CALIBN - A Multiocular Camera Calibration Tool

Johnny Chien

August 6, 2014

Abstract

This report accompanies the program CALIBN provided for multiocular stereo vision. It briefly explains the related theoretical fundamentals and the used procedures before illustrating the performance by a few experimental results.

Contents

1 Introduction

Calibration of geometric imaging parameters is a fundamental task in the field of computer vision. Obtaining the imaging parameters allows accurate modeling of the incident ray of each pixel, and once backprojected the light paths can be used to estimate 3D structure of the scene.

Calibrating an image sensor is essentially to find the inverse mapping from image space to world coordinates, given a set of 3D-to-2D correspondences $(x, y, z) \rightarrow (u, v)$. Any object that once imaged produces such correspondences can be selected as a calibration target. In this work we consider the use of a planar chessboard as the target, as the making of such target is easier and the manufacturing accuracy can be controlled.

Many mature camera calibration packages such as OpenCV calib3d module [1] and Matlab Camera Calibration Toolbox [2] are available. These tools, however, are originally designed for monocular and binocular configurations, and their straightforward extension to multiocular case leads to consistency issues. In this work, we demonstrate why directly applying a stereo calibration algorithm on each camera pair fails, and how the problem is solved.

This report is organised as follows. In Sections 1.1 and 1.2, classical monocular calibration approaches and an extension to binocular calibration are reviewed. In Section 2, the adopted nonlinear camera model is described. In Section 3 we propose our multiocular calibration strategy, which is then evaluated in Section 4. Section 5 concludes this report.

1.1 Monocular Calibration

In the very beginning of the development, the direct linear transform (DLT) technique was adopted to solve the calibration problem before any sophisticated algorithms are proposed [3]. The DLT method treats the imaging process as a linear transform and solves the projection matrix directly. Post-processing is therefore required to extract the intrinsic and extrinsic parts of the parameters

from the estimated matrix.

In 1987, R.Y. Tsai proposed perhaps the earliest popular c amera calibration algorithm [4]. Credited by his name the method has been well-known as *Tsai's method* by the community. The method first calibrates camera's intrinsic parameters and the x, y components of extrinsic parameters, then the z components is estimated by enforcing orthonormality. Tsai's method is able to calibrate a camera from a single image.

A decade later Z. Zhang published an approach that soon became the de facto standard based on the image of an abstract geometry entity - the *absolute conic* [5]. The imaging of the calibration plane can be described by a homography. Given a homography, the orthogonality and normality of the rotation matrix respectively determine two out of six degrees of freedom of the image of the absolute conic (IAC). Since the IAC is defined by the camera's intrinsic parameters and is invariant to the derived homography, a linear system can be constructed by stacking constrains of many (at least 3) observed homographies. Zhang's method first recovers the intrinsic parameters by solving the IAC, then the extrinsics are determined using the solved intrinsics.

In the practice the parameters estimated using any of the described techniques are never used directly. Instead, they are served as an *initial guess* for the nonlinear optimisation stage carried out right after applying a chosen calibration algorithm.

1.2 From Binocular to Multiocular

The monocular calibration is extended to calibrate stereo camera rigs. Provided by the OpenCV [1] library the subroutine stereoCalibrate implements a state-of-the-art stereo calibration used widely by the community. The algorithm first independently calibrates each camera's intrinsic parameters and the pose of the calibration target in each view. Since the target's poses estimated from the first and the second cameras can be inconsistent, the algorithm chooses the first camera-derived pose as the reference, and the target pose derived from the

second camera is used only for the estimation of extrinsic parameters. Again, one has to work out the inconsistency of the estimated extrinsic parameters. A workaround to alleviate the impact of inconsistency as implemented by the subroutine is to decide the extrinsic parameters by averaging[1].

