



2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Robust Factorization Methods Using A Gaussian/Uniform Mixture Model

Andrei Zaharescu and Radu Horaud

INRIA Grenoble Rhône-Alpes

655, avenue de l'Europe

38330 Montbonnot, France

Corresponding author: Radu.Horaud@inrialpes.fr

July 18, 2008

Abstract

In this paper we address the problem of building a class of robust factorization algorithms that solve for the shape and motion parameters with both affine (weak perspective) and perspective camera models. We introduce a Gaussian/uniform mixture model and its associated EM algorithm. This allows us to address robust parameter estimation within a data clustering approach. We propose a robust technique that works with any affine factorization method and makes it robust to outliers. In addition, we show how such a framework can be further embedded into an iterative perspective factorization scheme. We carry out a large number of experiments to validate our algorithms and to compare them with existing ones. We also compare our approach with factorization methods that use M-estimators.

Index terms – robust factorization, 3-D reconstruction, multiple camera calibration, data clustering, expectation-maximization, EM, M-estimators, outlier rejection.

1 Introduction

The problem of 3-D reconstruction from multiple images is central in computer vision [16, 23]. Bundle adjustment provides both a general method and practical algorithms for solving this reconstruction problem using maximum likelihood [41]. Nevertheless, bundle adjustment is non-linear in nature and sophisticated optimization techniques are necessary, which in turn require proper initialization. Moreover, the combination of bundle adjustment with robust statistical methods to reject outliers is not clear both from the points of view of convergence properties and of efficiency. Factorization was introduced by Tomasi & Kanade [39] as an elegant solution to affine multiple-view reconstruction; their initial solution based on SVD and on a weak-perspective camera model has subsequently been improved and elaborated by Morris & Kanade [30], Anandan & Irani [2], Hartley & Schaffalitzky [15] as well as by many others. These methods treat the non-degenerate cases. Kanatani [20], [21] investigated how to apply model selection techniques to deal with degenerate cases, namely when the 3-D points lie on a plane and/or the camera centers lie on a circle.

The problem can be formulated as the one of minimizing the following Frobenius norm:

$$\theta^* = \arg \min_{\theta} \|\mathbf{S} - \hat{\mathbf{S}}(\theta)\|_F^2 \quad (1)$$

where matrix $\mathbf{S} = [s_{ij}]$ denotes the measurement matrix containing matched 2-D image observations, $\hat{\mathbf{S}}(\theta) = \mathbf{M}\mathbf{P}$ denotes the prediction matrix that can be factorized into the *affine* motion matrix \mathbf{M} and the *affine* shape matrix \mathbf{P} . Hence, we denote by θ the affine motion **and** shape parameters collectively. In the error-free case, direct factorization of the observation matrix using SVD provides an optimal solution. More recently the problem of *robust* affine factorization has received a lot of attention and powerful algorithms that can deal with *noisy*, *missing*, and/or *erroneous* data were suggested.

Anandan & Irani [2] extended the classical SVD approach to deal with the case of directional uncertainty. They used the Mahalanobis norm instead of the Frobenius norm and they reformulated the factorization problem such that the Mahalanobis norm can be transformed into a Frobenius norm. This algorithm handles image observations with covariance up to a few pixels but it cannot cope with missing data, mismatched points, and/or outliers. More generally, a central idea is to introduce a weight matrix \mathbf{W} of the same size as the measurement matrix \mathbf{S} . The

minimization criterion then becomes:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \|\mathbf{W} \otimes (\mathbf{S} - \hat{\mathbf{S}}(\boldsymbol{\theta}))\|_F^2 \quad (2)$$

where \otimes denotes the Hadamard product ($A = B \otimes C \iff a_{ij} = b_{ij}c_{ij}$) and $\mathbf{W} = [w_{ij}]$ is matrix whose entries are weights that reflect the confidence associated with each image observation. The most common way of minimizing eq. (2) is to use alternation methods: these methods are based on the fact that, if either one of the matrices \mathbf{M} or \mathbf{P} is known, then there is a closed-form solution for the other matrix that minimizes eq. (2). Morris & Kanade [30] were the first to propose such an alternation method. The PowerFactorization method introduced by Hartley & Schaffalitzky [15], as well as other methods by Vidal & Hartley [42], and Brant [5] fall into this category. PowerFactorization is based on the PowerMethod for sparse matrix decomposition [11]. Notice that these techniques are very similar in spirit with PCA methods with missing data, Wiberg [44], Ikeuchi, Shum, & Reddy [35], Roweis [34], and Bishop [3]. Another way to alternate between the estimation of motion and shape is to use factor analysis, Gruber and Weiss [12], [13].

Alternatively, robustness may be achieved through *adaptive weighting*, i.e., by iteratively updating the weight matrix \mathbf{W} which amounts to modifying the data \mathbf{S} , such as is done by Aanaes et al. [1]. Their method uses eq. (1) in conjunction with a robust loss function (see Stewart [36] and Meer [27] for details) to iteratively approximate eq. (2), getting a temporary optimum. The approximation is performed by modifying the original data \mathbf{S} such that the solution to eq. (1) with *modified data* $\tilde{\mathbf{S}}$ is the same as the solution to eq. (2) with the original data:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \|\mathbf{W} \otimes (\mathbf{S} - \hat{\mathbf{S}}(\boldsymbol{\theta}))\|_F^2 = \arg \min_{\boldsymbol{\theta}} \|\tilde{\mathbf{S}} - \hat{\mathbf{S}}(\boldsymbol{\theta})\|_F^2 \quad (3)$$

In [1] the weights are updated via IRLS [36]. This may well be viewed as both an iterative and an alternation method because the motion matrix \mathbf{M} is estimated using SVD, then the shape matrix \mathbf{P} is estimated knowing \mathbf{M} , while the image residuals are calculated and the data (the weights) are modified, etc. A similar algorithm that performs outlier correction was proposed by Huynh, Hartley, & Heyden [18]. Indeed, if the observations are noisy, the influence of outliers can be decreased by iteratively replacing bad observations with “pseudo” observations. The convergence of such methods, as [1] or [18] is not proved but is tested through experiments with both simulated and real data.

An alternative to M-estimators are random sampling techniques developed independently in computer vision [9] and statistics [32] (see Meer [27] for a recent overview of these methods).

1
2
3
4
5 For example, Huynh & Heyden [19] and Tardif et al. [38] use RANSAC, Trajkovic and Hedley
6 use LMedS [40], and Hajder and Chetverikov [14] use LTS (Least Trimmed Squares) [33]. The
7 major drawback of these methods is that they must consider a large number of subsets sampled
8 from the observation matrix \mathbf{S} .
9

10
11
12
13 Generally speaking, robust regression techniques, such as the ones that we briefly discussed,
14 work well in conjunction with affine factorization algorithms. Factorization was initially de-
15 signed as a “closed-form solution” to multiple-view reconstruction, but robust affine factoriza-
16 tion methods are iterative in nature, as explained above. This has several implications and some
17 drawbacks. In the presence of a large number of outliers, proper initialization is required. The
18 use of an influence function (such as the truncated quadratic) that tends to zero too quickly cause
19 outliers to be ignored and hence, this raises the question of a proper choice of an influence func-
20 tion. The objective function is non-convex implying that IRLS will be trapped in local minima.
21 The generalization of affine factorization to deal with perspective implies the estimation of depth
22 values associated with each reconstructed point. This is generally performed iteratively [37], [6],
23 [24], [25], [29], [31]. It is not yet clear at all how to combine *iterative robust methods* with
24 *iterative projective/perspective factorization methods*.
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

49 In this paper we cast the problem of robust factorization into the framework of data clustering
50 [10]. Namely, we consider the problem of classifying the observed 2-D matched points into two
51 categories: inliers and outliers. For that purpose we model the likelihood of the observations
52 with a Gaussian/uniform mixture model. This leads to a maximum likelihood formulation with
53 missing variables that can be solved with the EM algorithm [7], [26], [10]. Notice that this
54 approach is different than the method proposed by Miller & Browning [28] requiring both labeled
55 and unlabeled data sets.
56
57
58
59
60
61
62
63
64
65

66 We devise an EM algorithm within the framework of 3-D reconstruction and within the spe-
67 cific mixture model just outlined. Each EM step guarantees that the likelihood is increased.
68 Hence EM may indeed lead to a local maximum of the likelihood. We show that in this particular
69 case (normally distributed inliers and uniformly distributed outliers) the posterior probabilities
70 have a very simple interpretation in terms of robust regression. We describe an affine factoriza-
71 tion algorithm that uses EM; This algorithm is robust and it shares the convergence properties
72 just outlined. We also describe an extension of this algorithm to deal with the perspective camera
73
74
75

1
2
3
4
5 model.

