XXI IMEKO World Congress "Measurement in Research and Industry"
August 30 - September 4, 2015, Prague, Czech Republic

NOVEL METHOD FOR ONLINE STEREO SELF-CALIBRATION

Joonas Melin [1], *Risto Ritala* [2]

[1] Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland, joonas.melin@tut.fi
[2] Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland, risto.ritala@tut.fi

Abstract - We present a novel method for correcting for the most critical errors in stereo camera calibration. Online self-calibration is often required as small errors in the vertical direction render many stereo vision algorithms useless. The method presented here is suitable for online self-calibration and does not require any external information. The results indicate that the developed algorithm is able to correct vertical errors to a sufficient degree.

Keywords: Stereo vision, Self-calibration, Robot

1. Introduction

Stereo vision is based on two cameras that capture images at the same time from different positions; this results in parallax that is in relation to the object's distance from the cameras. The parallax can be calculated using various techniques, however, usually some variant of block matching is employed. Block matching computes the disparities between the images by sliding a block over the other image and finding the minimal difference either by the sum of absolute differences or correlation. The disparity can be further converted to distance when camera calibration parameters are known.

The stereo cameras must be calibrated to determine the intrinsic and extrinsic parameters. The intrinsic parameters describe the internal parameters of the pinhole camera model and the distortion caused by the lens, while the extrinsic parameters describe the translation and rotation between the cameras. The extrinsic information is essential in metric disparity conversion. Once the calibration information is known image rectification can be performed to produces horizontal epipolar lines. The epipolar lines are lines that go through the corresponding features in both images. Horizontal epipolar lines are desirable as they reduce the computation needed.

The problem with real world stereo camera systems is that especially the extrinsic calibration parameters change when environmental conditions change. In many cases, this renders calibrated stereo camera systems nearly useless when used in the real world. Accurate calibration can also be challenging for some types of stereo camera systems, the small field of view and large baseline can make it challenging to get enough variety to the calibration images as the system can be designed for usage with significantly larger distances that are typically found in an office environment.

The section 2 outlines the necessary theory briefly and outlines the methods needed for stereo self-calibration. The section 3 presents the results obtained from stereo data with simulated errors and real world stereo data with unknown errors. In the final section 4 we will study the implications of the results and provide the outline of the future work.

2. Stereo self-calibration

Stereo camera self-calibration aims to tune the calibration information as the system is operating. The benefit of this approach is that changing conditions are constantly compensated. The importance of self-calibration increases as the run time of the system is increased as its often impractical to require recalibration of the system with arbitary time intervals just to keep the system performing adequately.

The self-calibration method described in this paper relies in features found in the environment. This method closely resembles the methods presented in [1] but with a few key differences. The authors compare self-calibration with epipolar geometry, bundle adjustment and trilinear constrains. We confirmed their results with epipolar constrains and determined that using only epipolar constrains is not a suitable method for robust self-calibration. Instead of bundle adjustment or trilinear constrains, we developed an algorithm that finds the optimal rectification for the images to minimize the feature errors in the vertical direction. All the phases of the algorithm are described in the Figure 1.

The design of this algorithm was motivated by the need to make existing stereo vision algorithms work reliably in case the stereo calibration changes during the operation as this is a common issue with practical stereo vision implementations as noted in [5] and [3]. Furukawa et al. note that even small errors in calibration parameters hinder the operation of the stereo vision system [2]. The goal of this work is to make the resulting source code publicly available for the robot operating system, ROS.

We decided to leave the intrinsic camera parameters out of the optimization as it was reasoned that their changes during operation are minimal as stereo vision systems commonly use fixed focal lenses that are rigidly mounted. This choice also simplifies the calculations required, making this algorithm better suited for an online algorithm. It was also decided that changes in the camera baseline will not

be considered as this would require information about either the camera movements or known distance in the image in case only one image pair is optimized at a time. Either of these measurements would have relatively high uncertainty, thus resulting in significantly more complex algorithm with little benefits as the changes in baseline are expected to be minimal on most rigidly mounted stereo vision systems. This way we are left with three rotations and two translations for our optimization. In addition, we will require only the intrinsic parameters and an initial guess for the extrinsic parameters.

2.1. Feature detection and matching

We did not note any major differences between different feature detection algorithms as long as the algorithm detected enough features to perform the matching. For our simulated data, we used SURF detectors as the FAST detector failed to detect clear edges on the generated surface. For our real world data sets, we have been using either ORB or FAST detectors, mainly due to their availability and suitable characteristics.