Obviously, several drawbacks exist in the stated algorithm. First, the target has to be observed by all cameras for each view. This can be an issue when calibrating a multiocular camera system, as the overlapped field of view shrinks as more cameras are considered. Second, estimating extrinsic parameters using a averaged pose is not an elegant solution and the accuracy heavily depends on the selection of reference camera. The development of improvements over the naive approach is therefore required. Key contributions of the proposed method are:

1. *Automatic pairing* of m cameras to establish $m-1$ stereo couples with maximised use of collected calibration data.

2. *Nonlinear bundle adjustment* in a unified coordinate system that simultaneously optimise all the system parameters to achieve a global consistency.

2 Camera Model

We follow the nonlinear camera model implemented by OpenCV. Given $E = (R, t)$ the extrinsics parameters consist of rotation matrix $R \in SO(3)$ and translation vector t, a 3D point (x, y, z) is first projected onto the ideal normalised image plane at $z = 1$ by

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1}$$

[1] stereoCalibrate actually calculates the median of pose vectors.

where $\mapsto$ denotes the projective equality (i.e. equivalent up to a non-zero scale). Then the nonlinear distortion applies to find the distorted pixel $(\breve{u}, \breve{v})$:

$$\begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} \dot{u} & 2\dot{u}\dot{v} & r^2 + 2\dot{u}^2 & 0 \\ \dot{v} & r^2 + 2\dot{v}^2 & 2\dot{u}\dot{v} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 \\ p_1 \\ p_2 \\ 1 \end{pmatrix} \tag{2}$$

where $r^2 = \dot{u}^2 + \dot{v}^2$ is the radial distance, and parameters $\kappa_{1..3}$ and $p_{1..2}$ respectively control the radial and tangential distortion of the image sensor. These parameters denote the distortion coefficients. Finally the distorted pixel is transformed to the observed pixel coordinates

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} = K \begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} \tag{3}$$

where f_u and f_v are the effective focal lengths in longitude and latitude directions respectively and (u_c, v_c) is the principal point where optical axis passes through the image plane.

Note that, the entry K_{12} of the intrinsic matrix is set zero. This entry controls the skewness factor of image formation, and is modeled when K is, more generally, an upper triangle matrix [3]. Alternatively, the skewness factor is modeled here by tangential parameters p_1 and p_2.

3 Multiocular Calibration

Calibrating m cameras with a calibration target placed in n different positions defines a m-camera n-view calibration problem. There are m sets of intrinsic parameters, $m-1$ sets of camera poses and n sets of target poses to be estimated. The pose parameters are defined by an arbitrarily selected camera space as the reference coordinate system (therefore we calibrate only $m - 1$ instead of m

camera poses, as the transformation from the selected camera to the reference is always the identity).

The selection of the referenced camera, however, can be tricky. Due to occlusion of the calibration target, one has to assume it is only observed partially by a subset of cameras for each view. As the result, the extrinsic parameters could not be derived directly using any single view. In this section we describe a multiocular technique that automatically finds a robust way to assemble complete extrinsic parameters using multiple views.

3.1 Calibration Graph and Parameters Initialisation

The extrinsic parameters of a multiocular camera system can be modeled by an *complete directed graph* $G = (V_G, E_G)$. A vertex $v \in V$ in the graph represents a camera, and an edge $e_{ij} \in E$ between vertices v_i, v_j denotes the coordinate transformation from v_i to v_j. Assume we have vertices v_i and v_j with their extrinsic parameters $E_i = (R_i, t_i)$ and $E_j = (R_j, t_j)$, the transformation on edge e_{ij} is therefore $E_{ij} = (R_j R_i^{-1}, t_j - R_i^{-1} t_i)$.

Explicitly estimating all $m(m-1)/2$ transformations E_{ij} is not necessary (and could be impossible due to view occlusion) for the recovery of extrinsic parameters $E_i, 1 \leq i \leq m$. As the transformation of coordinate system in Euclidean space is transitive, one can derive E_{ij} via an agent camera v_k by concatenating E_{ik} to E_{kj}. Calibrating edges $E_T \subset E_G$ in a spanning tree $T = (V_T, E_T)$ of G is therefore sufficient to recover all the extrinsic parameters.

