Geometric Calibration for Mobile, Stereo, Autofocus Cameras

Stephen DiVerdi
Google, Adobe
stephen.diverdi@gmail.com

Jonathan T. Barron
Google
barron@google.com

Abstract

Mobile, stereo, autofocus cameras present unique challenges for robust and efficient depth estimation. Specifically, existing approaches for calibration of the stereo camera intrinsic and extrinsic parameters are inadequate because of per-shot changes in the configuration, long-term mechanical drift, extremely constrained manufacturing processes, and the requirement for real-world robustness. We present a hybrid strategy that combines a single-photo of a calibration grid in an offline step with online blind refinement, which satisfies all of these goals.

1. Introduction

Reliable scene-aware processing for consumer mobile photography (e.g. cellphone cameras) requires an inexpensive, robust, small, and low-power depth sensor. Options include using active stereo with infrared structured light or using a time of flight sensor. While these approaches are robust and fast, they are generally large, consume significant power, put off excessive heat, and only work in areas with low ambient infrared illumination, thus making them inadequate for consumer photography.

An alternative is to compute a dense depth map from a stereo image pair. Recent techniques compute high resolution depth maps at far greater speeds than previously possible [3]. An important prerequisite for this type of approach is rectified images, which ensure that corresponding points lie on the same horizontal scanline of the left and right images, enabling dramatic improvements in the search for correspondences. Good rectification requires in turn good calibration of the parameters of the stereo camera.

Stereo camera calibration is usually considered to be a solved problem, but consumer mobile photography presents surprising challenges. Most consumer cameras are autofocus, which means it is insufficient to simply calibrate once offline and use this data for each photograph, because as the focus adjusts to a scene the intrinsics and extrinsics will change. Furthermore, the calibration needs to remain correct over the lifetime of the device, which may be two years and may include a number of physical stresses that alter the configuration. The moving parts of a sensor module have physical tolerances that also result in some amount of lens shift, impacting the calibration per-shot. Finally, to satisfy manufacturing requirements this calibration must take no more than a few seconds, because time during manufacturing is prohibitively expensive for camera manufacturers.

Therefore, we present a solution for calibration and rectification of mobile, stereo, autofocus cameras which enables the efficient creation of depth maps at full sensor resolution. Our solution is hybrid in the sense that it includes offline (during manufacture) and online (per-shot) components. It is specifically tailored to the set of hardware requirements that make consumer mobile photography challenging, which we enumerate in Section 2.

2. Challenges

Mobile, stereo, autofocus cameras have specific limitations that make calibration challenging.

Drift. Manufacturing has some acceptable tolerance for mounting sensors which is why calibration is necessary. After manufacturing, cellphones are actively used for over a year and the relative pose of the two cameras may change. First, repeated thermal expansion and contraction from heating and cooling can cause the sensor mount to change shape over time. Second, sudden impacts (such as dropping on the ground) can loosen the mount. Third, mechanical parts responsible for moving the lens may loosen or weaken over time, changing the focus behavior. Long-term robustness is a primary reason why online calibration is required.

Lens Jitter. Commonly available focus modules move the lens along its optical center in a line perpendicular to the sensor, staying parallel to the sensor, but the actual lens motion is more complex. While the lens moves within its barrel, the sides of the lens may stick, causing the lens to tilt randomly. The barrel itself may be tilted, causing the default tilt of the lens to be non-zero. Lens barrel tilt also means the lens is not moving perpendicular to the sensor and thus shifts parallel to the sensor through the focus range.

The tolerances for these behaviors are provided in manufacturer's specification sheets.

Lens tilt has only a small impact on calibration. It moves the center point of the image which changes the radial distortion correction but for low distortion lenses this is a negligible effect. It also causes regions of the image far from the center point to be out of focus, but for the small apertures of mobile cameras this effect is again negligible.

Lens shift is more problematic as shifting the lens is equivalent to moving the camera and then moving the sensor relative to the lens in the opposite direction. This can create a significant pixel shift in image features. The equation for the pixel coordinate shift of an imaged scene point is

$$\Delta x = t \left(\frac{1}{p} - \frac{f}{Z} \right) \tag{1}$$

where t is the shift of the lens, p is the sensor pixel size, f is the focal length, and Z is the depth of the scene point. For typical values $p = 1.77\,\mu\text{m}$, $f = 2000\,\text{px}$, $t = 1\,\mu\text{m}$, and $Z = 1\,\text{m}$, $\Delta x = 0.563\,\text{px}$. That is, a $1\,\mu\text{m}$ lens shift results in about a half pixel image shift. Lens shifts of up to $10\,\mu\text{m}$ can be within tolerances, which means that consumer cameras will commonly produce image shifts of up to 5 pixels due to lens jitter.

