CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

RESEARCH REPORT

ISSN 1213-2365

Global Camera Parameterization for Bundle Adjustment

C. Albl and T. Pajdla

{alblcene,pajdla}@cmp.felk.cvut.cz

CTU–CMP–2013–17

September 24, 2013

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/alblcene/Albl-TR-2013-17.pdf

The work was supported by the EC project FP7-SME-2011-285839 De-Montes, Technology Agency of the Czech Republic project TA2011275 ATOM and Grant Agency of the CTU Prague project SGS12/191/OHK3/3T/13. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

Research Reports of CMP, Czech Technical University in Prague, No. 17, 2013

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Global Camera Parameterization for Bundle Adjustment

C. Albl and T. Pajdla

September 24, 2013

Abstract

Bundle adjustment is an important optimization technique in computer vision. It is a key part of Structure from Motion computation. An important problem in Bundle Adjustment is to choose a proper parameterization of cameras, especially their orientations. In this paper we propose a new parameterization of a perspective camera based on quaternions, with no redundancy in dimensionality and no constraints on the rotations. We conducted extensive experiments comparing this parameterization to four other widely used parameterizations. The proposed parameterization is non-redundant, global, and achieving the same performance in all investigated parameters. It is a viable and practical choice for Bundle Adjustment.

1 Introduction

Structure from motion (SfM) reconstruction received a lot of attention resulting in many practical applications such as Photosynth [15] and Bundler [18]. Current research aims at providing more precise reconstruction as well as the ability to handle larger datasets [1], [5].

Bundle Adjustment (BA) [19] is an important part of SfM as it optimizes the resulting estimates of 3D point coordinates and the position, orientation, and calibration of cameras (Figure 1). Detailed analysis of BA optimization methods, parameterizations, error modeling and constraints has been given in [19]. An efficient and comprehensive algorithm that utilizes the sparsity of BA has been developed by Lourakis and Argyros [14] and the code was made freely available. This algorithm has been further used in [18] to build a full structure from motion pipeline. An extended version of [14] has been developed in [12] utilizing the sparsity even further in order to reduce computation time. Recently, the performance of BA on large datasets has been scrutinized [1]. The use of conjugate

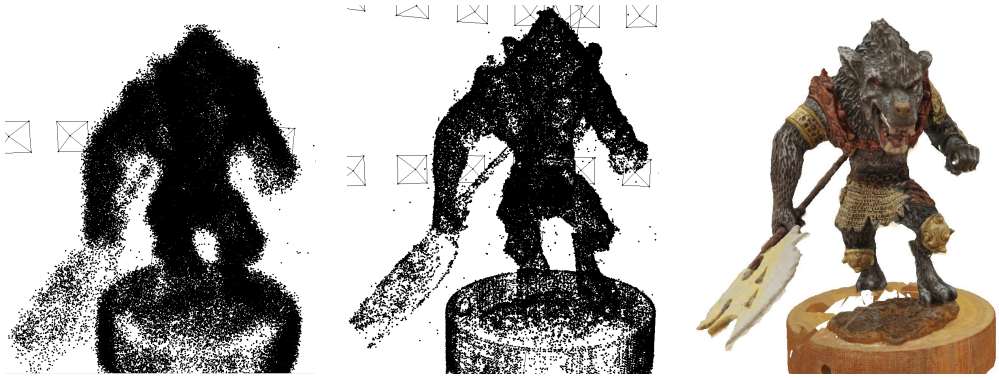

Figure 1: Bundle adjustment in action. After initial reconstruction, the refined parameters can be used to create a textured 3D model using dense reconstruction.

gradients and its effect on performance has been investigated in [3]. In [11], significant performance improvements using multiple techniques, such as embedded point iterations and preconditioned conjugate gradients were shown.