6
7 We performed several experiments in two different scenarios: multiple-camera calibration
8 and 3-D reconstruction using turn-table data. Our method was compared to other methods on an
9 equal footing: it performs as well as bundle adjustment to estimate external camera parameters. It
10 performs better than IRLS (used in conjunction with the truncated quadratic) to eliminate outliers
11 in some difficult cases.
12
13
14
15

16
17 The remainder of this paper is organized as follows. Section 2 describes the probabilistic
18 modelling of inliers and outliers using a mixture between a Gaussian and an uniform distribution.
19 Section 3 explains how this probabilistic model can be used to derive an affine factorization al-
20 gorithm and section 4 extends this algorithm to iterative perspective factorization. Sections 5 and
21 6 describe experiments performed with multiple-camera calibration and with multi-view recon-
22 struction data sets. Section 7 compares our approach to M-estimators and section 8 draws some
23 conclusions and gives some directions for future work.
24
25
26
27
28
29
30

31 **2 Probabilistic modelling of inlier/outlier detection**

32
33 The 2-D image points \mathbf{s}_{ij} ($1 \leq i \leq k, 1 \leq j \leq n$) are the observed values of an equal number of
34 random variables s_{ij} . We introduce another set of random variables, z_{ij} which assign a category
35 to each observation. Namely there are two possible categories, an *inlier* category and an *outlier*
36 category. More specifically $z_{ij} = \text{inlier}$ means that the observation \mathbf{s}_{ij} is an inlier while $z_{ij} =$
37 outlier means that the observation \mathbf{s}_{ij} is an outlier.
38
39
40
41
42
43
44
45
46

47 We define the prior probabilities as follows. Let A_i be the area associated with image i and
48 we assume that all the images have the same area, $A_i = A$. If an observation is an inlier, then it is
49 expected to lie within a small circular image patch a of radius σ_0 , $a = \pi\sigma_0^2$. The prior probability
50 of an inlier is the proportion of the image restricted to such a small circular patch:
51
52
53
54

$$55 P(z_{ij} = \text{inlier}) = \frac{a}{A} \quad (4)$$

56
57 Similarly, if the observation is an outlier, its probability should describe the fact that it lies outside
58 this small patch:
59
60

$$61 P(z_{ij} = \text{outlier}) = \frac{A - a}{A} \quad (5)$$

Moreover, an observation \mathbf{s}_{ij} , given that it is an inlier, should lie in the neighborhood of an estimation $\hat{\mathbf{s}}_{ij}$. Therefore, we will model the probability of an observation \mathbf{s}_{ij} given that it is assigned to the inlier category with a Gaussian distribution centered on $\hat{\mathbf{s}}_{ij}$ and with a 2×2 covariance matrix \mathbf{C} . We obtain:

$$P_{\boldsymbol{\theta}}(\mathbf{s}_{ij} | z_{ij} = \text{inlier}) = \frac{1}{2\pi(\det \mathbf{C})^{1/2}} \exp\left(-\frac{1}{2}d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))\right) \quad (6)$$

where we denote by $d_{\mathbf{C}}$ the Mahalanobis distance:

$$d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) = (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^{\top} \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) \quad (7)$$

Whenever the observation is an outlier, it may lie anywhere in the image. Therefore, we will model the probability of an observation \mathbf{s}_{ij} given that it is assigned to the outlier category with a uniform distribution over the image area:

$$P(\mathbf{s}_{ij} | z_{ij} = \text{outlier}) = \frac{1}{A} \quad (8)$$

Since each variable z_{ij} can take only two values, marginalization is straightforward and we obtain:

$$\begin{aligned} P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}) &= P_{\boldsymbol{\theta}}(\mathbf{s}_{ij} | z_{ij} = \text{inlier})P(z_{ij} = \text{inlier}) + P(\mathbf{s}_{ij} | z_{ij} = \text{outlier})P(z_{ij} = \text{outlier}) \\ &= \frac{a}{2\pi(\det \mathbf{C})^{1/2}A} \exp\left(-\frac{1}{2}d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))\right) + \frac{A-a}{A^2} \end{aligned} \quad (9)$$

We already defined the small area a as a disk of radius σ_0 , $a = \pi\sigma_0^2$ and we assume that $a \ll A$. Using Bayes' formula¹, we obtain the posterior conditional probability of an observation to be an inlier. We denote this posterior probability by α_{ij}^{in} :

$$\alpha_{ij}^{in} = P_{\boldsymbol{\theta}}(z_{ij} = \text{inlier} | \mathbf{s}_{ij}) = \frac{1}{1 + \frac{2}{\sigma_0^2}(\det \mathbf{C})^{1/2} \exp\left(\frac{1}{2}d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))\right)} \quad (10)$$

The covariance matrix can be written as $\mathbf{C} = U\Lambda U^{\top}$ where U is a rotation and Λ is a diagonal form with entries λ_1 and λ_2 . Hence $\det(\mathbf{C}) = \lambda_1\lambda_2$. In order to plot and illustrate the shape of α_{ij}^{in} as a function of \mathbf{C} we consider the case of an isotropic covariance, i.e., $\lambda_1 = \lambda_2 = \sigma^2$ and

¹ $P(z_{ij} = \text{inlier} | \mathbf{s}_{ij})P(\mathbf{s}_{ij}) = P(\mathbf{s}_{ij} | z_{ij} = \text{inlier})P(z_{ij} = \text{inlier})$

inria-00446987, version 1 - 14 Jan 2010

one may notice that the rotation becomes irrelevant in this case. We have: $\mathbf{C} = \sigma^2 \mathbf{I}_2$. Eq. (10) writes in this case:

$$\alpha_{ij}^{in} = P_{\boldsymbol{\theta}}(z_{ij} = \text{inlier} | \mathbf{s}_{ij}) = \frac{1}{1 + \frac{2\sigma^2}{\sigma_0^2} \exp\left(\frac{\|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2}{2\sigma^2}\right)} \quad (11)$$

This posterior probability is shown on Figure 1, i.e., the function $f_{\sigma}(x) = 1/(1 + \sigma^2 \exp(x^2/2\sigma^2))$. Here σ takes discrete values in the interval $[0.05, 5]$ and $\sigma_0^2 = 2$, i.e., inliers lie within a circle of radius 2 pixels centered on a prediction. It is worthwhile to notice that, at the limit $\sigma \rightarrow 0$, we obtain a Dirac function:

$$f_0(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \quad (12)$$

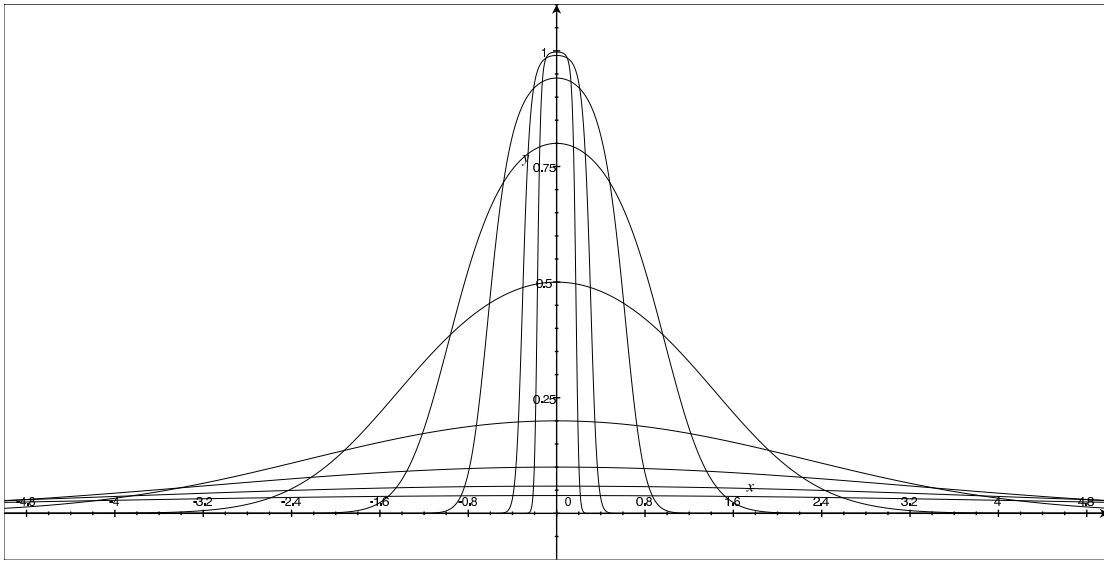


Figure 1: Various plots of the conditional posterior probability of an observation to be an inlier, i.e., $f_{\sigma}(x) = 1/(1 + \sigma^2 \exp(x^2/2\sigma^2))$. This function corresponds to eq. (11) with $\sigma_0^2 = 2$. As the variance decreases, i.e., $\sigma = 5, 4, 3, 2, 1, 0.5, 0.25, 0.1, 0.05$, the function becomes more and more discriminant. It is worthwhile to notice that $\lim_{\sigma \rightarrow 0} f_{\sigma}(x)$ is a Dirac.

