutely correct disparity maps, a more robust approach is clearly needed.



Resolution used for floating point comparison: 1.0 pixels



Resolution used for floating point comparison: 0.1 pixels

Figure 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.

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 arbitrary pixel sampling 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 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 $\max_{2\times2}(\text{disparities}) - \min_{2\times2}(\text{disparities})$ is minimized). This window selection therefore helps avoid the pointwise method's error of pointing from a slanted surface to the wrong object.

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. 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

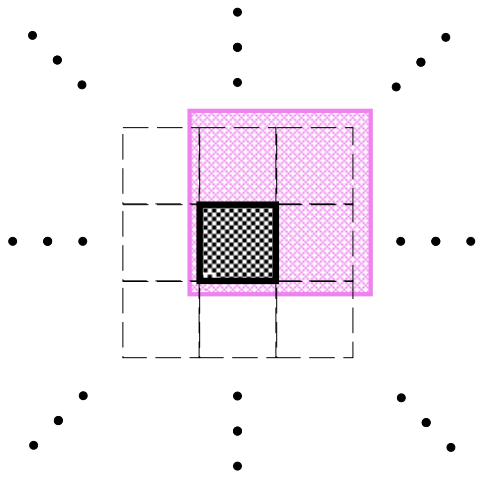


Figure 3: Finding the best-fit plane in the disparity map. There are four 2x2 windows that contain the center (dark) pixel; the upper right window is highlighted.

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: 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 [11] [13].

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

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.

4.1 Example 3: Virtual Checkerboard

Figure 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 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

