Practical Autocalibration

Riccardo Gherardi and Andrea Fusiello

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15, 37134 Verona (Italy)
name.surname@univr.it

Abstract. As it has been noted several times in literature, the difficult part of autocalibration efforts resides in the structural non-linearity of the search for the plane at infinity. In this paper we present a robust and versatile autocalibration method based on the enumeration of the inherently bounded space of the intrinsic parameters of two cameras in order to find the collineation of space that upgrades a given projective reconstruction to Euclidean. Each sample of the search space (which reduces to a finite subset of $\mathbb{R}^2$ under mild assumptions) defines a consistent plane at infinity. This in turn produces a tentative, approximate Euclidean upgrade of the whole reconstruction which is then scored according to the expected intrinsic parameters of a Euclidean camera. This approach has been compared with several other algorithms on both synthetic and concrete cases, obtaining favourable results.

Key words: Autocalibration, Self-calibration

1 Introduction

Autocalibration (a.k.a. self-calibration) has generated a lot of theoretical interest since its introduction in the seminal paper by Maybank and Faugeras [1]. The attention spawned by the problem however is inherently practical, since it eliminates the need for off-line calibration and enables the use of content acquired in an uncontrolled environment. Modern computer vision has partly sidestepped the issue using ancillary information, such as EXIF tags embedded in some image formats. Such data unfortunately is not always guaranteed to be present or consistent with its medium, and does not extinguish the need for reliable autocalibration procedures.

Lots of published methods rely on equations involving the dual image of the absolute quadric (DIAQ), introduced by Triggs in [2]. Earliest approaches for variable focal lengths were based on linear, weighted systems [3, 4], solved directly or iteratively [5]. Their reliability were improved by more recent algorithms, such as [6], solving super-linear systems while forcing directly the positive definiteness of the DIAQ. Such enhancements were necessary because of the structural non-linearity of the task: for this reason the problem has also been approached using branch and bound schemes, based either on the Kruppa equations [7], dual linear autocalibration [8] or the modulus constraint [9].

The algorithm described in [10] shares with the branch and bound approaches the guarantee of convergence; the non-linear part, corresponding to the localization of the plane at infinity, is solved exhaustively after having used the cheiral inequalities to compute explicit bounds on its location.

The technique we are about to describe is closely related to the latter: first, we derive the location of the plane at infinity given two perspective projection matrices and a guess on their intrinsic parameters, and subsequently use this procedure to iterate through the space of camera intrinsic parameters looking for the best collineation that makes the reconstruction Euclidean. The search space is inherently bounded by the finiteness of the acquisition devices; each sample and the corresponding plane at infinity define a collineation of space whose likelihood can be computed evaluating skew, aspect ratio, principal point and related constraints for each transformed camera. The best solution is eventually refined via non-linear least squares.

Such approach has several advantages: it's fast, easy to implement and reliable, since a reasonable solution can always be found in non-degenerate configurations, even in extreme cases such as when autocalibrating just two cameras.

2 Method

As customary, we assume being given a projective reconstruction $\{P_i; X_j\}$ $i = i \dots n;\ j = 1 \dots m$. The purpose of autocalibration is therefore to find the collineation of space H such that $\left\{P_i H; H^{-1} X_j\right\}$ is a *Euclidean* reconstruction, i.e., it differs from the true one by a similarity.

The set of camera matrices can always be transformed to the following canonical form by post-multiplying each P_i by the matrix $[P_1;\ 0\ 0\ 0\ 1]^{-1}$:

$$P_1 = [I \mid \mathbf{0}] \qquad P_i = [Q_i \mid \mathbf{q}_i]\,. \tag{1}$$

In this situation, the collineation of space H performing the Euclidean upgrade has the following structure:

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & \lambda \end{bmatrix} \tag{2}$$

where K_1 is the calibration matrix of the first camera, $\mathbf{v}$ a vector which determines the location of the plane at infinity and λ a scalar fixating the overall scale of the reconstruction.