The posterior conditional probability of an observation to be an outlier is given by:

$$\alpha_{ij}^{out} = P_{\boldsymbol{\theta}}(z_{ij} = \text{outlier} | \mathbf{s}_{ij}) = 1 - \alpha_{ij}^{in} \quad (13)$$

inria-00446987, version 1 - 14 Jan 2010

2.1 Maximum likelihood with inliers

The maximum likelihood estimator (ML) maximizes the log-likelihood of the joint probability of the set of measurements, $P_{\boldsymbol{\theta}}(\mathbf{S})$. Under the assumption that the observations are independent and identically distributed we have:

$$P_{\boldsymbol{\theta}}(\mathbf{S}) = \prod_{i,j} P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}) \quad (14)$$

Since we assume that all the observations are inliers, eq. (9) reduces to:

$$P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}) = P_{\boldsymbol{\theta}}(\mathbf{s}_{ij} | z_{ij} = \text{inlier}) \quad (15)$$

The log-likelihood of the joint probability becomes:

$$\log P_{\boldsymbol{\theta}}(\mathbf{S}) = -\frac{1}{2} \sum_{i,j} \left(d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) + \log(\det \mathbf{C}) \right) + \text{const} \quad (16)$$

which can be written as the following criterion:

$$Q_{ML} = \frac{1}{2} \sum_{i,j} \left((\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^{\top} \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) + \log(\det \mathbf{C}) \right) \quad (17)$$

The shape and motion parameters can be estimated by minimizing the above criterion with respect to $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^{\top} \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) \quad (18)$$

Once an optimal solution is found, i.e., $\boldsymbol{\theta}^*$, it is possible to minimize eq. (17) with respect to the covariance matrix which yields (see appendix A):

$$\mathbf{C}^* = \frac{1}{m} \sum_{i,j} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*))^{\top} \quad (19)$$

where $m = k \times n$ is the total number of observations for k images and n 3-D points.

Alternatively, if one uses an isotropic covariance, i.e., $\mathbf{C} = \sigma^2 \mathbf{I}$, By minimization of Q_{ML} with respect to $\boldsymbol{\theta}$ we obtain:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2 \quad (20)$$

The optimal variance is given by (see appendix B):

$$\sigma^{2*} = \frac{1}{2m} \sum_{i,j} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)\|^2 \quad (21)$$

2.2 Maximum likelihood with inliers and outliers

In the presence of outliers, the previous method cannot be applied. Instead, one has to use the *joint probability* of the observations and of their assignments. Again, by assuming that the observations are independent, we have:

$$\begin{aligned} P_{\boldsymbol{\theta}}(\mathbf{S}, \mathbf{Z}) &= \prod_{i,j} P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}, z_{ij}) \\ &= \prod_{i,j} P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij})P(z_{ij}) \\ &= \prod_{i,j} (P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij} = \text{inlier})P(z_{ij} = \text{inlier}))^{\delta_{in}(z_{ij})} \\ &\quad (P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij} = \text{outlier})P(z_{ij} = \text{outlier}))^{\delta_{out}(z_{ij})} \end{aligned} \quad (22)$$

The random variables $\delta_{in}(z_{ij})$ and $\delta_{out}(z_{ij})$ are defined by:

$$\delta_{in}(z_{ij}) = \begin{cases} 1 & \text{if } z_{ij} = \text{inlier} \\ 0 & \text{otherwise} \end{cases} \quad \delta_{out}(z_{ij}) = \begin{cases} 1 & \text{if } z_{ij} = \text{outlier} \\ 0 & \text{otherwise} \end{cases}$$

By taking the logarithm of the above expression and grouping constant terms, we obtain:

$$\begin{aligned} \log P_{\boldsymbol{\theta}}(\mathbf{S}, \mathbf{Z}) &= \sum_{i,j} (\delta_{in}(z_{ij}) \log(P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij} = \text{inlier})) \\ &\quad + \delta_{out}(z_{ij}) \log(P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij} = \text{outlier})) + \text{const}) \end{aligned} \quad (23)$$

This cannot be solved as previously because of the presence of the missing assignment variables z_{ij} . Therefore, they will be treated within an expectation-maximization framework. For this purpose we evaluate the *conditional expectation* of the log-likelihood over the random variables z_{ij} , given the observations \mathbf{S} :

inria-00446987_version 1 - 14 Jan 2010

inria-00446987_version 1 - 14 Jan 2010

$$\begin{aligned}
 E_Z [\log(P_{\boldsymbol{\theta}}(\mathbf{S}, \mathbf{Z})|\mathbf{S})] &= \sum_{i,j} (\log(P_{\boldsymbol{\theta}}(\mathbf{s}_{ij}|z_{ij} = \text{inlier}))E_Z [\delta_{in}(z_{ij})|\mathbf{S}] \\
 &+ \log(P(\mathbf{s}_{ij}|z_{ij} = \text{outlier}))E_Z [\delta_{out}(z_{ij})|\mathbf{S}]) \quad (24)
 \end{aligned}$$

In this formula we omitted the constant terms, i.e., the terms that do not depend on the parameters $\boldsymbol{\theta}$ and \mathbf{C} . The subscript Z indicates that the expectation is taken over the random variable z . From the definition of $\delta_{in}(z_{ij})$ we have:

$$\begin{aligned}
 E[\delta_{in}(z_{ij})] &= \delta_{in}(z_{ij} = \text{inlier})P(z_{ij} = \text{inlier}) + \delta_{in}(z_{ij} = \text{outlier})P(z_{ij} = \text{outlier}) \\
 &= P(z_{ij} = \text{inlier})
 \end{aligned}$$

Hence:

$$E_Z [\delta_{in}(z_{ij})|\mathbf{S}] = P(z_{ij} = \text{inlier}|\mathbf{S}) = P(z_{ij} = \text{inlier}|\mathbf{s}_{ij}) = \alpha_{ij}^{in}$$

and:

$$E_Z [\delta_{out}(z_{ij})|\mathbf{S}] = 1 - \alpha_{ij}^{in}$$

Therefore, after removing constant terms, the conditional expectation becomes:

$$E_Z [\log(P_{\boldsymbol{\theta}}(\mathbf{S}, \mathbf{Z})|\mathbf{S})] = -\frac{1}{2} \sum_{i,j} \alpha_{ij}^{in} \left(d_{\mathbf{C}}^2(\mathbf{s}_{ij}, \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) + \log(\det \mathbf{C}) \right) \quad (25)$$

This leads to the following criterion:

$$Q_{EM} = \frac{1}{2} \sum_{i,j} \alpha_{ij}^{in} \left((\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^{\top} \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) + \log(\det \mathbf{C}) \right) \quad (26)$$

3 Robust affine factorization with the EM algorithm

In this section we provide the details of how the robust affine factorization problem can be solved iteratively by maximum likelihood via the expectation-maximization (EM) algorithm [7], and how the observations can be classified into either inliers or outliers by maximum a-posteriori (MAP).

By inspection of equations (17) and (26) one may observe that the latter is a weighted version of the former and hence our formulation has strong similarities with M-estimators and

their practical solution, namely iteratively reweighted least-squares (IRLS) [36]. Nevertheless, the weights $\omega_{ij} = \alpha_{ij}^{in}$ were obtained using a Bayesian approach: they correspond to the posterior conditional probabilities of the observations (i.e., given that they are inliers), and such that the equality $\alpha_{ij}^{in} + \alpha_{ij}^{out} = 1$ holds for each observation. The structure and the shape of these posteriors are depicted by equations (10) and (11) and shown on Figure 1. These probabilities are functions of the residual but they are parameterized as well by the 2×2 covariance matrix \mathbf{C} associated with the normal probability distribution of the observations: One advantage of our formulation over IRLS is that this covariance is explicitly taken into consideration and estimated within the EM algorithm.

It is worthwhile to remark that the minimization of eq. (18) over the affine shape and motion parameters, i.e., $\boldsymbol{\theta}$, can be solved using an affine camera model and a factorization method such that the ones proposed in the literature [1, 15]. In practice we use the PowerFactorization method proposed in [15]. The minimization of eq. (26) can be solved in the same way, provided that estimates for the posterior probabilities α_{ij}^{in} are available. This can be done by iterations of the EM algorithm:

- The **E-step** computes the conditional expectation over the assignment variables associated with each observation, i.e., eq. (25). This requires a current estimate of both $\boldsymbol{\theta}$ and \mathbf{C} from which the α_{ij}^{in} 's are updated.
- The **M-step** maximizes the conditional expectation or, equivalently, minimizes eq. (26) with fixed posterior probabilities. This is analogous, but not identical, with finding the means $\boldsymbol{\mu}_{ij}$ and a common covariance \mathbf{C} of $m = k \times n$ Gaussian distributions, with $\boldsymbol{\mu}_{ij} = \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})$. Nevertheless, the means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_{11}, \dots, \boldsymbol{\mu}_{kn}\}$ are parameterized by the global variables $\boldsymbol{\theta}$. For this reason, the minimization problem needs a specific treatment (unlike the classical mixture of Gaussians approach where the means are independent).