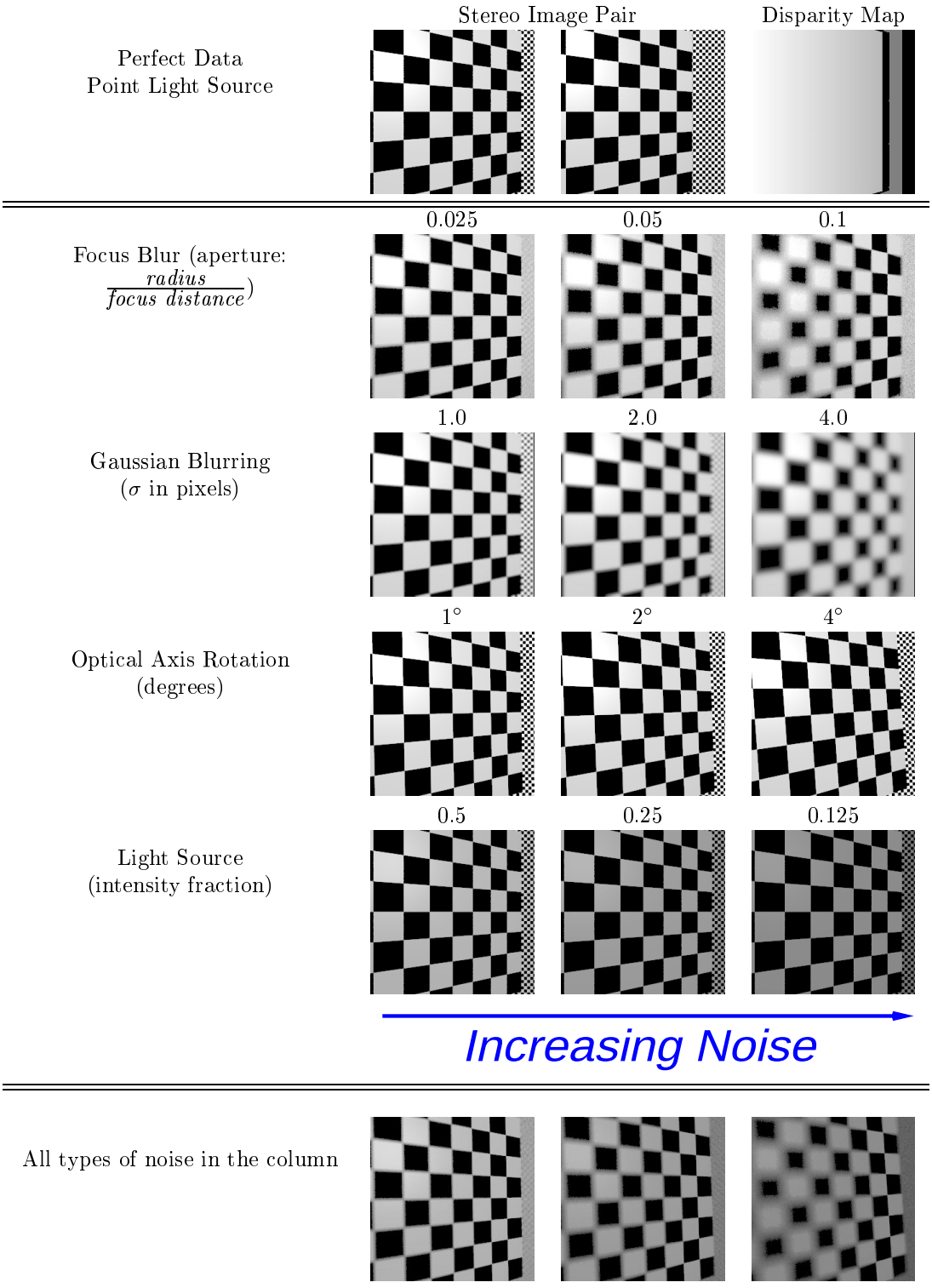


Figure 4: Examples of noise easily modeled with synthetic data.

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 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 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.

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 [14]. 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 [15]).

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.

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 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.

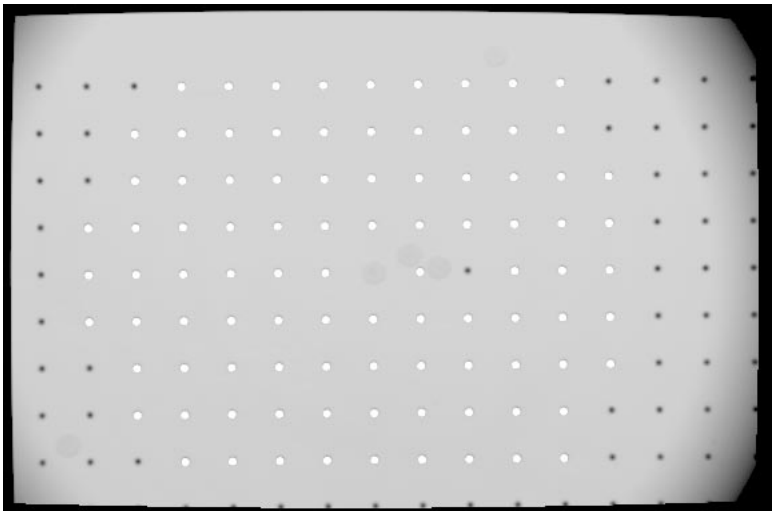


Figure 5: Actual image of the Calibrated Imaging Laboratory (CIL) calibration target with virtual rendering overlaid. The grey background and grid of black dots are part of the original picture, the white dots are rendered dots located at the 3D grid point locations. The dots were rendered as spheres using a virtual camera with the same parameters as those computed from the real image.

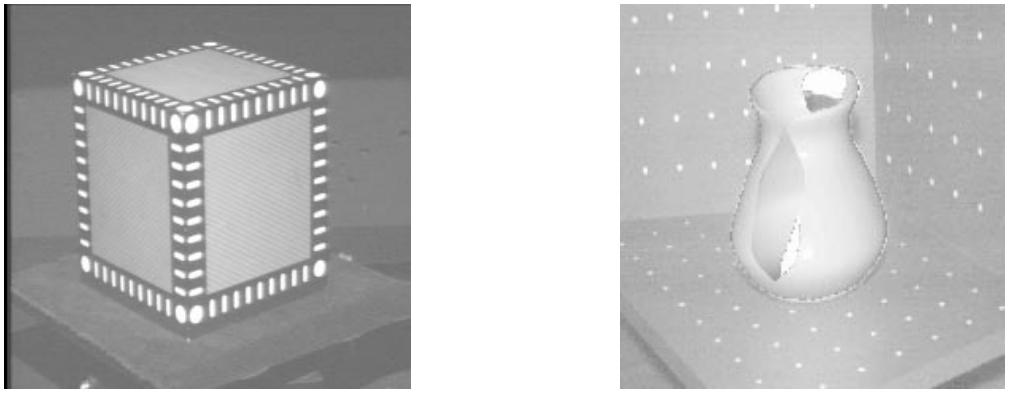


Figure 6: More sample calibration targets. The left image is the calibration cube from [17], right is an image of the MOVI (INRIA) "inverted cube" calibration pattern from [18].

