ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/258712744

Unassisted 3D camera calibration

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · February 2012

DOI: 10.1117/12.909616

CITATIONS
0

READS
156

4 authors, including:



Vikas Ramachandra
Stanford University
90 PUBLICATIONS **519** CITATIONS
SEE PROFILE



Sergio Goma
Qualcom
36 PUBLICATIONS **229** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Machine learning View project

All content following this page was uploaded by Vikas Ramachandra on 15 July 2016.

The user has requested enhancement of the downloaded file.

Unassisted 3D camera calibration

Kalin Atanassov, Vikas Ramachandra, James Nash and Sergio R. Goma,
Qualcomm Incorporated.

ABSTRACT

With the rapid growth of 3D technology, 3D image capture has become a critical part of the 3D feature set on mobile phones. 3D image quality is affected by the scene geometry as well as on-the-device processing. An automatic 3D system usually assumes known camera poses accomplished by factory calibration using a special chart. In real life settings, pose parameters estimated by factory calibration can be negatively impacted by movements of the lens barrel due to shaking, focusing, or camera drop. If any of these factors displaces the optical axes of either or both cameras, vertical disparity might exceed the maximum tolerable margin and the 3D user may experience eye strain or headaches. To make 3D capture more practical, one needs to consider unassisted (on arbitrary scenes) calibration. In this paper, we propose an algorithm that relies on detection and matching of keypoints between left and right images. Frames containing erroneous matches, along with frames with insufficiently rich keypoint constellations, are detected and discarded. Roll, pitch yaw , and scale differences between left and right frames are then estimated. The algorithm performance is evaluated in terms of the remaining vertical disparity as compared to the maximum tolerable vertical disparity.

Keywords: 3D video, calibration, rectification, homography

1. INTRODUCTION

Recently, 3D capture and display technologies have become popular in consumer electronic and mobile devices. On the display side, modern TVs are capable of displaying 3D content. Also, contemporary mobile phones are equipped with high quality cameras while newer phones include stereo 3D cameras. As a result, 3D image and video capture have become a critical part of the feature set on mobile phones. 3D image quality is affected by scene geometry as well as on-the-device processing. It is known that for a pleasant viewing experience, no vertical disparity should exist between left and right camera images. To achieve this most 3D systems use an initial calibration step (based on a special chart) to estimate the relative camera pose. Left and right images are then aligned so that the stereo epipolar lines are parallel and there is no resultant vertical disparity. Very often the camera sensor contains a variable (programmable) auto focus mechanism to achieve better focus capability. As a consequence, the pose parameters estimated by factory calibration can be negatively impacted by movements of the lens barrel due to shaking, focusing, or camera drop. If any of these factors displaces the optical axes of either or both cameras, the vertical disparity might exceed the maximum tolerable margin and the 3D viewing experience may deteriorate. This is uncomfortable and has been known to cause eye strain and headaches. To avoid this, one needs to consider real time unassisted (on arbitrary scenes) calibration. In summary there are two steps needed to ensure high quality 3D content:

- **Factory based sensor calibration** - Done at the sensor manufacturing site. The purpose of this process is to screen out sensor parts that cannot be corrected. The latter is caused by the fact that typically a certain outer region, say 10 percent, is allocated for the pose correction (image warping budget). This budget needs to be split between: 1)camera pose deviation due to imprecise sensor mounting (static pose mismatch) and 2) camera poses deviations due to lens movement (dynamic pose mismatch). Sensor parts that exceed the static budget during factory calibration are discarded. It is to be noted that factory calibration is needed to identify sensor parts that are "way off" and for which the calibration, although still possible, produces a significantly reduced effective sensor field of view (due to a large amount of warping required to align the two images). Another possible role of factory calibration would be to assign sensor parts to bins based on the field of view loss due to image alignment via projective warping.

Qualcomm Inc., 5775 Morehouse Drive, San Diego CA 92121.

- **Real time calibration** - Typically done before or even during 3D content capture. It is desirable that real time calibration run without any user assistance, either periodically or on specific events. It is known that the lens barrel will move (thus changing the optical axis) when the focusing mechanism is engaged, due to gravitational force or excessive camera shake/drop. Since the algorithm runs very often, low processing power, reliability, and unsupervised operation are critical.