Therefore $\min_{\boldsymbol{\mu}} Q_{EM}$ in the standard EM method must be replaced by $\min_{\boldsymbol{\theta}} Q_{EM}$ and it does depend on \mathbf{C} in this case:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} \alpha_{ij}^{in} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) \quad (27)$$

inria-00446987, version 1 - 14 Jan 2010

Moreover, the covariance that minimizes eq. (26) can be easily derived from [3]:

$$\mathbf{C}^* = \frac{1}{\sum_{i,j} \alpha_{ij}^{in}} \sum_{i,j} \alpha_{ij}^{in} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*))^\top \quad (28)$$

In many practical situations it is worthwhile to consider the case of an *isotropic covariance*, in which case the equations above reduce to:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} \alpha_{ij}^{in} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2 \quad (29)$$

and

$$\sigma^{2*} = \frac{1}{2 \sum_{i,j} \alpha_{ij}^{in}} \sum_{i,j} \alpha_{ij}^{in} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)\|^2 \quad (30)$$

This may well be viewed as a special case of model-based clustering [10]. It was proved [26] that EM guarantees convergence, i.e., that $Q_{EM}^{q+1} < Q_{EM}^q$, where the overscript q denotes the q^{th} iteration, and that this implies the maximization of the joint probability of the observations: $P_{\boldsymbol{\theta}}(\mathbf{S})^{q+1} > P_{\boldsymbol{\theta}}(\mathbf{S})^q$. To conclude, the algorithm can be paraphrased as follows:

Affine factorization with EM:

Initialization: Use the PowerFactorization method to minimize eq. (18). This provides initial estimates for $\boldsymbol{\theta}$ (the affine shape and motion parameters). Estimate \mathbf{C} (the covariance matrix) using eq. (19).

Iterate until convergence

Expectation: Update the values of α_{ij}^{in} according to eq. (10) or eq. (11).

Maximization: Minimize Q_{EM} over $\boldsymbol{\theta}$ (affine factorization) using either eq. (27) or eq. (29). Compute the covariance \mathbf{C} with eq. (28) or the variance σ^2 with eq. (30).

Maximum a posteriori: Once the EM iterations terminate, choose in between inlier and outlier for each observation, i.e., $\max\{\alpha_{ij}^{in}; \alpha_{ij}^{out}\}$.

The algorithm needs initial estimates for the shape and motion parameters from which an initial covariance matrix can be estimated. This guarantees that, at the start of EM, all the residuals have equal importance. Nevertheless, “bad” observations will have a large associated residual and, consequently, the covariance is proportionally large. As the algorithm proceeds, the covariance adjusts to the current solution while the posterior probabilities α_{ij}^{in} become more and more discriminant as depicted on Figure 1. Eventually, observations associated with small residuals will be classified as inliers, and observations with large residuals will be classified as outliers.

The overall goal of 3-D reconstruction consists of the estimation of the shape and motion parameters: As just explained, we embed affine reconstruction in the M-step. Therefore, with our algorithm, robustness stays *outside* the factorization method at hand – is it iterative or not – and hence one can plug into EM any factorization procedure.

4 Robust perspective factorization

In this section we address the problem of 3-D reconstruction using intrinsically calibrated cameras. Moreover, we consider both the weak-perspective and the perspective camera models, and we explain how the affine solution provided by factorization can be upgraded to Euclidean reconstruction. We describe an algorithm that combines the EM affine factorization algorithm described above with an iterative perspective factorization algorithm [6, 46]. This results in a robust method for solving the 3-D Euclidean reconstruction problem as well as the multiple-camera calibration problem.

An image point $\mathbf{s} = (x, y)$ is the projection of a 3-D point $\tilde{\mathbf{X}}$:

$$x_{ij} = \frac{\mathbf{r}_i^x \cdot \tilde{\mathbf{X}}_j + t_i^x}{\mathbf{r}_i^z \cdot \tilde{\mathbf{X}}_j + t_i^z} = \frac{\mathbf{a}_i^x \cdot \tilde{\mathbf{X}}_j + b_i^x}{\varepsilon_{ij} + 1} \quad (31)$$

$$y_{ij} = \frac{\mathbf{r}_i^y \cdot \tilde{\mathbf{X}}_j + t_i^y}{\mathbf{r}_i^z \cdot \tilde{\mathbf{X}}_j + t_i^z} = \frac{\mathbf{a}_i^y \cdot \tilde{\mathbf{X}}_j + b_i^y}{\varepsilon_{ij} + 1} \quad (32)$$

We introduced the following notations: 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 motion parameters and they are also denoted the external camera parameters. Dividing the above equations with the *depth* t_i^z we obtain a similar set of scaled equations. We have: $\mathbf{a}_i^x = \mathbf{r}_i^x / t_i^z$, $\mathbf{a}_i^y = \mathbf{r}_i^y / t_i^z$, $b_i^x = t_i^x / t_i^z$ and $b_i^y = t_i^y / t_i^z$.

We denote by ε_{ij} the *perspective distortion* parameters, namely the following ratios:

$$\varepsilon_{ij} = \frac{\mathbf{r}_i^z \cdot \tilde{\mathbf{X}}_j}{t_i^z} \quad (33)$$

Finally, the perspective equations, i.e., eqs. (31) and (32) can be written as:

$$\mathbf{s}_{ij}(1 + \varepsilon_{ij}) = \mathbf{A}_i \mathbf{X}_j \quad (34)$$

where $\mathbf{X} = (\tilde{\mathbf{X}}, 1)$ and \mathbf{A}_i denotes the following 2×4 matrix:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_i^x & b_i^x \\ \mathbf{a}_i^y & b_i^y \end{bmatrix}$$

From now on we can replace the parameter vector $\boldsymbol{\theta}$ with the affine shape and motion parameters, namely the point set $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_j, \dots, \mathbf{X}_k\}$ and the matrix set $\mathcal{A} = \{\mathbf{A}_1, \dots, \mathbf{A}_j, \dots, \mathbf{A}_n\}$. Using these notations, eq. (27) can now be written as:

$$\min_{\mathcal{A}, \mathcal{X}} \frac{1}{2} \sum_{i,j} \alpha_{ij}^{in} (\mathbf{s}_{ij}(1 + \varepsilon_{ij}) - \mathbf{A}_i \mathbf{X}_j)^\top \mathbf{C}^{-1} (\mathbf{s}_{ij}(1 + \varepsilon_{ij}) - \mathbf{A}_i \mathbf{X}_j) \quad (35)$$

which can be solved via the EM affine factorization algorithm with $\varepsilon_{ij} = 0, \forall (i, j)$. A weak-perspective camera model can then be used for upgrading to Euclidean reconstruction.

The introduction of the perspective camera model adds non null perspective-distortion parameters ε_{ij} , i.e., eq. (33). One fundamental observation is the following: If estimates for the parameters $\varepsilon_{ij}, \forall i \in [1 \dots k], \forall j \in [1 \dots n]$ are available, then this corresponds to a weak-perspective camera model that is closer to the true perspective model. If the true values of the perspective-distortion parameters are available, the corresponding weak-perspective model corresponds exactly to the perspective model. Hence, the problem reduces to affine factorization followed by Euclidean upgrade. Numerous iterative algorithms have been suggested in the literature for estimating the perspective-distortion parameters associated with each 2-D observation, both with uncalibrated and calibrated cameras [37], [6], [24], [25], [29], [31] to cite just a few. One possibility is to perform *weak-perspective iterations*. Namely, the algorithm starts with a *zero-distorsion* weak-perspective approximation and then, at each iteration, it updates the perspective distortions using eq. (33). To conclude, the robust perspective factorization algorithm can be summarized as follows:

inria-00446987, version 1 - 14 Jan 2010

Robust perspective factorization:

Initialization: Set $\varepsilon_{ij} = 0, \forall i \in [1 \dots k], \forall j \in [1 \dots n]$. Use the same initialization step as the **affine factorization with EM** algorithm.

Iterate until convergence:

Affine factorization with EM: Iterate until convergence the E- and M-steps of the algorithm described in the previous section.

Euclidean upgrade: Recover the rotations, translations, and 3-D Euclidean coordinates from the affine shape and affine motion parameters.

Perspective update: Estimate new values for the parameters $\varepsilon_{ij}, \forall i \in [1 \dots k], \forall j \in [1 \dots n]$. If the current depth values are identical with the previously estimated ones, then terminate, else iterate.

Maximum a posteriori: After convergence choose in between inlier and outlier for each observation, i.e., $\max\{\alpha_{ij}^{in}, \alpha_{ij}^{out}\}$.

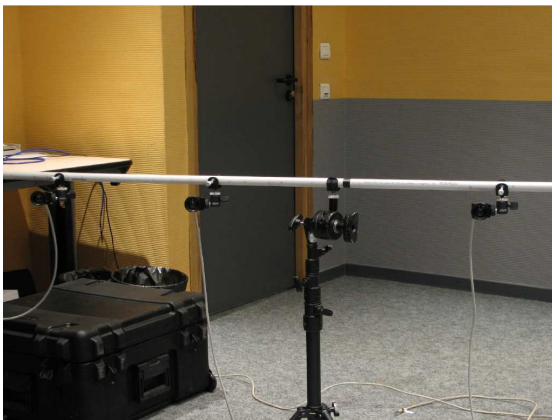
5 Multiple-camera calibration

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 are many papers available that address the problem of internal camera calibration either from 3-D reference objects [8], 2-D planar objects [47], 1-D objects [48] or self-calibration, e.g., from point-correspondences [22], [16], [23].

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Figure 2 shows a partial view of a multiple-camera setup as well as the one-dimensional object used for calibration. In practice we used three different camera configurations as depicted in Figure 4: two 30 camera configurations and one 10 camera configuration. These camera setups will be referred to as the *Corner Case*, the *Arc Case*, and the *Semi-Spherical Case*. Finding point correspondences across 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 maximizes the number of points that are matched over all the views. Nevertheless, in practice there are missing observations as well as badly detected image features, bad matches, etc. The problem of missing data has already been addressed. Here we concentrate on the detection and rejection of outliers.

