Multiple Camera Calibration using Robust Perspective Factorization

Andrei Zaharescu, Radu Horaud, Rémi Ronfard and Loic Lefort
INRIA Rhone-Alpes, 655, Avenue de l'Europe, 38330 Montbonnot, France
{andrei.zaharescu, radu.horaud, remi.ronfard, loic.lefort}@inrialpes.fr

Abstract

In this paper we address the problem of recovering structure and motion from a large number of intrinsically calibrated perspective cameras. We describe a method that combines (1) weak-perspective reconstruction in the presence of noisy and missing data and (2) an algorithm that updates weak-perspective reconstruction to perspective reconstruction by incrementally estimating the projective depths. The method also solves for the reversal ambiguity associated with affine factorization techniques. The method has been successfully applied to the problem of calibrating the external parameters (position and orientation) of several multiple-camera setups. Results obtained with synthetic and experimental data compare favourably with results obtained with non-linear minimization such as bundle adjustment.

1. Introduction

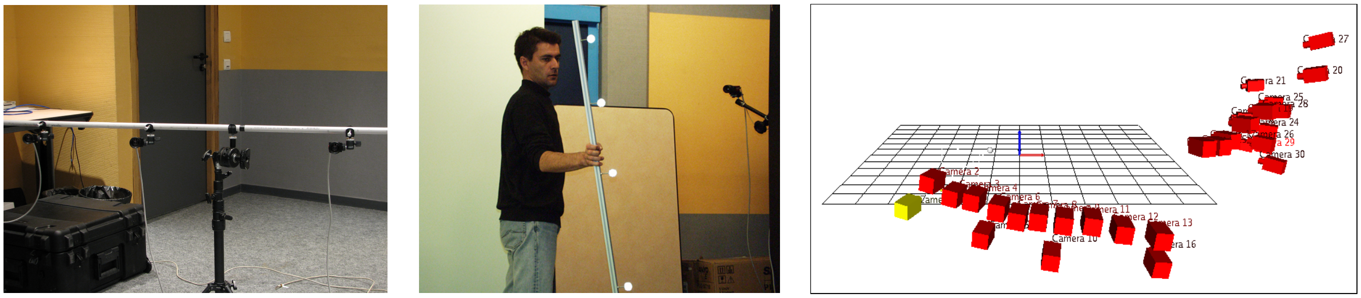
The problem of 3-D reconstruction using affine factorization has received a lot of attention. The method initially introduced by Tomasi & Kanade [1] and subsequently elaborated by many others – see section 1.2 – is attractive for a number of reasons: (i) it provides a closed-form solution, (ii) it can deal with noisy, uncertain, erroneous, and missing data, (iii) and it is upgradable to Euclidean reconstruction. Although initially designed to solve for rigid shape, it has been succesfully extended to deformable shape as well as to multiple rigid shapes (motion segmentation). Nevertheless, affine factorization has a number of limitations as well. First, it uses weak-perspective or paraperspective camera models which are approximations of the perspective model. Second, it recovers orientation and shape up to a reversal ambiguity. Third, it recovers translation and shape up to a scale ambiguity.

The problem of multiple-camera calibration has received a lot of attention in the past. This problem is two-fold: estimate both the intrinsic (internal camera parameters) and the external parameters (rotation and translation) with respect to a global reference frame. Intrinsic parameters can be estimated independently for each camera and several software packages are available today. However, it is more complex and more tedious to estimate the extrinsic parameters, especially when a very large number of cameras must be calibrated – thirty or more cameras. In practice, the cameras must simulteneously observe a large number of points. Point correspondences must be established accross the images under a multiple wide-baseline setup. Based on these point matches, both the rotation/translation parameters and the 3-D point coordinates must be recovered. The method of choice for multiple-camera calibration is bundle adjustment [2], [3]. It is a non linear minimization technique that requires proper initialization.

In this paper we propose a robust *incremental perspective factorization* method that features the following properties:

- It can deal with noisy and missing 2-D data;
- It provides accurate Euclidean reconstruction by using a perspective camera model;
- It solves for the orientation/shape reversal ambiguity;
- It exploits the motion parameters – **recovered in closed form** through the factorization process – to provide a solution for the multiple-camera external calibration problem, and
- It compares favourably with the results obtained with *ground-truth* methods such as bundle adjustment.

Paper organization. The remaining of this paper is organized as follows. In section 1.1 we recall the shape-and-motion recovery problem using a factorization formulation and, in particular, we establish the link between full perspective and weak-perspective. Section 1.2 briefly describes previous work with emphasis on robust affine factorization and on several attempts to deal with the projective case. Section 2 describes the robust method that we propose for incrementally solving for perspective factorization. Section 3 describes a practical method for estimating the external parameters of a multiple-camera setup. We show results



(a) Partial view of a 30-camera setup

(b) Calibration data gathered by moving a stick

(c) External multi-camera calibration

Figure 1: Outline of the multiple-camera calibration method described in this paper.

obtained with 30 cameras in two different layouts. We analyse the performance of our method in the presence of noise and of missing data.