The technique we are about to describe is based on two stages:

1. Given a guess on the intrinsic parameters of two cameras compute a consistent upgrading collineation. This yields an estimate of all cameras but the first.
2. Score the intrinsic parameters of these $n-1$ cameras based on the likelihood of skew, aspect ratio and principal point.

The space of the intrinsic parameters of the two cameras is enumerated and the best solution is eventually refined via non-linear least squares.

2.1 Estimation of the plane at infinity

In this section we will show how to compute the plane at infinity given two perspective projection matrices and their intrinsic parameters. This procedure is, in a sense, the dual of the second step of the stratified autocalibration [11] in which the intrinsic parameters are recovered given the plane at infinity. This problem has been dealt with for the first time in [12] where it has been turned into a linear least squares system. We shall derive here a closed form solution.

Given two projective cameras

$$P_1 = [I \mid \mathbf{0}] \qquad P_2 = [Q_2 \mid \mathbf{q}_2] \tag{3}$$

and their intrinsic parameters matrices K_1 and K_2 respectively, the upgraded, Euclidean versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{4}$$

$$P_2^E = K_2 [R_2 | \mathbf{t}_2] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top | \lambda \mathbf{q}_2 \right] \tag{5}$$

with the symbol $\simeq$ meaning "equality up to a scale". The rotation R_2 can therefore be equated to the following:

$$R_2 \simeq K_2^{-1} \left(Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top \right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{6}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term.

Using the constraints on orthogonality between rows or columns of a rotation matrix, one can solve for $\mathbf{v}$ finding the value that makes the right hand side of (6) equal up to a scale to a rotation. The solution can be obtained in closed form by noting that there always exists a rotation matrix R^* such as: $R^* \mathbf{t}_2 = [\|\mathbf{t}_2\| \; 0 \; 0]^\top$. Left multiplying it to (6) yields:

$$R^* R_2 \simeq \overbrace{R^* \; K_2^{-1} Q_2 K_1}^{W} + [\|\mathbf{t}_2\| \; 0 \; 0]^\top \mathbf{v}^\top \tag{7}$$

Calling the right hand side first term W and its rows $\mathbf{w}_i^\top$, we arrive at the following:

$$R^* \; R_2 = \begin{bmatrix} \mathbf{w}_1{}^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2{}^\top \\ \mathbf{w}_3{}^\top \end{bmatrix} / \|\mathbf{w}_3\| \tag{8}$$

in which the last two rows are independent from the value of $\mathbf{v}$ and the correct scale has been recovered normalizing to unit norm each side of the equation.

Since the rows of the right hand side form a orthonormal basis, we can recover the first one taking the cross product of the other two. Vector $\mathbf{v}$ is therefore equal to:

$$\mathbf{v} = \left(\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1 \right) / \|\mathbf{t}_2\| \tag{9}$$

The upgrading collineation H can be computed using (2); the term λ can be arbitrarily chosen, as it will just influence the overall scale of the reconstruction. Its sign however will affect the cheirality of the reconstruction, so it must be chosen positive if cheirality was previously adjusted.

2.2 Estimation of the intrinsic parameters

In the preceding section we showed how to compute the location of the plane at infinity given the calibration parameters of two of the cameras of the projective reconstruction to upgrade. When these calibration parameters are known only approximately, we are not guaranteed anymore that the right hand side of (8) will be a rotation matrix because $\mathbf{w}_2$ and $\mathbf{w}_3$ will not be mutually orthogonal, nor have equal, unit norm. However, (9) will still yield the value of $\mathbf{v}$ that makes the right hand side of (8) closest to a rotation in Frobenius norm. Hence, the derived upgrading collineation H will produce an *approximate* Euclidean reconstruction.

The autocalibration algorithm we propose consists in enumerating through all possible matrices of intrinsics of two cameras K_1 and K_2 checking whether the entire resulting reconstruction has the desired properties in terms of $K_2 \dots K_n$. The process is well-defined, since the search space is naturally bounded by the finiteness of the acquisition devices.