For a stereo camera, the two images will have independent shifts which will change the calibration. The two image shifts can be decomposed into a translation, scale, and rotation of the stereo configuration. The translation component has no impact on the calibration (it is equivalent to simply moving the stereo camera as a unit and only impacts the image center point, similar to lens tilt). The scale component will change the magnitude of the disparities found but not impact the rectification. The rotation component will cause the epipolar lines to no longer be horizontal.

Stabilization. Optical Image Stabilization (OIS) improves photographs by adjusting the lens to reduce the impact of camera motion during exposure. In a simple form, OIS works by measuring the camera's angular velocity and then shifts the "floating" lens (spring-mounted) to compensate. This motion is on the order of $100\,\mu\text{m}$ per lens, or 50 or more pixels, and has the same impact as lens jitter.

Open-loop Focus. A reasonable strategy to calibrate an autofocus camera is to focus at different depths, performing standard single-focus calibration and storing a lookup table of calibrations indexed by depth [30], but this is inadequate.

In a "closed-loop" focus module, a register setting between e.g. 0 and 100 is mapped to a target position of the lens along the barrel. The spring-mounted lens is moved by applying a voltage-controlled force. The voltage is adjusted until another sensor indicates that the desired lens position is achieved, creating a closed control loop. Manufacturing tolerances result in variance of the lens position at some setting and the change in lens position per change in setting.

It is possible to calibrate the relationship between focus depths and register settings, though it adds additional calibration complexity. However, some focus modules are actually "open-loop", where there is no lens position sensor and the register setting maps to a constant force. If the camera is tilted, gravity exerts an additional force, changing the lens position. Therefore, a different voltage achieves the target focus depth, invalidating focus depth calibration. Currently, open-loop modules are more common for mobile cameras, as closed-loop modules are more expensive.

Auto Focus. Stereo camera sensors must be focused independently and therefore, may be focused differently. Aside from the differences between the two lenses and focus modules, contrast-based autofocus [28] depends on the image content which varies between the two sensors. Therefore, the peak contrast for each sensor may be when focusing at different effective scene depths. It may be possible for stereo cameras to tightly couple the autofocus between the two sensors, but inter-sensor communication in hardware is difficult and software camera APIs are limited in this regard.

Manufacturing. A constraint unique to consumer devices is that the total calibration burden cannot be more than perhaps five seconds and one to two photos. Manufacturing time and complexity directly impact cost and reliability. The acquired images can then be processed over a longer period of time, but whatever images are needed must fit within these requirements. Furthermore, steps that are demanding of operators are more likely to be done incorrectly, so complex instructions (such as a series of poses to photograph a calibration board) also impact cost. A fast calibration procedure can save a significant amount of money.

3. Existing Approaches

Existing approaches do not satisfy our constraints.

Offline Calibration. Offline stereo calibration [31] commonly uses around ten views of a 2D checkerboard. In each view, the camera poses are found, and over all views, the intrinsics and stereo configuration are computed. The focus range could be divided into n bins, and then n calibrations computed [30]. Work has been done to limit the number of focus samples that are needed [2]. In the limit, $n = 2$ but this is still more photos than desired for consumer manufacturing, and it makes no accommodation for per-shot error. Short-term (lens jitter and OIS) and long-term (drift) issues will not be corrected.

Fundamental Matrix. The fundamental matrix describes the geometry of a stereo camera and can be computed from 2D correspondences using the normalized 8-point algorithm [13], and made robust with RANSAC [9]. Rec-

tifying homographies are computed from the fundamental matrix [14]. There are an infinite number of rectifying homographies because x-shear does not modify the epipolar geometry, so there are techniques to compute the "correct" one [11, 19, 22]. This enables a completely online (or "blind") method of stereo calibration: per-shot, find 2D correspondences and then compute the fundamental matrix and rectifying homographies from them. This has the appealing quality that any short- or long-term sources of error will automatically be accounted for. However, there are serious distortion and robustness issues in many of the approaches [11, 21, 22]. Also, it is difficult to formulate regularization strategies for fundamental matrix estimation.

Bundle Adjustment. Another online approach is bundle adjustment, a workhorse algorithm for multiview stereo (MVS) that simultaneously estimates camera parameters and 3D world points [1]. The limitation is that it often needs more than two observations to be effective. While the formulation does not explicitly require more than two images, there are ambiguities that quickly produce incorrect results, even though the reprojection error can reliably be optimized to less than one pixel. Even if the radial distortion and center point intrinsics are known, the focus depth, model point depths, and z-offset of the stereo configuration are highly correlated with one another such that very similar reprojections can be computed with very different focus depths and z-offsets. MVS systems normally solve this problem with more observations of scene points, which yields more information about the 3D positions and camera motion.