Costs on the edges have to be defined to form a spanning tree. In the practice, different formations of spanning tree lead to numerically distinguishing parameters due to data imperfections. To ensure the robustness, we define a cost function subject to the number of shared views between cameras, which designating to each edge a cost $c_{ij} = 1/\gamma_{ij}$ where

$$\gamma_{ij} = \sum_{1 \leq r \leq n} b_r(i, j) \tag{4}$$

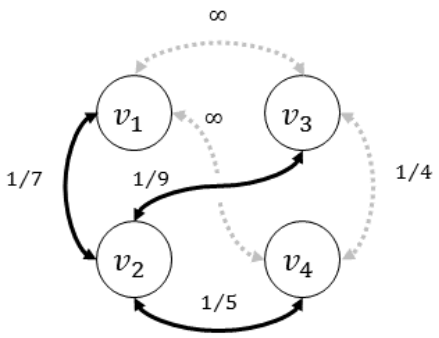

Figure 1: Example of a calibration graph and its MST.

and

$$b_r(i,j) = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ see the target in } r\text{-th image} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This way, cost between strongly connected cameras are low, and the *minimum spanning tree (MST)* yields a more robust initialisation of extrinsic parameters.

Figure 1 demonstrates an example of calibration graph and its MST formed using the described cost function. In the shown calibration problem the number of shared views between (v_1, v_2), (v_2, v_3), (v_2, v_4) and (v_3, v_4) are 7, 9, 5 and 4 respectively, while pairs (v_1, v_3) and (v_1, v_4) share no views.

We estimate the initial parameters of a multiocular system as follows. First the intrinsic parameters are calibrated independently for each camera. The intrinsics are then used to estimate the transformations pairwise in the MST. The calibration fails if the MST of the calibration graph does not exist (i.e. isolated vertex presents). CALIBN implements Prim's algorithm to find spanning trees.

Two problems remain with the initialised parameters. First, they are solved in a linear manner whereas only algebraical errors are minimised, hence lacked for geometrical meaning. Second, the estimations are done locally without taking global conditions into account. This could lead to highly biased estimations. The global adjustment as described in next section is therefore suggested to be

carried out.

3.2 Nonlinear Optimisation

The linearly initialised parameters are further tuned using our implementation of the Levenberg-Marquardt (LM) optimiser. LM algorithm iteratively searches the optimal parameters β that minimises a nonlinear objective function $\phi : \mathbb{R}^d \to \mathbb{R}$ in the sum-of-square form $\phi(\beta) = \|Y - f(X;\beta)\|^2$, given the observation $X \to Y$ and the parametrised measurement function $f : \mathbb{R}^n \to \mathbb{R}^m$.

In iteration k the estimate $\hat{\beta}_k$ is updated as $\hat{\beta}_{k+1} = \hat{\beta}_k + \Delta\beta_k$ by solving the augmented normal equations

$$\Delta\beta_k = [H + \lambda \mathrm{diag}(H)]^{-1}\delta \tag{6}$$

with $H = J^T J$ the Hessian matrix approximated by J the Jacobian matrix (i.e. $J_{ij} = \partial f_i(X;\hat{\beta}_k)/\partial\beta_j$), $\delta = J^T[Y - f(X;\hat{\beta}_k)]$ the error gradient and λ the damping variable. The differentiation can be achieved using either symbolic or numerical approach, while CALIBN calculates numerical differentiation.

The new estimate $\hat{\beta}_{k+1}$ is then assessed by checking $\phi(\beta_{k+1}) < \phi(\beta_k)$. For a better result the update is accepted and the damping variable is magnified to $\eta\lambda$ where η is a defined multiplier, turning the optimiser toward the gradient descent algorithm. Otherwise we reject the update and decrease the damping variable to λ/η such that the optimiser will behave more like the Gauss-Newton approach.

