Variable Baseline/Resolution Stereo

David Gallup[1], Jan-Michael Frahm[1], Philippos Mordohai[2], Marc Pollefeys[1,3]
[1]University of North Carolina [2]University of Pennsylvania [3]ETH Zurich
Chapel Hill, NC Philadelphia, PA Zurich, Switzerland
{gallup, jmf}@cs.unc.edu, mordohai@seas.penn.edu, marc.pollefeys@inf.ethz.ch

Abstract

We present a novel multi-baseline, multi-resolution stereo method, which varies the baseline and resolution proportionally to depth to obtain a reconstruction in which the depth error is constant*. This is in contrast to traditional stereo, in which the error grows quadratically with depth, which means that the accuracy in the near range far exceeds that of the far range. This accuracy in the near range is unnecessarily high and comes at significant computational cost. It is, however, non-trivial to reduce this without also reducing the accuracy in the far range. Many datasets, such as video captured from a moving camera, allow the baseline to be selected with significant flexibility. By selecting an appropriate baseline and resolution (realized using an image pyramid), our algorithm computes a depthmap which has these properties: 1) the depth accuracy is constant over the reconstructed volume, 2) the computational effort is spread evenly over the volume, 3) the angle of triangulation is held constant w.r.t. depth. Our approach achieves a given target accuracy with minimal computational effort, and is orders of magnitude faster than traditional stereo.*

1. Introduction

Stereo is a well-studied problem in computer vision [14]. Recent work has been very successful in solving the correspondence problem, which is to decide which pixels in one image correspond to which pixels in another. Techniques employing graph cuts and belief propagation can achieve error rates of less than 1% (on laboratory data). However, for many applications the goal is ultimately not pixel correspondence but depth accuracy. Even with perfect correspondences, the depth error in traditional stereo grows *quadratically* with depth, which means that the accuracy in the near range far exceeds that of the far range. While the accuracy in the far range is unusably bad, the accuracy in the near range is unnecessarily high and comes at significant computational cost. Accuracy can be improved by incorporating multiple views. These views provide additional information which aids in the correspondence problem, but they can also improve the depth accuracy *geometrically* by increasing the angle of triangulation. In many applications, such as structure from motion from video [12], or recently reconstruction from community photo collections [4], the choice of views for stereo is quite flexible. Our technique focuses on selecting the best cameras, as well as the most appropriate sampling in the images, to compute a depthmap that meets the desired geometric accuracy with minimal computation. Specifically, we increase the baseline to increase accuracy in the far range, and we reduce the resolution (using a gaussian pyramid) to reduce computational effort in the near range. Additionally our novel algorithm is compatible with most matching and optimization strategies, and will work with any higher level post-processing typically used in stereo, *e.g.* depth fusion [10, 19].

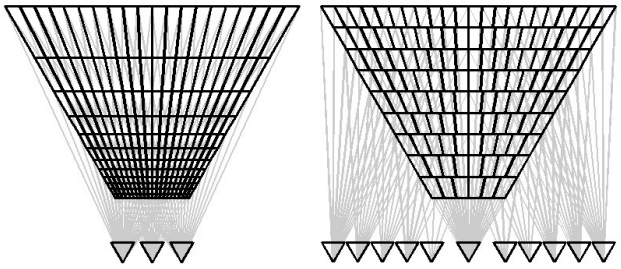


Figure 1. *Left*: Standard stereo. Note that the distance between depths increases quadratically. *Right*: Variable Baseline/Resolution Stereo. The distance between depths is held constant by increasing the baseline and selecting the appropriate resolution.

Many applications today require accurate 3D models of real-world scenery. For example, mapping applications such as Google Earth and Microsoft Virtual Earth have recently incorporated textured 3D models of cities. These models are typically extracted from aerial and satellite images and lack ground level detail. Several research projects aim to produce 3D reconstructions of cities from photographs or video acquired from ground level. The amount of image data required to observe an entire city can be enormous. Therefore, processing speed, as well as accuracy, is an important consideration. Furthermore, scenes captured

from ground level typically exhibit a large depth range. Standard stereo is ill-suited to such scenes since the error grows quadratically with depth.

The motivation for our approach derives from the point of view of a system designer wishing to employ stereo as a measuring device. Often, the definition of the stereo problem assumes the camera parameters are given and fixed, but these parameters have a significant effect on the depth accuracy of stereo. The system designer has some accuracy requirements in mind and, with traditional stereo methods, must carefully select baseline, focal length, and field of view in order to meet these requirements. Furthermore, computation time is important in real systems, and so the designer must be conservative. It is unacceptable to spend large amounts of time obtaining accuracy that far exceeds the minimum requirement. Balancing accuracy and efficiency for standard stereo is difficult indeed due to its quadratic error characteristics. Our novel algorithm enhances stereo to be able to efficiently use the additional information contained in dense image sets such as video by dynamically selecting the appropriate baseline and image scale for each depth estimate. In contrast to traditional stereo our technique guarantees a constant target accuracy throughout the maximal possible volume with orders of magnitude less computational effort.