We performed multiple camera calibration with two algorithms: The robust perspective factorization method previously described and bundle adjustment. We report a detailed comparison between these two methods. We further compare our robust method with a method based on M-estimators.



(a)



(b)

Figure 2: (a): Partial view of a 30-camera setup. (b): The calibration data are gathered by moving a one-dimensional object in the common field of view of the cameras.

As already mentioned, we use a simple 1-D object composed of four identical markers with known 1-D coordinates. These coordinates form a projective-invariant signature (the cross-ratio) that is used to obtain 3-D to 2-D matches between the markers and their observed image locations. With finely synchronized cameras it is possible to gather images of the object while the latter is freely moved in order to cover the 3-D space that is commonly viewed by all cameras. In the

three examples below we used 73, 58, and 32 frames, i.e., 292, 232, and 128 3-D points. The number of cameras in each setup is 30, 30 and 10, respectively. Therefore, in theory there should be 8760, 6960, and 1280 2-D observations.

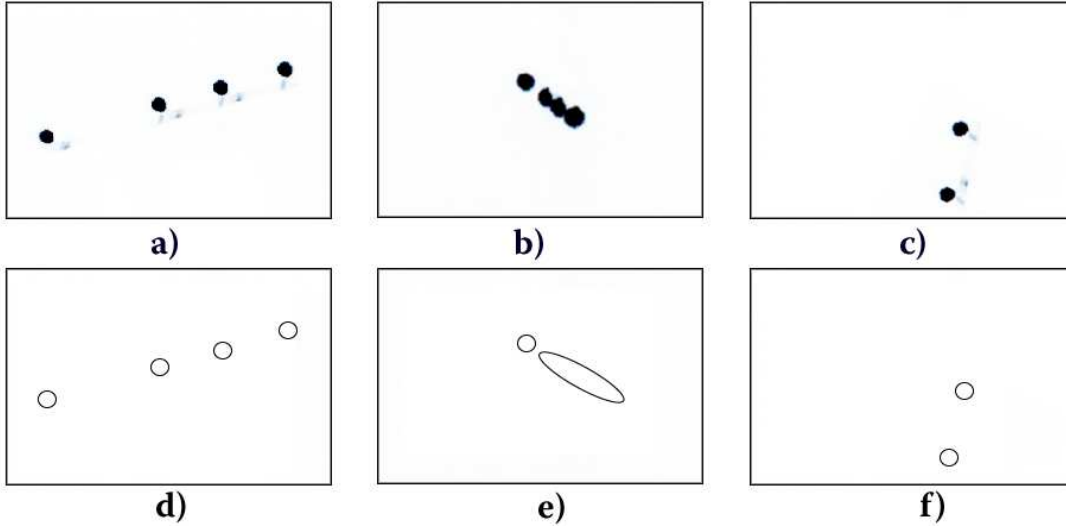


Figure 3: Top: These are typical images where the number of connected components depend on the position and orientation of the calibrating object with respect to the cameras. Bottom: Detected blobs with their centers and associated covariance, i.e., second-order moments.

Figure 3 depicts three possible image configurations: (a) four distinct connected components that correspond without ambiguity to the four markers, (b) a degenerate view of the markers, due to strong perspective distortion, that results in a number of connected components that cannot be easily matched with the four markers, and (c) only two connected components are visible in which case one cannot establish a reliable match with the four markers. In practice we perform a connected-component analysis that finds the number of blobs in each image. Each such blob is characterized by its center and second order moments, i.e., Figure 3 (d), (e), and (f). These blobs are matched with the object markers. In most of the cases the match is unambiguous, but in some cases a blob may be matched with several markers.

Let as before, s_{ij} denote the center of a blob from image i that matches marker j . The second order moments of this blob can be used to compute an initial 2×2 covariance matrix \mathbf{C}_{ij}^0 for each such observation. Moreover, we introduce a binary variable, μ_{ij} , which is equal to 0 if the observation s_{ij} is missing and equal to 1 otherwise. The multiple-camera calibration algorithm

can now be paraphrased as follows:

Multiple camera calibration:

Initialization: Use eq. (18) to estimate the affine shape and motion parameters in the presence of some missing data:

$$\boldsymbol{\theta}^0 = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} \mu_{ij} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top (\mathbf{C}_{ij}^0)^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))$$

Estimate the initial covariance matrix using eq. (19):

$$\mathbf{C}^0 = \frac{1}{m} \sum_{i,j} \mu_{ij} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^0)) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^0))^\top$$

Set $\varepsilon_{ij} = 0, \forall i \in [1 \dots k], \forall j \in [1 \dots n]$

Iterate until convergence:

Affine factorization with EM: Iterate until convergence the E- and M-steps of the algorithm described in section 3.

Euclidean upgrade: Recover the rotations, translations, and 3-D Euclidean coordinates from the affine shape and affine motion parameters.

Perspective update: Estimate new values for the parameters $\varepsilon_{ij}, \forall i \in [1 \dots k], \forall j \in [1 \dots n]$. If the current depth values are identical with the previously estimated ones, terminate, else iterate.

Both the initialization and the M steps of the above algorithm perform affine factorization in the presence of uncertainty and missing data. In [46] we compared several such algorithms and we came to the conclusion that the PowerFactorization algorithm outperforms the other tested algorithms. In order to assess quantitatively the performance of our algorithm, we compared it with an implementation of the bundle adjustment method along the lines described in [16].

This comparison requires the estimation of the rotations and translations allowing the alignment of the two reconstructed 3-D sets of points with the cameras. We estimate these rotations and translations using a set of control points. Indeed, both the robust perspective factorization and the bundle adjustment algorithms need a number of control points with known Euclidean 3-D coordinates. In practice, the calibration procedure provides such a set. This set of control points allows one to define a global reference frame. Let P_j^c denote the 3-D coordinates of the control points estimated with our algorithm, and let Q_j^c denote their 3-D coordinates provided in advance. Let λ , \mathbf{R} , and \mathbf{t} be the scale, rotation and translation allowing the alignment of the two sets of control points. We have:

$$\min_{\lambda, \mathbf{R}, \mathbf{t}} \sum_{j=1}^8 \|\lambda \mathbf{R} Q_j^c + \mathbf{t} - P_j^c\|^2 \quad (36)$$

The minimizer of this error function can be found in closed form either with unit quaternions [17] to represent the rotation \mathbf{R} or with dual-number quaternions [43] to represent the rigid motion \mathbf{R}, \mathbf{t} . Similarly, one can use the same procedure to estimate the scale λ' , rotation \mathbf{R}' , and translation \mathbf{t}' associated with the 3-D reconstruction obtained by bundle adjustment.

Finally, in order to evaluate the quality of the results we estimated the following measurements:

The 2D error is measured in pixels and corresponds to the RMS error between the observations and the predictions weighted by their posterior probabilities:

$$\left(\frac{\sum_{i,j} \alpha_{ij}^{in} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)\|^2}{\sum_{i,j} \alpha_{ij}^{in}} \right)^{1/2}$$

The 3D error is measured in millimeters and corresponds to the RMS error between the two sets of 3-D points obtained with our algorithm and with bundle adjustment:

$$\left(\frac{\sum_{j=1}^n \|P_j - P'_j\|^2}{n} \right)^{1/2}$$

The error in rotation is measured in degrees and depicts the average angular error of the rotation matrices over all the cameras. With the same notations as before, let \mathbf{R}_i and \mathbf{R}'_i be the rotations of camera i as obtained with our algorithm and with bundle adjustment. Let \mathbf{v} be

inria-00446987, version 1 - 14 Jan 2010

an arbitrary 3-D vector. The dot product $(\mathbf{R}_i \mathbf{v}) \cdot (\mathbf{R}'_i \mathbf{v})$ is a reliable measure of the cosine of the angular discrepancy between the two estimations. Therefore the RMS error in rotation can be measured by the angle:

$$\arccos \left(\frac{180}{\pi} \sqrt{\frac{\mathbf{v}^\top \left(\sum_{i=1}^k \mathbf{R}_i^\top \mathbf{R}'_i \right) \mathbf{v}}{k}} \right)$$

The error in translation is measured in millimeters with:

$$\left(\frac{\sum_{i=1}^k \|\mathbf{t}_i - \mathbf{t}'_i\|^2}{k} \right)^{1/2}$$

As already mentioned, we used three camera setups. All setups use identical 1024×768 Flea cameras from Point Grey Research Inc.² The intrinsic parameters were estimated in advance. Two of the setups use 30 cameras, whereas the third one uses 10 cameras. We denote these setups as *Corner Case*, *Arc Case* and *Semi-Spherical Case*, based on the camera layout.