Termination criteria indicating convergence of the optimisation process are assessed over time. The error drop and step size are considered good indicators of convergence. In particular, we check the conditions

$$\frac{\phi(\beta_k) - \phi(\beta_{k+1})}{\phi(\beta_k)} < \epsilon \tag{7}$$

and

$$\frac{\|\Delta\beta_k\|}{\|\beta_k\|} < \epsilon \tag{8}$$

with the machine precision ϵ, which is usually set to a small positive value below which the change is numerically meaningless [6]. Setting higher ϵ tends to terminate the optimisation prematurely, while having a low ϵ could be a waste of time.

In the context of camera calibration, the 3D-2D correspondences of control points are used to instantiate the minimisation problem, f actualises the camera's projection function, and the least square reprojection error (RPE) is attained by the optimal parameters. In Bayesian terms, LM yields the maximum likelihood estimation of the parameters with the highest probability given the observations, if the noises of calibration data are Gaussian.

To apply the LM algorithm, CALIBN vectorises the parameters including

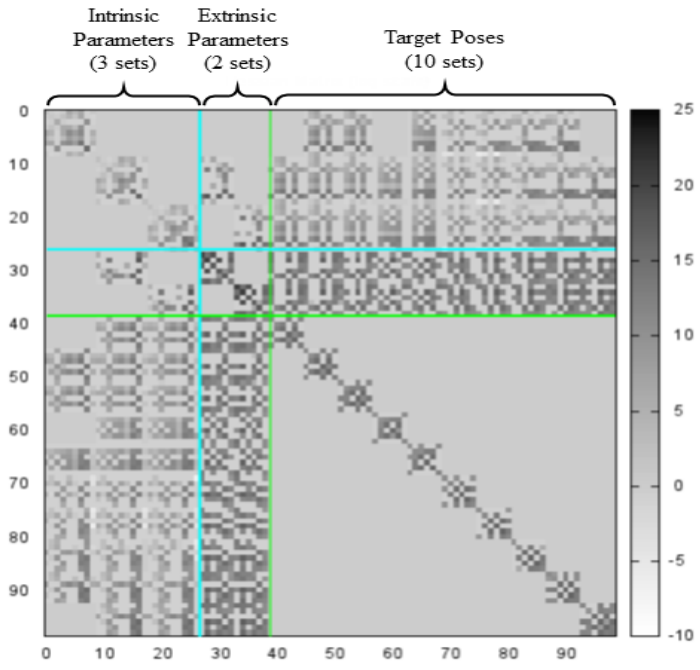

Figure 2: Hessian matrix (in logarithmic scale) of a 3-camera 10-view calibration problem.

m-set intrinsics, $(m-1)$-set extrinsics and n-set target poses as

$$\beta = (\mu_1, \mu_2, ...\mu_m, \xi_1, \xi_2, ..., \xi_{m-1}, \rho_1, \rho_2, ..., \rho_n) \tag{9}$$

where each intrinsic vector $\mu = (f_u, f_v, u_c, v_c, \kappa_1, \kappa_2, \kappa_3, p_1, p_2)$ encodes camera's intrinsic parameters, while ξ_i and ρ_i respectively describe the extrinsic parameters and target poses in the 6-vector form $(a_x, a_y, a_z, t_x, t_y, t_z)$ with first 3 components (a_x, a_y, a_z) define the angle-axis representation of rotation, and last 3 components (t_x, t_y, t_z) represent the translation. We therefore have $d = 9m + 6(m-1) + 6n$ the dimension of parameter space. Figure 2 visualises the Hessian matrix in a real case with respect to β of the described vector form.