After calculating the features in both images, they need to be matched together. In our algorithm, this is a two-stage process. First the features are matched with their descriptors to produce potential matching features. This is done without considering the camera or scene geometry. The next step is to calculate the fundamental matrix with the random sample concensus (RANSAC) method for finding the inlier points that result in a valid fundamental matrix. As stated earlier, extracting the extrinsic parameters from only the fundamental matrix was noted to be too prone to errors. The epipolar inliers resulting from the RANSAC calculation often contain roughly 0.5-1% of outliers that will result in errors while running the optimization.

Our algorithm employs simple thresholding to detect the outliers from the matches. The match is seen as an outlier if the vertical coordinate difference is significantly larger than the median difference of all the points. This method relies on the fact that the errors in rigidly mounted stereo rigs are generally small and less than half of the points are actual outliers. The difference threshold used in our algorithm was 0.3% of the image height. Outliers generally have differences in the order of 10% of the image height. This phase of the algorithm is described as "Feature culling" in the Figure 1.

2.2. Optimization

The pinhole camera model and stereo rectification is based on the OpenCV implementation [4] as the algorithm is designed to run online in ROS. The pinhole model for the rectification is shown in Equation 1 where M is the original homogenized 3D point in the form of $M = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ where the Z coordinate is also set as 1 as we are only interested in 2D coordinates. The term m describes the resulting homogenized 2D coordinate in the form of $m = \begin{bmatrix} sx & sy & s \end{bmatrix}^T$. The term T represents the translation of the stereo pair, it is described as $T = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$, in our case, we are optimizing only the Y and Z coordinates. The $3x3$ rotation matrix R represents the rotation of the stereo pair. The rotation matrix is constructed from three axis Euler rotations that are all parameters of the optimization. The matrix K is the intrinsic camera calibration matrix which is presented in Equation 2 where the f_x and f_y are the focal lengths in horizontal and vertical direction. The terms c_x and c_y in the Equation 2 are the principal points along the X and Y axes of the image. This point is often located roughly at the image center.

$$m = K[R \mid T]M \tag{1}$$

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The error function used for the optimization with Levenberg-Marquardt algorithm is shown in Equation 3 where $\hat{y}$ is the rectified Y-coordinate of a feature and e is the error. Levenberg-Marquardt is not essential for this algorithm to work as many other least squares algorithms perform just as well as the optimization is generally performed close to the global minimum. We are optimizing only for the error in Y-direction as one image pair does not provide information about the error in horizontal X-direction if we don't have additional information about the depth of the scene.

$$e = (\hat{y_{left}} - \hat{y_{right}}) \tag{3}$$

The optimal rectification parameters R and T are updated iteratively as new images arrive. The update step is shown in equations 4 and 5.

$$R_{t+1} = R_t R_{t-1} \tag{4}$$

$$T_{t+1} = T_t + T_{t-1} \tag{5}$$

3. Results

This section describes the results obtained with simulated data with known relative motion between the cameras. We will also present the algorithms performance with real world data that has been collected with a micro air vehicle, MAV.

3.1. Synthetic data

The simulated data was generated with 3D modeling suite to simulate a case where the cameras are facing down at the scene with a well-featured surface. This makes it possible to have controlled errors in the camera calibration. This test was intended for testing how much distance error is left in the system when the optimization is done one step behind the measurements as in the real system.

The setup and the camera movement is described in the Figure 2. The setup mimics the cameras used for