1.1. Problem formulation

We consider a calibrated perspective camera model in which case the relationship between 3-D points and image points expressed in camera coordinates can be written as:

$$x_{ij} = \frac{\mathbf{r}_i^x \cdot \mathbf{P}_j + t_i^x}{\mathbf{r}_i^z \cdot \mathbf{P}_j + t_i^z} = \frac{\mathbf{R}_i^x \cdot \mathbf{P}_j + x_i^0}{1 + \varepsilon_{ij}} \tag{1}$$

$$y_{ij} = \frac{\mathbf{r}_i^y \cdot \mathbf{P}_j + t_i^y}{\mathbf{r}_i^z \cdot \mathbf{P}_j + t_i^z} = \frac{\mathbf{R}_i^y \cdot \mathbf{P}_j + y_i^0}{1 + \varepsilon_{ij}} \tag{2}$$

where a 3-D point $\mathbf{P}_j$ is projected onto the i-th camera at location $\mathbf{s}_{ij}$. The rotation matrix $\mathbf{r}_i^\top = [\mathbf{r}_i^x \ \mathbf{r}_i^y \ \mathbf{r}_i^z]$ and the translation vector $\mathbf{t}_i^\top = (t_i^x \ t_i^y \ t_i^z)$ correspond to the external camera parameters. Dividing the above equations with the *depth* t_i^z we obtain a similar set of scaled equations. The following notations have been introduced: $\mathbf{R}_i^x = \mathbf{r}_i^x / t_i^z$, $\mathbf{R}_i^y = \mathbf{r}_i^y / t_i^z$. Moreover, $x_i^0 = t_i^x / t_i^z$ and $y_i^0 = t_i^y / t_i^z$ are the camera coordinates of a point $\mathbf{s}_i^0$ which is the projection of $\mathbf{P}_0$ (the origin of the 3-D frame). Also, ε_{ij} denotes the following ratio:

$$\varepsilon_{ij} = \frac{\mathbf{r}_i^z \cdot \mathbf{P}_j}{t_i^z} \tag{3}$$

We also introduce the *perspective depth* λ:

$$\lambda_{ij} = 1 + \varepsilon_{ij} \tag{4}$$

The perspective projection equations (1) and (2) can now be rewritten as:

$$\lambda_{ij} \mathbf{s}_{ij} - \mathbf{s}_i^0 = \begin{bmatrix} \mathbf{R}_i^{x\top} \\ \mathbf{R}_i^{x\top} \end{bmatrix} \mathbf{P}_j \tag{5}$$

For k cameras and n points, $i \in \{1 \ldots k\}$, $j \in \{1 \ldots n\}$ one can write:

$$\mathbf{S}_\lambda = \mathbf{MP} \tag{6}$$

So far we considered a perspective camera model. If an affine camera model is used instead, the model above becomes simpler. In particular, under *weak perspective* $\lambda_{ij} = 1, \forall (i,j)$ and we have:

$$\mathbf{S} = \mathbf{MP} \tag{7}$$

1.2. Previous work

There are several ways for solving the affine factorization problem, i.e., eq. (7). Initially it was proposed to minimize the Frobenius norm by Tomasi & Kanade [1]:

$$\min_{\mathbf{M},\mathbf{P}} \|\mathbf{S} - \mathbf{MP}\|_F^2 \tag{8}$$

In the absence of noise, the singular value decomposition of the measurement matrix provides an optimal solution. The latter is defined up to an affine transformation $\mathbf{T}$, $\mathbf{MP} = \mathbf{MTT}^{-1}\mathbf{P}$, that can be determined using Euclidean constraints. It is worthwhile to notice that both $\mathbf{T}$ and $-\mathbf{T}$ can be used. This sign ambiguity corresponds to the well known reversal ambiguity when the 3-D world is observed with affine cameras.