In order to sample the space of calibration parameters we can safely assume, as customary, null skew and unit aspect ratio: this leaves the focal length and the principal point location as free parameters. However, as expected, the value of the plane at infinity is in general far more sensitive to errors in the estimation of focal length values rather than the image center. Thus, we can iterate just over focal lengths f_1 and f_2 assuming the principal point to be centered on the image; the error introduced with this approximation is normally well-within the radius of convergence of the subsequent non-linear optimization. The search space is therefore reduced to a bounded region of $\mathbb{R}^2$.

To score each sampled point (f_1, f_2), we consider the aspect ratio, skew and principal point location of the resulting transformed camera matrices and aggregate their respective value into a single cost function:

$$\{f_1, f_2\} = \arg\min_{f_1, f_2} \sum_{\ell=2}^{n} \mathcal{C}^2(K_\ell) \tag{10}$$

where K_ℓ is the intrinsic parameters matrix of the ℓ-th camera after the Euclidean upgrade determined by (f_1, f_2), and

$$\mathcal{C}(K) = \overbrace{w_{sk}|k_{1,2}|}^{\text{skew}} + \overbrace{w_{ar}|k_{1,1} - k_{2,2}|}^{\text{aspect ratio}} + \overbrace{w_{u_o}|k_{1,3}| + w_{v_o}|k_{2,3}|}^{\text{principal point}} \tag{11}$$

where $k_{i,j}$ denotes the entry (i, j) of K and w are suitable weights, computed as in [4]. The first term of (11) takes into account the skew, which is expected to be 0, the second one penalizes cameras with aspect ratio different from 1 and the last two weigh down cameras where the principal point is away from the image centre. If a sufficient (according to the autocalibration "counting argument" [13]) number of cameras is available, the terms related to the principal point can be dropped, thereby leaving it free to move.

As an example, Fig. 1 shows the aggregated cost for a ten camera synthetic dataset, obtained with the aforementioned method. More in detail, Fig. 2 depicts

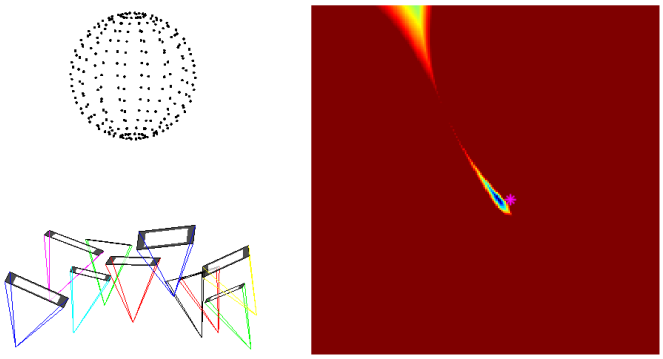

Fig. 1. A ten camera synthetic reconstruction and the resulting aggregated cost function. An asterisk marks the correct solution.

the profiles of each of the term of (11) for two sample cameras. As it can be seen, the cost profiles have very clear valleys and collectively concur to identify the correct solution, displayed in the graphs as an asterisk.

Even the aggregate cost from just a single camera can still identify a unambiguous minima. This situation is equivalent to the task of identifying the focal lengths of two cameras from their fundamental matrix. This problem, studied extensively in [12, 14, 15], was demonstrated to be essentially ill-conditioned. Our approach is stabler since it structurally requires the solution to be in a valid region of the parameter space. The solution clearly improves as more and more cameras are added.

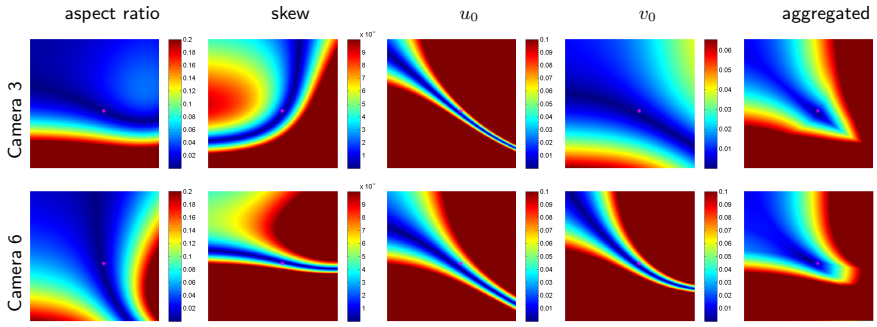

Fig. 2. Cost functions. The two rows refer to cost functions relative to different cameras of a same dataset. From left to right, are shown the profiles of aspect ratio, skew, principal point u_0 and v_0 coordinates and their aggregated value as function of the focal lengths of the reference cameras. Cooler colors correspond to lower values of the cost function. A asterisk marks the correct solution.