Another important aspect of the 3D capture system is an autoconvergence algorithm which distributes the 3D effects within a prescribed screen area (e.g. our earlier work on automatic stereo 3D auto convergence[1]). As a consequence, the calibration methodology need only achieve a vertical disparity below a certain limit. The condition for the optical axes to remain parallel need not be enforced since it will be overridden by the subsequent auto convergence adjustment process.

1.1 BACKGROUND

Previous efforts towards real time calibration (also called 'self calibration' in the literature) focused on estimating intrinsic as well as extrinsic pose parameters of either camera by using images captured during camera translation[2,3,.4] This requires joint processing of two or more stereo video frames. Moreover, some of these methods require special structures to be present in images.[5] Another issue encountered by some methods is sensitivity to noise.

Our contention is that full self calibration is not practical and is unnecessary due to the fact that factory calibration, which is done in a controlled environment, can very accurately (at the time of the calibration) estimate camera intrinsic parameters and align the left and right camera images to make the epipolar lines parallel. The only moving element in the camera system affecting the calibration results is the lens and as corroborated by the camera and physical mount maker's specifications, this causes only 3D rotations and scale differences. Furthermore, based on our experiments, we have found that the real time yaw, pitch and roll deviations are small (less than 5 degrees). In this special case, estimation of each rotation angle can be done independently of the other angles as is explained in later sections. Our algorithm takes advantage of these small deviations. Also, video frames can be processed independently of each other since the proposed scheme evaluates and uses one stereo image at a time to estimate rotations. Our modifications result in faster processing, lower memory requirements, and reduced power consumption. The proposed scheme can be thought of as a specialized and efficient homography solver.

1.2 ALGORITHM SUMMARY

Our proposed algorithm relies on detection and matching of key points between left and right images (no special structures are assumed to be present in the scene). Frames containing erroneous matches between the left and right images are discarded. Also, the quality of the key point constellation is evaluated for each input frame and a decision is made whether to use the frame for pose estimation or not. This step explicitly handles sensitivity to noise. Roll, pitch yaw, and scale differences between left and right frames are then estimated, and the right camera image is corrected with a projective transform to account for these relative rotations. Algorithm performance is evaluated in terms of the remaining vertical disparity as compared to the maximum tolerable vertical disparity, which was found to be 10 pixels at 1080p resolution based on our earlier perceptual studies.[6]

This paper is organized as follows. Section 2 contains a detailed explanation of the various processing stages followed by a discussion of the verification procedure in section 3. Section 4 contains some results and conclusions are presented in section 5.

2. ALGORITHMIC DETAILS

The proposed algorithm has the following processing steps:

- Key point detection and matching.
- Key point constellation evaluation.
- Rotation/scale parameters extraction.

- Right image warping using the estimated 3x3 projective matrix.

The above mentioned steps are discussed below.

2.1 Key point detection and matching

Key point detection is done with a modified Harris corner detector. The important points to be stressed here are:

- Resolution - width of 240 pixels for a typical horizontal field of view of 53deg
- High pass filter support 1x3 and 3x1 for the vertical and horizontal HP filters correspondingly
- Power averaging filter support - 7x7 using a Hamming window
- Another power averaging block with support 11x11 used to make detection not so much dependent of the edge differential.

The proposed modification to the Harris corner detector is described as follows. The original Harris detector computes the local matrix M

$$M = \sum_{(x,y)} w(x,y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{1}$$

and detects a corner when,

$$R = det(M) - k(trace(M))^2 > 0$$

In contrast, the modified detector detects a key point when

$$\frac{R}{max(1, trace(M_l/C)^2)} > threshold$$

M_l has a larger support than M, which helps detect corners in low contrast images. Detected key points based on Harris detector versus the proposed scheme are compared in Fig. 1.

Once the key points are detected in either image, they are matched using local window based normalized cross correlation. The search space for each local window is a 2D neighborhood around the key point location. At the end of this step, vertical and horizontal disparities are obtained for each key point.