1.1 Motivation

The choice of the camera parameterization has an important impact on BA performance. It directly influences the shape and the number of local minima of the objective function, which is minimized. In gradient based iterative optimization, e.g. in BA, the shape of the function has impact on the reduction of error within iteration, can lead to finding a better local minima or getting stuck in a worse one. Reducing the degrees of freedom can improve the convergence and the conditionality of the Jacobian matrix. Some parameterizations impose constraints on the actual values and therefore require special treatment. Last but not least, it is appealing to aim at a low number of parameters to reduce the computational demand of the BA.

The question which everyone must ask when designing BA parameterization is how to describe camera orientation. We believe this question still remains unanswered, since several parameterizations are being used in various BA softwares and there is no general rule which one to choose.

1.2 Parameterizing camera orientation

A standard perspective camera model, which uses a rotation matrix to describe camera orientation, is described in [7]. The rotation matrix can be parametrized in different ways.

In [20], quaternions [8] are investigated and their advantages and drawbacks are identified as well as a need for parameter scaling in gradient based optimization methods. In [2], authors use unit quaternions and rotation matrices and show how to update the parameters perserving their constraints. We will call this parameterization *4-quaternion* in the rest of the paper.

A non-redundant, local parameterization using a tangential hyperplane to the unit quaternion space, which will be further referred to as *3-quaternion-tangent*, is presented in [16] and compared to the angle/axis representation [4] (further denoted as *angleaxis*), with rather inconclusive results in terms of performance.

In terms of practical applications, well known state of the art BA solvers support or use several different parameterizations. A general graph solver, which can be used for BA, described in [13], uses *4-quaternion*, *3-quaternion-tangent*, rotation matrices and euler angles. Method [14] uses local parameterization, where only three components of the quaternion are optimized, which we will denote as *3-quaternion-local*. Method [18] uses either *angleaxis*, *4-quaternion* or *3-quaternion-tangent* parameterizations. Google Ceres [6] provides support for any projection function and its embedded BA solver offers the *angleaxis*, *4-quaternion* or or *3-quaternion-tangent*.

All parameterizations which were emphasized will be described in greater detail in section 2.2.

1.3 Contribution

In this paper we describe a new way how to parameterize cameras inside BA using non-unit quaternions, while keeeping the dimensionality as low as when using *3-quaternions-local*, *3-quaternions-tangent* or euler angles. Compared to the *angleaxis* representation, quaternions in general are easier to handle in terms of computation, which holds true also in our case. Compared to *3-quaternions-tangent*, there is no need in our case for extra care when updating the parameters. Our parameterization of rotation does not posses singularities as euler angles do and does not have to care about the border of the parameter space as in the case of *3-quaternions-local*. The performance in experiments on real datasets is the same as for other most common parameterizations. We present a global, non-redundant (i.e. minimal) and practical camera parameterization for Bundle Adjustment.

2 Camera parameters

In this section, we show a standard way how to describe a perspective camera and describe common ways how to parameterize camera orientation, which are later used in the experiments. Then we introduce the new parameterization.

2.1 A standard camera parameterization

A perspective camera with radial distortion can be described [7] as follows. A 3D point represented by coordinates $\mathtt{X} \in R^3$ in a Cartesian world coordinate system is transformed to the camera Cartesian coordinate system as

$$\mathtt{Y} = \mathtt{R}(\mathtt{X} - \mathtt{C}) = \begin{bmatrix} \mathtt{r}_1^\top \\ \mathtt{r}_2^\top \\ \mathtt{r}_3^\top \end{bmatrix} (\mathtt{X} - \mathtt{C}) \tag{1}$$

by a 3×3 rotation matrix $\mathtt{R}$ with rows $\mathtt{r}_1, \mathtt{r}_2, \mathtt{r}_3$ and camera center $\mathtt{C} \in R^3$. Then, it is projected to an image Cartesian coordinate system

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\mathtt{r}_1^\top (\mathtt{X} - \mathtt{C})}{\mathtt{r}_3^\top (\mathtt{X} - \mathtt{C})} \\ \frac{\mathtt{r}_2^\top (\mathtt{X} - \mathtt{C})}{\mathtt{r}_3^\top (\mathtt{X} - \mathtt{C})} \end{bmatrix} \tag{2}$$