1.1. Variable Baseline/Resolution Stereo

The motivation for our algorithm is derived from the depth error in stereo, which can be shown to be

$$\epsilon_z = \frac{z^2}{bf} \cdot \epsilon_d \tag{1}$$

where ϵ_z is the depth error, z is the depth, b is the baseline, f is the focal length of the camera in pixels, and ϵ_d is the matching error in pixels (disparity values). Dense image sets, such as video, allow the baseline b to be selected with great flexibility, and because f is measured in pixels, the focal length can be varied by selecting the appropriate scale in a gaussian pyramid (up to the maximum value of f at full resolution). The principal idea of our algorithm is to set both b and f proportionally to z throughout the depthmap computation, thereby canceling the quadratic z term, and leaving ϵ_z *constant* w.r.t. depth. Thus matching scores for depths in the near range are computed using a narrow baseline and coarse resolution, while depths in the far range use a wider baseline and finer resolution.

Our algorithm, which we call Variable Baseline/Resolution Stereo, exhibits three important properties:

1. By selecting the baseline and resolution proportionally to the depth, we can match the quadratic term in the depth error, and achieve constant accuracy over the reconstructed volume.
2. Because the accuracy is constant throughout the reconstructed volume, the computational effort is also evenly spread throughout the volume.
3. The baseline grows linearly with depth, therefore the angle of triangulation remains constant[1].

To the best of our knowledge, our method is the first to exhibit all three properties.

In the following sections we consider previous work, analyze the error and time complexity of our algorithm as compared to traditional stereo, discuss the implementation of our algorithm in detail, and present results.

2. Previous Work

Early research on multi-baseline stereo includes the work of Okutomi and Kanade [11] who use both narrow and wide baselines, which offer different advantages, from a set of cameras placed on a straight line with parallel optical axes. A layer-based approach that can handle occlusion and transparency was presented by Szeliski and Golland [16]. The number of layers, however, is hard to estimate a priori and initialization of the algorithm is difficult. Kang et al [6] explicitly address occlusion in multi-baseline stereo. For each pixel of the reference view a subset of the cameras with the minimum matching cost is selected under the assumption that the pixel may be occluded in the other images. We also use this scheme in our approach. Sato et al. [13] address video-based 3D reconstruction using hundreds of frames for each depth map computation. The median SSD between the reference and all target views is used for robustness against occlusions. In general, using multiple images improves matching, and employing wider baselines can increase the depth accuracy. Our approach is unique in that we use a different set of images throughout the computation such that the baseline grows proportionally to depth.

Multi-resolution approaches are used for stereo to either speed up computation or to combine the typically less ambiguous detection at coarse resolution with the higher precision of fine resolution. The latter was the motivation for the approach of Falkenhagen [3] in which disparities are propagated and refined as processing moves from coarse to fine levels of image pyramids. Yang and Pollefeys [18] presented an algorithm in which cost functions from several different resolutions were blended to take advantage of the reduced ambiguity coming from matching at coarse levels of the image pyramids and the increased precision coming from matching at fine levels. Koch et al. [7] use a multi-resolution stereo algorithm to approximately detect the surfaces quickly, since processing speed is important for large scale reconstruction systems which operate on large disparity ranges. Reducing image resolution results in an equiv-

[1] The angle of triangulation is constant w.r.t. to depth. It varies slightly from pixel to pixel.

alent reduction of the disparity range. Sun [15] presented a method that aims at improving both the speed and reliability of stereo. It operates in bottom-up fashion on an image pyramid in which stripes are adaptively merged to form rectangular regions based on disparity similarity. A two-stage dynamic programming optimization stage produces the final depth map. In these approaches, multiple resolutions are used for speed and/or improved matching, but depth accuracy is not addressed. A key component of our algorithm is that we use different resolutions at different depths. Specifically, we use lower resolutions to estimate depths in the near range in order to avoid unnecessary computations for accuracy that far exceeds what is required.

Algorithms that take geometric uncertainty explicitly into account include [9] and [7]. Matthies *et al*. [9] introduced an approach based on Kalman filtering that estimates depth and depth uncertainty for each pixel using monocular video inputs. These estimates are refined incrementally as more frames become available. Koch et al. [7] proposed a similar approach that computes depth maps using pairs of consecutive images. Support for each correspondence in the depth maps is found by searching adjacent depth maps both forward and backward in the sequence. When a match is consistent with a new camera, the camera is added to the chain that supports the match. The position of the reconstructed 3D point is updated using the wider baseline. While these methods are successful in reducing the error in the reconstruction, they do not exhibit the properties from Section 1.1. In particular, the computational effort is concentrated in the near range, and as a result the depth accuracy in the near range exceeds that of the far range.