2.2 Key point constellation evaluation

The key point constellation needs to be sufficiently rich to provide means to estimate pitch, yaw, roll, and scale. If the constellation is not rich enough , the solution is not unique. The uniqueness of the solution is tested in the following manner:
The angle/scale estimation procedure is run repeatedly for a given constellation while a small amount of noise is added to the key point positions. If the final result is relatively independent of small noise perturbations, i.e., the resultant variance in the solution is small, the constellation is deemed acceptable with very high reliability. This can be written as follows: The positional variations are a function of the rotation angles and scale.

$$P_y + \delta_y = F(P_x, P_y, roll, pitch, yaw, scale) \tag{2}$$

Thus, the angles and scale can be written as an inverse function of the positional differences between the left and right images.

$$(roll, pitch, yaw, scale) = \hat{F}(P_x, P_y, P_y + \delta_y) \tag{3}$$

The estimated solution for the angles and scale is stable if:

$$[roll + n_{roll}, pitch + n_{pitch}, yaw + n_{yaw}, scale + n_{scale}] = \hat{F}(P_x, P_y, P_y + \delta_y + n_y) \tag{4}$$

and

$$\sigma_{roll} < k_1\sigma_{Py}, \sigma_{pitch} < k_2\sigma_{Py}, \sigma_{yaw} < k_3\sigma_{Py}, \sigma_{scale} < k_4\sigma_{Py} \tag{5}$$

where n is the noise term.