A vector function f parametrised over β has to be given to the optimiser to evaluate an estimate $\hat{\beta}$. Let $(x, y, z) \to (u, v)$ be a correspondence observed by i-th camera in j-th image, we define reprojection error vector as

$$f_{ij}(x, y, z; \beta) = (u, v) - \Pi[g_{ij}(x, y, z; \beta); \mu_i] \tag{10}$$

where $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the camera projection function introduced in Section 2, $g_{ij} : \mathbb{R}^3 \to \mathbb{R}^3$ is the coordinate transformation function

$$g_{i,j}(x, y, z; \beta) = \begin{pmatrix} R(\xi_i) & t(\xi_i) \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} R(\rho_j) & t(\rho_j) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{11}$$

and $R(\square)$ and $t(\square)$ respectively represent rotation matrix and translation vector of a given pose parameter. The parameters μ_i, ξ_i and ρ_j are extracted from β. To allow finer access to the error term, we separate the x and y components of the reprojection error vector in the implementation of f [7].

4 Experimental Results

We applied CALIBN to calibrate a trinocular camera system, and the results are compared with the direct extension of the stereo calibration comes with OpenCV, which estimates parameters pairwise. A chessboard has been placed in 38 poses and simultaneously imaged by 3 cameras. Figure 3 shows the positions of the calibration target observed by the first camera.

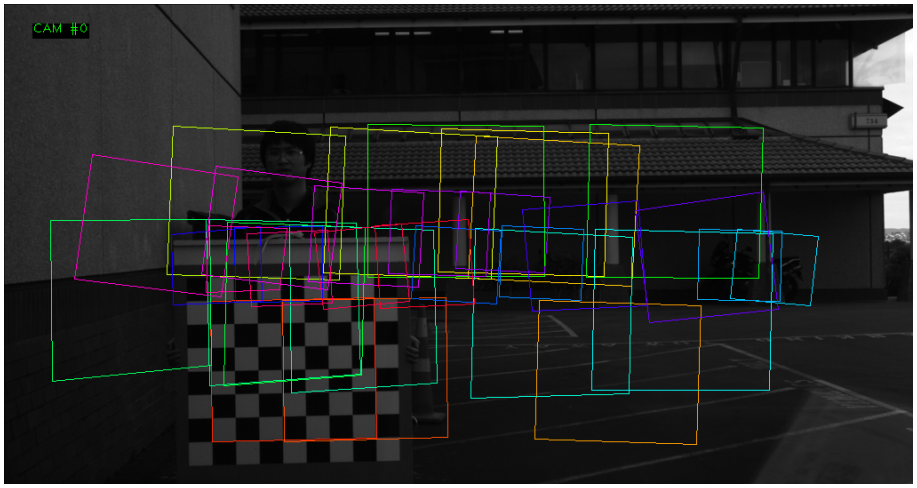

Figure 3: Coverage of 38 placements of the calibration target.

The global optimisation runs for at most 30 iterations, and we set machine epsilon $\epsilon = 10^{-8}$ for convergence check. The root-mean-square errors (RMSEs) of reprojected control points are plotted in Fig. 4. Significant error drops are observed through iteration 4 to 8. In the first 4 iterations the improvement is not

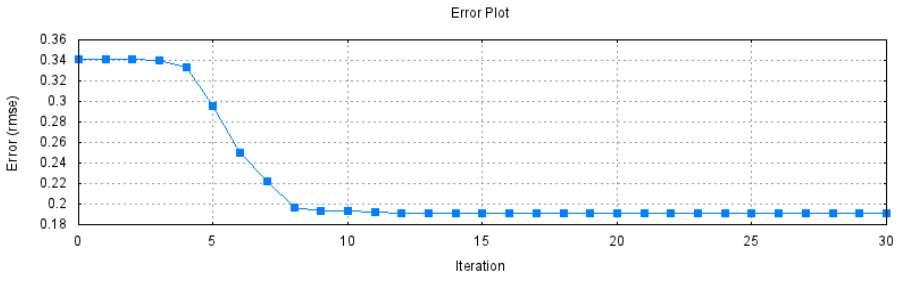

Figure 4: Error plot through the global optimisation process.