3. Analysis

Before analyzing the accuracy and time complexity of our stereo algorithm, we shall briefly address the issue of depth sampling. Stereo seeks to determine the depth of the surface along the rays passing through each pixel in a reference image. Each point along the ray is projected into any number of target images, and a measure of photoconsistency is computed. This defines a function in depth, the minimum (or maximum) of which indicates the depth of the surface and is discovered by sampling the function at various depths. The number and location of the samples should be defined by the pixels in the target images (disparities). While supersampling can obtain a more accurate minimum, the minimum itself does not necessarily accurately locate the surface. Because the frequency content of the images is limited by the resolution, the photoconsistency function is also limited, and thus the surface can only be localized to within one pixel. (Sub-pixel accuracy up to 1/4 pixel is possible, but the depth accuracy is still proportional to the pixels.) Note also that subsampling the function without properly filtering the images will lead to aliasing, and the minimum of the aliased function can often be far from the minimum of the original function. Therefore the sampling rate must be on the order of one pixel. In stereo, one cannot expect to obtain greater depth accuracy simply by finer disparity sampling, and in order to use coarser sampling (to reduce computation time and accuracy), filtered lower-resolution images must be used. For more details on sampling in stereo, see [17].

3.1. Accuracy and Time Complexity

We now analyze the time complexity of traditional stereo and compare it to the time complexity of our variable baseline/resolution algorithm. Our analysis assumes that the cameras are separated by lateral translation and no rotation, so that all cameras share a common image plane, and pixel correspondences have the same vertical image coordinate. This setup, which is convenient for analysis, can be somewhat relaxed in the actual implementation of our algorithm.

In our analysis we assume that the system designer specifies a desired accuracy: a maximum error ϵ_z, and a maximum range z_{far}. Stereo is expected to deliver depth measurements with error less than ϵ_z for all depths $z \leq z_{far}$.

Consider two cameras with focal length f separated by distance b. Let d be the difference in x coordinates, called disparity, of two corresponding pixels. The depth z of the triangulated point is given by $z = \frac{bf}{d}$ (where f is measured in pixels). The depth error can be written in terms of the disparity error ϵ_d:

$$\epsilon_z = \frac{bf}{d} - \frac{bf}{d+\epsilon_d} = \frac{z^2\epsilon_d}{bf + z\epsilon_d} \approx \quad \tfrac{z^2}{bf} \cdot \epsilon_d. \qquad (2)$$

The final step is obtained by taking the first order taylor series approximation about $\epsilon_d = 0$.

Here we separate the error into two factors: *correspondence error*, ϵ_d, and *geometric resolution*, $z^2/(bf)$. Geometric resolution describes the error in terms of the geometry of the stereo setup, namely baseline, focal length, and depth. We see that geometric resolution is *quadratic* in depth. Correspondence error describes the error from incorrect matches and the sub-pixel accuracy of the correct matches. In stereo, correspondence error depends on image noise, scene texture, and other scene properties such as occlusions and non-Lambertian surfaces. In this paper we focus on geometric resolution, and assume that the number of incorrect matches is reasonably low, and that matching accuracy is bounded to within one pixel. Therefore, meeting our target error bound ϵ_z at depth $z \leq z_{far}$ depends on the baseline and focal length of the cameras.

We will now analyze the effect of baseline and focal length separately, then combined, followed by an analysis of our algorithm. We focus our analysis on the target accuracy parameters ϵ_z and z_{far}.

Fixed-baseline stereo. For a fixed-baseline stereo system, the accuracy can be adjusted by varying the focal length parameter f. Since f here is measured in pixels, it can be increased either by narrowing the field of view (zoom), or by increasing the resolution of the sensor. We assume the field of view θ_{fov} has been carefully chosen for the application, meaning f describes the resolution as

$$w = 2f \tan \frac{\theta_{fov}}{2} \qquad h = \frac{w}{a}, \tag{3}$$

where w, h and a are the width, height and aspect ratio of the image. We can determine the resolution needed to meet the target accuracy by solving for f in equation (2). (Here we have assumed $\epsilon_d = 1$.)

$$f = \frac{{z_{far}}^2}{b\epsilon_z} \tag{4}$$

$$\begin{aligned} \textit{number of pixels} &= wh = \frac{w^2}{a} \\ &= \frac{{z_{far}}^4}{{\epsilon_z}^2} \frac{4 \tan^2 \frac{\theta_{fov}}{2}}{b^2 a} \end{aligned} \tag{5}$$

This shows that increasing the resolution alone to meet the target accuracy requires the image resolution to grow proportionally to ${z_{far}}^4$! Note that a higher resolution sensor does not necessarily increase the effective resolution. Higher quality lens optics may also be required, making it prohibitively expensive, or impossible, to increase the resolution at this rate. Another prohibitive factor is the processing time. In stereo, each pixel must be tested against the pixels along the corresponding epipolar line within the disparity range of the scene. Because the *depth* range is defined by the scene, the *disparity* range is some fraction of the image width, and thus increases with resolution. Letting D be the ratio of the disparity range to the image width, the number of pixel comparisons needed is

$$\begin{aligned} T_{\textit{fixed}} &= Dw^2 h = \frac{Dw^3}{a} \\ &= \frac{{z_{far}}^6}{{\epsilon_z}^3} \frac{8D \tan^3 \frac{\theta_{fov}}{2}}{b^3 a} \\ &= O({z_{far}}^6 {\epsilon_z}^{-3}). \end{aligned} \tag{6}$$

This means the system designer is severely limited by depth range. For example, extending the depth range by a factor of 2 would require $2^6 = 64$ times more computational effort!

