Efficient Computation of Relative Pose for Multi-Camera Systems

Laurent Kneip and Hongdong Li
Research School of Engineering, Australian National University
laurent.kneip@anu.edu.au and hongdong.li@anu.edu.au

Abstract

We present a novel solution to compute the relative pose of a generalized camera. Existing solutions are either not general, have too high computational complexity, or require too many correspondences, which impedes an efficient or accurate usage within Ransac schemes. We factorize the problem as a low-dimensional, iterative optimization over relative rotation only, directly derived from well-known epipolar constraints. Common generalized cameras often consist of camera clusters, and give rise to omni-directional landmark observations. We prove that our iterative scheme performs well in such practically relevant situations, eventually resulting in computational efficiency similar to linear solvers, and accuracy close to bundle adjustment, while using less correspondences. Experiments on both virtual and real multi-camera systems prove superior overall performance for robust, real-time multi-camera motion-estimation.

1. Introduction

One of the most fundamental problems in structure from motion consists of the computation of the relative pose between two camera viewpoints. The present paper deals with the challenge of extending relative pose computation to generalized cameras. By a *generalized camera*, we understand an imaging device where the spatial rays that correspond to interest points in the image are no longer concurrent. Generalized—or non-central—cameras can be identified in various practically relevant cases. For instance, a camera looking at an arbitrarily shaped mirror forms an imaging system where the observation rays pointing at 3D points no longer meet at a single point. In other words, such a system no longer permits the identification of a *camera projection center*. Another popular case—and the main subject of the present paper—is given by vehicle-mounted multi-camera systems pointing into all directions. As illustrated in Figure 1, such systems have a potentially larger field of view, and—given that the distance between the cameras is known in correct scale—produce metric results.

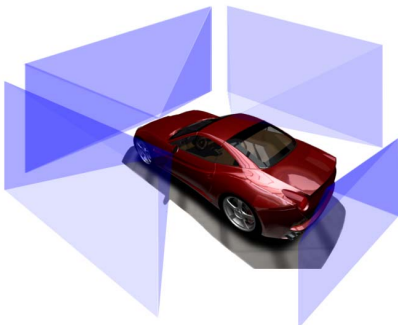


Figure 1. Example of a multi-camera system mounted on a car.

From a more abstract and geometric point of view, a generalized camera consists of a Euclidean reference frame in which measurements are represented by rays in space, described by a suitable parametrization such as Plücker line vectors. The generalized camera as such represents a calibrated case, where the intrinsic parameters of the imaging system are hidden from the problem by first having the interest points in the image space transformed into spatial rays. A generalized camera can therefore describe any calibrated imaging device. Even if there exist less general—but potentially suited—models such as *General Linear Cameras* [17], we will in this paper use the fully generalized approach as outlined in [16], which—to the best of our knowledge—presents the only minimal solver to the generalized relative pose problem known to date. This solver is based on the Gröbner basis theory, and uses 6 ray-correspondences in order to come up with 64 solutions.

While the generalized camera model is in theory able to handle plenoptic functions with ultimate complexity, practically relevant cases such as mirror-based omni-directional cameras and multi-camera systems are usually less "chaotic". However, [13] provides an extension to the original linear solver presented in [14], and demonstrates that such regularity can also lead to singularities if not handled properly [8]. The linear solution requires 17 correspondences in general, and 16 or 14 correspondences in certain special situations such as multi-camera systems where the camera centers are aligned. Similar *specialized* solutions

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.64

have been presented in [3] and [6] for systems with two or three rigidly attached central cameras. The latter work is also summarized in [7] along with a geometrically optimal L_∞-solution, which is however not efficient, and thus not suited for real-time motion estimation. Furthermore, [12] presents an algorithm that is specialized for pure Ackermann motion.

Robust real-time estimation is gaining practical relevance in the robotics and automotive industry, where vehicles show an increasing number of onboard cameras. A practical example would be the vehicle in Figure 1, which contains one camera in the front, one in the back, and two in the side-mirrors. We therefore need to restrict ourselves to solutions that are able to handle an arbitrary number of cameras. There currently exist two solutions, which we both consider unsatisfying for inclusion into Ransac [4]:

- The 6-point solver presented in [16]: This algorithm has high computational complexity, and—having to disambiguate 64 solutions for each hypothesis—leads to high overall time consumption.
- The 17-point solver presented in [13]: This algorithm has poor noise resilience, and leads to extreme numbers of iterations for increasing outlier ratios.

Although it is known that 7 points are enough to compute a unique solution, we know of no efficient and general way to combine them in case we have an arbitrary number of cameras and evenly distributed observations (less than 5 in each camera). We present a solution to this problem that is based on a recently proposed strategy for efficient, iterative rank minimization directly over the three parameters of the frame-to-frame rotation [10]. The optimization procedure is well-conditioned in the case of omni-directional observations, and therefore tailored to the above mentioned multi-camera systems. In this context, our algorithm combines a number of advantages:

- Each step in the optimization has constant computational complexity, independently of the number of features. The computational efficiency is similar to [13].
- Even if using less than half of the points, our solver drastically outperforms in terms of noise resilience.
- Ability to compute a unique solution from any number of points bigger than 6, any number of cameras, and any distribution across the cameras. Unlike the generalized essential matrix, the solution does not degenerate for special, multi-camera configurations.

Section 2 summarizes our method and its origins in epipolar geometry and direct optimization of frame-to-frame rotation [10]. Section 3 presents a comparison in terms of computational efficiency and noise resilience, as well as the overall gain in performance when embedded into Ransac. Section 4 finally demonstrates the practical usefulness in real-world scenarios.