Assuming that the symmetry axis of the camera optical system is perpendicular to the image plane, radially symmetric "distortion" parameterized by ρ_1 and $\rho_2$is applied

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} (1 + \rho_1 \|\mathtt{d}\|^2 + \rho_2 \|\mathtt{d}\|^4) \text{ with } \mathtt{d} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

and, finally, the result is measured in an image coordinate system

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{4}$$

giving the image coordinates u, v. Parameters $k_{11}, \ldots, k_{23}$ are elements of the camera calibration matrix $\mathtt{K}$ [7].

We parameterize rotation matrix $\mathtt{R}$ by quaternion $\mathtt{q} = [q_1, q_2, q_3, q_4]^\top$ as

$$\mathtt{R} = \frac{S}{\|\mathtt{q}\|^2} \tag{5}$$

with

$$\mathtt{S} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 + q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{6}$$

where rows of $\mathtt{R}$ become $\mathtt{r}_i^\top = \mathtt{s}_i^\top / \|\mathtt{q}\|^2$, $i = 1, 2, 3$ as a function of rows $\mathtt{s}_i^\top$ of $\mathtt{S}$. Matrix $\mathtt{S}$ is parameterized by the quaternion and represents a composition of a rotation and a non-negative scaling.

Let us now observe an interesting fact. We substitute the quaternion parameterization to Eq. 2

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\mathtt{r}_1^\top(\mathtt{X}-\mathtt{C})}{\mathtt{r}_3^\top(\mathtt{X}-\mathtt{C})} \\ \frac{\mathtt{r}_2^\top(\mathtt{X}-\mathtt{C})}{\mathtt{r}_3^\top(\mathtt{X}-\mathtt{C})} \end{bmatrix} = \begin{bmatrix} \frac{\mathtt{s}_1^\top/\|\mathtt{q}\|^2(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top/\|\mathtt{q}\|^2(\mathtt{X}-\mathtt{C})} \\ \frac{\mathtt{s}_2^\top/\|\mathtt{q}\|^2(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top/\|\mathtt{q}\|^2(\mathtt{X}-\mathtt{C})} \end{bmatrix} = \begin{bmatrix} \frac{\mathtt{s}_1^\top(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top(\mathtt{X}-\mathtt{C})} \\ \frac{\mathtt{s}_2^\top(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top(\mathtt{X}-\mathtt{C})} \end{bmatrix} \tag{7}$$

and observe that the size of the quaternion has no effect on the projection. Non-unit quaternions hence give a redundant parameterization of camera rotations [19]. The redundancy is often removed by (i) imposing $\|\mathtt{q}\|^2 = 1$ [19, 20], (ii) using a parameterization that is not completely global [19, 14], or (iii) using a very local parameterization in the tangent space around the identity [19, 18, 1].

2.2 Common parameterizations

4-quaternions

One way to approach the parameterization of rotation matrix R inside BA is to optimize all four elements of the quaternion. This parameterization does not suffer from singularities. It has, however, one extra degree of freedom since, Eq. (7), the magnitude of the quaternion does not have effect on the projection function. Since unit quaternions are subject to $||\mathtt{q}||^2 = 1$ constraint, we need to normalize it to obtain a rotation. Usually, the drawback of having four parameters and extra degree of freedom using quaternion is solved in one of the two following ways.

3-quaternions-tangent

First, it is possible to use a local approximation to the unit quaternion by calculating the tangent space of the unit quaternion manifold at each iteration [16]. When moving in the tangential hyperplane, we obtain a vector v which needs to be projected back onto the unit quaternion manifold.