Figure 5 shows a sample calibration image used in the Tsai-derived calibration procedure developed by Willson [16], and used in this work as well. The target is a grid 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 5. This projection back into 2D agrees nicely with the original data.

Figure 6 shows some more calibration targets. While CMU's Calibrated Imaging Laboratory (CIL) target from Figure 5 must be imaged at several locations to sweep out a 3D volume, these other targets have inherent 3D structure. Shum's cube [17] (leftmost in Figure 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 the target. The MOVI "inverted cube" target [18] 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 6 is a set of precisely-machined metal plates, and the CIL target in Figure 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.

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 [19] [16] [20], some use correspondences between images to compute the stereo epipolar geometry without complete Euclidean knowledge [21] [22], and others control or restrict camera motion instead [23] [24] [25]. We will summarize here the requirements for the first type of procedure (e.g., the one described in [16, Chapter 5]) for acquiring Euclidean geometry without restricting camera motion, and then extend it to include the acquisition of arbitrary ground truth in Section 6.

5.2.1 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 7 along with a range frame (explained in Section 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. 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.

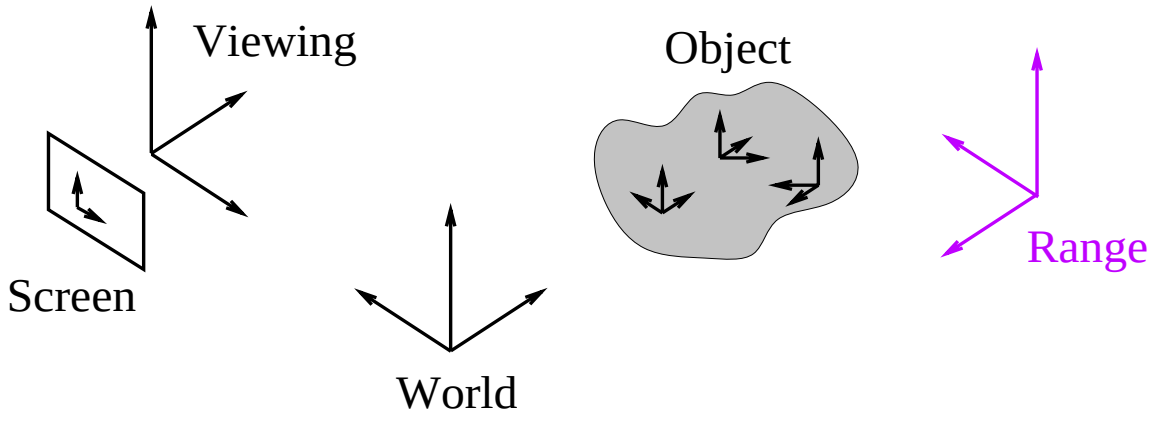


Figure 7: Coordinate frames used in dataset acquisition.

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

5.2.2 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 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 5.1 all use circular feature points for which the centroid can be determined to subpixel precision [16, 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 6.

5.2.3 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 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.

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 [26]) or a single camera platform with highly reliable positioning capabilities (e.g., [16, Appendix A]. In addition to imagery, completely dense ground truth will also be acquired for each camera.

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 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 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.

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. 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 [27], 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 [28] and [29] 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.

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 5, but the additional requirements and some example datasets are described below.

6.1.1 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).

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.

6.1.2 Define and Register Coordinate Frames

As in Section 5.2.1, 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 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.

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.

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 [30]. This dataset consists of one stereo image pair, from which an image is reproduced in Figure 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