Anandan & Irani [4] extended the classical SVD approach to deal with the case of directional uncertainty. They use the Mahalanobis norm instead of the Frobenius norm. They reformulate the factorization problem such that the Mahalanobis norm can be transformed into a Frobenius norm. Finally they develop a modified factorization technique. More generally, a central idea is to introduce a weight matrix $\mathbf{W}$ of the same size as the measurement matrix $\mathbf{S}$. The Frobenius norm becomes in this case:

$$\min_{\mathbf{M},\mathbf{P}} \|\mathbf{W} \otimes (\mathbf{S} - \mathbf{MP})\|_F^2 \tag{9}$$

where $\otimes$ is the Hadamard product: $A = B \otimes C \iff a_{ij} = b_{ij} c_{ij}$. The weights w_{ij} describe the confidence that we have in an image datum and the values of these weights vary from 1 (highly confident data) to 0 (missing data). The most common way of minimizing eq. (9) is to use alternation methods: these methods are based on the observation that if either $\mathbf{M}$ or $\mathbf{P}$ is known, there is a closed-form solution for the other that minimizes eq. (9). The PowerFactorization method [5], [6], [7] as well as many other methods [8], [9], [10] fall into this category. PowerFactorization [5] presents a number of attractive features: It converges by starting with any random input, it requires a lot less computation than SVD (5% of the time for a 500×500 matrix) and it deals well with missing data (60-90%). Another way to alternate between motion and shape estimation is to use the EM algorithm and a nice formulation that uses factor analysis is provided by Gruber and Weiss et al. [11], [12].

Alternatively, robustness both to noisy data and to outliers can be achieved through *adaptive weighting* by iteratively re-estimating the weight matrix $\mathbf{W}$ which amounts to iteratively modify the data $\mathbf{S}$, [13]. The method proposed in [13] uses eq. (8) in conjunction with a robust weighting matrix (truncated quadratic or Huber's M-estimator) to iteratively approximate eq. (9), getting a temporary optimum. The approximation is performed by modifying the original data $\mathbf{S}$ such that the solution to eq. (8) with modified data $\tilde{\mathbf{S}}$ is the same as eq. (9) with the original data:

$$\min_{\mathbf{M},\mathbf{P}} \|\mathbf{W} \otimes (\mathbf{S} - \mathbf{MP})\|_F^2 = \min_{\mathbf{M},\mathbf{P}} \|\tilde{\mathbf{S}} - \mathbf{MP}\|_F^2 \qquad (10)$$

The weights are updated at each iteration, using an M-estimator. This method may well be viewed as both an iterative and an alternation method because the matrix $\mathbf{M}$ is estimated using SVD, than the $\mathbf{P}$ is estimated knowing $\mathbf{M}$, and finally the image residuals are calculated and the data (the weights) are modified. Buchanan & Fitzgibbon [14] propose to solve eq. (9) through non-linear minimization over the motion and shape parameters. They review the alternation methods and they compare their speed of convergence with a damped Newton method (a variation of the Levenberg-Marquardt method). They propose a hybrid method that switches from alternation to damped Newton.

The factorization methods mentioned so far can only deal with affine camera models. The latters are not well suited for accurate estimation of motion and shape parameters. It is more difficult to solve for the projective multiple-camera calibration and/or reconstruction problem. The most accurate results are obtained through bundle adjustment methods [2], [3] which require proper initialization. Sturm and Triggs [15] introduced the use of factorization in multiple image projective structure from motion. Their formulation is similar in spirit to eq. (6) and their method relies on fundamental matrices to solve for the projective depths. Subsequently, Heyden et al. [16], Ueshiba & Tomita [17], and Mahamud & Hebert [18] proposed iterative methods to solve the Sturm/Triggs projective factorization problem.

A final class of methods uses the perspective camera model, i.e., the intrinsic parameters are known: this model is described by eqs. (1) and (2) or, equivalently by eq. (6). If the perspective depths λ_{ij} are known, the perspective factorization problem is identical to the affine factorization problem. Christy & Horaud [19] and Fujiki & Kurata [20] suggested a method that consists in incrementally estimating the depth parameters starting with a weak-perspective (or a paraperspective) approximation ($\lambda_{ij} = 1$). Each iteration updates the perspective depths and allows to solve for the reversal ambiguity. At the limit, these algorithms converge to an accurate *perspective* solution. There has been no attempt to incorporate robustness in such iterative perspective factorization procedures.

2. Robust perspective factorization

The perspective factorization method that we propose builds on the work of [5], [19], and [13] briefly described above. Our algorithm estimates values for the perspective depth iteratively in a robust fashion starting from a weak perspective camera model. The robustness is defined two-fold: with respect to the *a-priori* knowledge of both noise covariance and missing data, and with respect to the *adaptive* knowledge of the residual between weak-perspective and perspective image measurements of each of the points. The algorithm requires $\mathbf{S}$, $\mathbf{W}$ and the intrinsic camera parameters as input and provides the extrinsic camera parameters and the 3-D coordinates of the points as output. Since the proposed algorithm puts emphasis on the camera calibration (thus the $\mathbf{M}$), it can dispense of the 2-D points that are far from the center of projection and thus provide a bad initial estimate for weak perspective approximation.