Fixed-resolution stereo. If the resolution is held fixed, the depth error can only be reduced by increasing the baseline b. To meet the target accuracy, we solve equation (2) for b, yielding $b = \frac{{z_{far}}^2}{f\epsilon_z}$. One drawback of increasing the baseline is that the depth where the fields of view begin to overlap also increases, and the near range is lost. The depth where the overlap begins is $z_{near} = \frac{b}{\tan \theta_{fov}/2}$. Because z_{near} depends on b, and b grows quadratically with z_{far}, there is a point at which z_{near} surpasses z_{far}, meaning that the depth where the target accuracy is met is no longer in the overlapping field of view. In general, one cannot rely on increasing the baseline alone to meet the target accuracy.

Variable baseline and resolution. In order to avoid z_{near} surpassing z_{far}, the baseline cannot grow faster than linearly with z_{far}. Thus we set $b = \beta z_{far}$ where β can be chosen to give a certain angle of triangulation at z_{far}. Given this constraint, we solve for the resolution needed to meet the target accuracy as follows:

$$f = \frac{{z_{far}}^2}{b\epsilon_z} = \frac{z_{far}}{\beta\epsilon_z} \tag{7}$$

$$\textit{number of pixels} = \frac{{z_{far}}^2}{{\epsilon_z}^2} \frac{4 \tan^2 \frac{\theta_{fov}}{2}}{\beta^2 a}. \tag{8}$$

From this equation we see that the baseline and the focal length both grow linearly with z_{far}, and the required resolution grows proportionally with ${z_{far}}^2$ rather than ${z_{far}}^4$. However, with a linearly growing baseline, z_{near} also grows linearly, and overlap in the near range is lost. Therefore, in order to accurately reconstruct the entire scene, wide baselines must be used in the far range, and narrow baselines must be used in the near range.

We now analyze our method which uses multiple baselines and resolutions to recover depths over the entire viewing volume with minimal computational effort. Unlike previous approaches which combine measurements among multiple baselines and resolutions, our method chooses a single baseline and resolution based on the depth being measured. This approach has several advantages mentioned in Section 1.1: 1) the depth error is constant for all depths, 2) the amount of computational effort is evenly distributed throughout the volume, 3) the depth angle of triangulation does not vary with depth.

For the sake of analysis, assume that the stereo setup consists of a continuous set of cameras with baselines given by the function $\mathcal{B}(x) = xb, 0 \leq x \leq 1$, where b is the required baseline from equation (7). For each of these cameras there is an image I_x which has been constructed as a scale pyramid, again, with a continuous set of scales. The focal length (in pixels) of the scales is given by the function $\mathcal{F}(x) = xf, 0 \leq x \leq 1$, where f is the required focal length from equation (7). In reality, baselines and scale pyramid levels are discrete; however, the set of baselines acquired from a moving camera is quite dense, and the continuous scale pyramid can be approximated by filtering between the two nearest discrete levels.

Since our method uses multiple images, it is more convenient to parameterize correspondences in terms of their triangulated depth z instead of their pixel coordinate disparity d. Our approach varies the baseline and resolution with z. The baseline is chosen as $\mathcal{B}(z/z_{far})$, and the reso-

lution is chosen such that the focal length is $\mathcal{F}(z/z_{far})$. By substituting this baseline and focal length into the stereo error equation (2), we see that the error is equal to our target error ϵ_z for all $z \leq z_{far}$.

To analyze the time complexity of our algorithm, we sum the number of pixel comparisons needed at each depth z. Since we step through depth at a constant rate ϵ_z, there are z_{far}/ϵ_z steps. Letting $k\epsilon_z/z_{far}$ be the proportion of the width and height required at depth $k\epsilon_z$, the time complexity can be expressed as a sum:

$$\begin{aligned} T_{variable} &= \sum_{k=1}^{\frac{z_{far}}{\epsilon_z}} wh\left(\frac{k\epsilon_z}{z_{far}}\right)^2 = \frac{wh{\epsilon_z}^2}{{z_{far}}^2}\sum_{k=1}^{\frac{z_{far}}{\epsilon_z}} k^2 \\ &= \frac{4\tan^2\frac{\theta_{fov}}{2}}{a\beta^2}\left(\frac{{z_{far}}^3}{3{\epsilon_z}^3} + \frac{{z_{far}}^2}{2{\epsilon_z}^2} + \frac{z_{far}}{6\epsilon_z}\right) \\ &= O({z_{far}}^3{\epsilon_z}^{-3}). \end{aligned} \tag{9}$$

This is a considerable improvement over standard stereo which is $O({z_{far}}^6{\epsilon_z}^{-3})$ as shown in equation (6). Note that the reconstructed volume is a frustum (pyramid) ranging from the camera to z_{far}. If we divide this volume into voxels with side length ϵ_z, the number of voxels in the volume is also $O({z_{far}}^3{\epsilon_z}^{-3})$. Without prior knowledge, each voxel must be visited, or at least a number of voxels proportional to the volume must be visited, to reconstruct the volume. Under these assumptions, $\Omega({z_{far}}^3{\epsilon_z}^{-3})$ is the asymptotic lower bound for stereo, which our algorithm achieves.