2. Theory

We will in the following summarize the origins of the approach in the central case, the extension to generalized cameras, and the details of the minimization of the resulting cost function.

2.1. Origins in epipolar geometry

Relative pose in the calibrated case is constrained by the geometry of two viewpoints, called *epipolar geometry*. Let $\mathbf{f}_i$ and $\mathbf{f}'_i$ be unit vectors pointing at the same 3D point from different viewpoints. The epipolar constraint is given by

$$\mathbf{f}_i^T \mathbf{E} \mathbf{f}'_i = 0 \Leftrightarrow \mathbf{f}_i^T [\mathbf{t}]_\times \mathbf{R} \mathbf{f}'_i = 0, \tag{1}$$

where $\mathbf{E}$ represents the essential matrix, $\mathbf{t}$ the position of viewpoint 2 w.r.t. viewpoint 1, and $\mathbf{R}$ the rotation from viewpoint 2 back to viewpoint 1 [5]. This can be rewritten as the scalar triple product $\mathbf{f}_i \cdot (\mathbf{t} \times (\mathbf{R}\mathbf{f}'_i))$.

Definition: *The scalar triple product (or mixed product) between three vectors is defined to be the dot-product between one of the vectors and the cross product of the other two. Geometrically, the scalar triple product is equal to the signed volume of the parallelepiped that is spanned by the three vectors.*

The epipolar constraint therefore forces the volume of the parallelepiped defined by $\mathbf{f}_i$, $\mathbf{t}$, and $\mathbf{R}\mathbf{f}'_i$ to be zero (e.g., it forces the vectors to be coplanar). The outlined geometrical intuition makes it easy to see that swapping the vectors in an arbitrary way can at most change the sign of the computed volume, but never the volume itself, which is given by the absolute value of the signed triple product. The sign of the triple product reflects the handedness of the coordinate frame defined by the three vectors. Any cyclic permutation of the three vectors of a scalar triple product therefore does not change its value at all, while any acyclic permutation of the three vectors returns its negative. In conclusion, we can easily change the original epipolar constraint (1) into various other forms, one of which is given by

$$-\mathbf{t} \cdot (\mathbf{f}_i \times (\mathbf{R}\mathbf{f}'_i)) = -(\mathbf{f}_i \times (\mathbf{R}\mathbf{f}'_i)) \cdot \mathbf{t} = -([\mathbf{f}_i]_\times \mathbf{R}\mathbf{f}'_i)^T \mathbf{t} = 0. \tag{2}$$

Let's define the vector $\mathbf{n}_i = [\mathbf{f}_i]_\times \mathbf{R}\mathbf{f}'_i$. The translation $\mathbf{t}$ needs to lie in the nullspace of any $\mathbf{n}_i^T$. Supposing that we have five correspondence pairs, we obtain the constraint

$$\mathbf{N}^T \mathbf{t} = \begin{pmatrix} \mathbf{n}_1 & \dots & \mathbf{n}_5 \end{pmatrix}^T \mathbf{t} = 0 \tag{3}$$

The trivial solution $\mathbf{t} = 0$ is not allowed, which means that any 3×3 sub-matrix formed by stacking three different rows of $\mathbf{N}^T$ needs to have zero determinant. Now considering the fact that the introduced vectors $\mathbf{n}_i$ are nothing but normal vectors to epipolar planes as introduced in [11],

we have easily derived the translation independent epipolar plane normal coplanarity constraint via algebraic ways. An interesting observation is that the number of possible determinant constraints that can be formed from a minimum of 5 $(\mathbf{f}_i, \mathbf{f}'_i)$-correspondences equals to $C_5^3 = 10$, which—ignoring additional constraints that might be required for non-minimal parametrizations—coincides with the maximum number of algebraically independent constraints found in previous works on relative pose.

The recent work presented in [10] extends this idea to an arbitrary number of correspondences n by converting it into a rank minimization approach. The idea is that multiple epipolar plane normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$, ..., $\mathbf{n}_n$ must remain coplanar. In other words, the matrix $\mathbf{N} = (\mathbf{n}_1 \quad \mathbf{n}_2 \quad ... \quad \mathbf{n}_n)$ must be rank-defficient, and hence the smallest Eigenvalue of $\mathbf{M} = \mathbf{N}\mathbf{N}^T$ must be zero. One of the core contributions of [10] consists of a factorization of the rotation matrix $\mathbf{R}$ inside the 3×3-matrix $\mathbf{M}$, proving that the smallest Eigenvalue can be computed very efficiently and in closed-form as a function of the rotation only. The computational complexity remains independent of the number of correspondences, which is subsequently exploited in order to come up with a very efficient inter-frame rotation optimization scheme.

While a direct optimization of the rotation still bares a certain—managable—risk of convergence to local minima, the basin of attraction becomes fairly large in case of correspondences originating from omni-directional measurements, rendering direct rotation optimization a valid approach. A non-central multi-camera system similar to the one illustrated in Figure 1 naturally gives rise to omnidirectional correspondences. The remainder of this paper therefore unlocks the full potential of the idea by presenting a powerful extension to generalized cameras.

