Micro-Baseline Stereo
Neel Joshi  C. Lawrence Zitnick
Microsoft Research

Abstract

Tradeoffs exist between the baseline or distance between cameras and the difficulty of matching corresponding points in stereo and structure from motion. Smaller baselines result in reduced disparities reducing the accuracy of depth estimation. Larger baselines increase the range of observed disparities, but also increase the difficulty of finding corresponding points. In this paper, we explore the use of very small baselines, called micro-baselines. Micro-baselines, typically just a few millimeters, provide the advantage that they can be captured using a single camera. That is, a “static” camera that is either hand-held or mounted on a tripod will typically vibrate some small amount while capturing video. We take advantage of the vibrating motion to compute depth information. For hand-held cameras a small amount of motion is generally always present, while many surveillance applications involve cameras mounted outside or on high poles that exhibit this type of motion. Even indoor cameras mounted on tripods move due to human traffic and machine vibrations.

1 Introduction

The baseline or distance between cameras is an important factor in stereo and structure from motion. Smaller baselines reduce the accuracy of computed depths, since the observed disparity of corresponding points is reduced relative to changes in depth. Larger baselines increase the observed disparities, but increase the difficulty of finding corresponding points.

Several works address this issue by using multiple cameras. For instance, multi-baseline stereo uses cameras with large disparities to increase the accuracy of depth estimation, while cameras with smaller baselines are used to disambiguate correct correspondences [Okutomi and Kanade 1991]. When matching images with small baselines, simple window matching costs may be used such as sum of squared distances or normalized correlation [Hirschmuller and Scharstein 2007; Tombari et al. 2008]. Many multi-view stereo algorithms also take advantage of these techniques [Seitz et al. 2006]. The use of large baselines [Pritchett and Zisserman 1998] requires more sophisticated matching measures, such as SIFT [Lowe 2004] or MSER [Matas et al. 2002]. Even with these measures, robust correspondence algorithms such as RANSAC are necessary. In structure from motion approaches [Triggs et al. 2000], images with varying baselines can be used to refine correspondences across several images. The robustness of these techniques has been demonstrated in several recent papers using large databases of images [Snavely et al. 2006; Agarwal et al. 2009].

We explore the use of very small baselines, called micro-baselines. Micro-baselines, typically just a few millimeters, provide the advantage that they can be captured using a single camera. That is, a “static” camera that is either hand-held or mounted on a tripod will typically vibrate some small amount while capturing video. We take advantage of the vibrating motion to compute depth information. For hand-held cameras a small amount of motion is generally always present, while many surveillance applications involve cameras mounted outside or on high poles that exhibit this type of motion. Even indoor cameras mounted on tripods move due to human traffic and machine vibrations.
The use of micro-baselines provides three main challenges. First the disparity between images or frames in the video is typically a small number or even a fraction of a pixel. As a result, accurate sub-pixel disparity estimates must be computed. Second, even with accurate sub-pixel estimates, large numbers of images must be obtained to offset inherent noise in the disparity estimation. Finally, the motion of the camera is from random vibrations, so extrinsic calibration information is not known.

2 Previous Work

A broad comparison of stereo vision techniques is the paper by Scharstein and Szeliski [2002]. Several methods have been proposed for sub-pixel correspondence. The work of Tomasi et al [2004] use a Phase-Only correlation technique to align two windows. The approach of Psarakis et al [2005] handles both sub-pixel alignments as well as photometric distortions. Thevenaz et al [1998] use a pyramid based approach for sub-pixel registration. Shimizu and Okutomi [2005] analyze sub-pixel estimation error using different functions.

Previous works have also addressed computing depth with a single image [Hoiem et al. 2005].

3 Micro-Baseline Stereo

We will now describe the framework for micro-baseline stereo. Let \( \{I_1, \ldots, I_j, \ldots, I_n\} \) denote a sequence of \( n \) images of an object where each image is a different frame of an input video-sequence. We assume each frame of the sequence is captured from a slightly different viewpoint. For a point \( p \), the observed intensity is denoted as \( I_j(p) \). Given a reference world-space coordinate system, which we choose to be coincident with coordinate system of a reference image \( I_0 \), correspondence between reference-coordinate points and coordinates in an arbitrary image is established given the points relative depth, \( z(x, y) \), corresponding to the object’s surface and the camera-projection matrix \( P_j \) at time \( j \). Observations across all \( n \) views for all points \( \{p_1, \ldots, p_j, \ldots, p_m\} \) can be related to the reference coordinate system by:

\[
I_0(x, y) = I_j(P_j(x_j, y_j, z(x_j, y_j))).
\]

In our work, the central goal in our method is to solve for correspondence and, in turn, the unknown disparity map. The unknowns in this system are the correspondence, disparity map, and camera projection matrices \( \{P_1, \ldots, P_j, \ldots, P_m\} \). To recover these components we use a structure from motion approach that derives camera-projection matrices relative to \( P_j \) where \( \hat{c}_j \) is a 2D component as the change in depth of \( c_j \) is a global (non-depth dependent) camera translation are captured by \( T_j \). We note that \( T_j \) is a 2D component as the change in depth of the camera relative to the scene is negligible in our setup.

Our setup is quite similar to the second approach, but has some subtle differences. Particularly as our baseline is extremely small, on the order of millimeters or less, an in plane rotation and translation alone are a very good approximations to the full planar homography. Under this assumption, the revised projection model is:

\[
\hat{p} = R_j \ast \hat{p}_j + T_j + \Delta c_j \ast \hat{z},
\]

where \( R_j \) is the in-plane rotation and \( T_j \) is a global \( (x, y) \) translation in the image plane. In this model yaw and pitch rotation and global (non-depth dependent) camera translation are captured by \( T_j \). We note that \( T_j \) is a 2D component as the change in depth of the camera relative to the scene is negligible in our setup.