3-quaternions-local

Another way is to use the fact that $||\mathtt{q}||^2 = 1$, optimize only three components of a quaternion and calculate the remaining component as

$$q_1 = \sqrt{1 - q_2^2 + q_3^2 + q_4^2} \tag{8}$$

This, however, limits us only to rotations by $\langle -\frac{\pi}{2}; \frac{\pi}{2} \rangle$, since it does not allow for negative q_1 and $q_1 = cos(\phi)$. Therefore, it is a common practice to save the initial orientation of a camera before the optimization and then to optimize only the

difference from the initial orientation. This also prevents from dealing with the border of the paramter space in practical situations since a local update is never close to any rotation by 180°.

Angleaxis

A widely used [9, 10, 17] alternative to the quaternion is the *angleaxis* representation of rotations. It describes rotations by a vector $\mathtt{a}$ representing the axis of rotation and the angle ϕ by which to rotate around it. Since only the direction of the rotation axis vector is important, we can use its length to store the rotation angle, such that $\phi = \frac{a}{\|a\|}$. This approach is almost equivalent to using unit quaternions. One difference is that the rotation matrix can be constructed as a polynomial function of a unit quaternion while it is necessary to use transcendental functions, i.e. sin and cos, when constructing the rotation matrix from a given *angleaxis* representation. Another difference is that in the *angleaxis* representation, one either has to deal with the boundary of the parameter space or one has to allow infinite possible representations of a rotation (due to the periodicity of sin and cos).

2.3 Global non-redundant camera parameterization

We will next introduce a new parameterization of a general perspective camera with radial distortion, which is global and it is not redundant. This parameterization can be used in cases where focal length of the camera is one of the parameters being estimated.

The idea is simple. Since $\|\mathtt{q}\|^2$ has no impact on the value of x, y in Eq. 7, we can use it to parameterize *any remaining positive parameter*.

Now, it is always possible to change the coordinate system in images to have $k_{11} > 0$. For instance, assuming intial parameters in the bundle adjustment $\mathtt{q}_0, \mathtt{C}_0, \mathtt{K}_0, \rho_0$, we can choose a new coordinate system in each image with its origin in the principal point [7] and with k_{11} close to 1 by passing form u, v to u', v' by

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} k_{11} & 0 & k_{13} \\ 0 & k_{11} & k_{23} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{k_{12}}{k_{11}} & 0 \\ 0 & \frac{k_{22}}{k_{11}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{9}$$

and from $\mathtt{K}_0$ to

$$\mathtt{K}_0' = \begin{bmatrix} 1 & \frac{k_{12}}{k_{11}} & 0 \\ 0 & \frac{k_{12}}{k_{11}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Notice that this change of the image coordinate system is a similarity transformation, i.e. a composition of a rotation, translation and scaling, and hence it does not change the distribution of image errors.

Now, with such a choice of image coordinate system, it is natural to set $k_{11} = \|\mathtt{q}\|^2$. Since $\mathtt{q}_0$ is initiated from an initial rotation matrix $\mathtt{R}_0$, it has the norm equal to one, i.e. $\|\mathtt{q}_0\|^2 = 1$.

Our camera parameterization can now be written as

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \|\mathtt{q}\|^2 & k'_{12} & k'_{13} \\ 0 & k'_{22} & k'_{23} \end{bmatrix} \begin{bmatrix} x(1+\rho_1\|\mathtt{d}\|^2+\rho_2\|\mathtt{d}\|^4) \\ y(1+\rho_1\|\mathtt{d}\|^2+\rho_2\|\mathtt{d}\|^4) \\ 1 \end{bmatrix} \tag{11}$$

with

$$\mathtt{d} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\mathtt{s}_1^\top(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top(\mathtt{X}-\mathtt{C})} \\ \frac{\mathtt{s}_2^\top(\mathtt{X}-\mathtt{C})}{\mathtt{s}_3^\top(\mathtt{X}-\mathtt{C})} \end{bmatrix} \tag{12}$$

where $\mathtt{s}_i$ are given by Eq. 5 and ρ_1 and ρ_2 are coefficients of the radial distortion model used in [18]. For a typical consumer camera we will get $k'_{12}, \approx 0$ and $k'_{22} \approx 1$.