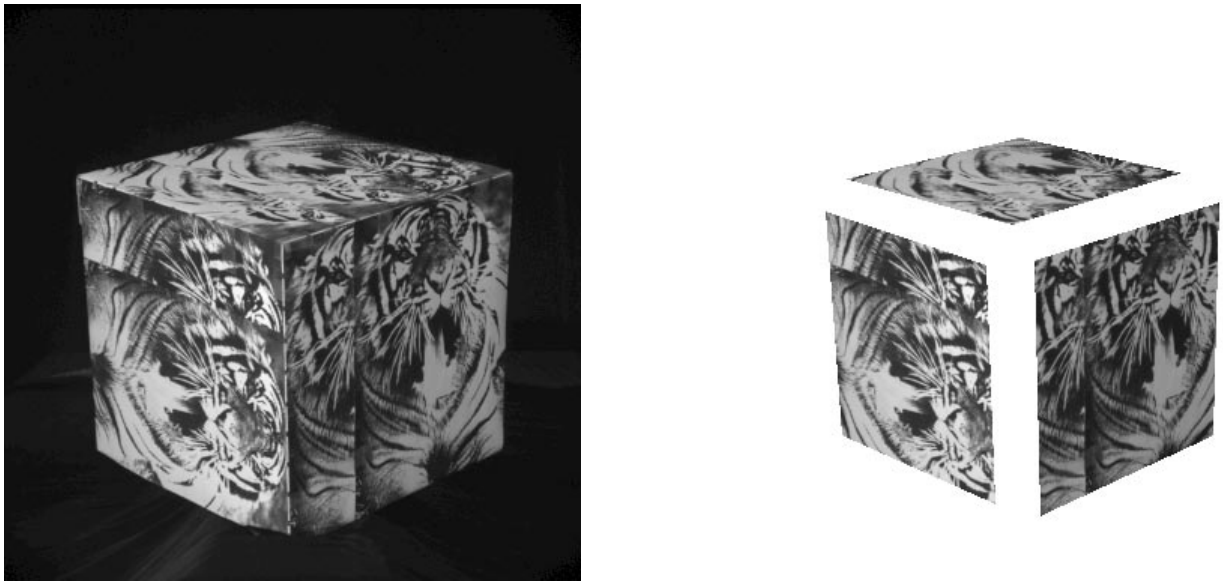


Figure 8: Textured cube image and the piecewise-planar patches used by Xiong for error analysis [30, Figure 3.20]. Used with permission.

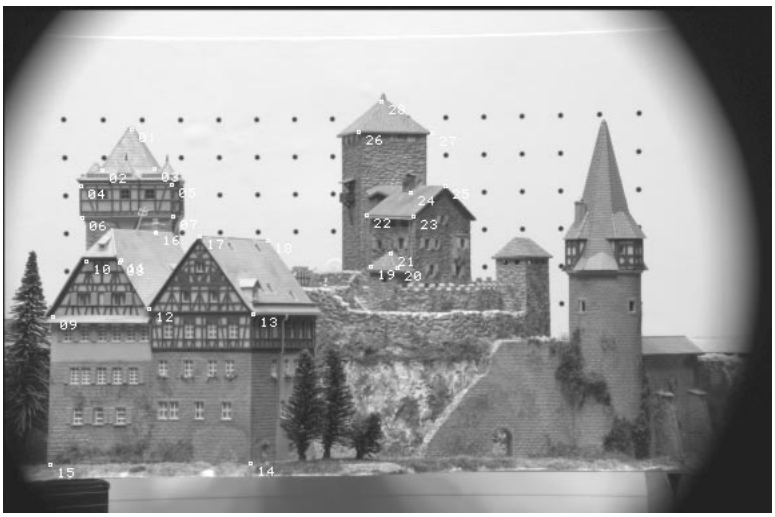


Figure 9: Image from CIL-0001 dataset with the locations of ground truth measurements.

quantify his algorithm's ability to recover shape information from stereo disparity, using the residuals from the planar fit as the error measure.

6.4 Example 6: Model Train Set

Figure 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 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 the following section.

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

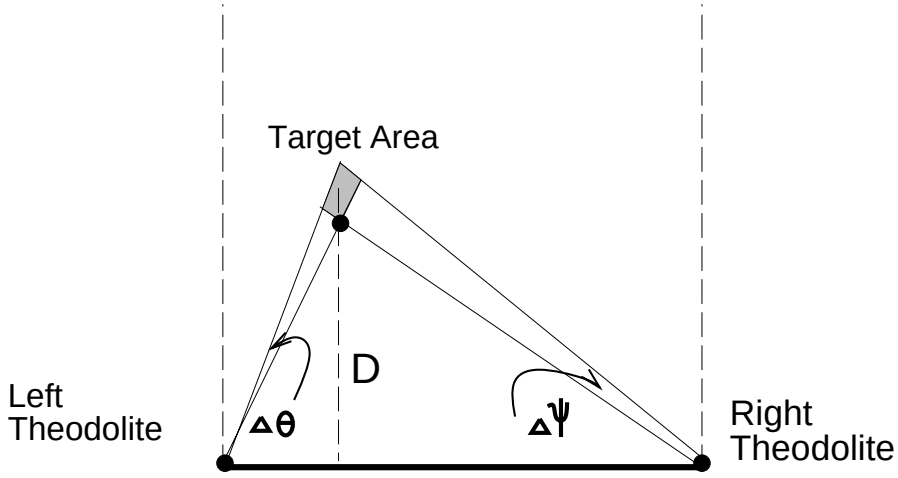


Figure 10: Region of error. The greatest possible error occurs across one of the Target Area diagonals (see Figure 12 for a close up).