An outline of the algorithm is provided below:

1. $\forall i,\ i \in \{1...k\}$ and $\forall j,\ j \in \{1...n\}$ set: $\varepsilon_{ij} = 0$;
2. Update the values of $\boldsymbol{s}_{ij}$ using ε_{ij} as shown in (5);
3. Perform an Euclidean reconstruction with a weak perspective camera:
   a) Perform a robust affine reconstruction up to a affine transformation using PowerFactorization (given $\mathbf{S}$ and $\mathbf{W}$).
   b) Perform an Euclidean reconstruction by imposing orthonormality constraints on the 3×3 matrix $\mathbf{T}$.
4. $\forall i,\ i \in \{1...k\}$ and $\forall j,\ j \in \{1...n\}$ estimate new values for ε_{ij} using(3);

5. Re-weight the values of $\boldsymbol{w}_{ij}$ corresponding to non-missing data using a robust error function and the estimate $\varepsilon_{ij}\boldsymbol{s}_{ij}$.

6. Check the values of ε_{ij}:

   if $\forall(i,j)$ the values of ε_{ij} just estimated at this iteration are identical with the values estimated at the previous iteration,
   then stop;

   else go to step 2.

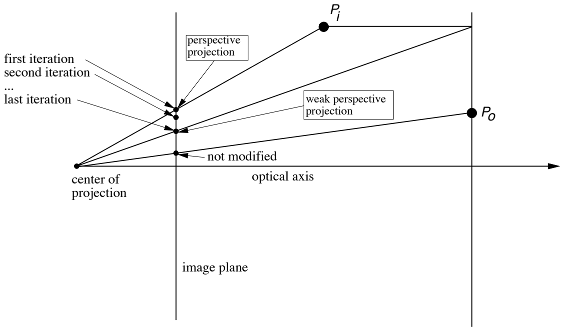


Figure 2: The iterative algorithm modifies the projection of a 3-D point from true perspective to weak perspective. Taken from [19]

A visual representation of the iterations of the algorithm can be observed in figure 2. Step 1 initializes $\varepsilon_{ij} = 0, \forall(i,j)$, which is the same as approximating the camera with the weak perspective model. Step 3 adapts the weights w_{ij} based on the 2-D reprojection error e_{ij} obtained between the perspective model and the weak-perspective camera model.

$$\boldsymbol{e}_{ij} = ||((1{+}\varepsilon_{ij})\boldsymbol{s}_{ij} - \boldsymbol{s}_i^0) - (\boldsymbol{s}_{ij} - \boldsymbol{s}_i^0)||_2 = ||\varepsilon_{ij}\boldsymbol{s}_{ij}||_2 \quad (11)$$

The entries w_{ij} are reweighted using the truncated quadratic as the robust error function[1]:

$$w_{ij}(e_{ij}) = \begin{cases} 1 & e_{ij} < h \\ \sqrt{\frac{h^2}{e_{ij}^2}} & e_{ij} > h \end{cases} \quad (12)$$

At initialization, there is no knowledge about the perspective depths, therefore $e_{ij} = 0$ for all the points and all the cameras. Once an affine reconstruction is estimated, one can measure the quality of the weak-perspective model through eq. (11) and incrementally reweight the measurements in order to favour points that best satisfy the weak-perspective model.

Step 4 is broken into two stages. The affine reconstruction is recovered using PowerFactorization as the robust affine factorization method of choice (other methods have also been considered - see section 3.1) As detailed earlier, such a reconstruction is obtained only up to an affine transformation (**T**). The Euclidean reconstruction is obtained by imposing orthonormality constraints on the 3×3 matrix **T**. We have adopted the elegant solution proposed by [13] to recover the **T** matrix.

A reversal ambiguity is inherent to any factorization based technique, since (7) can be written as $\mathbf{S} = \mathbf{MP} = (-\mathbf{M})(-\mathbf{P})$. In other words,

$$\varepsilon_{ij} = \pm\frac{\mathbf{r}_i^z \cdot \mathbf{P}_j}{t_i^z} \quad (13)$$

Therefore, two values for ε_{ij} are estimated at each iteration of the algorithm. After N iterations however, there will be 2^N possible solutions. Fortunately, not all the solutions are consistent with the image data and a verification technique (the smallest cumulative 2-D reprojection error) allows to check this consistency and to avoid the combinatorial explosion of the number of solutions.