The results are summarized on Figures 4 and 5 and on Tables 1 and 2. Let us analyze in more detail these results. In the *Corner Case* there are 30 cameras and 292 3-D points. Hence, there are 8760 possible predictions out of which only 5527 are actually observed, i.e., 36% predictions correspond to missing 2-D data. The algorithm detected 5202 2-D inliers. An inlier is an observation with a posterior probability greater than 0.4. Next, the outliers are marked as missing data. Eventually, in this example, 285 3-D points were reconstructed (out of a total of 292) and all the cameras were correctly calibrated. The number of iterations of the robust perspective factorization algorithm (referred to as affine iterations) is equal to 7. On an average, there were 2.4 iterations of the EM algorithm. The obtained reconstruction has a smaller 2-D reprojection error (0.30 pixels) than the one obtained by bundle adjustment (0.58 pixels).

In any of the 3 calibration scenarios, the proposed method outperforms bundle adjustment results, as it can be observed in Table 2. This is in part due to the fact that the bundle adjustment algorithm does not have a mechanism for outlier rejection.

Figure 5 shows the evolution of the algorithm as it iterates from a weak-perspective solution to the final full-perspective solution. At convergence, the solution found by our method (shown in blue or dark in the absence of colors) is practically identical to the solution found by bundle adjustment (which is shown in grey).

²<http://www.ptgrey.com/products/flea/>

inria-00446987, version 1 - 14 Jan 2010

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

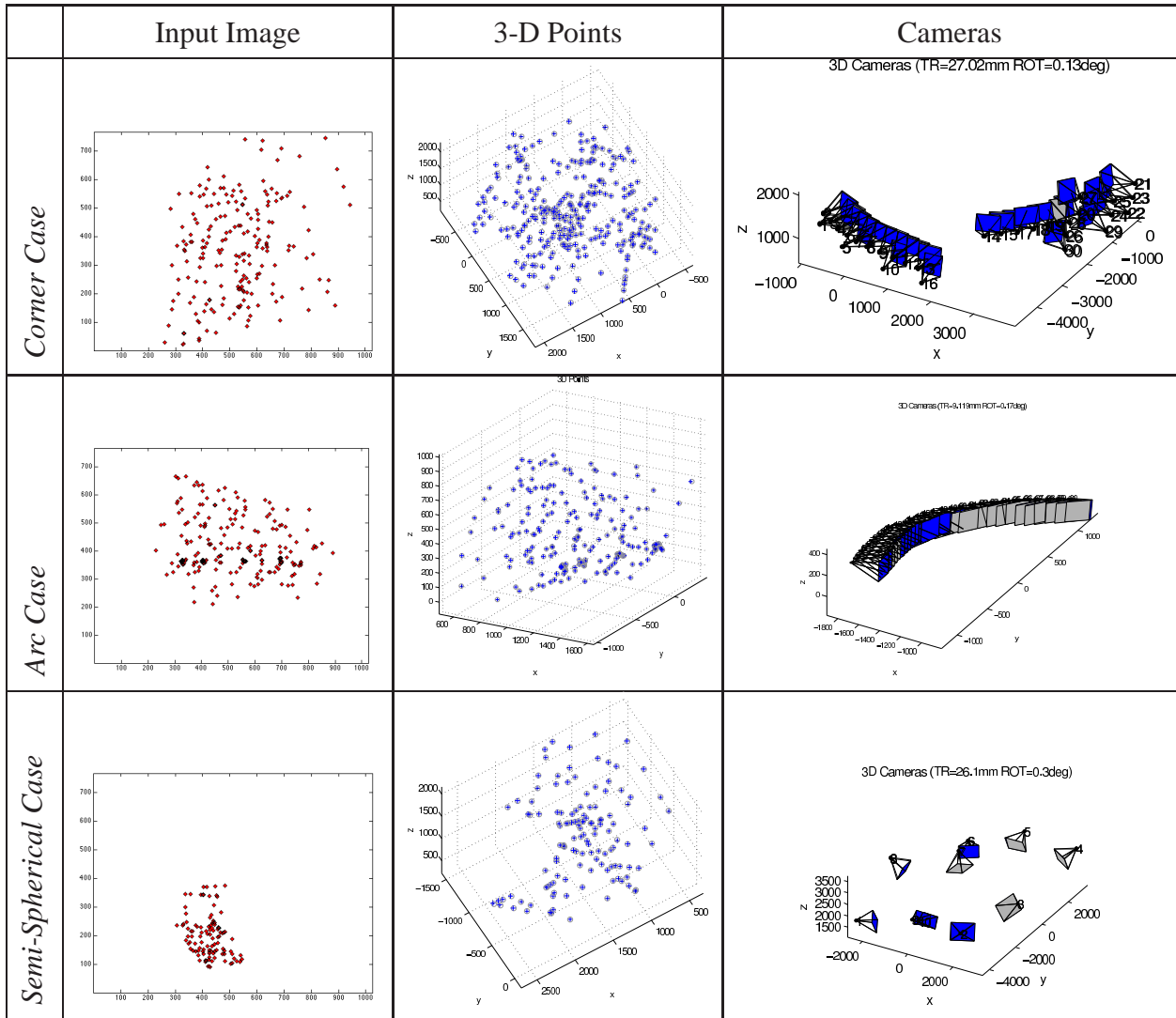


Figure 4: Multiple camera calibration results. Left: A typical set of 2-D observations associated with one camera. Middle: Reconstructed 3-D points with our method (blue) and with bundle adjustment (grey). Right: Camera calibration results obtained with our method (blue) and with bundle adjustment (grey).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Multi-camera calibration		<i>Corner Case</i>	<i>Arc Case</i>	<i>Semi-Spherical Case</i>
Input	# Cameras	30	30	10
	# 3-D Points	292	232	128
	# 2-D Predictions	8760	6960	1280
	# Missing observations	36%	0%	33%
	# 2-D Observations	5527	6960	863
Results	# 2-D Inliers	5202	6790	784
	# 3-D Inliers	285	232	122
	2D error (pixels)	0.30	0.19	0.48
	3D error (mm)	6.91	2.65	4.57
	Rot. error (degrees)	0.13	0.18	0.27
	Tr. error (mm)	27.02	9.37	24.21
	# Aff. iter. (# EM iter.)	7 (2.4)	11 (2)	8 (3.2)

Table 1: Summary of the camera calibration results for the three setups.

Multi-camera calibration	<i>Corner Case</i>	<i>Arc Case</i>	<i>Semi-Spherical Case</i>
Proposed Method - 2D error (pixels)	0.30	0.19	0.48
Bundle Adjustment - 2D error (pixels)	0.58	0.61	0.95

Table 2: Comparison between the proposed method and bundle adjustment for the three camera calibration setups.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

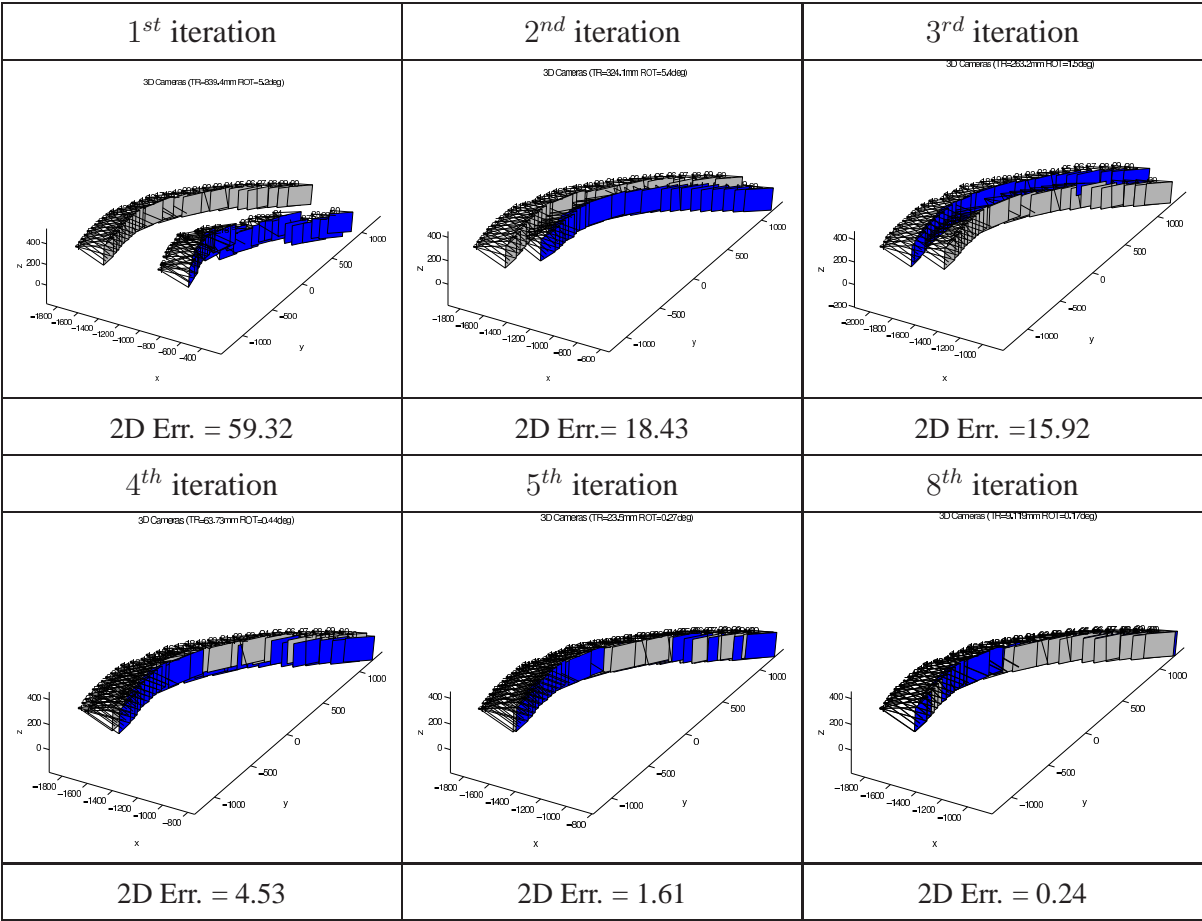


Figure 5: Iterations of the robust perspective factorization algorithm in the *Arc Case* and comparison with bundle adjustment. The first iteration corresponds to weak-perspective factorization. The bundle adjustment solution is shown in grey. The bundle adjustment solution does not perform outlier treatment.