6.5 Theodolite Error Analysis

Both stereo disparity and ground truth measurements have finite precision which should be made explicit. As a first step toward extending our notion of ground truth to include this precision, we present in this section an analysis of the resolution obtainable using surveyor's theodolites in their present configuration as part of the Calibrated Imaging Laboratory (CIL).

The Calibrated Imaging Laboratory theodolites [31] can repeatably measure angles to within about 20 seconds of arc. That is, during a single test run, an individual can repeatedly aim the theodolite site at a target, unlock it, then aim again and be confident that the difference between successive measurements will never be more than 20 seconds.

We would like to know how accurate subsequent **X-Y-Z** computations can be, under this limitation. A simple two-dimensional (**X-Z**) analysis will give us a rough idea of the magnitude of the precision in the horizontal plane. Figure 10 shows the overall model: depending the angles measured, the computed depth D might lie anywhere within the shaded region. Since that region is polygonal, we know that the largest possible error (i.e., the maximum distance between any two points in the region) will occur between the endpoints of one of its two diagonals. Just how long are the diagonals? To determine that, we need to derive equations for the horizontal and depth coordinates.

6.5.1 Deriving equations for the coordinate axes

Figure 11 shows the geometry of the scene. We will treat the two theodolites (as well as the target) as points. θ is the angle measured by the left theodolite, ψ is that measured by the right. B is the length of the baseline between the two theodolites; the baseline is split in two at the projection of the target point: $B = L + R$. D is the distance from the baseline to the target point. If we define the left theodolite to be the origin of a coordinate system with horizontal axis along the baseline, we have D as the vertical *depth* coordinate, and L as the horizontal coordinate.

Now we need to express D and L as functions of the two angles and baseline alone. The left and right angles bear a simple relationship to the two baseline components:

$$L = D \tan\theta \qquad \textit{and} \qquad R = D \tan\psi \tag{3}$$

Recalling that L and R sum to the baseline B, we have by substitution:

$$B = L + R = D\,(\tan\theta + \tan\psi)$$

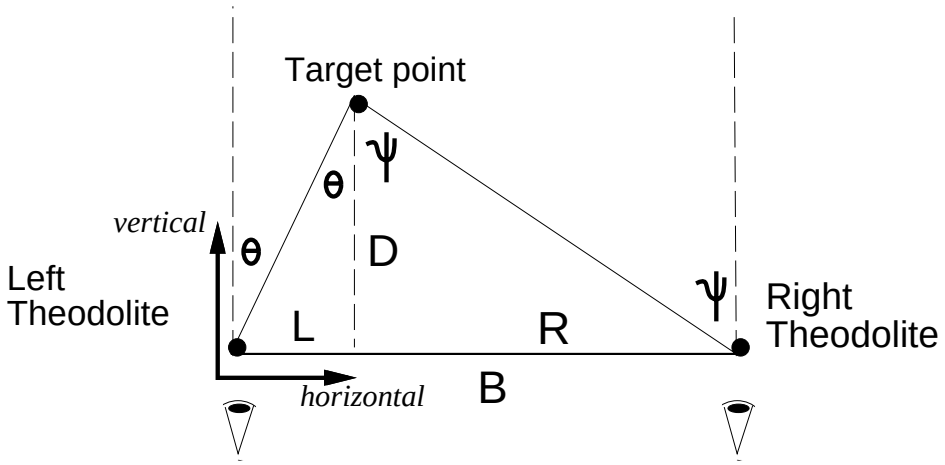


Figure 11: Two dimensional view of the theodolite imaging process.