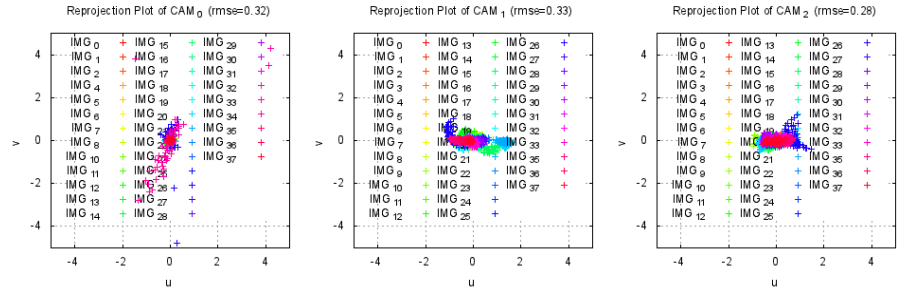

Figure 5: Control points reprojected using pairwise optimised parameters without global bundle adjustment.

significant. A possible explanation is that, the linearly initialised parameters are very close to a local optimum. In this case, an insufficient termination criteria, say $\epsilon = 10^{-4}$, could lead to an early stop of the optimisation process.

In Fig. 5 the control points are reprojected using parameters obtained and locally optimised by the OpenCV calibration subroutine, without global adjustment. Compared to Fig. 6 which shows the reprojections optimised according to Section 3.2, an improvement is indicated by the drops of RMSE from 0.3 to 0.07 pixels for the second and the third cameras respectively, despite the error drop is not significant for the first camera .

Image rectification are applied to study the accuracy of calibrated parameters. The rectified trinocular views are shown in Fig. 7 and Fig. 8, respectively of pairwise and globally optimised parameters. As can be seen, the epipolar constraint is well preserved near the marked control points in both cases. However,

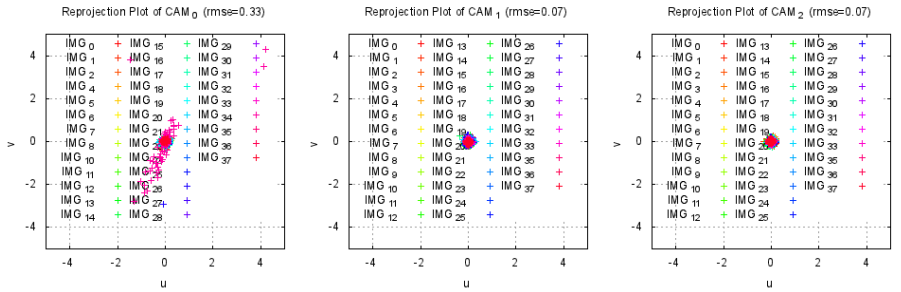

Figure 6: Control points reprojected using globally optimised parameters.