3 Experiments

We tested the different parameterizations on various real datasets. As a baseline, we used the publicly available datasets from [1]. These datasets consist of individual stages of incremental SfM reconstruction for four different scenarios. In order to speed up the experiments, we limited the amount of datasets while preserving the variety of data. We also added six additional datasets from our own database.

The solver used to perform BA was Ceres from Google [6], which is freely available and implements the state of the art BA techniques to achieve optimal performance.

We compared our new parameterization proposed in section 2.3 to four commonly used parameterizations mentioned in section 2.2. In order to be able to compare how parameterizations converge, we forced all the optimizations to run for 30 iterations. No changes have been observed in additional iterations.

Two versions of experiments were performed. First, without any prior image coordinate normalization and second, using the image normalization described by Eq.(9) and (10). In our case, as in [18, 1], we assumed square pixels, zero skew and the image center to be at $[0,0]^\top$. Therefore, the matrix $\mathtt{K}$ reduces to

$$\mathtt{K}'_0 = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{11} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Parameterization	Parameters	No. par.
Angleaxis	k_{11},a,C	7
4-quaternion	k_{11},q,C	8
3-quaternion-tangent	k_{11},v,C	7
3-quaternion-local	k_{11},q_2,q_3,q_4,C	7
New	q,C	7

Table 1: Parameterizations used in experiments.

We then optimize only k_{11}, i.e. the focal length, which is in our parameterization replaced by $||\mathtt{q}||^2$. The summary of the parameterizations can be found in table 1.

3.1 Results

The results for both normalized and non-normalized data are shown in Figure 2. We compared the evolution of the reprojection error over each iteration as well as its final value. One can see in Figures 2(c) and 2(d) that the new parameterization is converging to the same value of reprojection error as all the other parameterizations. The numbers on the x-axis denote the index of a dataset and the labels separate different datasets. The same behaviour is observed for all the parameterizations, with the exception of several outliers.

The convergence curves are not always identical for different parameterizations, Figures 2(a) and 2(b). We have found that the normalized data are slightly better suited for BA, as suggested in [20], judging from the convergence which was slightly faster and also more correlated between different parameterizations.

The histograms in Figures 2(e) and 2(f) show the relative difference in reprojection error achieved by our parameterization compared to all other parameterizations on all datasets, where by a run we denote a result of one parameterization on one of the datasets. In vast majority of cases our parameterization achieved exactly the same final reprojection error value.

In absence of ground truth data, we compared the resulting parameters of cameras, i.e. the focal lengths, camera centers and orientations only among the tested parameterizations. As in the case of reprojection error, the resulting parameters after BA were exactly the same for all parameterizations. The parameters sometimes differed by a similarity transformation and after registering them, they were the same. Since the quantitative results would not be interesting, we show at least the final reconstruction of one of the datasets using all parameterizations in Figure 3. Original data is labeled by black color and the results using different parameterizations are colored accordingly to previous figures. The results are almost indistinguishable, which was also the case for the rest of the datasets.