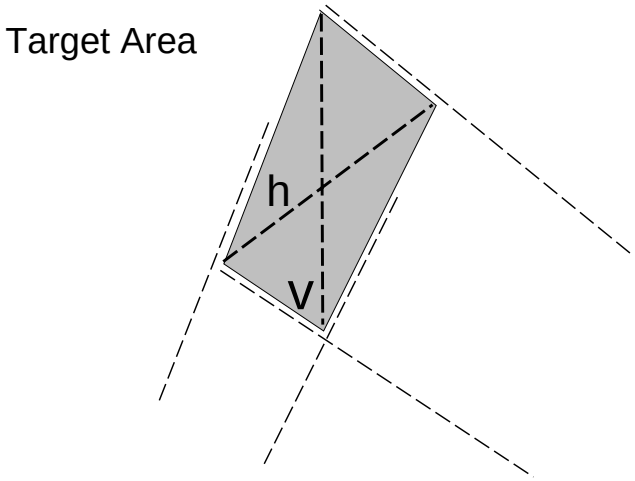


Figure 12: Target Area Precision (zoom in on Figure 10): the largest error is the length of one of the diagonals h and v.

Which gives us a solution for the depth coordinate D that depends only on angles θ, ψ and the baseline:

$$D = \frac{B}{\tan\theta + \tan\psi} \tag{4}$$

To solve for the horizontal coordinate L, simply plug this expression for D back into Equation 3:

$$L = \frac{B}{1 + \frac{\tan(\psi)}{\tan(\theta)}} \tag{5}$$

So now we have both the horizontal coordinate L and depth coordinate D. Without loss of generality, we assume the baseline is a constant factor and write $D = d(\theta, \psi)$ and $L = l(\theta, \psi)$. Now we can compute the lengths of the two diagonals in the Target Area; the larger one will give us the maximum possible error.

To compute the lengths of the diagonals h and v (shown in Figure 12), we find the Euclidean distance between their endpoints. Call the theodolite measurement error δ: for the CIL theodolites δ is $20''$. Then the length of the "horizontal" diagonal h (it's not really horizontal) is:

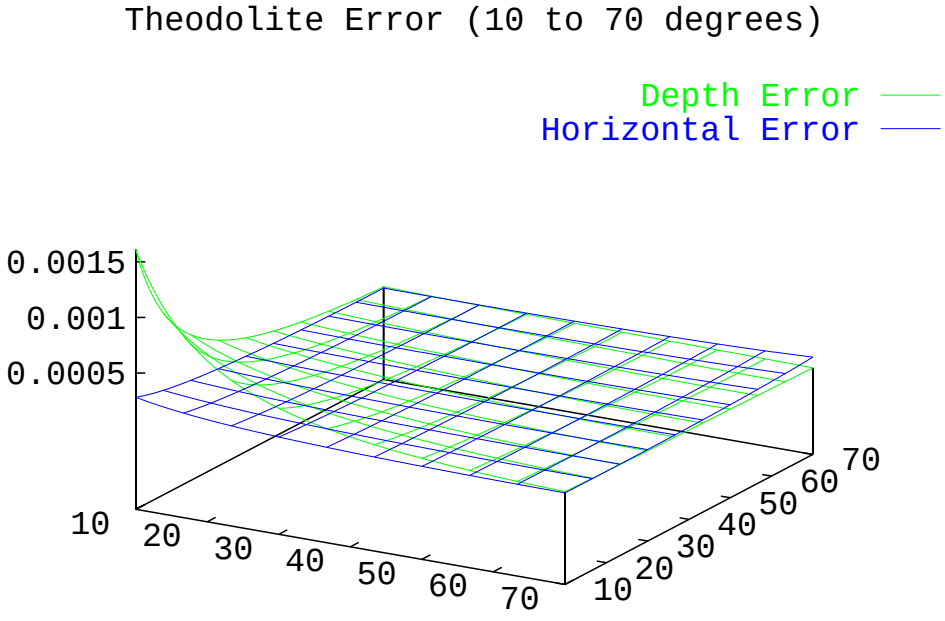


Figure 13: Error Space for $\delta = 20''$ with unit baseline: depth error $v(\theta, \psi)$ is greater than horizontal error $h(\theta, \psi)$ for all but the nearest points (when the sum of the angles is greater than about 90 degrees)

$$h(\theta, \psi) = \sqrt{(d(\theta, \psi - \delta) - d(\theta - \delta, \psi))^2 + (l(\theta, \psi - \delta) - l(\theta - \delta, \psi))^2} \tag{6}$$

The vertical diagonal v (it's not really vertical) is computed in the same way:

$$v(\theta, \psi) = \sqrt{(d(\theta - \delta, \psi - \delta) - d(\theta, \psi))^2 + (l(\theta - \delta, \psi - \delta) - l(\theta, \psi))^2} \tag{7}$$

Now we're done; the length of the horizontal diagonal is the maximum error in the horizontal direction, the vertical diagonal is the maximum error in depth. All that remains is to plug in the measurement error $\delta = 20''$.