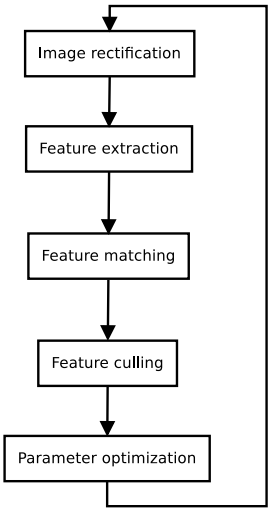

Fig. 1. Flowchart of the algorithm

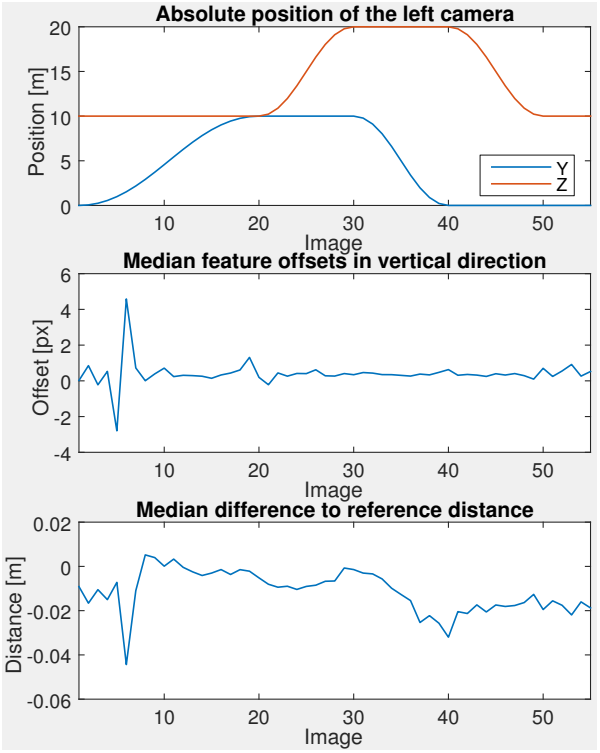

Fig. 3. The performance of the system with no relative movement of the right camera. The absolute position of the whole system is described at the top. The presumed errors in feature localization is described in the middle and the bottom describes the difference from the real distance.

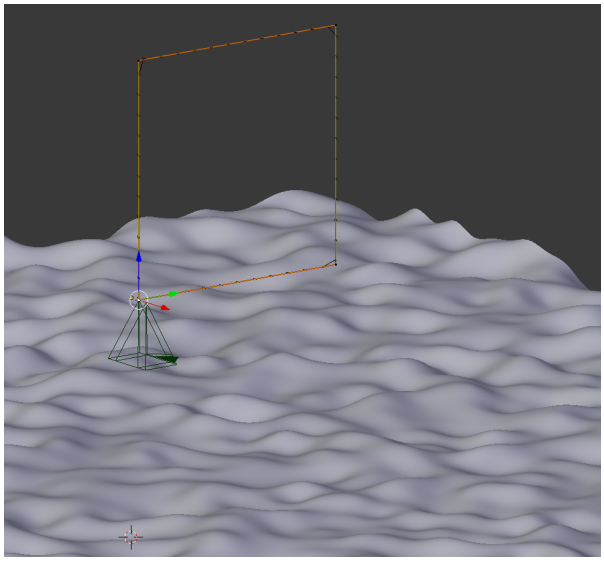

Fig. 2. Setup for data generation. Two cameras are separated by a 0.5m baseline. The green arrow is the Y-axis of the camera which points vertically up in the image direction. The red arrow is the X-axis of the cameras which points to the right in the images. The blue arrow is the Z-axis of the cameras which parallel to the optical axis of the cameras.

the real data as closely as possible. The setup was first evaluated without the relative movement of the cameras. The absolute movement of the system is can be seen in the Figure 3. Translation in Y direction was chosen so that the performance of the system with changing texture could be evaluated, this was considered to be equal to translation in X-direction so the X-directional translation was not considered in these tests. The translation in Z-direction was included to test the systems performance in case the distance to the objects change. The distance range was chosen to represent the distances present in the real world data.

The Figure 3 is the baseline test that presents the best case performance of the system with no errors introduced to the calibration. The algorithm occasionally detects some difference in the vertical direction and proceeds to correct these errors which are then compensated in the next image. The reference distance is from the depth buffer of the virtual camera but the optimization algorithm generates the the distances from the disparities between the images which leads to unavoidable differences between the two, especially when the distance increases and one pixel of disparity represents increasingly larger distances. The optimization algorithm itself is also introducing some of the error as badly detected features can cause lateral offsets to images. The worst case performance of the algorithm is causing few centimeters of error in distances ranging from 10 to 20 m.