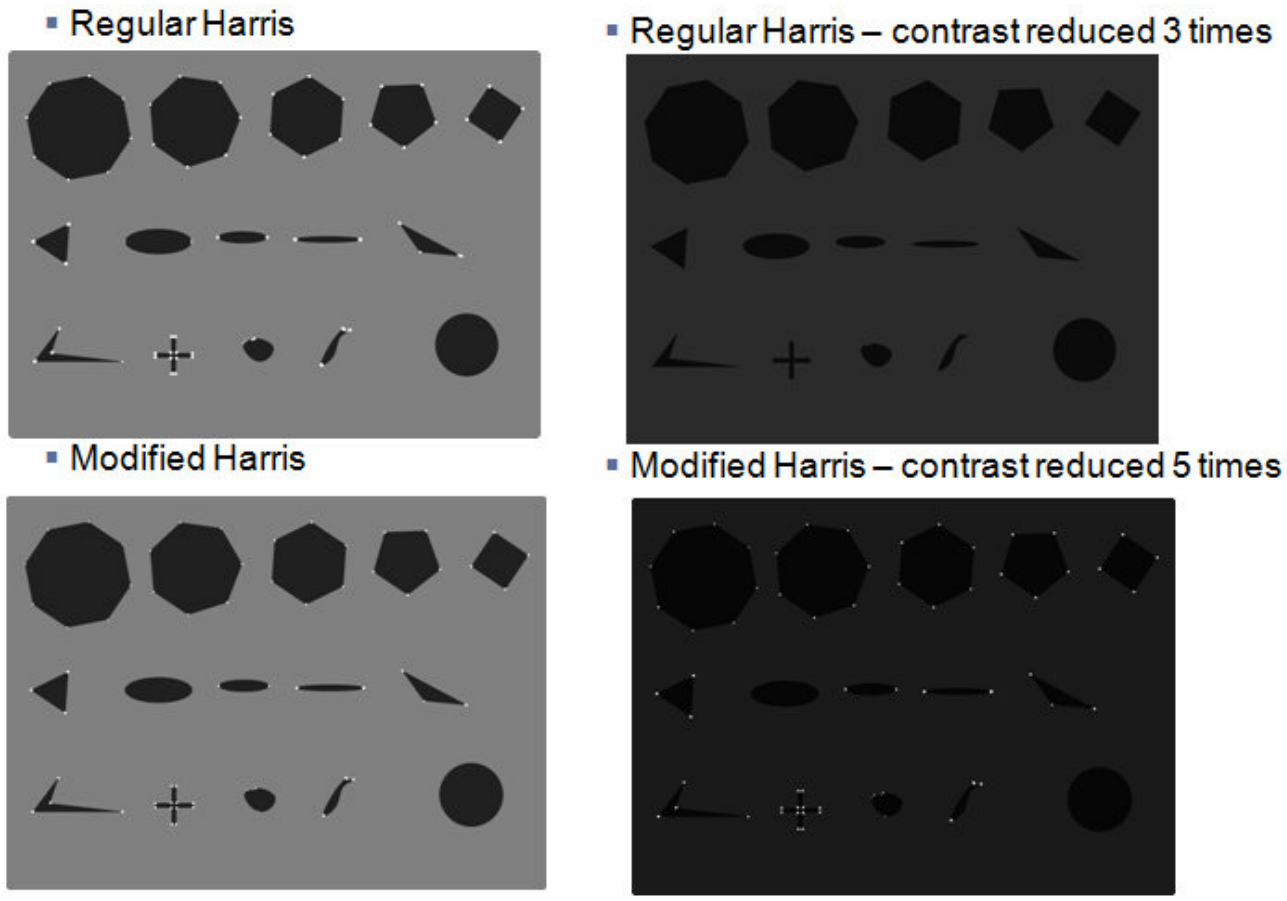


Figure 1. Harris versus proposed scheme

2.3 Rotation/scale parameters extraction

It is assumed that the left and right camera images differ by a scale and/or a 3D dimensional rotation (yaw, pitch and roll), which results in a 3x3 projective relationship between them. To make the problem tractable, an affine transformation is first estimated to account for roll, pitch and scale. This can be expressed mathematically as follows. Let (X,Y,Z) be the real world 3D coordinates of a point, whose corresponding image pixel coordinates P_x, P_y are given through the perspective projection (a is the pitch):

$$\begin{bmatrix} P_x W \\ P_y W \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{6}$$

For small pitch, the vertical pixel coordinates in the image can be related to the real world coordinates by:

$$P_y(a) = \frac{Y - aZ}{aY - Z} = \frac{Y/Z - a}{aY/Z + 1} \tag{7}$$

where P_y is the vertical pixel coordinates, a is the (small) pitch angle, and (X,Y,Z) are the 3D coordinates of a point in the 3D scene. Thus, for the plane sufficiently far away, the change in the vertical pixel coordinates between corresponding key points in the left and right camera images is given by:

$$P_y(a) - P_y(0) = \frac{Y - aZ}{aY - Z} - Y/Z = \frac{Y/Z - a}{aY/Z + 1} - Y/Z = \frac{-a - a(Y/Z)^2}{aY/Z + 1} \approx -a \tag{8}$$

Thus the vertical disparities caused by pitch angle difference , can be approximated as a constant shift. Also, the roll R and scale sc are determined through the following approximate relationship between corresponding key points in the left and right images:

$$\begin{bmatrix} P_x + \delta(x) \\ P_y + \delta(y) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(R)sc & -\sin(R)sc & shift_h \\ \sin(R)sc & \cos(R)sc & shift_v \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} \tag{9}$$

Using the second row only, we have:

$$\begin{bmatrix} \sin(R)sc & \cos(R)sc & shift_v \end{bmatrix} = \begin{bmatrix} P_y + \delta(y) & \cdots & \cdots \end{bmatrix} (\begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix})^{-1} = \begin{bmatrix} A & B & C \end{bmatrix} \tag{10}$$

Then, $shift_v = C = -a$ where a is the pitch, $sc = \sqrt{(A^2 + B^2)}$, $R = \tan^{-1}(A/B)$

Note that to perfectly correct for the roll and make epipolar lines in both images parallel to the horizontal axis, we would need to warp both images. However, for small roll angles, warping only the right image approximately corrects for the roll rotation, although it does not remove the vertical disparities completely.

The right image is warped based on this affine fit. The yaw angle is then estimated with a projective transform. The fact that the yaw changes the vertical disparities around the corners (from one horizontal end of the image to the other) is taken advantage of in this step. This relation can be written as follows (for an object at depth D).

$$\begin{bmatrix} (P_x + \delta_x)W/D \\ (P_y + \delta_y)W/D \\ W \end{bmatrix} = \begin{bmatrix} \cos(b) & 0 & \sin(b) \\ 0 & 1 & 0 \\ sin(b) & 0 & \cos(b) \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \\ D \end{bmatrix} \tag{11}$$

where b is the yaw. Thus, $[(P_y + \delta_y)/D \cdots] = [P_y/(-bP_x + D) \cdots]$

Yaw $b = -((P_y + \delta_y)P_x)^{-1}(P_yD - (P_y + \delta_y)D)$

Note that the yaw angle introduces large vertical disparities (and visual discomfort due to keystoning) only when its value is large. For smaller yaw values, the stretch or compression of the horizontal disparities is the main effect, which can be compensated for by the auto convergence algorithms.[1] Hence, the yaw angle estimation step need not be perfectly accurate.

The estimation and correction steps are run repeatedly causing the quality of the affine fit to become better with each iteration. The complete algorithm can be summarized pictorially through the flowchart in Fig. 2.