Our goal is to recover the relative depths \( \hat{z} \). Once \( R_j \) and \( T_j \) are computed, \( \hat{z} \) can be recovered using an rank-1 factorization [Tomasi and Kanade 1992]. We recover these transformations using a RANSAC process on the results of correspondences computed using an dense optical flow alignment.

There is extensive literature on optical flow [Baker et al. 2007] for tracking pixels that move small amounts from frame to frame. Our case is relatively simple compared to finding general flow since the motion is small with minimal occlusions. We have found both HornSchunck [Horn and Schunck 1980] and patch based SSD methods to work well, if global alignment is performed first.

By running optical flow between each image \( I_j \) and the reference image \( I_r \) we compute an alternate estimate for Equation 7:

\[
\hat{p} = \hat{p}_j + \hat{F}_j,
\]

where \( \hat{F}_j \) is per-view, vector of per-pixel \( (x, y) \) translations chosen to minimized the re-projection error between the reference view and a given frame:

\[
F(p_j) = \arg \min_{F(p_j)} \sum_{i \in N(p_j)} (I_i(p_j) - I_i(p_j + F(p_j)))^2,
\]

for each pixel \( p_j \), where \( N(p_j) \) is the set of neighboring pixels – we use \( 5 \times 5 \) pixel windows. Sup-pixel accuracy if found by re-sampling the image on a 1/10 pixel grid using bi-cubic interpolation, and we search by shifting each window by the same 1/10 pixel stepping size.

By subtracting Equations 7 and 4, we get:

\[
F_j = (R_j \ast \hat{p}_j - \hat{p}_j) + T_j + \Delta c_j \ast \hat{z}.
\]
We estimate the rotations and translations for each view using a RANSAC procedure; however, instead of computing this in the direct way, i.e., independently for each view, we use a RANSAC process that uses the same inliers across all views. By using the same set of inliers and thus the same set of corresponding points, we ensure that the same reference plane is fit across all views. With separate inliers, different reference planes could be fit, and this would violate the plane+parallax model.

The final step in our algorithm is to compute the relative depth by factorizing the residual local flow:

$$\left(F_j - \left( R_{j} \ast \hat{p}_j - \hat{p}_j \right) + T_j \right) = \Delta c_j \ast \hat{z}. \quad (7)$$

The left side is the residual flow, computed using the fit rotations, translations, and per-pixel flows. We stack this system of equations for all images $j = 1...N$, creating a significantly over-constrained system, and compute $\hat{z}$ using a rank-1 factorization using SVD.

### 4 Results

To validate our method, we present several experimental scenes. Our first experiment is a proof-of-concept result in a lab setup. While the remaining results are in real-world, uncontrolled settings.

We filmed datasets with two different setups. For the results in Figure 3, 4, and 5, we filmed approximately 1000 frames at 30 FPS (about 30 seconds of video) from a video camera mounted on an unstable tripod. We imparted small vibrations to the tripod to jitter the camera to acquire micro-baseline views. For the results in Figure 1, we filmed approximately 100 frames at 30 FPS (about 3 seconds of video) using a hand-held smart-phone.

For each method we ran our algorithm to compute relative depth. For our flow-based alignment, we have tried and SSD based approach and Horn-Schunck [Horn and Schunck 1980]. For the SSD approach, we initially compute a global offset using a whole-image SSD search to remove large global translation, and then seed the local flow estimate with this. The local flow computation works on SSD search to remove large global translation, and then seed the approach, we initially compute a global offset using a whole-image approach and Horn-Schunck [Horn and Schunck 1980]. For the SSD based alignment, we have tried and SSD based approach.

The result is noisier than the first two to the many challenging aspects in the scene, yet the relative depth map is still reasonable.

In Figure 1, we show three more real-world scenes with a range of depths and textures. For these scenes, we use the relative depth map created from 100 input images, to create a synthetic shallow depth-of-field. These results are created by blurring the image with a pinhole (i.e. disk) point-spread function that is scaled as a function of the difference of the relative depth of a particular pixel and a chosen reference depth, i.e., the one that will remain in focus. The shallowness of the depth-of-field is a function of the mapping from depth difference to the blur kernel size. We create a set of focal sweep of images by using a fixed set of reference depths evenly swept through the range of recovered depths. In Figure 1, we show focus for three depths chosen so that the foreground, mid-ground, and background are respectively in focus for each dataset.

### 5 Conclusions and Future Work

We have shown how to recover depth maps from small natural variations that can occur in camera position when recording a short video clip. Our method recovers reasonable relative depth maps...
even for quite changing scenes and does this without any user intervention, external pre-calibration, or complex regularization procedures. The methods are computationally simple and fairly efficient. Our results suggest several directions for future work. While we intentionally did not explicitly include spatial regularization when computing the depth map; we, nevertheless, have no doubt that our results could be improved by including a regularization model such as Graph Cuts or Belief Propagation. We plan to incorporate one of these methods in the future.

Another extension is deploying this in an automated camera setup. Our method lends well to an iterative update rule, by aligning new input frames to the initial existing coordinate system, and using incremental PCA [Ross et al. 2004] to update the rank-1 factorization. We believe it would be very interesting to have a camera that continually refines its depth model as it records more data.

Results could be improved by including a regularization model such as Graph Cuts or Belief Propagation. We plan to incorporate one of these methods in the future.

**References**


Figure 5: Outdoor Scene: Top row, three images from the input video and bottom row, relative depth maps computed using 10, 200, and 500 images, respectively.