2.2. Generalization

The measurements in the generalized case can be elegantly expressed by Plücker line-vectors. A Plücker-vector is a 6-vector of which the first three entries correspond to the direction vector of the ray, and the latter three to the corresponding line's moment, which is given by taking the cross-product of a point on the line and the line's direction. As outlined in [14], the transformation rule as well as the intersection-constraint of Plücker line-vectors easily leads to the generalized epipolar constraint

$$\mathbf{l}_i^T \begin{pmatrix} \mathbf{E} & \mathbf{R} \\ \mathbf{R} & \mathbf{0} \end{pmatrix} \mathbf{l}'_i = 0, \tag{4}$$

where $(\mathbf{l}_i, \mathbf{l}'_i)$ denotes a pair of corresponding Plücker line-vectors pointing at the same 3D point from two different generalized cameras. Similar to the central case, this formulation allows us to solve linearly for the relative pose. However—unlike the central case—the linear solution is

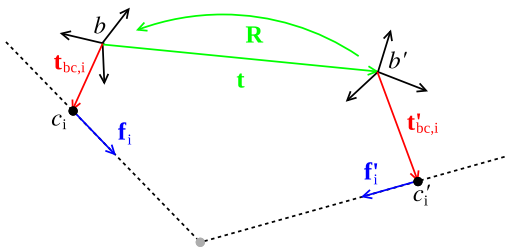


Figure 2. Geometry of the generalized relative pose problem for multi-camera systems. The unknowns are the transformation parameters between the two viewpoints b and b', given by $\mathbf{t}$ and $\mathbf{R}$. The measured or known variables are the landmark observation vectors $\mathbf{f}_i$ and $\mathbf{f}'_i$ and the position of the camera centers c_i and c'_i with respect to b and b', given by $\mathbf{t}_{bc,i}$ and $\mathbf{t}'_{bc,i}$, respectively.

only possible via a massively redundant parametrization, and requires 17 correspondences for solving only 6 DoF [13].

As illustrated in Figure 2, in the case of a multi-camera system, a point on each Plücker-line is easily given by the capturing camera's center c_i, seen from the origin of the multi-camera system b. If denoting this displacement by $\mathbf{t}_{bc,i}$, our Plücker vector results to

$$\mathbf{l}_i = \begin{pmatrix} \mathbf{f}_i \\ \mathbf{t}_{bc,i} \times \mathbf{f}_i \end{pmatrix}. \tag{5}$$

Note that we assume that—without loss of generality—c and b have identical orientation. Substituting (5) in (4), we easily arrive at the following, alternative generalized epipolar constraint:

$$\mathbf{f}_i^T \mathbf{E} \mathbf{f}'_i + \mathbf{f}_i^T (\mathbf{R}[\mathbf{t}'_{bc,i}]_\times - [\mathbf{t}_{bc,i}]_\times \mathbf{R}) \mathbf{f}'_i = 0. \tag{6}$$

By again using (1) to represent the essential matrix as a function of $\mathbf{t}$ and $\mathbf{R}$, and applying the permutation rule for triple scalar products outlined in (2), we arrive at

$$(\mathbf{f}_i \times \mathbf{R}\mathbf{f}'_i)^T \mathbf{t} + \mathbf{f}_i^T ([\mathbf{t}_{bc,i}]_\times \mathbf{R} - \mathbf{R}[\mathbf{t}'_{bc,i}]_\times) \mathbf{f}'_i = 0, \tag{7}$$

which can be easily rewritten as

$$\mathbf{g}_i^T \tilde{\mathbf{t}} = 0 \text{ , with} \tag{8}$$

$$\mathbf{g}_i = \begin{pmatrix} \mathbf{f}_i \times \mathbf{R}\mathbf{f}'_i \\ \mathbf{f}_i^T ([\mathbf{t}_{bc,i}]_\times \mathbf{R} - \mathbf{R}[\mathbf{t}'_{bc,i}]_\times) \mathbf{f}'_i \end{pmatrix} \text{ and } \tilde{\mathbf{t}} = \begin{pmatrix} w\mathbf{t} \\ w \end{pmatrix},$$

where $\mathbf{g}_i$ is called a *generalized epipolar plane normal vector*, and $\tilde{\mathbf{t}}$ the *homogeneous translation vector*, which has arbitrary scale. Note however that $\mathbf{t} = \frac{1}{w}\tilde{\mathbf{t}}$ is now completely defined, including metric scale. If having n generalized normal vectors–each one being a function of an individual quadruplet $(\mathbf{f}_i, \mathbf{f}'_i, \mathbf{t}_{bc,i}, \mathbf{t}'_{bc,i})$—, we obtain the constraint

$$\mathbf{G}^T\tilde{\mathbf{t}} = \begin{pmatrix}\mathbf{g}_1 & \dots & \mathbf{g}_n\end{pmatrix}^T \tilde{\mathbf{t}} = 0. \quad (9)$$

We can easily observe that this expression now constrains $\tilde{\mathbf{t}}$ by a $n \times 4$ matrix that again depends on the rotation only. Since the trivial solution is not allowed, the determinant of each 4×4 submatrix of $\mathbf{G}$ needs to vanish, which gives rise to translation independent constraints on the rotation. We know from earlier works that the minimum number of required correspondences for solving the generalized relative pose problem amounts to 6. Interestingly, the maximum number of algebraically independent 4×4 submatrices equals to $C_6^4 = 15$, which again agrees exactly with the findings in [16]. Furthermore, $\mathbf{G}$ has at most rank 3. Given an arbitrary number of correspondences n, we can thus again perform a rank minimization of $\mathbf{G}$ over $\mathbf{R}$ by minimizing the smallest Eigenvalue of

$$\mathbf{H} = \mathbf{G}\mathbf{G}^T = \sum_{i=1}^{n} \mathbf{g}\mathbf{g}^T. \quad (10)$$