Finally, the solution selected is refined by non-linear minimization; since it is usually very close to a minima, just a few iterations of a Levemberg-Marquardt solver are necessary for convergence. The employed cost function is the same reported in (10).

Algorithm 1: Autocalibration pseudo-code

input : a set of PPMs P and their viewports V
output: their upgraded, euclidean counterparts

foreach P **do** $P \leftarrow V^{-1}P/\|P_{3,1:3}\|$ /* normalization */
foreach K_1, K_2 **do** /* iterate over focal pairs */
  compute Π_∞
  build H *from* (2)
  foreach P **do** /* compute cost profiles */
    $P_E \leftarrow PH$
    $K \leftarrow$ *intrinsics of* P_E
    compute $\mathcal{C}(K)$ *from* (11)
  end
end
aggregate cost and select minimum
refine non-linearly
foreach P **do** $P \leftarrow VPH$ /* de-normalization, upgrade */

The entire procedure is presented as pseudo-code in Algorithm 1. With the perspective projection matrices the code presented takes as input also the viewport matrices of the cameras, defined as:

$$V = \frac{1}{2}\begin{bmatrix} \sqrt{w^2+h^2} & 0 & w \\ 0 & \sqrt{w^2+h^2} & h \\ 0 & 0 & 2 \end{bmatrix} \tag{12}$$

where w and h are respectively the width and height of each image. This piece of data is used inside the algorithm to normalize camera matrices. While this is not mandatory, we recommend it to improve the numerical conditioning of the algorithm.

The algorithm shows remarkable convergence properties; it has been observed to fail only when the sampling of the focal space was not sufficiently dense (in practice, less than twenty focals in each direction), and therefore all the tested infinity planes were not close enough to the correct one. Such problems are easy to detect, since they usually bring the final, refined solution outside the legal search space.

3 Experimental evaluation

We report here several tests on synthetic and concrete datasets. For the experiments, unless otherwise specified, we sampled the focal space using 50 logarithmically spaced divisions in the range $[0.3 \dots 3]$. Please note that, being cameras normalized, a focal length of 1 unit correspond to the length of the image diagonal in pixels.

3.1 Synthetic tests

For this series of tests, we generated several synthetic reconstructions with twenty cameras looking at the unit sphere. Each camera was chosen having different parameters except for skew, which was set equal to zero for all perspective projection matrices. The other characteristics were selected by a random process inside the valid parameter space. The virtual viewport size for each camera was $[1024, 768]$ units, leading to focal lengths and principal points coordinates of comparable magnitude. We built projectively equivalent reconstructions multiplying the Euclidean frame for a random collineation.

Sampling rate. The top two graphs of Fig. 3 shows the relationship between the number divisions used in the focal search phase and the error of the resulting autocalibration for focal length and skew respectively, averaged over 100 trials. The focal length error has the form:

$$\varepsilon = \frac{1}{n} \sum_{\ell=1}^{n} \Delta f \tag{13}$$

where Δf is defined in equation 14. The error function used for skew has a similar formulation.

For too low rates of sampling, corresponding to the left side of the diagram, the chance of picking a solution close to the correct one is very low. Most of the time the subsequent minimization outputs parameters outside the valid range, generally converging towards the trivial null focal solution. As soon as the focal lengths are sampled with a sufficient degree of accuracy, the residual of the recovered solution becomes and stay low. When this happens, the proposed solution is usually very near to the correct one, and the following non-linear minimization has no problem to converge to the correct, best calibration parameters.