While our analysis has focused on image-centered stereo, we briefly mention a different class of stereo, namely volumetric methods [2, 8]. By nature, the time complexity of volumetric stereo is proportional to the volume, and the computation time is spread evenly over the volume (property 2 from Section 1.1). However, these methods do not explicitly guarantee geometric accuracy. In order to do so, voxel size and camera selection must be chosen such that the projection of each voxel differs from the projection of the neighboring voxels by exactly one pixel in some camera. Assuming pixel accurate matching, this ensures that each voxel is visually distinguishable from its neighbors, and therefore the surface can be located to within one voxel in space. To the best of our knowledge, no volumetric method exists which guarantees uniform geometric accuracy over the entire volume.

We have assumed that we can step through depth at a constant rate equal to ϵ_z. However, we must ensure the proper spacing between depth steps in the image [17] in order to avoid aliasing. The projection of the tested points along the ray should be spaced *no farther than one pixel apart*, otherwise it is possible the correct match will be missed. Consider two consecutive depth samples z_1 and z_2. We shall measure the spacing of these projected depths in the image where the finest scale is used, which is the image corresponding to z_2. We denote the baseline and resolution used at z_2 as b_2 and f_2. We shall define $z_2 = z_1 + \Delta z$, and derive the spacing Δd as follows:

$$\Delta d = \frac{b_2 f_2}{z_1} - \frac{b_2 f_2}{z_1 + \Delta z} \tag{10}$$

We now replace b_2 and f_2 with the values used at z_2 (see equation (7)).

$$\begin{aligned} b_2 f_2 &= (z_1 + \Delta z)\beta \cdot \frac{z_1 + \Delta z}{\beta\epsilon_z} = \frac{(z_1 + \Delta z)^2}{\epsilon_z} \\ \Delta d &= \frac{(z_1 + \Delta z)^2}{z_1\epsilon_z} - \frac{z_1 - \Delta z}{\epsilon_z} \\ &= \frac{\Delta z}{\epsilon_z} + \frac{\Delta z^2}{\epsilon_z z_1} \end{aligned} \tag{11}$$

It is reasonable to assume that all depth hypotheses remain in front of the camera, which means we can assume the smallest depth tested is ϵ_z since it is the target accuracy. By solving for Δz such that $\Delta d \leq 1$ pixel, it can be shown that the step size is bounded as $\frac{\epsilon_z}{2} \leq \Delta z \leq \epsilon_z$. Since we step through depth at a rate bounded by constants, the time complexity of our algorithm is still $O({z_{far}}^3{\epsilon_z}^{-3})$. In practice, for all but the smallest depths, $\Delta z \approx \epsilon_z$.

4. Algorithm

We use a plane-sweeping approach to compute a depthmap for a reference view using multiple target views. Plane-sweeping tests entire depth planes by warping all views according to the plane, comparing the images to a reference view and computing a per-pixel matching score or *cost*, and storing these in a *cost volume* from which the depthmap is then extracted. For more details on plane-sweeping, see [1, 18]. Since there is no need for rectification, plane-sweeping can easily use multiple views. In our algorithm, for each depth plane z, we choose a constant number of images whose baselines are evenly spread between $-\mathcal{B}(\frac{z}{z_{far}})$ and $\mathcal{B}(\frac{z}{z_{far}})$. To handle occlusions, we select the 50% best matching scores [6]. Finally, because pixel-to-pixel matching is inherently ambiguous, additional constraints such as surface smoothness must be imposed to compute the depthmap. In our implementation we use semi-global optimization [5] which is both efficient and accurate.

In the actual implementation of our algorithm, the set of cameras and their associated image pyramids are finite and discrete, and we do not require that cameras be strictly constrained to lateral translation and no rotation. Thus we cannot compute the accuracy and pixel motion by the simple formulas previously mentioned. Instead we measure the image sampling and accuracy directly by projecting the hypothesized depths into the leftmost and rightmost views. Since our depth planes do not intersect the convex hull of

Algorithm 1 Variable Baseline/Resolution plane-sweep. The baseline and resolution increase from narrow to wide and from coarse to fine as necessary to maintain the target error bound. Error and depth step are computed by directly measuring pixel motion in the images.