2.3. Optimization

If $\lambda_{\mathbf{H},min}$ denotes the smallest Eigenvalue of $\mathbf{H}$, the optimization problem is given by

$$\mathbf{R} = \text{argmin}_{\mathbf{R}} \lambda_{\mathbf{H},min}. \quad (11)$$

Similarly to [10], the rotation matrix $\mathbf{R}$ can be factorized inside the expression for $\mathbf{H}$. The equations for the constant complexity composition of $\mathbf{H}$—independently of the number of correspondences n—can be found in Appendix A. $\mathbf{H}$ is a positive-definite matrix, and its Eigenvalues are therefore positive. They are given by the roots of $\det(\mathbf{H} - \lambda\mathbf{I}_{4\times4})$, and the closed-form solution for the smallest root of this fourth order polynomial $\lambda_{\mathbf{H},min}$ is given in Appendix B. If chosing the Cayley parameters $\mathbf{v} = \begin{pmatrix}x & y & z\end{pmatrix}^T$ such that

$$\mathbf{R} = 2(\mathbf{v}\mathbf{v}^T - [\mathbf{v}]_\times) + (1 - \mathbf{v}^T\mathbf{v})\mathbf{I}, \quad (12)$$

the optimization problem is solved by minimizing a cost expressed by $\lambda_{\mathbf{H},min}$ in a 3-dimensional space of rotations, each step having constant computational complexity. The above rotation matrix has wrong scale, which does however not affect the validity of the homogeneous constraint.

An example cost function is illustrated in Figure 3. One can easily observe that there is a second minimum besides the true minimum in the plane $z \approx -0.2$, namely at $\mathbf{v} = (0,0,0)^T$. An analysis of the algebraic constraint quickly reveals that if $\mathbf{v} = (0,0,0)^T$ and $\mathbf{t}_{bc,i} = \mathbf{t}'_{bc,i}$—which is the case in multi-camera situations where correspondences always arise from the same camera in both viewpoints—the fourth component of $\mathbf{g}_i$ is always zero, and hence $\mathbf{H}$ is always rank-deficient. Moreover, the presence of local minima is of course not completely impossible, which is why a

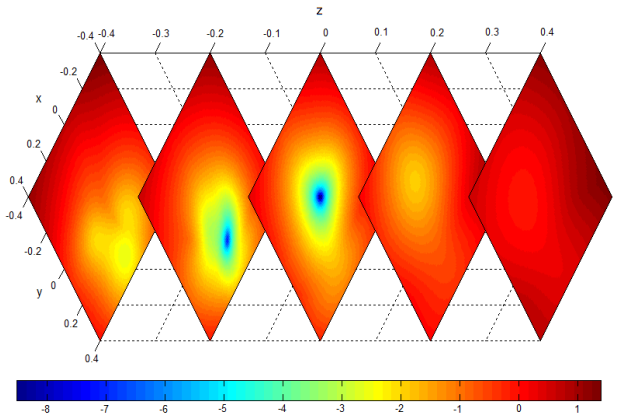


Figure 3. Example of a cost-volume defined by $\lambda_{\mathbf{H},min}$ over the 3-dimensional Cayley-space of rotations. For the sake of better visility, the figure displays the log-value of the cost.

blind application of gradient descent does not return satisfying performance. Our optimization strategy is as follows:

- Find a good starting value by first ignoring the effects of the distance between the cameras and the translation, and compute an initial guess based on a pure-rotation model. We apply the method presented in [1] to our unit bearing vectors as if they were originating from a central viewpoint.
- We then apply gradient descent to $\lambda_{\mathbf{H},min}$ based on an efficient numerical computation of all partial derivatives by the Cayley parameters.
- In case we end up close to the origin, we consider the second smallest Eigenvalue $\lambda_{\mathbf{H},sec}$. If $(0,0,0)$ (e.g., identity) is a solution, the translation becomes unobservable, and $\mathbf{H}$ needs to have at most rank 2. In other words, $\lambda_{\mathbf{H},sec}$ needs to be zero as well. If this is not the case, we know that we converged to a wrong minimum, and rerun gradient descent with a slightly perturbed initial value.

The proposed strategy works well for problems with multi-directional correspondences. The ratio between $\lambda_{\mathbf{H},sec}$ and $\lambda_{\mathbf{H},min}$ tells us how well the scale of the problem is defined, an information that is hard to extract from alternative solutions. Note that the Eigenvector corresponding to the smallest Eigenvalue returns the translation. Also note that—in comparison to [10]—the implementation of a Levenberg-Marquardt scheme has been omitted, due to numerical inaccuracies of the quickly increasing computational complexity. Finally, a bound on the variation of the smallest Eigenvalue based on interval arithmetics can be derived much in the same way this has been done in [10], which however will be very conservative and thus of limited usefulness.

3. Application to a 4-camera system

Our iterative rank minimization approach works best in case of an omni-directional distribution of observations. Our solver is therefore very well suited for motion estimation with multi-camera systems, where the cameras are pointing into opposite directions. We restrict our evaluation to such systems, and present results on noise resilience, accuracy of the initial guess, computational efficiency, and overall performance within a random sample consensus scheme.

3.1. Outline of the experiments

The multi-camera system we investigate is illustrated in Figure 4. It contains 4 cameras in a body frame b such that they are shifted by 1m along the positive and negative directions of the x and y axes. We create random problems by first—without loss of generality—fixing the position of the first viewpoint to the origin of the world frame, and its orientation to identity. The position of the second viewpoint is set randomly such that the magnitude of the distance to the first viewpoint does not exceed 2.0m. The rotation is bounded such that none of the Euler angles exceeds 0.5 rad (30 deg). This creates random transformations as they would appear in practical online motion estimation scenarios. Random points are created for each camera individually with a uniformly varying depth of 5.0m to the origin of viewpoint 1, and then shifted by the respective camera's position vector inside b multiplied by a factor of 10. This ensures that the average ratio between the depth of a point and the baseline of a camera stays around 10, which agrees with the practical application in mind. Noise is added to the measurements by extracting the orthogonal plane of each bearing vector, and adding noise based on a virtual spherical camera with focal length 800 pixels. Since all algorithms compute at least the rotation, errors are expressed in terms of the norm of the difference between the axis-angle representation of the ground truth and the recomputed rotation. Outliers are added by resetting unit vectors $\mathbf{f}'_i$ such that they point into the direction of randomly generated landmarks.