3. Multiple-camera calibration

In this section we describe how the solution obtained in the previous section is used within the context of multiple-camera calibration. As already described above, we are interested in the estimation of the external camera parameters, i.e., the alignment between a global reference frame (or the calibration frame) and the reference frame associated with each one of the cameras. We assume that the internal camera parameters were accurately estimated using available software. There may be many cameras. In practice we tested our algorithm with two setups, each composed of 30 cameras. Finding point correspondences accross the images provided by such a setup is an issue in its own right because one has to solve for a multiple wide-baseline point correspondence problem. We will briefly describe the practical solution that we retained and which avoids badly matched points. The Euclidean shape and motion solution found in section 2 must be aligned with the global reference frame associated with the multiple-camera setup, i.e., *the calibration frame*: rotation, translation and scale. We will explain how this alignment can be robustly estimated.

Multiple-camera data acquisition. In order to perform multiple-camera acquisition and multiple-image matching, we use a simple linear object composed of four identical markers with known one-dimensional coordinates. Using the one-dimensional coordinates of the markers and their

[1] In all our experiments we set $h = 0.02$.

cross-ratio (a projective invariant) we obtain 3-D to 2-D assignments between the markers and their observed image locations. From these assignments one can easily obtain 2-D to 2-D matches over the camera frames, at some time t. The cameras are finely synchronized, therefore the linear object can be freely moved through the 3-D space and cover the common field of view of the cameras. In practice we accumulate data over several frames. In the two examples below we used 58 frames and 74 frames, i.e., 232 points and 296 points, respectively. Figure 3 depicts several situations. The top row shows the ideal case where the four markers can be easily identified in the images. The middle row shows degenerate views due to perspective foreshortening. The bottom row shows a situation where markers are missing. Uncertain 2-D locations as well as missing data are described by an a priori weight matrix $\mathbf{W}$.

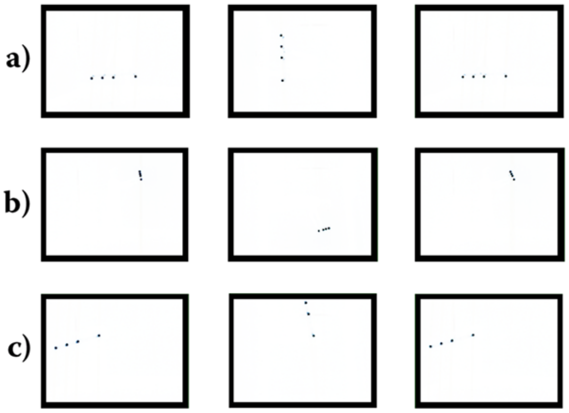


Figure 3: Examples of calibration input data: (a) typical frames; (b) frames where due to the perspective distortion the view is degenerate; (c) frames where some of the control points are missing due to the limited field of view of the camera

Alignment. We denote by $\mathbf{r}_i$ and $\mathbf{t}_i$ (the rotation matrix and translation vector) the parameters of camera i found by factorization. The 3×4 projection matrix associated with camera i writes:

$$\left[\begin{array}{cc} \mathbf{r}_i & \mathbf{t}_i \end{array}\right] \tag{14}$$

We also denote by $\mathbf{r}$, $\mathbf{t}$, and s the alignment parameters – rotation, translation, and scale – allowing to map the calibration frame onto the factorization frame. We also consider m control points, $m << n$, i.e., the total number of points. Let $\mathbf{P}_j$ be the coordinates of the control points in the factorization frame and $\mathbf{Q}_j$ be their coordinates in the calibration frame. The optimal alignment parameters may be found by minimizing:

$$\min_{s,\mathbf{r},\mathbf{t}} \sum_{j=1}^{m} \|s\mathbf{r}\mathbf{Q}_j + \mathbf{t} - \mathbf{P}_j\|^2 \tag{15}$$

The minimizer of this error function can be found in closed form using unit quaternions [21] to represent the rotation $\mathbf{r}$ or with dual-number quaternions [22] to represent the rigid displacement. Therefore, the new projection matrix associated with camera i, that aligns the calibration frame with the image-plane coordinates, writes as:

$$\left[\begin{array}{cc} \mathbf{r}_i & \mathbf{t}_i \end{array}\right] \left[\begin{array}{cc} s\mathbf{r} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{array}\right] \simeq \left[\begin{array}{cc} \mathbf{r}_i & \frac{\mathbf{t}_i}{s} \end{array}\right] \left[\begin{array}{cc} \mathbf{r} & \frac{\mathbf{t}}{s} \\ \mathbf{0}^\top & 1 \end{array}\right] \tag{16}$$

3.1. Simulated data

In all our experiments we used two camera setups. Both setups use 30 identical cameras whose intrinsic parameters were estimated in advance. We also estimated their external parameters using a non-linear method based on bundle adjustment. The external parameters obtained by bundle adjustment are the *ground truth*. The first setup is shown in figure 8 and is referred to as the *line case*. The second setup is illustrated in figure 1 and is referred to as the *corner case*). We simulated various configurations of 200 3-D data points which are projected onto the 30 images, using the camera parameters provided by bundle adjustment (the ground truth).