The total elapsed time follows a quadratic law, as expected. At the far right of the diagram, corresponding to fifty divisions for each focal, the total time spent (search plus refinement) is roughly 3 seconds, implemented as a MATLAB script. The omitted graphs for aspect ratio and principal point location show equivalent behaviour.

Number of cameras. In this section we verify the stability of the algorithm as the number of cameras varies from two to twenty. For uniformity all reported

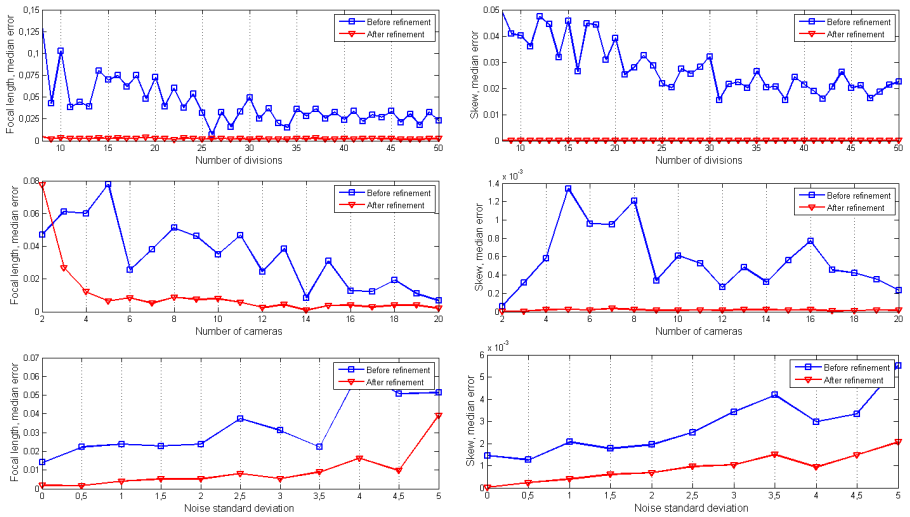

Fig. 3. Synthetic tests. Median autocalibration error ε as a function of: the number of sampling divisions (top), the number of cameras (middle), the standard deviation of noise for both focal length (left) and skew (right).

results were obtained with the full cost function described in (11), even for experiments which, having a sufficient number of cameras, could use fewer constraints. Results reported in the middle graphs of Fig. 3 are averaged over 100 runs of the algorithm. As shown, the algorithm is able to converge to the correct calibration parameters for all but the two-cameras setup, in which it trades focal length accuracy for a lower magnitude of skew. The resulting solution is still very close to the ground truth. From three cameras onwards the method successfully disambiguates the uncertainty.

Noise resilience. Our final synthetic test verifies the resilience to noise; several reconstructions were built from the ground truth perturbing the point projections with Gaussian noise and recovering each camera by DLT based resection [16]. The bottom plots of Fig. 3 shows the dependency of the error ε on the standard deviation of the added noise. Again, the results were averaged over 100 runs of the algorithm. As it can be seen the method is fairly stable, degrading quite gracefully as the standard deviation of noise increases.

Again, omitted graphs for aspect ratio and principal point location behave similarly.

3.2 Comparative tests

We compare our approach to a classical, linear technique based on the DIAQ constraints and a recent stratified method based on least squares minimization of the modulus constraint embedded in a branch and bound framework.

The first algorithm is our implementation of the iterative dual linear autocalibration algorithm described in [5], modified to use the weights of [4] and to enforce at every iteration the positive (negative) semi-definitess of the DIAQ. As explained in [17], the closest semi-definite approximation of a matrix in Frobenius norm can be obtained, assuming a single offending value, zeroing the eigenvalue with sign different from the others. This can be easily done during the rank 3 approximation step of the original algorithm. Several informal tests, not reported here, demonstrated this algorithm to have better convergence properties of both its parents [5, 4]. We report also the results obtained by this method when coupled with the preliminary quasi-affine upgrade step detailed in [18].