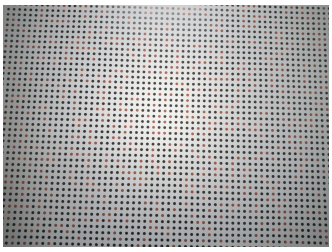
Other Approaches. Fitzgibbon et al. [10] generate priors for stereo camera configurations and use them to robustly estimate the actual configuration of other cameras from a few correspondences in one or more images, but they do not address autofocus. Pollefeys et al. [23] propose self-calibration with varying camera intrinsics, but not continuous calibration and not for stereo cameras. Liu et al. [18] formulate self-calibration for the pose of a stereo camera, but assume known intrinsics. Kurz et al. [17] use bundle adjustment to calibrate all the parameters of a stereo camera, but only for image sequences. Dang et al. [7] demonstrate continuous self-calibration to deal with long-term drift, but they also depend on image sequences. This result is extended by Hansen et al. [12] to show that per-frame self calibration is possible if at least 1000 correspondences are available, but some scenes have many fewer.

4. Hybrid Algorithm

Our contribution is a hybrid offline/online approach to stereo calibration. We use a single photo offline for an initial estimate and then refine it with online information per-shot. This strategy minimally impacts manufacturing while handling short- and long-term errors.

4.1. Offline

For the offline stage, we take a single photograph of a calibration pattern at about $1\,\mathrm{m}$ so it fills the image. Each sensor is auto-focused independently to create the sharpest possible images.



Our calibration pattern is a regular grid of small black and red dots (inset), where the color pattern uniquely identities each dot. While the specific calibration pattern is not important, we require that it consists of straight rows and columns of known coordinates, and that it fully covers the images (no "dead zones" near the borders with no data). Our calibration pattern detector is based on the work of Bruce, Balch, and Veloso [5, 6], extended to IDs on sliding windows as per M-arrays by Salvi et al. [27]. Small circles are used to reduce the impact of bias in the centers of projection [16].

This calibration pattern has many advantages over alternatives such as checkerboards or QR codes. We are able to find many more points in an image with dots. Any sub portion of the pattern can be visible and uniquely identified, so we do not need to carefully place the calibration board in the camera's field of view. Because the dots are small, we are able to find points even at the image borders.

4.1.1 Radial Distortion

Cellphone camera lenses tend to be low distortion to create pleasing photographs. However, pixel-accurate rectification still usually requires some distortion correction. We do not use the standard Oulu polynomial model [16] (Equation 2), because higher order polynomials often create artifacts around the perimeter, due to the scarcity of data near the image borders and the tendency of polynomials to overfit and diverge near scarce data. Instead we use a radial piecewise cubic spline (Equation 3)

$$\text{polynomial:} \quad \rho(r) = 1 + k_1 r^2 + k_2 r^4 \tag{2}$$

$$\text{spline:} \quad \rho(r) = 1 + \begin{bmatrix} 1 & r & r^2 & r^3 \end{bmatrix} \mathbf{A} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3}$$

where r is the distance from the center of the image, k_1 and k_2 are the Oulu coefficients [16], $\mathbf{A}$ is the matrix of cubic interpolation coefficients[1], and x_i are the spline control points. As the spline is piecewise cubic, the sequence of control points $\{x_0..x_n\}$ provide local control so each x_i

[1] https://en.wikipedia.org/wiki/Cubic_Hermite_spline

adjusts the correction over just a portion of the range of r, allowing better behavior near the image borders.

The standard approach is to minimize the reprojection error which depends on knowing the focal length and center point of the camera and the pose of the calibration pattern. We perform radial distortion correction in image space instead and remove the dependency on correct intrinsic and pose estimates. We use the rows and columns of the grid pattern as straight line priors and formulate our optimization such that the radial distortion parameters cause these lines to be straight (Equation 4), similar to the formulation of Devernay and Faugeras [8]:

$$\underset{\pi}{\operatorname{argmin}} \sum_i \sum_{j \in \Omega_i} d(\ell_i, \mu(\mathbf{p}_j, \pi)) \tag{4}$$

where ℓ_i is the line for a grid row or column, Ω_i is the set of points in ℓ_i, $d(\ell, \mathbf{p})$ is the point-line distance, and $\mu(\mathbf{p}, \pi)$ is the undistortion applied to point $\mathbf{p}$ with parameters π.