6.5.2 Results

In the current laboratory configuration, the theodolite measurements will vary from 10 to 70 degrees, if all objects of interest lie on or above the object optical table. Figure 13 shows the shape of the error surface for this configuration, with $\delta = 20''$ and assuming a unit baseline. How do we interpret this?

The largest error in Figure 13 is about 0.0015, when both theodolites have angles of 10°. What does this really mean? Assuming a baseline of 86.1 inches (219cm), it means an object 6.21m away can only by measured to within 3.3mm.[2] However, the far end of the optical table is only about 3m away from the baseline, and according to the model the center point on the far edge gives angles of about 20° for each theodolite. The error for those angles is 0.0004, which means the best precision for the far end of the table is 0.876mm= $0.0004 \cdot 219$cm. Thus it's safe to say our theodolite measurements over the optical table are in general accurate to within a millimeter.

Nearer measurements have better accuracy. For example, in the **CIL-0001** Stereo Dataset the left horizontal angles range from 41 to 50 degrees, and right angles from 31 to 42 degrees. The maximum error in that range is $0.296mm = 0.000135 \cdot 219cm$.

[2] 6.21m= $\frac{2.19\text{m}}{\tan 10° + \tan 10°}$ and 3.3mm= 219cm·0.0015

It is in fact possible to compute the precision at each point. Simply plug the angles measured into the error terms defined above. Or to estimate the precision for a whole region, run the code in Figure 14 through the GNUplot package with appropriate limits (instead of 10° to 70°), and visually pick out the largest error.

```
d2r = pi / 180.0
hor(x,y) = 1/(1+tan(x * d2r)/tan(y * d2r))
ver(x,y) = 1/(tan(x * d2r) + tan(y * d2r))
d = 20.0/3600
splot [10:70] [10:70] \
      sqrt((hor(x-d,y-d)-hor(x,y))**2 + \
           (ver(x-d,y-d)-ver(x,y))**2) title "Depth Error", \
      sqrt((hor(x-d,y)-hor(x,y-d))**2 + \
           (ver(x-d,y)-ver(x,y-d))**2) title "Horizontal Error"
```

Figure 14: GNUplot commands that generated Figure 13

Limitations: The acquisition of ground truth requires that static objects be imaged in a controlled environment. However there are many applications for which dynamic imagery is required, and stereo systems must be tested using comparable data. The requirements for these data imply that no ground truth will be available.

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.[3]

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 performed in our laboratory. Both synthetic and real imagery were used to develop algorithms, debug their implementations, and characterize their performance. Several examples are given throughout this paper.

Rayshade, a freeware ray tracing program, provided the foundation for the synthetic datasets used throughout this paper. 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.[4]

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 [16] 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 general research community: high-quality images with piecewise-dense ground truth.[1] Publications by other researchers using these datasets have already appeared in peer-reviewed publications, e.g., [32].

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

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

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.

10 Acknowledgements

We gratefully acknowledge the contributions of Paul Heckbert and Henry Rowley to the discussion of occlusion masks, and the suggestions of Yalin Xiong, Chris McGlone and our anonymous reviewers to the overall paper.

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.

References

[1] M. Shaw, "Prospects for an engineering discipline of software," Tech. Rep. CMU-CS-90-165, Carnegie Mellon University Computer Science Department, Sept. 1990. Also IEEE Software, Nov 1990.

[2] L. Matthies, *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University Computer Science Department, Oct. 1989.

[3] R. C. Bolles, H. H. Baker, and M. J. Hannah, "The JISCT Stereo Evaluation," in *ARPA Image Understanding Workshop*, pp. 263–274, Apr. 1993.

[4] E. Gülch, "Results of test on image matching of ISPRS WG III/4," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 46, pp. 1–18, 1991.

[5] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin, "Phase-based disparity measurement," *CVGIP: Image Understanding*, vol. 53, pp. 198–210, Mar. 1991.

[6] T. D. Sanger, "Stereo disparity computation using gabor filters," *Biological Cybernetics*, vol. 59, pp. 405–418, 1988.

[7] J. Weng, "Image matching using the windowed Fourier phase," *International Journal of Computer Vision*, vol. 11, pp. 211–236, Dec. 1993.