6 3-D Reconstruction

The robust perspective factorization algorithm was also applied to the problem of 3-D reconstruction from multiple views. For this purpose we used images of objects using a single camera and a turning table. More specifically, we used the following data sets:

- The “Dino” and “Temple” data sets from the Middlebury’s evaluation of Multi-View Stereo reconstruction algorithms;³
- The “Oxford dinosaur” data set,⁴ and
- The “Square Box” data set.

We used the OpenCV⁵ pyramidal implementation of the Lucas & Kanade interest point detector and tracker [4] to obtain the initial set of 2-D observations. This provides the $2k \times n$ measurement matrix \mathbf{S} as well as the missing-data binary variables μ_{ij} associated with each observation. Figures 6 and 7 and Table 3 summarize the camera calibration and reconstruction results. For both the Middlebury data sets (Dino and Temple) and for the Oxford data set (Dinosaur) we compared our camera calibration results with the calibration data provided with the data sets, i.e., we measured the error in rotation and the error in translation between our results and the data provided in advance.

The first row in Table 3 introduces the test cases. The second row corresponds to the number of views. The third and fourth rows provide the size of the measurement matrix based on the number of views and on the maximum number of observations over all the views. The sixth row provides the number of actually observed 2-D points while the seventh row provides the number of 2-D inliers (observations with a posterior probability greater than 0.4). The eighth row provides the number of actual 3-D reconstructed points. One may notice that, in spite of missing data and of the presence of outliers, the algorithm is able to reconstruct a large percentage of the observed points. In the “Dino” and “Temple” example we compared our camera calibration results with the groundtruth calibration parameters provided with the Middlebury multi-stereo dataset. Please

³<http://vision.middlebury.edu/mview/data/>

⁴<http://www.robots.ox.ac.uk/~vgg/data/data-mview.html>

⁵<http://www.intel.com/technology/computing/opencv/>

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65


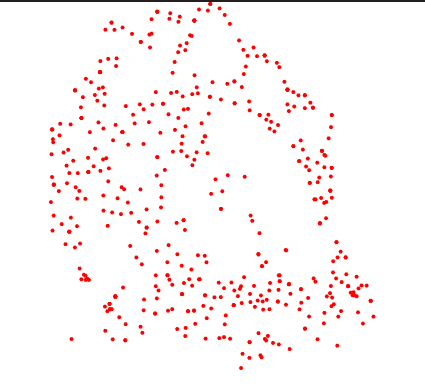
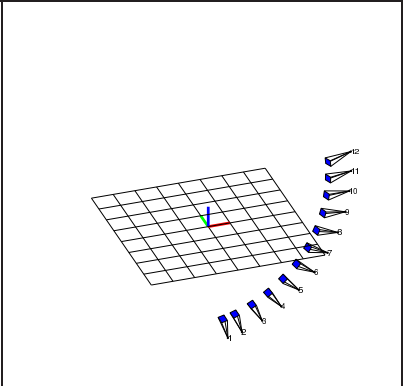

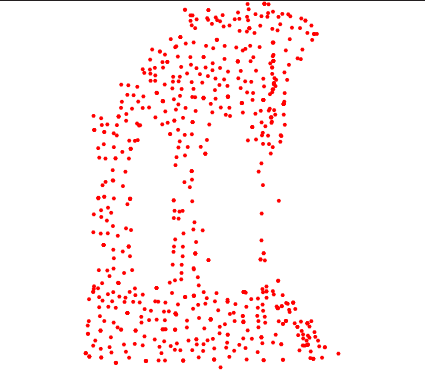
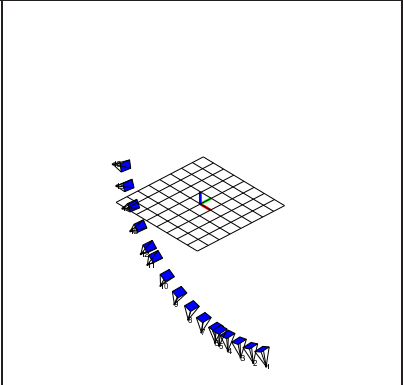

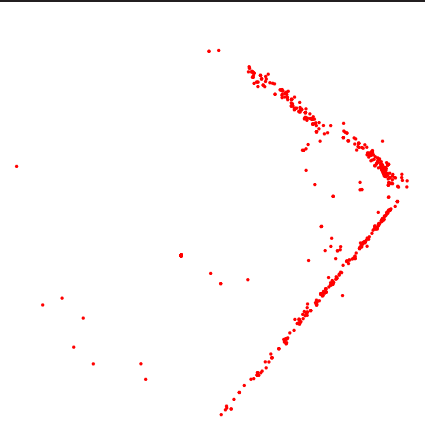
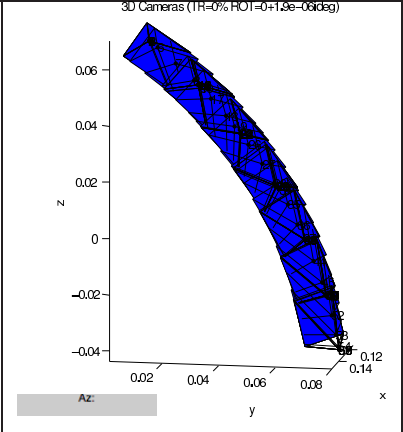
	Input Image	3-D Points	Cameras
Middlebury Dino			
Middlebury Temple			
INRIA Box			

Figure 6: Groundtruth data is represented in gray (light) colour, whereas reconstruction results are represented in blue (dark) colour.

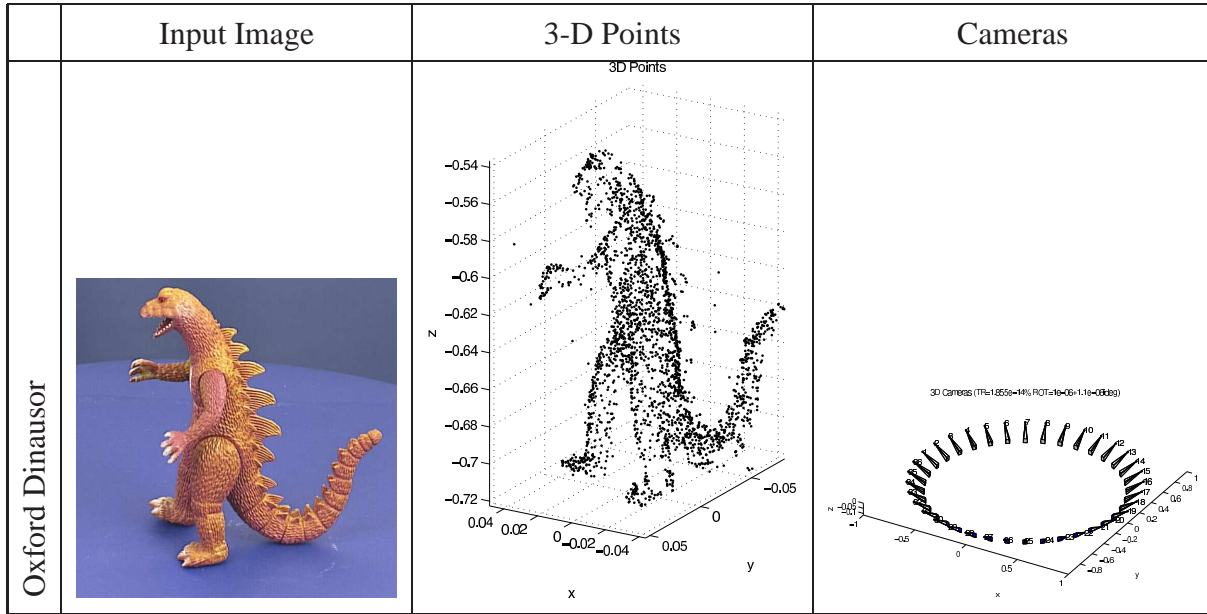


Figure 7: Groundtruth data is represented in gray (light) colour, whereas reconstruction results are represented in blue (dark) colour.