4.1.2 Stereo Configuration

Once radial distortion has been corrected, we compute the stereo configuration. Rather than parameterize the configuration as a rotation and translation, we use two 3D rotations, one for each image, similar to Monasse et al. [22]. We store the rotations in the Rodrigues format [15], so each has three degrees of freedom (DOF). There are only five DOF for the stereo configuration, as the scale of the translation is defined to be unit length. In our formulation, the extra DOF is the baseline rotation common to both cameras—regularization provides a way to uniquely select among solutions.

The rectifying homography is the product of the intrinsic parameters (focal length f_c, center point u_c, v_c), rotation (Rodrigues vector $\mathbf{r}_c$), and combined intrinsics (the average of the intrinsics of the two cameras), for cameras $c \in \{0, 1\}$:

$$\mathbf{K}_c = \begin{bmatrix} f_c & 0 & u_c \\ 0 & f_c & v_c \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\mathbf{R}_c = \text{Rodrigues}(\mathbf{r}_c) \tag{6}$$

$$\mathbf{K} = \frac{1}{2}(\mathbf{K}_0 + \mathbf{K}_1) \tag{7}$$

$$\mathbf{H}_c = \mathbf{K}\mathbf{R}_c\mathbf{K}_c^{-1} \tag{8}$$

Applying the homography $\mathbf{H}_c$ to the ith scene point $\mathbf{p}_c^i$ for camera c allows us to define the cost function as the difference between the y-coordinate of the rectified correspondences, over $i \in \{1..n\}$:

$$\mathbf{q}_c^i = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H}_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}_c\mathbf{p}_c^i \tag{9}$$

$$\delta_i = (\mathbf{q}_0^i)_y - (\mathbf{q}_1^i)_y \tag{10}$$

The offline optimization fixes the left (reference) image's intrinsics to nominal parameters (a provided focal length and center point equal to the center of the image) and minimizes the cost function over the right image's intrinsics and the rotations for both:

$$\underset{\mathbf{K}_1, \mathbf{R}_0, \mathbf{R}_1}{\operatorname{argmin}} \left(\sum \delta_i^2 + C_f + C_R \right) \tag{11}$$

The terms C_f and C_R are the regularization terms for the focal lengths and the rotations, respectively:

$$C_f = \lambda_f (f_0 - f_1)^2 \tag{12}$$

$$C_R = \lambda_R \left(\|\mathbf{r}_0\|^2 + \|\mathbf{r}_1\|^2 \right) \tag{13}$$

where C_f ensures the two focal lengths are near one another, and C_R biases the rotations towards zero, and λ_f and λ_R tune the relative influence of each.

4.2. Online

In the online step, we refine the previously optimized stereo configuration as well as the current focal lengths of the sensors. First we apply the optimized radial distortion parameters to undistort the images. Then we use FAST corner detection [25] to find salient features in the left image, and a sparse, pyramidal implementation of the Lucas-Kanade optical flow algorithm [4, 20] to propagate the features to the right image. From the correspondences, we estimate the fundamental matrix using the normalized 8-point algorithm [13] within a RANSAC framework [9] to remove outliers that do not satisfy the epipolar constraint, $\mathbf{x}_1^\intercal \mathbf{F} \mathbf{x}_0 = 0$. Once we have a set of inlier correspondences between the two images, we perform the optimization:

$$\underset{f_0, \mathbf{K}_1, \mathbf{R}_0, \mathbf{R}_1}{\operatorname{argmin}} \left(\sum \delta_i^2 + C_f + C'_R \right) \tag{14}$$

where the two focal lengths vary but are constrained to be near one another, and the center point of the right image varies to allow lens shifts. The rotations are also allowed to vary, but C'_R has a modified formulation:

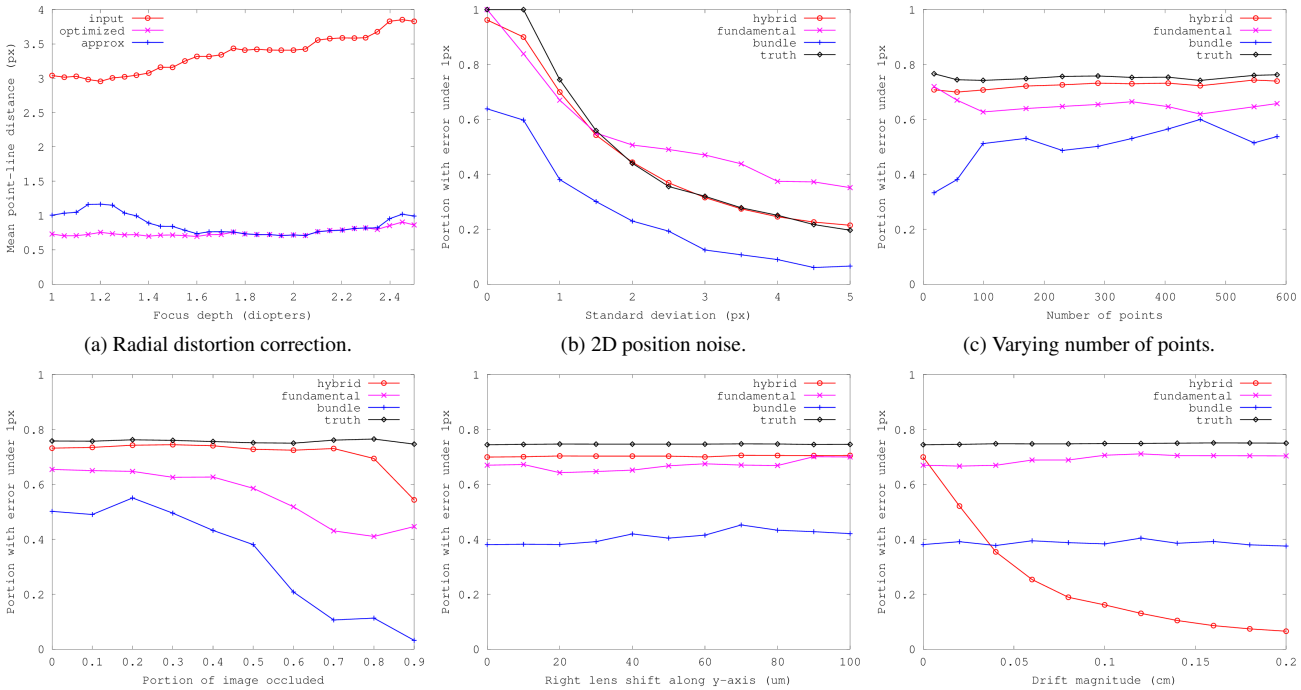
$$C'_R = \lambda'_R \left(\|\mathbf{r}_0 - \mathbf{r}'_0\|^2 + \|\mathbf{r}_1 - \mathbf{r}'_1\|^2 \right) \tag{15}$$

where $\mathbf{r}'_c$ is the optimized rotation from the offline step for camera c, and λ'_R is adjusted to allow smaller variations, thereby biasing the result towards the offline calibrated stereo configuration.

Finally, the optimized parameters yield the rectifying homographies which are used to warp the images.

5. Results

We conducted experiments with synthetic and real data to validate the performance of our algorithm. Our hardware prototype has two sensors separated by $20\,\text{mm}$ which capture 4208x3120 pixel images in RAW format.



(a) Radial distortion correction.

(b) 2D position noise.

(c) Varying number of points.

(d) Poor point distribution.

(e) Lens shift.

(f) Stereo configuration drift.

Figure 1: Experimental results. *(a)* Real images of the calibration board were captured at different focus depths, to test radial distortion correction. Results are reported as the mean distance in pixels between each point and its straight line approximation, lower numbers are better. *(b-f)* Simulated data is used to compare performance under different conditions. Results are reported as the percentage of points that have rectification less than 1 px, higher numbers are better.

5.1. Radial Distortion Validation

To validate our approach to radial distortion, we use a set of images of our calibration board focused at different depths. We measure the distortion as the mean point-line distance in pixels for the points in each grid row or column in the image. Figure 1a shows that the input images for a typical sensor have a distortion of between 3 and 4 pixels, depending on the focus depth, which is linearly sampled in diopters (inverse meters).

When we apply our calibration to the image at each focus depth and then undistort the image according to its optimized spline parameters, the residual distortion error is less than 1 px (Figure 1a "optimized"). For our single-shot offline calibration, we only use the distortion parameters estimated at a single focus depth and apply them to the images from all the other focus depths. To test this result, we select the calibration from the middle of the range, at 1.75 diopters, use it to undistort all the images, and show that the residual distortion error is only slightly increased at the extremes and still mostly under 1 px (Figure 1a "approx"). This means that we can use our approximation in place of a more expensive calibration with a negligible impact.

5.2. Simulated Data

We artificially create a stereo camera with intrinsics, extrinsics, and stereo configuration with random variations, plus a set of 3D scene points distributed throughout the frustum volume, and project the points to the two sensor views (so ground truth is known a-priori). We assume correspondence and only evaluate the ability of the different algorithms to rectify the two sets of points. Success is reported as the portion of the points that have vertical disparity (rectification error) of less than 1 px. We compare our algorithm ("hybrid") with estimation of the fundamental matrix ("fundamental") and with bundle adjustment ("bundle"), as well as rectification computed from the ground truth stereo configuration ("truth"). All experiments were conducted 10 times and averaged.