$z \Leftarrow z_{near}$
while $z \leq z_{far}$ **do**
  compute matching scores for depth plane z
    and store in cost volume
  $\Delta z \Leftarrow$ compute depth step
  while error$(z + \Delta z) > \epsilon_z$ and baseline $\leq (z + \Delta z)\beta$ **do**
    increase baseline
    $\Delta z \Leftarrow$ recompute depth step
  end while
  while error$(z + \Delta z) > \epsilon_z$ and resolution $<$ finest **do**
    increase resolution (pyramid level)
    $\Delta z \Leftarrow$ recompute depth step
  end while
  $z \Leftarrow z + \Delta z$
end while
compute depthmap from cost volume

the camera centers, the projection of the image border polygon remains convex, and we only need to measure at the vertices of the projected and clipped image border polygon, which bounds the measurements of all interior points.

Algorithm 1 describes our method. The plane-sweep begins with the narrowest baseline and coarsest resolution. As the sweep moves from near to far, the baseline and resolution (pyramid scale) increase from narrow to wide and from coarse to fine as necessary to maintain the target error bound. Once the full image resolution is attained, the plane-sweep can continue with increasing baselines, but the error bound will no longer be met. Matching scores are computed at each depth and stored in a cost volume, from which an optimized surface is then extracted. Depth error and depth step are computed by measuring pixel motion directly.

In our method we compare matching costs computed at different baselines and resolutions and expect that the minimum score indicates the correct match. For multiple baselines, we expect this to be true based on the brightness constancy constraint. Although appearance changes are a known problem in wide baseline matching, our method is *not* wide baseline since the angle of triangulation is kept approximately constant and relatively small. In Figure 2, we show an example of matching costs computed from various resolutions and baselines. This figure shows that the cost minimum is roughly the same for all cost functions. We have evaluated this for a variety of scenes and pixels and found the same general behavior in all of them.

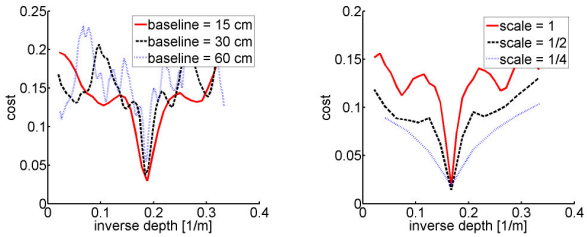


Figure 2. Matching cost functions for varying baselines (left) and resolutions(right). Cost minima are roughly the same value at different resolutions and baselines, which makes stereo matching possible across different resolutions and baselines.

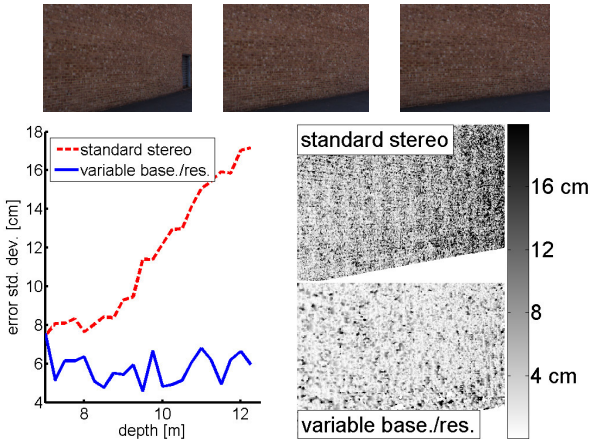


Figure 3. We compared standard stereo and our algorithm against a scene for which ground truth was acquired with a laser range finder. The depth of the wall ranges from 7 to 12 meters. *Top*: Some original images. *Bottom Left*: Standard deviation from ground truth w.r.t. depth. The error of standard stereo increases with depth while the error of our algorithm remains roughly constant. *Bottom Right*: Absolute error images (darker means larger error).

5. Results

We have evaluated our method and standard stereo against a scene for which ground truth was acquired with a laser range finder. The scene, shown in Figure 3, is simple by design, so that the focus is depth accuracy, not matching accuracy. The scene features a slanted brick wall which ranges from 7 to 12 meters in depth. For our method, we set $\epsilon_z = 10cm$ and $\beta = \tan 10°$ (*i.e.* $10°$ angle of triangulation). To evaluate error w.r.t. depth, we divided the computed depths into bins spaced $25cm$ apart, and computed the standard deviation of the (signed) difference from ground truth. As expected, the error in standard stereo grows with depth, whereas the error from our method remains constant. Note that our target accuracy is $10cm$, whereas the average standard deviation is $5cm$, from which we can deduce the standard deviation of the correspondence error ϵ_d is 0.5 pixels (see equation (2)).