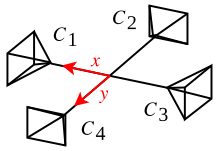


Figure 4. A regular 4-camera system.

The comparison algorithms are given by alternative generalized solutions that are able to handle an arbitrary number of cameras. They are given by the six-point algorithm presented in [16], and the 17-point algorithm presented in [13]. We use 2, 2, 1, and 1 points in each camera for the six-point algorithm (6pt), 2, 2, 2, and 2 for our generalized eigenvalue minimization framework (ge), and 5, 4, 4, and 4 points for the 17-point algorithm (17pt).

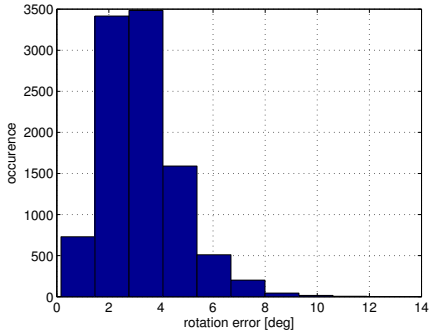


Figure 5. Histogram of errors of our initial guess based on a pure-rotation model. The experiment has been repeated 10000 times.

3.2. Accuracy of the initial guess

Figure 5 shows the error in rotation when assuming a pure rotation situation, and ignoring the effects of the distance between the cameras and the translation. We simply apply [1] on all $(\mathbf{f}_i, \mathbf{f}'_i)$-pairs, and observe that the error generally remains roughly below 3 degrees, which brings us close enough to the global minimum for most of the time. The method of course only works well in case of an omni-directional distribution of the observations.

3.3. Noise resilience

We evaluate each algorithm for various noise levels reaching from 0 to 5 pixels and 1000 random experiments per noise level. We also include the result obtained from nonlinear optimization, which illustrates the optimium we can achieve for a certain noise level using geometric error minimization. The initial guess for the nonlinear optimization is set by a small perturbation of the ground truth value, ensuring that we converge to the global minimum. The initial guess for (ge) is found automatically using the above mentioned method. We also include (ge) over all 17 points to compare the minimal algebraic error with the minimal geometric one.

The mean and median error of all methods is shown in Figure 6. As expected, non-linear iterative minimization of the reprojection error outperforms all other solutions. We also note that the median error of (ge) clearly outperforms both (6pt) and (17pt), and stays very close to the geometric minimum obtained from nonlinear optimization (e.g., bundle adjustment). Moreover—even though using most points—the linear solver has poor noise resilience in comparison to the nonlinear methods. The best solution for (6pt) is each time selected based on a comparison to ground truth, whereas the other methods simply return a unique solution.

The behavior of the mean error is not much different. It however shows that—in case of using only 8 correspondences—there is a residual error for (ge) at zero noise, which indicates occasional convergence into local

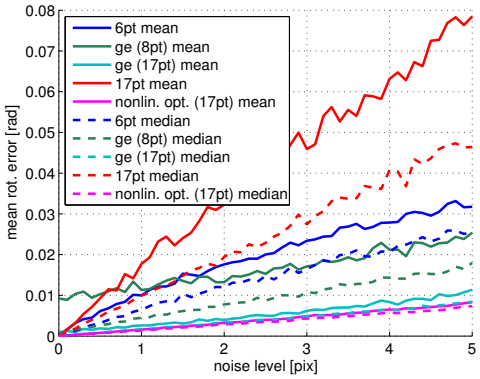


Figure 6. Mean and median error in rotation for different generalized relative pose methods and measurement noise between 0 and 5 pixels. (6pt) denotes the method presented in [16], (ge) the generalized eigenvalue minimization framework presented in this paper, and (17pt) the linear solver presented in [13].

minima. The probability of converging to a local minimum can be easily reduced by further tuning of the optimization parameters, however going to the cost of the computational efficiency. We opted for higher efficiency based on the fact that these algorithms are likely to be embedded into a random sample consensus scheme, which includes a model-verification step that eventually identifies convergence into wrong minima based on a low inlier-ratio. In other words, it is more efficient to drop the occasional hypotheses that suffered from wrong convergence via the regular model-verification step, than slowing the computation of each and every hypothesis generation just to avoid these occasional dropouts.

3.4. Computational efficiency

We tested the execution time by again averaging over 1000 random experiments. All algorithms are implemented in C++, and we reused the original code whenever available. (6pt) is very inefficient and is slower than (17pt) by almost two orders of magnitude, whereas (ge) consumes only 3 times the time of (17pt). The mean execution times are 8.95ms (6pt), 0.28 ms (ge), and 0.09 ms (17pt). The median execution times remain close to these values, proving that the iterative nature of our algorithm does not bare a risk for occasionally slow convergence.

3.5. Overall performance within Ransac

The most relevant performance measure consists of testing all algorithms as part of a random sample consensus framework. We use the classical Ransac approach presented in [4], and the same model verification for all three methods. The noise is kept at 0.5pixels. Our implementation maintains a maximally homogeneous sampling of points across the images. For (6pt), we use an additional 3 points per hypothesis to disambiguate the solution multiplicity. This has no effect on the cost of the disambiguation, and is safer than disambiguation with only one point, especially regarding the high number of solutions and the cost of hypothesis generation.