The fundamental matrix algorithm was implemented using `findFundamentalMat` and `stereoRectifyUncalibrated` from OpenCV[2]. The bundle adjustment algorithm was implemented using the Ceres Solver[3], an open-source optimization framework for large, complex problems, used by Google.

[2] http://opencv.org/

[3] http://ceres-solver.org/

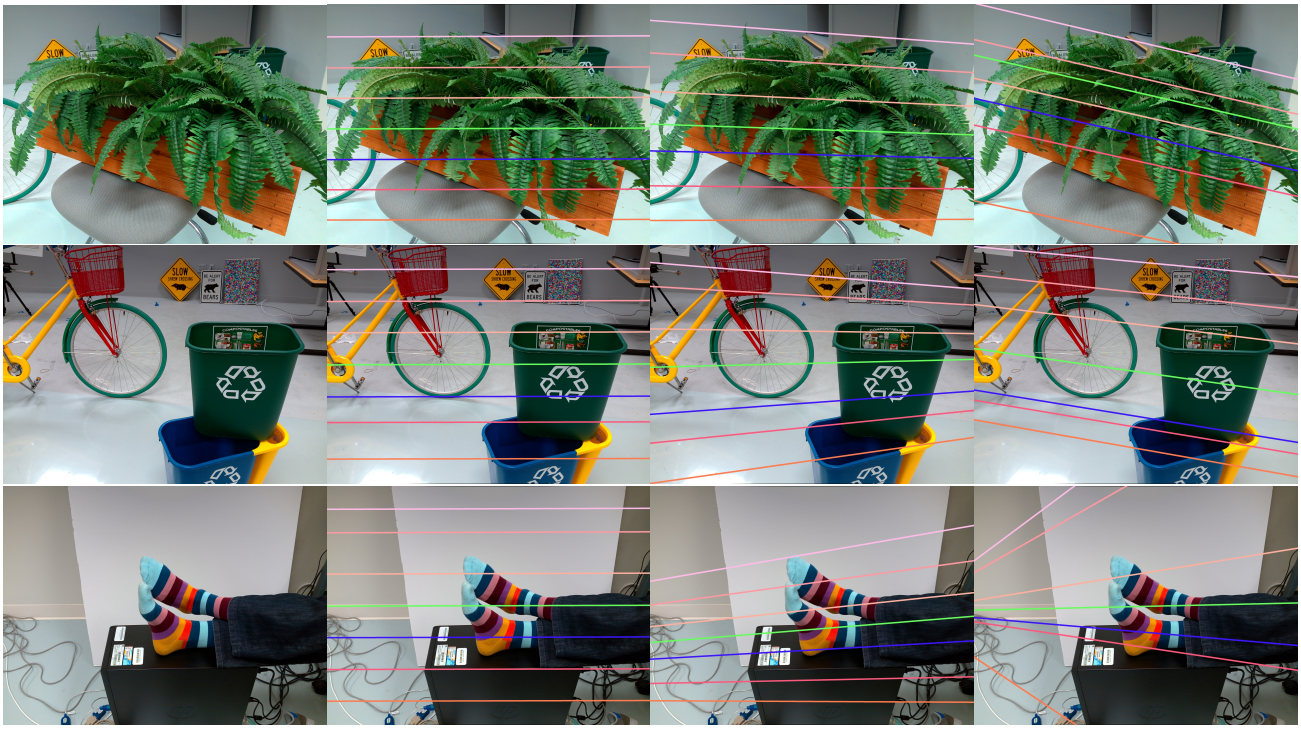


Figure 2: Results comparison. We show the optimized epipolar lines for the left sensor images in three scenes, where the actual epipolar lines are nearly horizontal. From left to right,the columns are the input, our results, fundamental matrix results, and bundle adjustment results.

Simulation of noise. Figure 1b shows the effect of increasing gaussian random noise added to the 2D point positions of each stereo pair. The ground truth performance drops with noise, because the noise moves points off the epipolar lines. Our performance is similar to that of the ground truth. It is interesting that the fundamental matrix approach eventually does better than the ground truth, presumably because it is finding (erroneously) consistent subsets of all the points. The noise simulation mimics what will happen in poor imaging conditions such as low light or short exposures, or for very small sensors.

In Figure 1c we show how performance changes based on the number of points in the scene, starting from 18. For all the simulations, gaussian random noise of standard deviation of 1 px is added to the 2D positions. What is important to note is that the hybrid results are roughly consistent even when there are few points, which can frequently occur in real scenes, especially if there is some blur or a dirty lens.