3. ALGORITHM VERIFICATION PROCEDURE

Verification of the proposed algorithm can be conducted in the following manner.

- Obtain a perfectly calibrated 3D camera
- Record different video clips with typical hand motion and panning
- Corrupt video clips by applying pitch, yaw, roll, and scale to the right image.
- Run real time calibration algorithm on corrupted video and extract pitch, yaw, roll, and scale.
- Compute pitch, yaw, roll, and scale errors
- Using remaining pitch, yaw, roll and scale, compute maximum vertical disparity.
- Compare the estimated worst case vertical disparity with disparity budget.

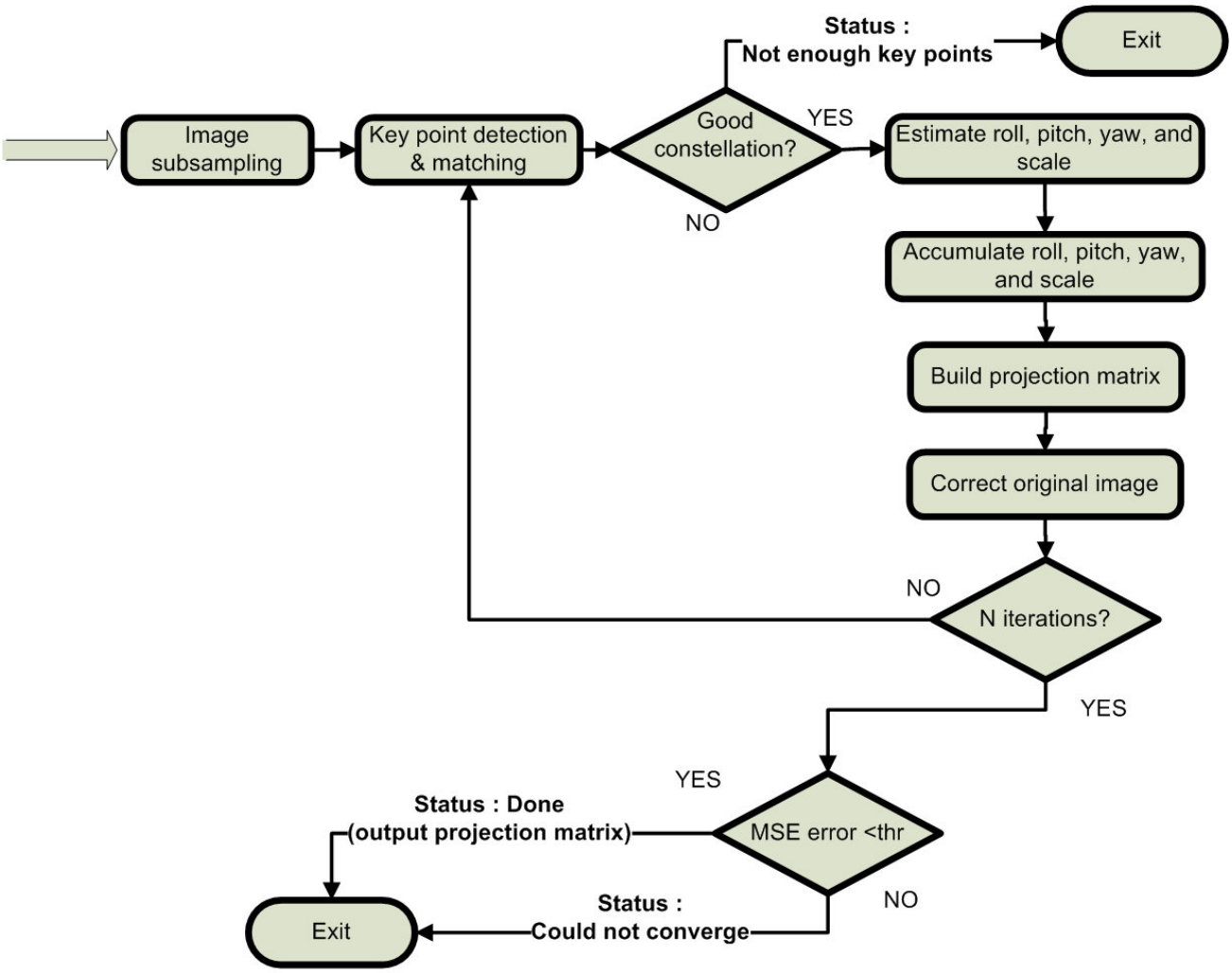


Figure 2. The proposed algorithm in a nutshell.