It is a well-known fact that the number of required iterations increases as the number of required correspondences for generating a hypothesis is growing. Figures 7(a) and 7(b) show the theoretical and practical evolution of the required iterations as a function of the outlier fraction, proving that (17pt) can potentially become very expensive. The most relevant evaluation criterion however consists of the overall execution time, which combines the number of iterations with the efficiency of each model computation. The result is indicated in Figure 7(c), showing that (6pt) remains the slowest method. The literal explosion of the overall computation time made it very difficult to analyze outlier fractions beyond 30%. More interestingly, we can also observe that in practice, although (17pt) is the most efficient method, (ge) performs most gracefully when moving to higher outlier ratios. The difference is increasing as the noise level goes up.

4. Results on a real multi-camera system

In order to demonstrate the practical usefulness of our algorithm in real-world scenarios, we applied it to a sequence of images captured by the custom-made, synchronized multi-camera system illustrated in Figure 8. It consists of two synchronized global shutter WVGA cameras pointing into opposite directions, and we apply simple frame-to-frame matching in each camera individually. A threshold on the median disparity then triggers the computation of generalized relative pose. We use Ransac with 4 correspondences in each camera, followed by interleaving two-view bundle adjustment (with a generalized point-projection equation) over the identified inlier subset. Our system uses FAST corners [15] and BRIEF descriptors [2], thus consuming only 80ms in average per multi-camera relative pose computation (e.g., extraction, matching, disparity computation, Ransac, and non-linear refinement). The concatenation of all relative rotations including a comparison to ground truth obtained by a Vicon motion capture system is indicated in Figure 9. Although the magnitude of the translation is occasionally unobservable (e.g. in situations with almost identity rotation), we note that this simple frame-to-

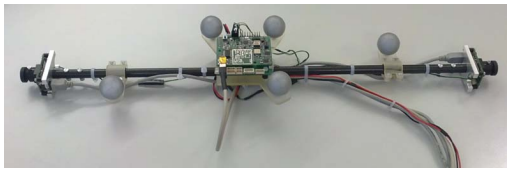
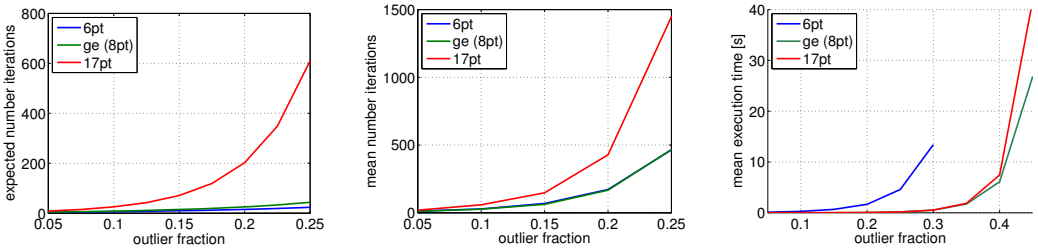


Figure 8. Custom multi-camera system.



(a) Theoretically required number of iterations as a function of outlier fraction. (b) Experimentally required number of iterations as a function of outlier fraction. (c) Experimentally required time as a function of outlier fraction.

Figure 7. Overall performance of the different generalized relative pose methods included in a Ransac scheme [4] (viewed best in color).

frame tracking scheme robustly tracks the rotation of the system. The small drift over time indicates the good accuracy of the estimation of relative rotation. We were unable to reconstruct similar performance with the 17-point algorithm presented in [13], which is related to the degeneracy of linear solvers in the two-camera case.

5. Discussion

The present paper follows the current trend in geometric vision of proposing a novel factorization of a fundamental structure-from-motion problem. The idea behind such works often consists of a theoretical benefit such as global optimality or simply a new way of looking at a problem, however often going to the cost of increased complexity. The low-dimensional factorization of the generalized relative pose problem presented here is different in the sense of providing superior computational efficiency in practically relevant situations. The combination of constant-complexity of each optimization step as well as a relatively low number of required correspondences leads to an accurate compromise between minimal and linear solvers outperforming in the context of random sample consensus schemes. It thus reinforces the importance of direct iterative optimization in algebraic geometry. The algorithm remains general in that it permits any combination of correspondences, cameras, and distribution. Future efforts consist of analysing the possibility of a closed-form solution, as well as an extension of the real-time pipeline to a full-scale, generic multi-camera structure-from-motion system.

Note: All algorithms are publically available through the OpenGV library [9].

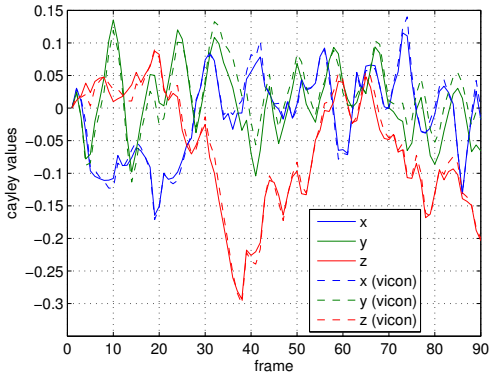


Figure 9. Evolution of the cayley parameters (x, y, z) for motion of a real multi-camera rig (viewed best in color). (vicon) denotes the ground-truth value obtained by a Vicon motion capture system.

ACKNOWLEDGMENT

The research leading to these results has received funding from ARC grants DP120103896 and DP130104567.