Simulation of poor distributions. We also simulate poor distributions of points around the image. This can frequently occur in real world images, particularly when outdoors with large, featureless skies occupying the upper half of the image, or when snow is on the ground, or indoors with featureless walls. Frequently, image points will only occupy some small portion of space in the image. While the various algorithms will still find reasonable solutions for those points, because the points only represent portion of the image, the result will be biased towards that region and errors may diverge in other image regions. To test this, we amend our simulation to run the rectification algorithm only on the bottom nth percentile of points, and then to measure the rectification error for the remaining top $(100-n)$th percentile of points. These results are shown in Figure 1d, where for example, the x value of 0.1 means the bottom 90% of points were optimized and the top 10% were tested. The important note is that our performance is consistent until the most dramatic of tests.

Simulation of lens shift. Per-shot lens shift can have any number of causes, and can result in images being shifted by up to 100 µm. To test this condition, we simulated motion of the right sensor's lens "upward", that is, along the image y-axis, perpendicular to the sensor's view vector, by up to 100 µm. The results are in Figure 1e. It is expected in this experiment that the fundamental matrix and bundle adjustment approaches will maintain their performance for different lens shifts—it is important to note that our approach is also consistent.

Lens shift also has an effect on the metric results of the depth map. Because of the nature of projective geometry, there is a fundamental inability to determine the absolute scale of a scene based solely on images, without some known length. Stereo vision solves this problem by having a known stereo baseline, often defined to be of unit length, which then sets the units for the depth estimation. As we have a known baseline of 20 mm, we can reconstruct absolute depth values in the scene. However, lens shift introduces an error in the baseline by translating the optical centers of the two sensors. In the worst case, they may both move 100 µm away from another, thus expanding the baseline by 200 µm. Con-

servatively, we can call this an error of less than 1% in the baseline.

The equations to relate disparity and depth are:

$$Z = \frac{fb}{d} \qquad (16) \qquad\qquad \frac{dZ}{db} = \frac{f}{d} \qquad (17)$$

for depth Z and baseline b in meters, and focal length f and disparity d in pixels. The derivative shows the relationship between depth and disparity is constant and a 1% error in the baseline results in a 1% error in the metric accuracy of the estimated depths. Intuitively, this makes sense as the problem is the lack of an absolute scale for the world, so if the reference unit (the stereo baseline) changes magnitude, the same change applies to the scale of the rest of the world as well. Unfortunately, without additional information (e.g. a scene prior) there is no solution to this problem and all stereo cameras with non-fixed lenses are subject to it, regardless of the algorithm used.

Simulation of camera drift. Similar to lens shift, camera drift can change the stereo configuration, though gradually over time and the induced transform is less constrained. To simulate the effect, we randomly vary the translation and rotation of the stereo configuration by adding in vectors of a set magnitude. A magnitude of 0.1 for the translation offset is relative to the baseline's unit length, or 10% which for our baseline of 20 mm would be 2 mm. Similarly, a magnitude of 0.1 for the rotation offset is roughly 5°. The results of this experiment can be seen in Figure 1f. Because our regularization tries to keep the optimized stereo configuration near the offline computed version, as the drift increases our error increases accordingly while fundamental and bundle stay roughly the same.

We do not have a prior for what the magnitude of the drift may be in a real camera, as extensive lifecycle testing of a number of units is necessary. That said, the linear thermal expansion coefficients for many common metals are on the order of 10^{-5} K^{-1} [29], so we believe 1% (0.01 on the x-axis of Figure 1f) to be a very conservative estimate. The parameter λ'_R (Equation 15) should be adjusted appropriately for the actual measured amount of drift. This drift also has the same impact on the metric accuracy of depth estimation as lens shift.

5.3. Real Images

We also performed our comparison with real world images acquired by our hardware prototype. Features were found using OpenCV's `FeatureDetector` interface which allows rapid experimentation with different algorithms. We use a pyramidal variant of FAST [24] with ORB descriptors [26] and `calcOpticalFlowPyrLK` to find correspondences between images. The same implementations for the fundamental matrix and bundle adjustment algorithms were used as in the simulation tests.

Figure 2 shows results for three stereo pairs, for our algorithm, fundamental matrix estimation, and bundle adjustment. A few of the optimized epipolar lines are marked for each algorithm's result. Clearly, fundamental matrix and bundle adjustment can generate unusable results even for straightforward scenes. This is likely because of the high resolution of the images resulting in surprisingly few features being found, poorly distributed throughout the images. While at low resolution there seem to be many feature points throughout the images, at high resolution corners appear softer and sparser, and available feature detection methods in OpenCV are unable to find more even distributions. A workaround is to find corners at lower resolutions (larger scales) but then the goal of pixel-accurate rectification at the original resolution is no longer achievable. We emphasize that this is a surprising result—we confirmed over many images in many different scenes and conditions that fundamental matrix estimation and bundle adjustment both frequently failed in the manner shown in Figure 2.