4. RESULTS

Results based on the verification procedure outlines above are presented in this section. Video clips were captured using the Sony stereo handycam which uses calibrated stereo cameras. The following angles and scale were added to the right camera video frames- pitch: 1.5 deg, yaw: 2 deg, roll: 1deg, and scale: 0.95. The left and right image pair corrupted with these rotations are shown in Fig. 3. Corrected images based on the proposed scheme are shown in Fig. 4. After warping the right image based on the proposed scheme, the vertical disparity histogram plot is shown in Fig. 5. In this figure,the red bar shows the vertical disparity budget. The green lines illustrate the residual vertical disparities for different video frames (corresponding to different scenes captured by the cameras). The blue bar shows the initial vertical disparities. By comparing the green and blue bars, it is clear that the proposed scheme succeeds in reducing the vertical disparities such that they fall below the acceptable limit.

5. CONCLUSIONS

Much effort has been expended on developing realtime calibration methods which are necessary for maintaining a pleasant viewing experience. Many conventional approaches are constrained by image structure requirements, the need for processing multiple stereo frames simultaneously, or suffer from noise sensitivity. These constraints increase the computational burden, power requirements, and cost. Our proposed method is a robust, low complexity, highly flexible realtime autocalibration system which yields corrected vertical disparities consistently below tolerance. Because typical alignment parameters are described by small deviations, the precise factory calibration is leveraged to simplify the calibration process. Noise sensitivity, especially in low contrast images, is reduced by using a modified Harris corner detector and taking into account the quality of key point constellations through an iterative process. These simplifications convert the general calibration process into a specialized homography solver. The algorithm was tested on real video clips according to a detailed algorithm verification procedure and was found to produce remaining vertical disparity values below the maximum acceptable level. This approach, therefore, avoids many of the barriers to widespread adoption of realtime calibration in mobile devices.

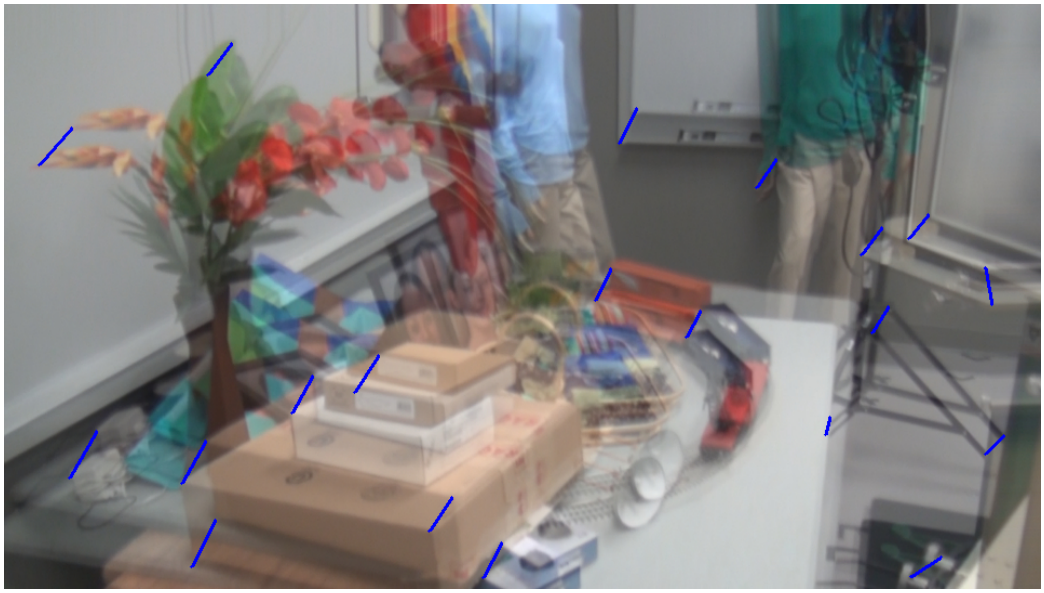


Figure 3. A stereo frame corrupted by yaw, pitch, roll and scale. Left and right images are overlaid with estimated disparity vectors in blue. Note that different regions of the image have different disparity vectors (both horizontal and vertical).