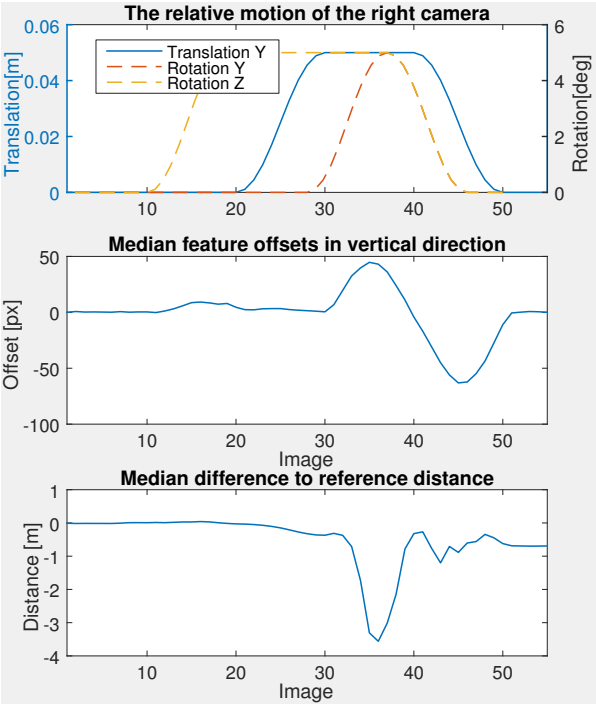

Fig. 4. The performance of the system with the optimization algorithm when the right camera has rotation and translation relative to the left camera. The relative motion is described in the top. The middle describes the feature offsets and the bottom describes the difference to the reference distance.

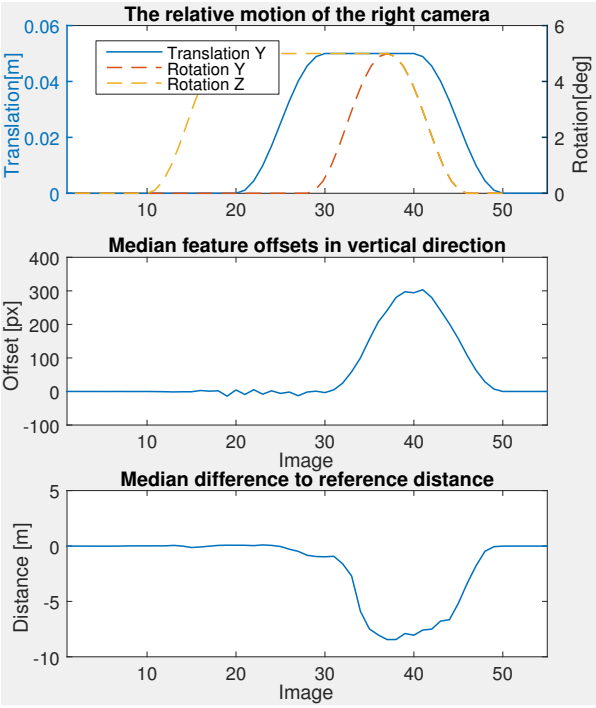

Fig. 5. The performance of the system with no parameter optimization in a case where the right camera has rotation and translation relative to the left camera. The relative motion is described in the top. The middle describes the feature offsets and the bottom describes the difference to the reference distance.

Fig. 6. MAV with the stereo camera system mounted in the bottom

The next two tests were performed with the optimization algorithm enabled and disabled. The tests had relative motion between the cameras. The results of the test with the optimization algorithm enabled can be seen in the Figure 4 and the case where the algorithm was disabled can be seen in the Figure 5. Initial motion is around the optical Z-axis of the camera which is the easiest to correct for the optimization as no translational component causes the same kind of errors. Rotation around the Z-axis will cause degradation of the disparity image in a favorable way as the errors are symmetrical thus resulting in only small median errors. With only the Z-rotation the disparity image is missing values in most of the pixel which also contributes to the small errors seen in both figures as only the pixels with valid values are considered for error evaluation. The next error that is introduced is the translation along the Y-axis. This will not create large errors in distance measurements as the features are only shifted vertically in the image. However the quality of the disparity image degrades further as less and blocks find a valid correlation. The final motion introduced is the rotation around the Y-axis which causes the features to move in a horizontal direction which is the most detrimental to the stereo vision algorithm as this directly alters the disparity values resulting in large errors in computed distance.

It can be seen that the optimization algorithm can correct for all the errors introduced to the system in these tests as seen in the middle of the Figure 4 the feature offset increases initially when an new error is introduced but then approaches zero when the errors are staying constant. It is noteworthy that the optimization algorithm is also able to reduce the errors in distance measurements. The relatively large errors even with the optimization are mainly caused by the online nature of the algorithm as the correction is always done with the data computed from the previous image.