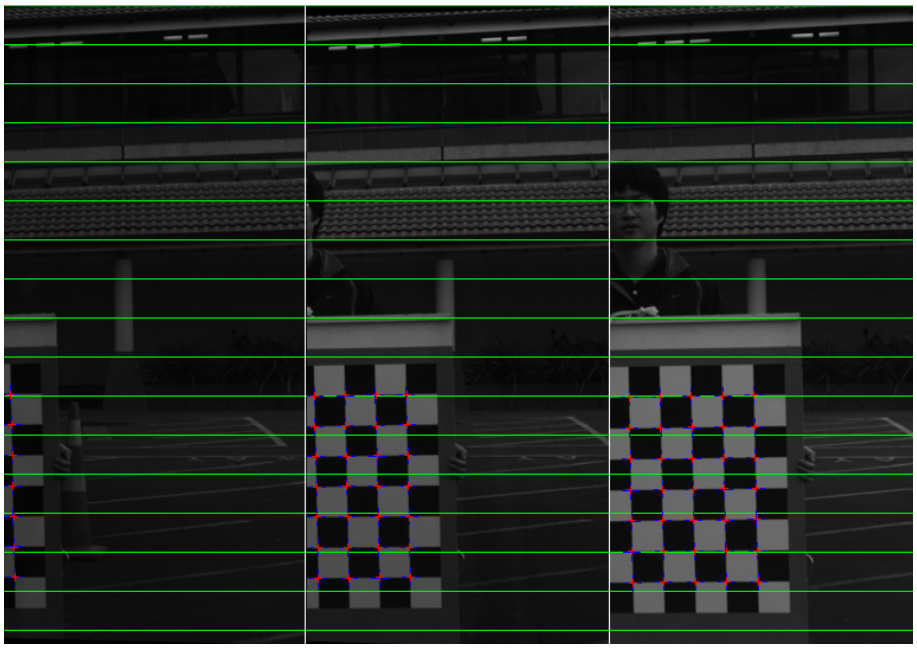

Figure 7: Views rectified using pairwise optimised parameters without global bundle adjustment.

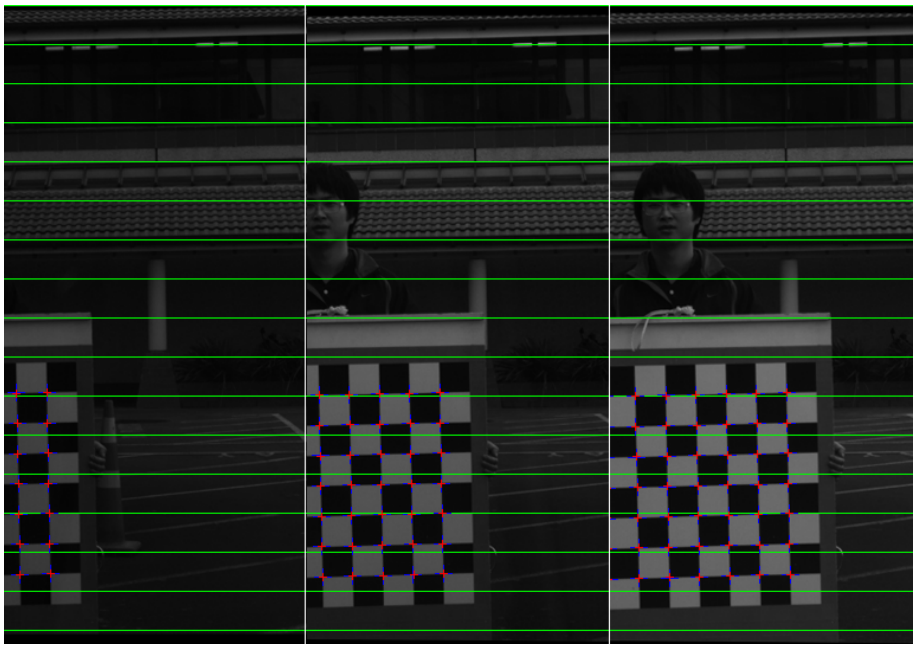

Figure 8: Views rectified using globally optimised parameters.

in the remaining (e.g. the top margin of the image) significant misalignment is obvious in Fig. 7. This demonstrates the generalisation error due to the bias of locally optimised parameters.

5 Conclusion

In this work we built a multiocular camera calibration tool. The proposed method automatically establishes robust calibration pairs by solving the MST problem. The initialised parameters are further optimised in global scale using our implemented LM algorithm. As supported by the experimental results, the proposed method is able to attain a globally consistent parameters. Compared to the direct extension of stereo calibration algorithm, an improvement of 44% in RPE is attainable.

Acknowledgement

I acknowledge the collaboration with Haokun Geng while implementing CALIBN.

References

[1] G. Bradski. "The OpenCV library," *Dr. Dobb's J. Software Tools*, Nov. 2000.

[2] J. Y. Bouguet. "Camera calibration toolbox for Matlab," `www.vision.caltech.edu/BOU2013j/calib_doc/`, 2013.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, Cambridge, 2004.

[4] R. Tsai. "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robotics and Automation*, vol. 3, pp. 323–344, 1987.

[5] Z. Zhang. "A flexible new technique for camera calibration," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*, 3rd edition, Cambridge University Press, New York, 2007.

[7] B. K. P. Horn. "Tsai's camera calibration method revisited," `people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf`, 2000.