The second method we compare to is the algorithm described in [9], a stratified autocalibration approach based on a branch and bound framework using convex relaxations minimizations. We tested the implementation of the authors (available from http://vision.ucsd.edu/stratum/), coupled with the SeDuMi [19] library version $1.1R3$ which was used in the original article (the latest version 1.21 is not compatible with the code) under MATLAB R2009a.

The synthetic test dataset, the same used in [9], is composed of twenty projective cameras and points, with known ground truth and Gaussian noise of standard deviation σ added to image coordinates. We report results obtained by our and the aforementioned methods over a hundred trials in the case of $\sigma = 0.1\%$ using the same metric defined in the original article:

$$\Delta f = \left| \frac{f_x + f_y}{f_x^{GT} + f_y^{GT}} - 1 \right| \tag{14}$$

where f_x and f_y are the focal entries of the calibration matrix and f_x^{GT} and f_y^{GT} the respective ground truth values. Results are reported in Tab. 1. The linear algorithm, which we pick as baseline case, achieves good results in terms of Δf but shows poor convergence properties, especially for lower number of cameras. Similar numerical results are unsurprisingly obtained coupling the method with the quasi-affine upgrade of [18], with slightly higher percentages of success. Both the algorithm described in [9] and our method never failed on this dataset, with a slight numerical advantage of our proposal.

3.3 Real world example

We finally tested our algorithm on two real reconstructions, *Pozzoveggiani* and *Duomo*, composed respectively of 52 and 333 cameras (data available from http://profs.sci.univr.it/~fusiello/demo/samantha/). These reconstructions, refined through bundle adjustment, have relatively low noise levels and were used as ground truth for the subsequent tests. Again, a total of a hundred trials were conducted for each set, multiplying the projective reconstructions for a random collineation while discarding the ones with very low condition number. In our method we also picked at random the reference views to be used for the estimation of the plane at infinity.

Results are reported in Tab. 2. With respect to the synthetic case, we can note a substantial decrease of the success rate of both linear algorithms which

Table 1. Comparison of results obtained on the dataset from [6].

Algorithm	Cameras	Δf	Success rate	Time
Dual linear	5	5.4012e-2	57%	0.39
	10	2.6522e-3	84 %	0.45
	20	1.5433e-3	90 %	0.78
DL + QA upgrade	5	2.7420e-2	63 %	0.41
	10	1.8943e-3	83 %	0.43
	20	1.1295e-3	92 %	0.68
Chandraker *et al* [9]	5	9.9611e-3	100 %	584.12
	10	4.7925e-3	100 %	560.56
	20	1.0461e-3	100 %	602.32
Our method	5	2.7546e-3	100 %	0.35
	10	1.3005e-3	100 %	0.72
	20	8.2266e-4	100 %	1.62

Table 2. Comparison of results obtained on real reconstructions.

Algorithm	*Pozzoveggiani*		*Duomo*	
	Δf	Succ. rate	Δf	Succ. rate
Dual linear	3.0815e-2	19 %	9.3255e-2	8 %
DL + QA upgrade	8.9261e-3	22 %	7.6403e-2	13 %
Our method	3.9733e-3	100 %	2.9293e-3	100 %

was instead expected to increase with the number of cameras. An informal audit of the code showed the effect to be caused both by noise and by the larger number of iterations required for convergence, which in turn increase the chance of encountering a failure case.

Algorithm [9] is missing from Tab. 2 because we were not able to obtain valid solutions on these data, even by varying the tolerance ϵ and the maximal number of iterations for both the affine and metric upgrade steps.

Our approach achieves on both datasets flawless success rate. Instances of the upgraded reconstructions can be qualitatively evaluated in Fig. 4.

4 Conclusions

We presented a practical autocalibration algorithm showing results comparable to the state of the art. Our approach is fast, easy to implement and shows remarkable convergence properties.

Future research will be aimed at developing sub-linear search strategies in the space of calibration parameters, which is made possible by the structure of the cost profiles.

Acknowledgments

The use of code and data from [6] is gratefully acknowledged.