APPENDIX A

This section presents the composition of the 4×4 matrix $\mathbf{H}$ in constant time, assuming that the following elements are precomputed (they depend on known or measured variables only, and may be computed in linear time).

$$\mathbf{F}_{xx} = \sum_{i=1}^{n} f_{x,i}^2 \mathbf{f}_i' \mathbf{f}_i'^T \quad , \mathbf{F}_{xy} = \sum_{i=1}^{n} f_{x,i} f_{y,i} \mathbf{f}_i' \mathbf{f}_i'^T$$

$$\mathbf{F}_{xz} = \sum_{i=1}^{n} f_{x,i} f_{z,i} \mathbf{f}_i' \mathbf{f}_i'^T \quad , \mathbf{F}_{yy} = \sum_{i=1}^{n} f_{y,i}^2 \mathbf{f}_i' \mathbf{f}_i'^T$$

$$\mathbf{F}_{yz} = \sum_{i=1}^{n} f_{y,i} f_{z,i} \mathbf{f}_i' \mathbf{f}_i'^T \quad , \mathbf{F}_{zz} = \sum_{i=1}^{n} f_{z,i}^2 \mathbf{f}_i' \mathbf{f}_i'^T$$

$$\mathbf{F}_{x1} = \sum_{i=1}^{n} f_{x,i} \mathbf{f}_i' \mathbf{f}_i'^T [\mathbf{t}'_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i^T \end{pmatrix}$$

$$\mathbf{F}_{y1} = \sum_{i=1}^{n} f_{y,i} \mathbf{f}_i' \mathbf{f}_i'^T [\mathbf{t}'_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i^T \end{pmatrix}$$

$$\mathbf{F}_{z1} = \sum_{i=1}^{n} f_{z,i} \mathbf{f}_i' \mathbf{f}_i'^T [\mathbf{t}'_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i^T \end{pmatrix}$$

$$\mathbf{F}_{x2} = \sum_{i=1}^{n} f_{x,i} \mathbf{f}_i' \mathbf{f}_i^T [\mathbf{t}_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i'^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i'^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i'^T \end{pmatrix}$$

$$\mathbf{F}_{y2} = \sum_{i=1}^{n} f_{y,i} \mathbf{f}_i' \mathbf{f}_i^T [\mathbf{t}_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i'^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i'^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i'^T \end{pmatrix}$$

$$\mathbf{F}_{z2} = \sum_{i=1}^{n} f_{z,i}\mathbf{f}'_i\mathbf{f}_i^T[\mathbf{t}_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i'^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i'^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i'^T \end{pmatrix}$$

$$\mathbf{F}_{11} = \sum_{i=1}^{n} \begin{pmatrix} \mathbf{f}_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i \end{pmatrix} [\mathbf{t}'_{bc,i}]_\times \mathbf{f}'_i\mathbf{f}_i'^T [\mathbf{t}'_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i^T \end{pmatrix}$$

$$\mathbf{F}_{12} = \sum_{i=1}^{n} \begin{pmatrix} \mathbf{f}'_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}'_i \end{pmatrix} [\mathbf{t}_{bc,i}]_\times \mathbf{f}_i\mathbf{f}_i'^T [\mathbf{t}'_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i^T \end{pmatrix}$$

$$\mathbf{F}_{22} = \sum_{i=1}^{n} \begin{pmatrix} \mathbf{f}'_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}'_i \end{pmatrix} [\mathbf{t}_{bc,i}]_\times \mathbf{f}_i\mathbf{f}_i^T [\mathbf{t}_{bc,i}]_\times \begin{pmatrix} \mathbf{f}_i'^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_i'^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i'^T \end{pmatrix}$$

If $\mathbf{r}_i$ denotes row i of $\mathbf{R}$, $\mathbf{c}_i$ column i of $\mathbf{R}$, $\mathbf{r} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix}$, and $\mathbf{c} = \begin{pmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T & \mathbf{c}_3^T \end{pmatrix}^T$, the elements of the symmetric matrix $\mathbf{G}$ are finally given by