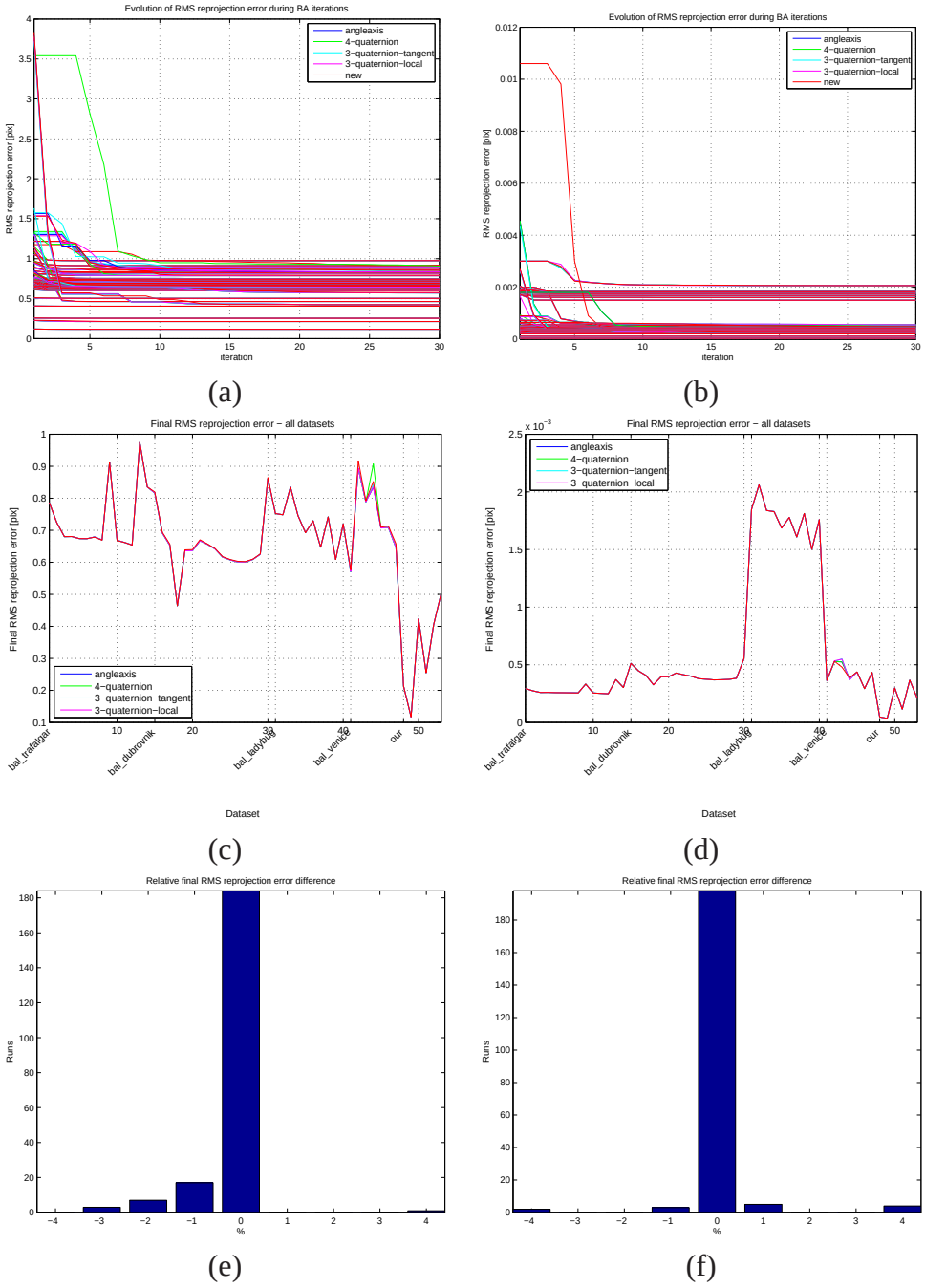

Figure 2: Results using non-normalized (a,c,e) and normalized (b,d,f) data for all datasets. Figures (a) and (b) show the evolution of the reprojection error over the BA iterations for all datasets. Figures (c) and (d) show the final reprojection error over all the datasets. Different data sets are denoted by their name. Figures (e) and (f) contain histograms of the relative difference of the final reprojection error for the new parameterization compared to all other parameterizations.