Next we have evaluated our algorithm on challeng-

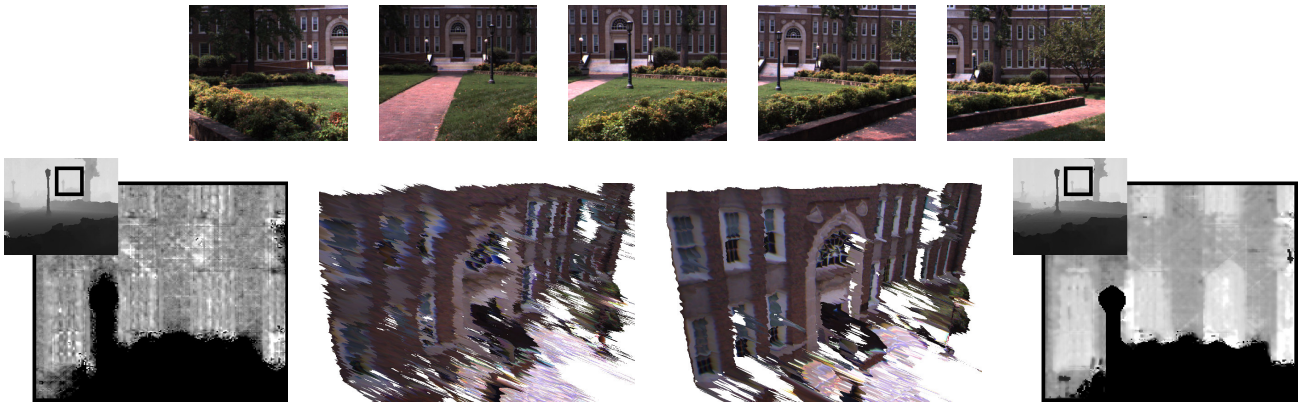
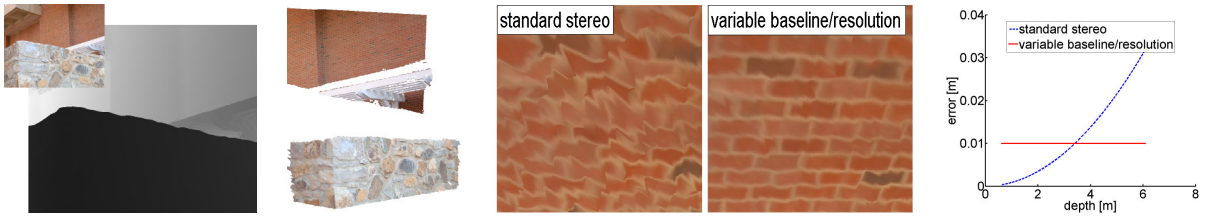


Figure 4. *First Row*: Some original images. The middle image is the reference view. *Second Row, Left*: Depthmap and 3D model views computed using standard stereo. *Second Row, Right*: Depthmap and 3D model views computed using our method. Because the correspondence accuracy is similar for both methods, the full views of the depthmaps appear similar. However, a close-up view of the standard stereo depthmap at the far range reveals the poor depth accuracy. In contrast, the close-up view of the depthmap computed using our method is much smoother, the indentations from the windows are more defined, and consequently the 3D model views are much cleaner, especially in the far range.



(a) (b) (c) (d)

Figure 5. *(a)*: The reference view image and depthmap from our method. *(b)*: 3D model view from our method. *(c)*: Close-up 3D model views of the far range for standard stereo and our method. *(d)*: Geometric depth resolution plot.

ing outdoor scenes. The first scene was acquired with a 1024x768 pixel video camera undergoing lateral motion capturing at 30 Hz. The field of view was 40 degrees. For this scene, we desired an accuracy of $\epsilon_z = 30cm$, and have found matching to be accurate at angles up to 6 degrees, *i.e.* $\beta = \tan 6°$. Given our resolution, the target accuracy can be maintained up to $45m$. We used a gaussian pyramid where the scaling factor between levels is 1/2, and filtered between the two nearest levels to handle variable resolution. Depthmaps were computed using the previously described plane-sweep, using 11 views, followed by semi-global optimization. We have compared our results with standard stereo, also using 11 views. For a fair comparison, we allowed standard stereo to use the widest baseline possible, while still keeping the objects in the near range in view. The near range in this scene is $3m$ which limits the baseline to $2.5m$. This baseline is in fact not sufficient to meet the target accuracy at the far range. Except for the differences in baseline and resolution, all other settings are the same for both methods. The two methods are compared in Figure 4. Our method is more than 6 times faster, and analysis predicts that it is more than 4 times more accurate at the far range. While no ground truth is available for this scene, the reconstruction produced using our method is clearly many times more accurate.

For the second result, shown in Figure 5, we captured a series of images with a 10 megapixel camera. We processed the images with a target accuracy of $\epsilon_z = 1cm$ at $z_{far} = 6.1m$, a maximum triangulation angle of 6°, and used 7 views for each plane. Compared to standard stereo using the widest baseline possible, our method is more than 6 times faster, and more than 3 times more accurate at the far range.