The simulated experiments compare the results of our method (robust perspective factorization) with the weak-perspective factorization methods propsed by Tomasi & Kanade [1] – the SVD method, Hartley & Schaffalitzky [5] – PowerFactorization, Aanaes et al. [13] – adaptive weighting, and Buchanan & Fitzgibbon [14] – damped Newton. Through these experiments we measured the RMS of the 2-D reprojection errors when increasing noise is added to the data and when the number of missing entries in the measurement matrix increases as well.

Figures 4 and 5 summarize the results of these simulations with the first camera setup (the *line case*) and figures 6 and 7 summarize the results obtained with the second camera setup (the *corner case*).

The best and most consistent results are obtained with affine PowerFactorization and with our method (perspective factorization that uses PowerFactorization in its inner loop). It is interesting to notice that with increasing noise, the performance of these two methods are comparable. This is due to the fact that when high-amplitude noise is present in the data, the incremental perspective algorithm cannot currently

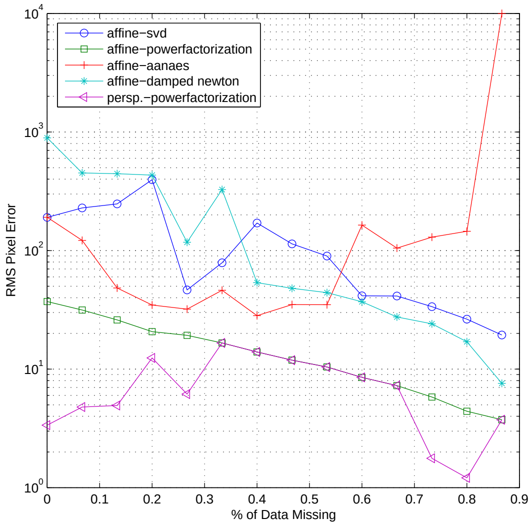


Figure 4: Line Case - The algorithm described in section 2 is compared with different affine factorization techniques: robustness towards missing features. The 2-D reprojection error is measured in in pixels.

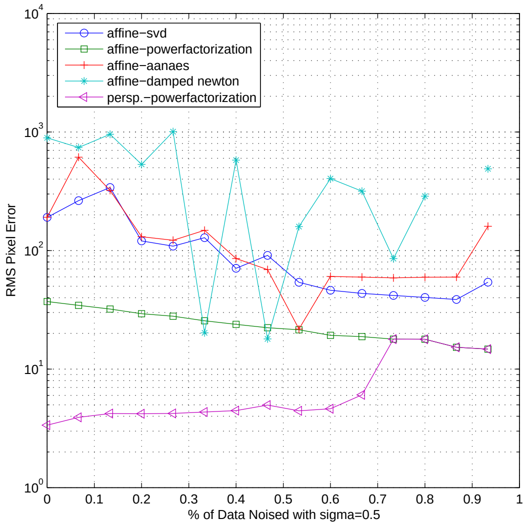


Figure 5: Line Case - The algorithm described in section 2 is compared with different affine factorization techniques: robustness towards noise in measurements.

distinguish between *projection errors* and errors to to noisy measurements.

eq

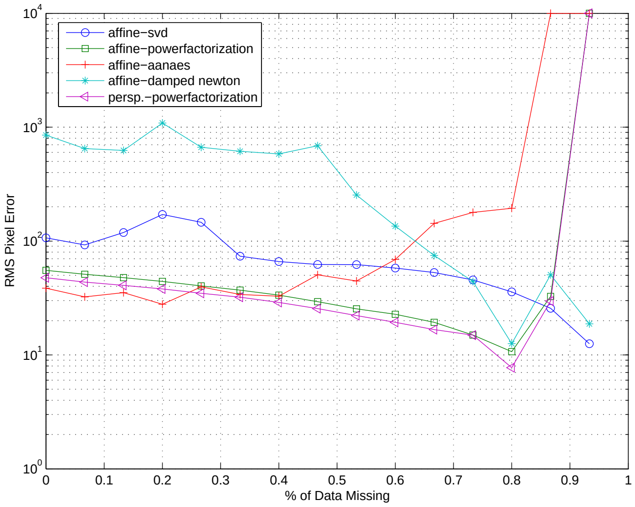


Figure 6: Corner Case - The algorithm described in section 2 is compared with different affine factorization techniques: robustness towards missing features.