We have also collected performance information, running in a single thread on a desktop CPU. For our images (4208x3120) finding the corresponding points between the stereo images (FAST detection, optical flow, and RANSAC) takes 1300 ms on average. From those correspondences, our algorithm takes 20 ms, fundamental matrix estimation takes 0.7 ms, and bundle adjustment takes 410 ms.

6. Conclusion

We demonstrate a practical mechanism for rectifying images captured by mobile, autofocus, stereo cameras over the course of their planned operation and within manufacturing constraints, which enables high performance, high resolution scene geometry reconstruction via dense stereo depth estimation.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Comm. ACM*, 54(10):105–112, 2011. 3

[2] R. Atienza and A. Zelinsky. A practical zoom camera calibration technique: An application of active vision for human-robot interaction. In *Australian Conf. on Robotics and Automation*, pages 85–90, 2001. 2

[3] J. T. Barron, A. Adams, Y.-C. Shih, and C. Hernández. Fast bilateral-space stereo for synthetic defocus. *CVPR*, 2015. 1

[4] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker: Description of the algorithm. *Intel Corporation*, 5:1–10, 2001. 4

[5] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Intelligent Robots and Systems*, pages 2061–2066, 2000. 3

[6] J. Bruce and M. Veloso. Fast and accurate vision-based pattern detection and identification. In *ICRA*, pages 1277–1282, 2003. 3

[7] T. Dang, C. Hoffmann, and C. Stiller. Continuous stereo self-calibration by camera parameter tracking. *Trans. Img. Proc.*, 18(7):1536–1550, 2009. 3

[8] F. Devernay and O. Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured enviroments. *Mach. Vision Appl.*, 13(1):14–24, 2001. 4

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981. 2, 4

[10] A. Fitzgibbon, D. Robertson, A. Criminisi, S. Ramalingam, and A. Blake. Learning priors for calibrating families of stereo cameras. In *ICCV*, pages 1–8, 2007. 3

[11] A. Fusiello and L. Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *ICPR*, pages 1–4, 2008. 3

[12] P. Hansen, H. S. Alismail, P. Rander, and B. Browning. Online continuous stereo extrinsic parameter estimation. In *CVPR*, pages 1059–1066, 2012. 3

[13] R. I. Hartley. In defense of the eight-point algorithm. *Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997. 2, 4

[14] R. I. Hartley. Theory and practice of projective rectification. *Int. J. Comput. Vision*, 35(2):115–127, 1999. 3

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4

[16] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR*, pages 1106–1112, 1997. 3

[17] C. Kurz, T. Thormählen, and H.-P. Seidel. Bundle adjustment for stereoscopic 3d. In *MIRAGE*, pages 1–12, 2011. 3

[18] R. Liu, H. Zhang, M. Liu, X. Xia, and T. Hu. Stereo cameras self-calibration based on sift. In *ICMTMA*, pages 352–355, 2009. 3

[19] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *CVPR*, 1999. 3

[20] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981. 4

[21] J. Mallon and P. F. Whelan. Projective rectification from the fundamental matrix. *Image Vision Comput.*, 23(7):643–650, 2005. 3

[22] P. Monasse, J.-M. Morel, and Z. Morel. Three-step image rectification. In *BMVC*, pages 89.1–89.10, 2010. 3, 4

[23] M. Pollefeys, R. Koch, and L. V. Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *Int. J. Comput. Vision*, 32(1):7–25, 1999. 3

[24] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443, 2006. 7

[25] E. Rosten, R. Porter, and T. Drummond. FASTER and better: A machine learning approach to corner detection. *Trans. Pattern Anal. Mach. Intell.*, 32:105–119, 2010. 4

[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, pages 2564–2571, 2011. 7

[27] J. Salvi, J. Pageès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004. 3

[28] M. Subbarao, T.-S. Choi, and A. Nikzad. Focusing techniques. *Optical Engineering*, 32(11):2824–2836, 1993. 2

[29] Y. S. Touloukian. *Thermophysical Properties of Matter, Vol. 12: Thermal Expansion*. IFI/Plenum, New York, 1970. 7

[30] R. G. Willson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1994. CMU-RI-TR-94-03. 2

[31] Z. Zhang. A flexible new technique for camera calibration. *Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 2