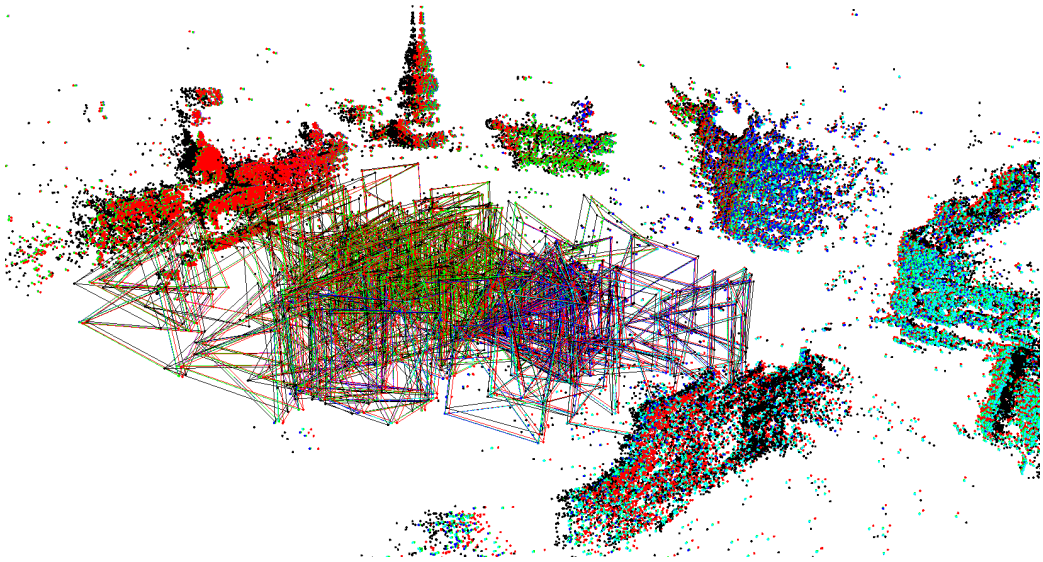

Figure 3: A visualization of a sample from the Trafalgar dataset with 126 cameras after BA using different . The original data is labeled by black color. Results obtained for other parameterizations are colored as in the previous figures.

4 Conclusion

In this paper, we proposed a new, global, and non-redundant (i.e. minimal) parameterization of a perspective camera for the Bundle Adjustment. We discussed the advantages of this parameterization in comparison to other commonly used parameterizations. Experiments evaluating the performance in terms of reducing the reprojection error were conducted on real datasets. The results showed that the proposed parameterization is achieving the same performance as the other investigated parameterizations and therefore we conclude that the new parameterization is a viable and practical option in BA.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. R. Szeliski. Bundle adjustment in the large. In *ECCV (2)*, pages 29–42, 2010.

[2] T. Barfoot, J. R. Forbes, and Paul T. Furgale. Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68:101 – 112, 2011.

[3] M. Byröd and K. Åström. Conjugate gradient bundle adjustment. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV'10, pages 114–127, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson Education, Incorporated, 2005.

[5] D. Crandall, A. Owens, and N. Snavely. Discrete-continuous optimization for large-scale structure from motion. *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, 286(26):3001–3008, 2011.

[6] Google. Ceres solver, 2012.

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

[8] M. Hazewinkel. *Encyclopaedia of Mathematics (1)*. Encyclopaedia of Mathematics: An Updated and Annotated Translation of the Soviet "Mathematical Encyclopaedia". Springer, 1987.

[9] A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. pages 438–443, 1997.

[10] A. Heyden and K. Åström. Flexible calibration: minimal cases for auto-calibration. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 350–355 vol.1, 1999.

[11] Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and In-So Kweon. Pushing the envelope of modern methods for bundle adjustment. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1474–1481, 2010.

[12] K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conference*, Aberystwyth, Wales, 08/2010 2010.

[13] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[14] M. Lourakis and A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36:1–30, 2009.

[15] Microsoft. Photosynth, 2008.

[16] J. Schmidt and H. Niemann. Using quaternions for parametrizing 3-d rotations in unconstrained nonlinear optimization. In *VISION, MODELING, AND VISUALIZATION 2001*, pages 399–406. AKA/IOS Press, 2001.

[17] H. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: a hierarchical approach to multi-frame structure from motion. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –543 Vol. 2, 1999.

[18] N. Snavely. Bundler: Structure from motion (sfm) for unordered image collections. http://phototour.cs.washington.edu/bundler/, 5 2011.

[19] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[20] M. Wheeler and K. Ikeuchi. Iterative estimation of rotation and translation using the quaternion, 1995.