3.2. Real data

The real data used for the calculations is a set of 70 images. The scene is a flat sand field with enough texture for the stereo vision algorithm and feature detection. The calibration data is roughly six months old and taken indoors. The stereo vision system used for capturing the data can be seen in the Figure 6.

The resulting optimized translation and rotation can be seen in the Figure 7. Even though there is noise in the

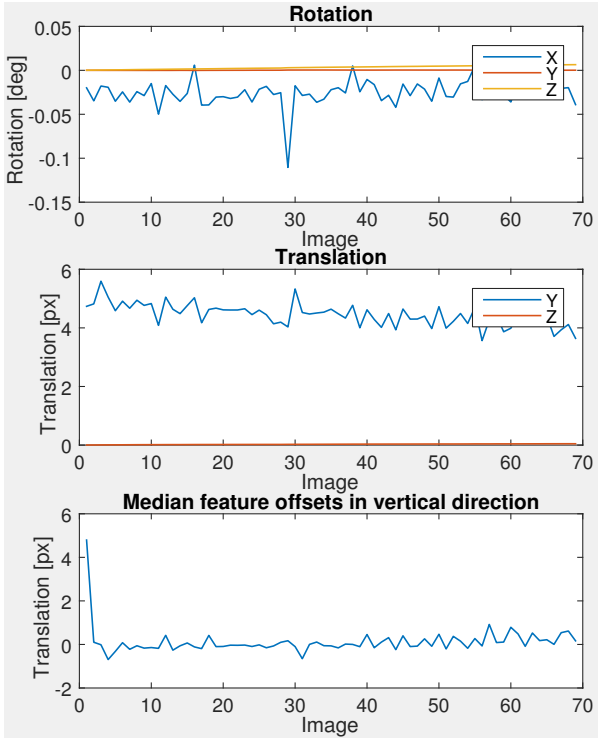

Fig. 7. The test case with real stereo data. The extracted relative rotation is shown in the top and the relative translation in the middle. The median offsets of the features can be seen in the bottom.

optimized rotation and translation, the error in the vertical direction is generally under one pixels even when using the transformation from the previous image. This means that the effects on the stereo calculation are negligible. It can be seen that the errors are fairly large initially but quickly converge to a small value. The noise in the extracted motion between the cameras is likely caused by inaccurate feature matches seen in the simulated results in addition to vibration of the system.

The magnitude of the original error in the vertical image direction is more than than ten pixels, which is more than enough to render the standard stereo vision algorithms unusable. However, as seen in the 7 the algorithm is able to bring the vertical error to under one pixel with in three images.

4. Conclusions and future work

We demonstrated that the developed algorithm is able to rectify the errors in the vertical direction that are the most critical in getting a dense stereo match with standard stereo vision algorithms. In addition, we demonstrated that the algorithm is able to correct for errors in the distance calculation caused by the relative motion between the cameras.

During the testing with the simulated data it was noted that the algorithm does not always recover the same motion input to the simulation, this is assumed to be caused by the fact that the optimization algorithm is correcting errors one image pair at a time instead of full bundle adjustment. This is a trade off between the algorithm complexity and accuracy of the recovered calibration information.

The main contribution of this work is presenting a novel and simple to implement algorithm for correcting small errors in the stereo calibration that plague many of the real life stereo vision systems. The simplicity is due to low requirements for external information and optimizing only for the camera rotation and 2D translation.

The next step for this work is to clean up of the proof of concept ROS implementation and have the source code published. In addition to this, a more complex version of this algorithm with bundle adjustment and distance information could be developed.

Acknowledgements

The authors express their gratitude to support by the Academy of Finland, grant Optimal operation of observation systems in autonomous mobile machines (O3-SAM).

References

[1] T. Dang, C. Hoffmann, and C. Stiller. Continuous stereo self-calibration by camera parameter tracking. *Image Processing, IEEE Transactions on*, 18(7):1536–1550, July 2009.

[2] Yasutaka Furukawa and Jean Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.

[3] Joonas Melin, Mikko Lauri, and Risto Ritala. Stereo vision with consumer grade high resolution cameras for a micro air vehicle. In *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014*. Delft University of Technology, 2014.

[4] OpenCV. Open computer vision(opencv), camera calibration and 3d reconstruction. `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`, 2011. Accessed: 2015-2-25.

[5] Michael Warren, David McKinnon, and Ben Upcroft. Online calibration of stereo rigs for long-term autonomy. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3692–3698. IEEE, 2013.