$$\begin{aligned}
g_{11} &= \mathbf{r}_3\mathbf{F}_{yy}\mathbf{r}_3^T - 2\mathbf{r}_3\mathbf{F}_{yz}\mathbf{r}_2^T + \mathbf{r}_2\mathbf{F}_{zz}\mathbf{r}_2^T \\
g_{12} &= \mathbf{r}_3\mathbf{F}_{yz}\mathbf{r}_1^T - \mathbf{r}_3\mathbf{F}_{xy}\mathbf{r}_3^T - \mathbf{r}_2\mathbf{F}_{zz}\mathbf{r}_1^T + \mathbf{r}_2\mathbf{F}_{xz}\mathbf{r}_3^T \\
g_{13} &= \mathbf{r}_3\mathbf{F}_{xy}\mathbf{r}_2^T - \mathbf{r}_3\mathbf{F}_{yy}\mathbf{r}_1^T - \mathbf{r}_2\mathbf{F}_{xz}\mathbf{r}_2^T + \mathbf{r}_2\mathbf{F}_{yz}\mathbf{r}_1^T \\
g_{14} &= \mathbf{r}_3\mathbf{F}_{y1}\mathbf{c} + \mathbf{r}_3\mathbf{F}_{y2}\mathbf{r}^T - \mathbf{r}_2\mathbf{F}_{z1}\mathbf{c} - \mathbf{r}_2\mathbf{F}_{z2}\mathbf{r}^T \\
g_{22} &= \mathbf{r}_1\mathbf{F}_{zz}\mathbf{r}_1^T - 2\mathbf{r}_1\mathbf{F}_{xz}\mathbf{r}_3^T + \mathbf{r}_3\mathbf{F}_{xx}\mathbf{r}_3^T \\
g_{23} &= \mathbf{r}_1\mathbf{F}_{xz}\mathbf{r}_2^T - \mathbf{r}_1\mathbf{F}_{yz}\mathbf{r}_1^T - \mathbf{r}_3\mathbf{F}_{xx}\mathbf{r}_2^T + \mathbf{r}_3\mathbf{F}_{xy}\mathbf{r}_1^T \\
g_{24} &= \mathbf{r}_1\mathbf{F}_{z1}\mathbf{c} + \mathbf{r}_1\mathbf{F}_{z2}\mathbf{r}^T - \mathbf{r}_3\mathbf{F}_{x1}\mathbf{c} - \mathbf{r}_3\mathbf{F}_{x2}\mathbf{r}^T \\
g_{33} &= \mathbf{r}_2\mathbf{F}_{xx}\mathbf{2}^T - 2\mathbf{r}_2\mathbf{F}_{xy}\mathbf{r}_1^T + \mathbf{r}_1\mathbf{F}_{yy}\mathbf{r}_1^T \\
g_{34} &= \mathbf{r}_2\mathbf{F}_{x1}\mathbf{c} + \mathbf{r}_2\mathbf{F}_{x2}\mathbf{r}^T - \mathbf{r}_1\mathbf{F}_{y1}\mathbf{c} - \mathbf{r}_1\mathbf{F}_{y2}\mathbf{r}^T \\
g_{44} &= -\mathbf{c}^T\mathbf{F}_{11}\mathbf{c} - \mathbf{r}\mathbf{F}_{22}\mathbf{r}^T - 2\mathbf{r}\mathbf{F}_{12}\mathbf{c}
\end{aligned}$$

APPENDIX B

Let $a\lambda^4 + b\lambda^3 + c\lambda^2 + d\lambda + e = 0$ be the fourth order polynomial of which the roots are the Eigenvalues of $\mathbf{H}$. a, b, c, d, and e are easily derived by evaluating $\det(\mathbf{H} - \lambda\mathbf{I}_{4\times4})$. It is important to note that the Eigenvalues of $\mathbf{H}$ are always real and positive. Applying Ferrari's solution, the smallest root is therefore given in closed-form by

$$\alpha = -\frac{3b^2}{8} + c, \quad \beta = \frac{b^3}{8} - \frac{bc}{2} + d, \quad \gamma = -\frac{3b^4}{256} + \frac{b^2c}{16} - \frac{bd}{4} + e,$$
$$p = -\frac{\alpha^2}{12} - \gamma, \quad q = -\frac{\alpha^3}{108} + \frac{\alpha\gamma}{3} - \frac{\beta^2}{8}, \quad h = -\frac{p^3}{27},$$
$$\theta_1 = h^{\frac{1}{6}}\cos(\frac{1}{3}\arccos(-\frac{q}{2\sqrt{h}})), \quad \theta_2 = h^{\frac{1}{3}},$$
$$y = -\frac{5\alpha}{6} - \frac{p\theta_1}{3\theta_2} + \theta_1, \quad w = \sqrt{\alpha + 2y},$$
$$\lambda_{\mathbf{H},min} = -\frac{b}{4} - \frac{w}{2} - \frac{1}{2}\sqrt{-3\alpha - 2y + \frac{2\beta}{w}}.$$

References

[1] K. Arun, T. Huang, and S. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9(5):698–700, 1987. 4, 5

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Heraklion, Greece, 2010. 6

[3] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 1–8, Washington, DC, USA, 2008. 2

[4] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 6, 7

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, second edition, 2004. 2

[6] J.-H. Kim, R. Hartley, J.-M. Frahm, and M. Pollefeys. Visual Odometry for Non-Overlapping Views using Second-Order Cone Programming. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 353–362, Tokyo, Japan, 2007. 2

[7] J.-H. Kim, H. Li, and R. Hartley. Motion Estimation for Nonoverlapping Multicamera Rigs: Linear Algebraic and L_∞ Geometric Solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(6):1044–1059, 2010. 2

[8] J.-S. Kim and T. Kanade. Degeneracy of the Linear Seventeen-Point Algorithm for Generalized Essential Matrix. *Journal of Mathematical Imaging and Vision (JMIV)*, 37(1):40–48, 2010. 1

[9] L. Kneip and P. Furgale. OpenGV: A Unified and Generalized Approach to Real-Time Calibrated Geometric Vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hongkong, 2014. 7

[10] L. Kneip and S. Lynen. Direct Optimization of Frame-to-Frame Rotation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013. 2, 3, 4

[11] L. Kneip, R. Siegwart, and M. Pollefeys. Finding the Exact Rotation Between Two Images Independently of the Translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Firenze, Italy, 2012. 2

[12] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion estimation for a self-driving car with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, 2013. 2

[13] H. Li, R. Hartley, and J.-H. Kim. A Linear Approach to Motion Estimation using Generalized Camera Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage, Alaska, USA, 2008. 1, 2, 3, 5, 6, 7

[14] R. Pless. Using many cameras as one. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 587–593, Madison, WI, USA, 2003. 1, 3

[15] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 430–443, Graz, Austria, 2006. 6

[16] H. Stewénius and D. Nistér. Solutions to Minimal Generalized Relative Pose Problems. In *Workshop on Omnidirectional Vision (ICCV)*, Beijing, China, 2005. 1, 2, 4, 5, 6

[17] J. Yu and L. McMillan. General linear cameras. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Prague, Czech Republic, 2004. 1