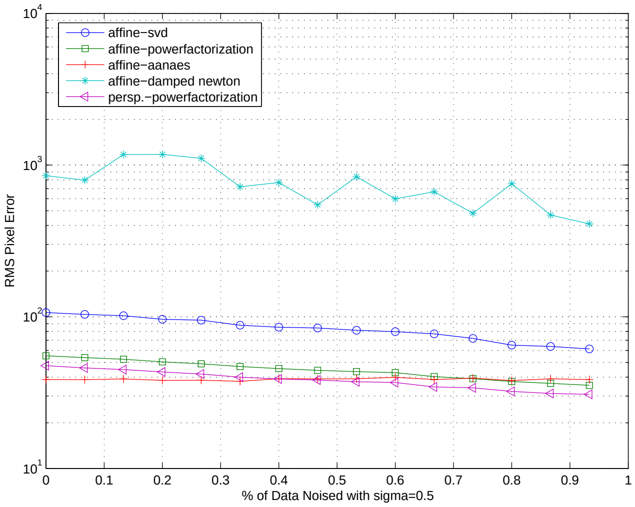


Figure 7: Corner Case - The algorithm described in section 2 is compared with different affine factorization techniques: robustness towards noise in measurements. The 2-D reprojection error is measured in in pixels.

3.2. Experimental data

In this subsection we present some of the calibration results obtained with the two camera setups and using the experimental data. The performance obtained with the method described in section 2 is compared to both weak-perspective algorithms and to the ground-truth.

The *corner case* uses 296 points accumulated over 74 frames. Therefore the measurement matrix has $2\times30\times296 = 17760$ entries, out of which 37% are missing. The recovered camera layout is shown in figure 1 and a comparison with the ground-truth is detailed in table 1. The translation error is represented as percentage error with respect to the groundtruth estimate, whereas the rotation error is represented in degrees. For completeness, table 2 shows the mean errors when applying the iterative perspective algorithm in conjunction with various affine factorization methods. This table also compares the results with the results obtained through purely weak-perspective methods. The 3-D point error is measured in milimeters, whereas the 2-D error is measured in pixels.

Camera	Rot.	Tr.	Camera	Rot.	Tr.
1	3.31	19.47	16	3.82	18.61
2	4.11	16.89	17	2.44	18.44
3	5.31	16.39	18	0.78	19.23
4	4.31	16.58	19	2.41	22.63
5	4.24	17.52	20	4.27	28.35
6	5.04	16.44	21	2.54	22.44
7	4.75	17.90	22	3.76	21.50
8	4.15	18.13	23	2.88	19.91
9	4.25	17.72	24	1.72	23.07
10	5.19	16.85	25	0.65	22.81
11	3.82	17.84	26	4.07	21.12
12	4.07	18.03	27	5.23	27.09
13	4.53	18.32	28	4.66	22.83
14	2.23	19.84	29	2.06	24.10
15	2.92	18.71	30	5.22	26.64

Table 1: Corner Case - Rotation and translation errors obtained for each individual camera. Rotation is measured in degrees, and the translation is measured in percentage error with respect to the original groundtruth

	Method	2-D	3-D	Cal.Rot	Cal.Tr
Affine	SVD	156	1405	11	48
	PowerFact	16	174	4	14
	Aanaes	-	-	-	-
	Damped Newton	892	325	77	119
Persp.	SVD	82	1558	56	41
	PowerFact	**13**	**285**	**4**	**20**
	Aanaes	-	-	-	-
	Damped Newton	480	793	10	90

Table 2: Comparison between affine and perspective for corner case

The *line case* uses 232 points accumulated over 58 frames and there is no missing data in this case. The resulting camera layout is shown in figure 9. This result may be compared with the ground-truth shown in figure 8. Table 3 summarizes the comparison of our method with various weak-perspective methods. Again, the figures indicated the discrepencies with respect to the ground-truth. As it can be observed in table 3, the perspective reconstruction provides clear benefits over the weak perspective model. The effect is diminished when the weak perspective approximations is close to the perspective camera model, as shown in table 2.

	Method	2-D	3-D	Cal.Rot	Cal.Tr
Affine	SVD	191	1058	80	58
	PowerFact	37	307	9	42
	Aanaes	191	1058	80	58
	Damped Newton	892	325	77	119
Persp.	SVD	125	1025	70	76
	PowerFact	**3**	**164**	**3**	**5**
	Aanaes	124	1031	77	75
	Damped Newton	255	357	50	127

Table 3: Comparison between affine and perspective for line case

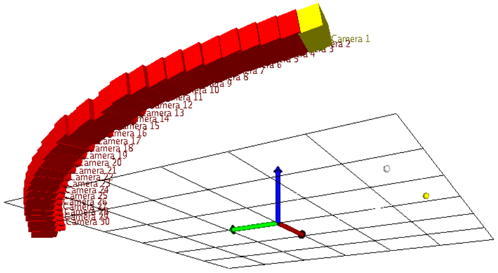


Figure 8: Line Case - Groundtruth Camera Layout (232 features - no missing data)