Figure 4. The same stereo frame corrected by the proposed algorithm. Note that the vertical disparities have been greatly reduced.

REFERENCES

1. K. Atanassov, V. Ramachandra, and S. Goma, "3d image processing architecture for camera phones," in *SPIE Three-Dimensional Imaging, Interaction, and Measurement Conference*, 2011.
2. Z. Zhang, Q. Luong, and O. Faugeras, "Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction," in *IEEE Robotics and Automation*, 1996.

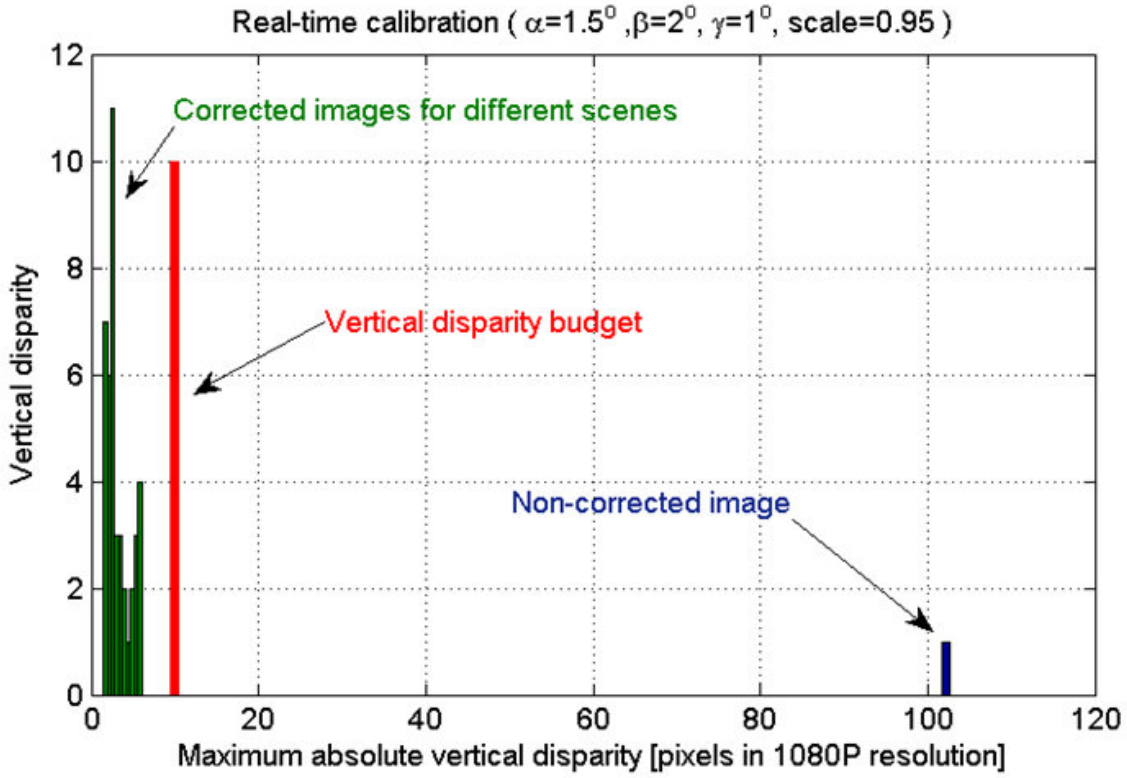


Figure 5. Histogram of the initial and residual vertical disparities (blue and green respectively) after the proposed algorithm acts on the images. It is clear that the proposed scheme is successful in lowering the vertical disparities across the image to acceptable values, as indicated by the red margin.

3. A. Zisserman, P. Beardsley, and I. Reid, "Metric calibration of a stereo rig," in *IEEE International Conference on Computer Vision*, 1995.
4. M. Pollefeys, R. Koch, and L. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," in *International Journal of Computer Vision*, 1999.
5. A. Habib, M. Morgan, and Y. Lee, "Bundle adjustment with selfcalibration using straight lines," in *The Photogrammetric Record*, 2002.
6. C. Tyler, L. Likova, K. Atanassov, V. Ramachandra, and S. Goma, "Measuring 3d discomfort from vertical and torsional disparities in natural images," in *SPIE Stereoscopic Displays and Applications Conference*, 2012.

View publication stats