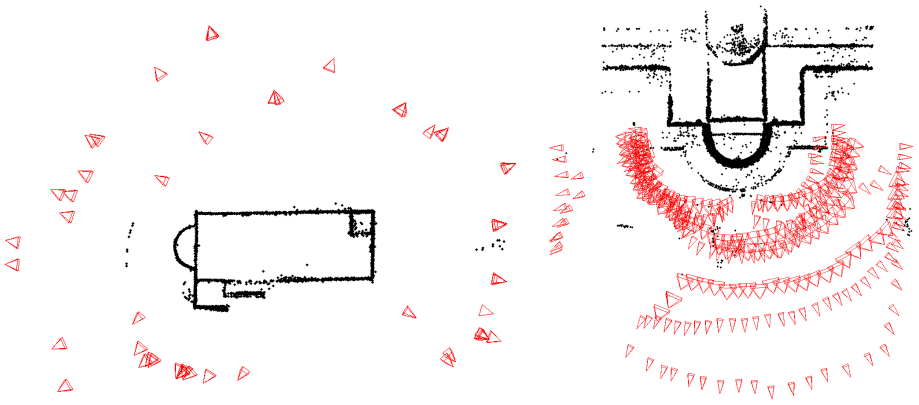

Fig. 4. *Pozzoveggiani* (left) and *Duomo* (right) reconstruction after the upgrade found by our method.

References

1. Maybank, S.J., Faugeras, O.: A theory of self-calibration of a moving camera. International Journal of Computer Vision **8** (1992) 123–151
2. Triggs, B.: Autocalibration and the absolute quadric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico (1997) 609–614
3. Pollefeys, M., Koch, R., Van Gool, L.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: Proceedings of the International Conference on Computer Vision, Bombay (1998) 90–95
4. Pollefeys, M., Verbiest, F., Van Gool, L.: Surviving dominant planes in uncalibrated structure and motion recovery. In: Proceedings of the European Conference on Computer Vision. (2002) 837–851
5. Seo, Y., Heyden, A., Cipolla, R.: A linear iterative method for auto-calibration using the dac equation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 880
6. Chandraker, M., Agarwal, S., Kahl, F., Nister, D., Kriegman, D.: Autocalibration via rank-constrained estimation of the absolute quadric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8
7. Fusiello, A., Benedetti, A., Farenzena, M., Busti, A.: Globally convergent auto-calibration using interval analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 1633–1638
8. Bocquillon, B., Bartoli, A., Gurdjos, P., Crouzil, A.: On constant focal length self-calibration from multiple views. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)
9. Chandraker, M., Agarwal, S., Kriegman, D., Belongie, S.: Globally optimal affine and metric upgrades in stratified autocalibration. In: Proceedings of the International Conference on Computer Vision. (2007) 1–8
10. Hartley, R., Hayman, E., de Agapito, L., Reid, I.: Camera calibration and the search for infinity. In: Proceedings of the International Conference on Computer Vision. (1999)

11. Faugeras, O.: Stratification of 3-D vision: projective, affine, and metric representations. Journal of the Optical Society of America A **12** (1994) 465–484
12. Bougnoux, S.: From projective to Euclidean space under any practical situation, a criticism of self-calibration. In: Proceedings of the International Conference on Computer Vision, Bombay (1998) 790–796
13. Luong, Q.T., Viéville, T.: Canonical representations for the geometries of multiple projective views. Computer Vision and Image Understanding **64** (1996) 193–229
14. Sturm, P.: On focal length calibration from two views. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume II., Kauai, USA (2001) 145–150
15. Newsam, G.N., Huynh, D.Q., Brooks, M.J., p. Pan, H.: Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. In: In Int. Arch. Photogrammetry & Remote Sensing. (1996) 575–580
16. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge University Press (2003)
17. Higham, N.J.: Computing a nearest symmetric positive semidefinite matrix. Linear Algebra and its Applications **103** (1988) 103 – 118
18. Hartley, R.I.: Chirality. Int. J. Comput. Vision **26** (1998) 41–61
19. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software **11–12** (1999) 625–653