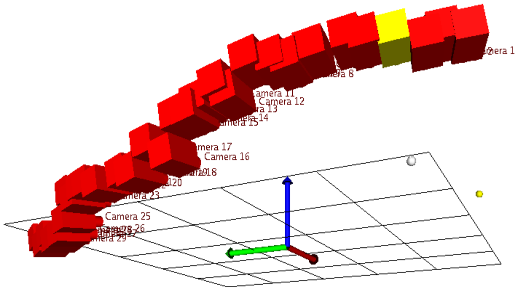


Figure 9: Line Case - Reconstructed Camera Layout (232 features - no missing data)

4. Discussion

In order to interpret these results, it is worth mentioning the experimental conditions. In the *line case* 3-D points are situated at approximatively 1.8 meters away from the cameras. In the *corner case* the 3-D points are at 5.1 meters away from the cameras. Therefore the second camera setup is consistent with both the weak-perspective and the perspective models. The advantage of perspective factorization is clearly shown with the first example. It is worthwhile to mention that in the *line case* the cameras' optical axes roughly lie in the same plane and they have a common rotation axis, and therefore the motion matrices associated with the extrinsic camera parameters are linearly dependent, which makes the **M** matrix a rank-2 matrix. Factorization methods assume the **M** matrix to be a rank-3 matrix.

We plan to statistically characterize the *perspective error* and to include such prior knowledge in the iterative perspective factorization algorithm.

Acknowledgments. This research has been supported by the European Project VISIONTRAIN.

References

[1] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization mehod," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, Nov. 1992.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[3] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon, *Bundle Ajustment — A Modern Synthesis*, vol. 1883 of *Vision Algorithms: Theory and Practice*, pp. 298–372, Spriner, lncs edition, 2000.

[4] P. Anandan and M.Irani, "Factorization with uncertainty," *International Journal of Computer Vision*, vol. 49, no. 2/3, pp. 101–116, 2002.

[5] R. Hartley and F. Schaffalitzky, "Powerfactorization: 3d reconstruction with missing or uncertain data," Tech. Rep., Australian National University, 2003.

[6] R. Hartley D. Q. Huynh and A. Heyden, "Outlier correction in image sequences for the affine camera," in *Proceedings ICCV*, 2003, vol. 1, pp. 585–590.

[7] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and gpca.," in *Proceedings CVPR*, 2004, vol. 2, pp. 310–316.

[8] K.Ikeuchi H. Shum and R. Reddy, "Principal component analysis with missing data and its application to polyhedral object modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 855–867, 1995.

[9] S. Brant, "Closed-form solutions for affine reconstruction under missing data," in *Proceedings of Statistical Methods for Video Processing (ECCV '02 Workshop)*, 2002, pp. 109–114.

[10] T. Wiberg, "Computation of principal components when data are missing," in *Proceedings Symposium of Computational Statistics*, 1976, pp. 229–326.

[11] A. Gruber and Y. Weiss, "Factorization with uncertainty and missing data: exploring temporal coherence," in *Proceedings Neural Information Processing Systems (NIPS'2003)*, 2003.

[12] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the em algorithm," in *Proceedings Conference on Computer Vision and Pattern Recognition*, 2004, pp. 707–714.

[13] R. Fisker H. Aanaes and K. Astrom, "Robust factorization," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1215–1225, Sept. 2002.

[14] A. M. Buchanan and A. W. Fitzgibbon, "Damped newton algorithms for matrix factorization with missing data," in *CVPR05*, 2005, vol. 2, pp. 316–322.

[15] Peter Sturm and Bill Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, April 1996, vol. 1065 of *LNCS*, pp. 709–720.

[16] A. Heyden R. Berthilsson and G. Sparr, "An iterative factorization method for projective structure and motion from image sequences," *Image and Vision Computing*, vol. 17, no. 13, pp. 981–991, November 1999.

[17] T. Ueshiba and F. Tomita, "A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation," in *Proceedings of Fifth European Conference of Computer Vision*, 1998, pp. 296–310.

[18] Shyjan Mahamud and Martial Hebert, "Iterative projective reconstruction from multiple views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, June 2000, vol. 2, pp. 430 – 437.

[19] S. Christy and R. Horaud, "Euclidean shape and motion from multiple perspective views by affine iterations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1098–1104, November 1996.

[20] J. Fujiki and T. Kurata, "Recursive factorization method for the paraperspective model based on the perspective projection," in *Proceedings of the International Conference on Pattern Recognition*, 2000, pp. 406–410.

[21] B.K.P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A.*, vol. 4, no. 4, pp. 629–642, 1987.

[22] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CGVIP-Image Understanding*, vol. 54, no. 3, pp. 358–367, November 1991.