[8] M. W. Maimone and S. A. Shafer, "Modeling foreshortening in stereo vision using local spatial frequency," in *International Robotics and Systems Conference (IROS)*, pp. 519–524, IEEE Computer Society Press, Aug. 1995. `http://www.ius.cs.cmu.edu/project/cil/fore/tr.html`.

[9] M. W. Maimone, *Characterizing Stereo Matching Problems using Local Spatial Frequency*. PhD thesis, Carnegie Mellon University Computer Science Department, May 1996. `http://www.cs.cmu.edu/~mwm/thesis`.

[10] E. Haines, "Free ray tracer summary," *Ray Tracing News*, vol. 6, p. article 4, Sept. 1993. `ftp://ftp-graphics.stanford.edu/pub/Graphics/RTNews/html/rtnv6n3.html#art4`.

[11] L. Williams, "Casting curved shadows on curved surfaces," in *ACM Computer Graphics SIGGRAPH*, pp. 270–274, 1978.

[12] D. G. Jones, *Computational Models of Binocular Vision*. PhD thesis, Stanford University, Aug. 1991.

[13] W. T. Reeves, D. H. Salesin, and R. L. Cook, "Rendering antialiased shadows with depth maps," in *ACM Computer Graphics SIGGRAPH*, pp. 283–291, 1987.

[14] Photometrics, Ltd., 3440 E. Britannia Drive, Tucson, AZ 85706, *Charge-Coupled Devices for Quantitative Electronic Imaging*, 1990.

[15] M. R. M. Jenkin and A. D. Jepson, "Recovering local surface structure through local phase difference measurements," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 59, pp. 72–93, Jan. 1994.

[16] R. G. Willson, *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Carnegie Mellon Electrical and Computer Engineering, Jan. 1994.

[17] H.-Y. Shum, M. Hebert, K. Ikeuchi, and R. Reddy, "An integral approach to free-form object modeling," Tech. Rep. CMU-CS-95-135, Carnegie Mellon University Computer Science Department, May 1995.

[18] E. Boyer and M. O. Berger, "3D Surface Recostruction Using Occluding Contours," in *International Conference on Computer Analysis of Images and Patterns*, Sept. 1995.

[19] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 323–344, Aug. 1987. `http://www.cs.cmu.edu/~rgw/TsaiCode.html`.

[20] A. Bani-Hashemi, "A Fourier Approach to Camera Orientation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1197–1202, Nov. 1993.

[21] Z. Zhang, O. Faugeras, and R. Deriche, "Calibrating a binocular stereo through projective reconstruction using both a calibration object and the environment," in *Europe-China Workshop on Geometrical modelling and Invariance for Computer Vision*, pp. 253–260, Apr. 1995.

[22] O. D. Faugeras and G. Toscani, "The calibration problem for stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15–20, 1986.

[23] R. I. Hartley, "Self-calibration from multiple views with a rotating camera," in *European Conference on Computer Vision*, pp. 471–478, 1994.

[24] D. E. Stevenson and M. M. Fleck, "Robot aerobics: Four easy steps to a more flexible calibration," in *International Conference on Computer Vision*, pp. 34–39, 1995.

[25] G. P. Stein, "Accurate internal camera calibration using rotation, with analysis of sources of error," in *International Conference on Computer Vision*, pp. 230–236, 1995.

[26] B. Ross, "A practical stereo vision system," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 148–153, 1993.

[27] M. Hebert and E. Krotkov, "3D measurements from imaging laser radars: How good are they?," *Intl. Journal of Image and Vision Computing*, vol. 10, pp. 170–178, Apr. 1992.

[28] S. Tada, A. Gruss, and T. Kanade, "CMU very fast range-imaging system," Tech. Rep. CMU-CS-93-179, Carnegie Mellon University Computer Science Department, Oct. 1993.

[29] S. K. Nayar, M. Watanabe, and M. Noguchi, "Real-time focus range sensor," in *International Conference on Computer Vision*, pp. 995–1001, 1995.

[30] Y. Xiong, *High Precision Image Matching and Shape Recovery*. PhD thesis, Carnegie Mellon Robotics Institute, Sept. 1995.

[31] Sokkisha Co., Ltd., Keio Yoyogi Building 5th Floor, No. 1, 1, 1-chome, Tomigaya, Shibuta-ku, Tokyo, 151 Japan, *Electronic Digital Theodolite DT20E Operation Manual*, 1984.

[32] Z. Wang and A. Jepson, "A new closed-form solution for absolute orientation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 129–134, 1994.