Again, we allowed standard stereo to use the widest baseline possible, so long as objects in the near range are kept in view. For the two scenes, the nearest object is 5-10% of the distance to the farthest object, which we have observed to be typical in outdoor ground-level imagery. Note that the near range has a significant effect on standard stereo, as it limits the baseline and increases disparity range. In contrast, this variable has negligible effect on our method because the baseline is variable, and because these near-range depths are processed at low resolution. For standard stereo, the resolution is insufficient to meet the target accuracy throughout the volume. In fact, the error surpasses ϵ_z at about 50% of z_{far}, and grows to nearly 3-4 times ϵ_z at z_{far}. Our method on the other hand maintains the target accuracy throughout the volume, while still performing about

	Scene 1 $z_{near} = 3m$, $z_{far} = 45m$, $\epsilon_z = 30cm$					**Scene 2** $z_{near} = 0.6m$, $z_{far} = 6.1m$, $\epsilon_z = 1cm$				
	Fixed1	**F1/V**	Fixed2	**F2/V**	Variable	Fixed1	**F1/V**	Fixed2	**F2/V**	Variable
resolution	0.8 Mp	**1**	15.2 Mp	**19.0**	0.8 Mp	10 Mp	**1**	103 Mp	**10.3**	10 Mp
# pixel comps	$3.76\text{x}10^8$	**6.16**	$3.19\text{x}10^{10}$	**523**	$6.10\text{x}10^7$	$1.75\text{x}10^{10}$	**6.65**	$5.76\text{x}10^{11}$	**219**	$2.63\text{x}10^9$
error at z_{far}	1.32m	**4.40**	0.3m	**1**	0.3m	0.032m	**3.20**	0.01m	**1**	0.01m
z where err. = ϵ_z	21.47m	**0.48**	45m	**1**	45m	3.41m	**0.56**	6.1m	**1**	6.1m

Figure 6. This table compares our algorithm, Variable, to two versions of standard stereo, Fixed1 and Fixed2. Both Fixed1 and Fixed2 use the widest baseline possible where the near range, z_{near} is still in view. In Fixed1, the resolution is not allowed to exceed that of the actual camera, while in Fixed2, the hypothetical resolution is computed so that the error bound ϵ_z is met at z_{far}. This resolution is much too high to be realized in practice. Compared to Fixed1, our algorithm is about 6 times faster and about 3-4 times more accurate at z_{far}.

6 times faster for both scenes. Now suppose the images were captured at a resolution sufficient for standard stereo to meet the target accuracy. This resolution would be 10 to 20 times greater than that required by our method, and the processing time would be 200 to 500 times greater. Recall that the time complexity of both algorithms is proportional to ${\epsilon_z}^{-3}$, so reducing the target accuracy does not change the relative processing time (see equations (6) and (9)). Even if we relax the target accuracy to that achieved by standard stereo, our method is still 200 to 500 times more efficient. Keep in mind that while the time complexity of both algorithms is proportional to ${\epsilon_z}^{-3}$, our algorithm is proportional to ${z_{far}}^3$ as opposed to ${z_{far}}^6$, which is a dramatic improvement, especially for scenes with large depth ranges.

In our experiments, we have found an angle of triangulation of between 6 and 10 degrees to work best for our scenes. Larger angles can reduce the resolution required to meet the accuracy goal (see β in equation (8)), but matching is more difficult, and mismatches are more frequent.

6. Conclusion

We have presented our Variable Baseline/Resolution Stereo algorithm, which varies the baseline and resolution proportionally with depth in order to maintain constant depth accuracy throughout the reconstructed volume. This is in contrast to traditional fixed-baseline stereo in which the error increases quadratically with depth. Our approach directly addresses the accuracy and efficiency needs of an application designer wishing to employ stereo as a measuring device, and produces depthmaps which meet the desired accuracy while requiring orders of magnitude less computation than standard stereo. We have demonstrated our algorithm on real scenes in which our algorithm performs many times more accurately and efficiently than that which is possible with standard stereo.

Acknowledgements: We gratefully acknowledge the support of IARPA under the VACE program, DARPA under the 3D Worlds for Location based Warfighter Assistance program, and NVIDIA Corporation.

References

[1] R. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996.

[2] C. R. Dyer. Volumetric scene reconstruction from multiple views. *Foundations of Image Analysis*, pages 469–489, 2001.

[3] L. Falkenhagen. Hierarchical block-based disparity estimation considering neighbourhood constraints. In *Int. workshop on SNHC and 3D Imaging*, 1997.

[4] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[5] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005.

[6] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR*, 2001.

[7] R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *ECCV*, 1998.

[8] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *ICCV*, 1999.

[9] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, 1989.

[10] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007.

[11] M. Okutomi and T. Kanade. A multiple-baseline stereo. *PAMI*, 1993.

[12] M. Pollefeys, D. Nister, ..., and H. Towles. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2007.

[13] T. Sato, M. Kanbara, N. Yokoya, and H. Takemura. Dense 3-d reconstruction of an outdoor scene by hundreds-baseline stereo using a hand-held video camera. *IJCV*, 2002.

[14] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[15] C. Sun. Fast stereo matching using rectangular subregioning and 3d maximum-surface techniques. *IJCV*, 2002.

[16] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 1999.

[17] R. Szeliski and D. Scharstein. Sampling the disparity space image. *PAMI*, 2004.

[18] R. Yang and M. Pollefeys. A versatile stereo implementation on commodity graphics hardware. *JRTI*, 2005.

[19] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *ICCV*, 2007.