3-D reconstruction		Dino	Temple	Box	Dinausor
Input	# Views	12	16	64	36
	Size of \mathbf{S} matrix	24×480	32×758	128×560	72×1516
	# 2-D predictions	5760	12128	35840	54576
	% Missing observations	11%	21%	17%	85%
	# 2-D Observations	5140	9573	29733	8331
Results	# 2-D Inliers	3124	6811	25225	7645
	# 3-D Inliers	370	720	542	1437
	2D error (pixels)	0.82	0.93	0.69	0.33
	Rot. error (degrees)	1.49	2.32	–	0.00
	Trans. error (mm)	0.01	0.01	–	0.00
	# Aff. iter. (# EM iter.)	7 (4)	7 (3.14)	9 (3)	7 (3.29)

Table 3: Summary of the 3-D reconstruction results for the four data sets.

note that these datasets are made available without the groundtruth 3-D data. They are typically used by the community to compare results for 3-D dense reconstructions. The rotation error stays within 3 degrees. The translation error is very small because, in this case we aligned the camera centers and not the 3-D coordinates of some reconstructed points.

The results obtained with the Oxford “Dinosaur” need some special comments. Because of the very large percentage of missing data, we have been unable to initialize the solution with the PowerFactorization method. Therefore, we provided the camera calibration parameters for initialization. However, this kind of problem can be overcome by using an alternative affine factorization algorithm [38].

In order to further assess the quality of our results, we used the 3-D reconstructed points to build a rough 3-D mesh and to further apply a surface-evolution algorithm to the latter in order to obtain a more accurate mesh-based 3-D reconstruction [45]. The results are shown on Figure 8.⁶

7 Comparison with other methods

As already mentioned in section 3, our robust ML estimator has strong similarities with M-estimators and their practical implementation, i.e., IRLS [36]. Previous work on robust affine factorization has successfully used the following reweighting function ϕ that corresponds to the truncated quadratic:

$$\phi(x) = \begin{cases} 1 & \text{if } |x| < k \\ \sqrt{\frac{k^2}{x^2}} & \text{otherwise} \end{cases} \quad (37)$$

It is therefore tempting to replace the EM procedure of our algorithm with an IRLS procedure, which amounts to replace the posterior probabilities of inliers α_{ij}^{in} given by eq. (10) with the weights ϕ_{ij} given by eq. (37). The latter tends to zero most quickly allowing aggressive rejection of outliers. One caveat is that the efficiency of IRLS depends on the tuning parameter k . Unfortunately the latter cannot be estimated within the minimization process as is the case with the covariance matrix. However, we noted that the results that we obtained do not depend on the

⁶They are also available at <http://vision.middlebury.edu/mview/eval/>.

inria-00446987, version 1 - 14 Jan 2010

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65



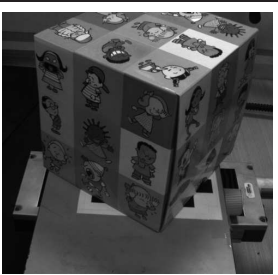

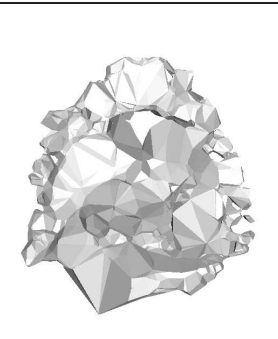
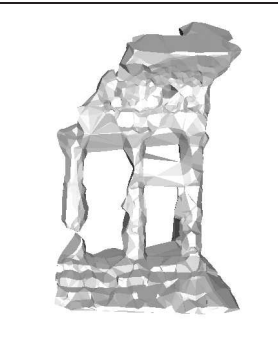
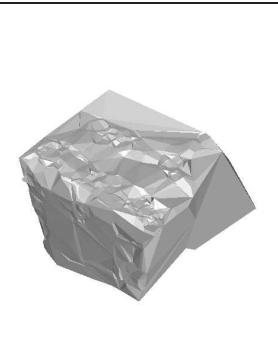
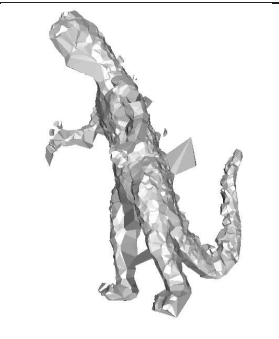
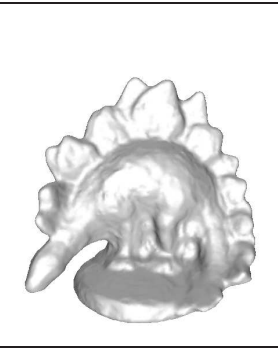
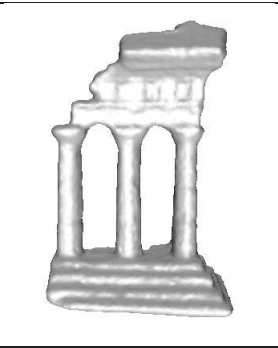
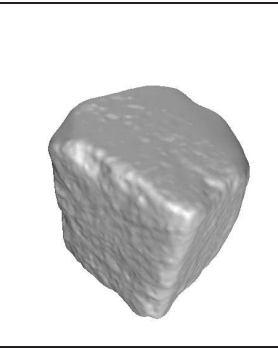
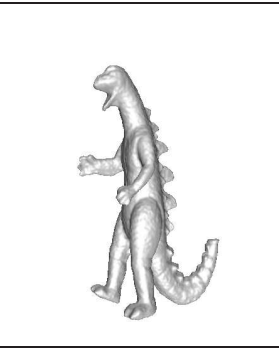
Dino	Temple	Box	Dinausor
			
			
			

Figure 8: Dense reconstruction results: A rough mesh obtained from the 3-D reconstructed points (middle) and the final dense reconstruction (bottom) after surface evolution using the method described in [45].

choice of k . In all the experiments reported below, we used $k = 1$. The plot of the truncated quadratic for different k values is plotted in Figure 9.

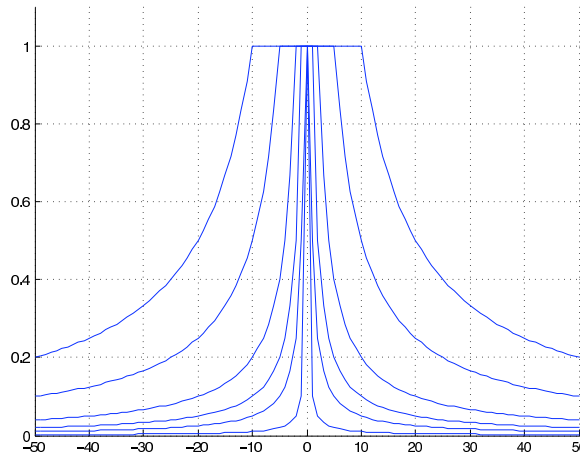


Figure 9: Reweighting function for $k = 0.1, 0.5, 1, 2, 5, 10$.

We compared the two robust methods (our EM-based robust perspective factorization algorithm and an equivalent IRLS-based algorithm) with five data sets for which we had the ground truth: Three multiple-camera calibrations data sets (the *Corner Case*, the *Arc Case* and the *Semi-Spherical Case*) and two multi-view reconstruction data sets (*Dino* and *Temple*). The results of this comparison are summarized in Table 4.

In the *Corner* case the quality of the results are very similar: our algorithm accepted 94% of the total number of observations as inliers and reconstructed 99.3% of the total number of 3-D points, while IRLS accepted all the observations as inliers and reconstructed all the 3-D points. Similar results are obtained in the *Arc* and *Semi-Spherical* cases, where the proposed method performs slightly better. Both algorithms were able to reconstruct the *Dino* and the *Temple*, but our algorithm yields more accurate results. Outlier detection is summarized in Table 5.

A more thorough comparison with robust as well as non robust 3-D reconstruction methods is provided in Figure 10. The proposed algorithm is denoted by "Persp. Power Factorization (Bayesian)", while the IRLS method is named "Persp. Power Factorization (IRLS - Truncated Quadratic)" and the non-robust method is called "Persp. Power Factorization (Not Robust)". Affine factorization algorithms are also presented, together with the results of bundle adjustment. The bundle adjustment method was always initialized using the PowerFactorization method. The robust perspective factorization method proposed in this paper is the most resilient

Dataset	Method	2-D Inliers	3-D Inliers	2-D err.	3-D err.	Rot. err.	Trans. err.
<i>Corner</i>	EM	5202 (5527)	285 (292)	0.30	6.91	0.13	27.02
	IRLS	5526 (5527)	288 (292)	0.40	6.91	0.14	26.61
<i>Arc</i>	EM	6790 (6960)	232 (232)	0.19	2.65	0.18	9.37
	IRLS	6960 (6960)	232 (232)	0.22	2.54	0.16	8.78
<i>Semi-Spherical</i>	EM	784 (863)	122 (128)	0.48	4.57	0.27	24.21
	IRLS	862 (863)	128 (128)	0.62	4.66	0.29	23.91
<i>Dino</i>	EM	3124 (5140)	370 (480)	0.82	–	1.49	0.01
	IRLS	3411 (5140)	390 (480)	2.57	–	2.13	0.01
<i>Temple</i>	EM	6811 (9573)	720 (758)	0.93	–	2.32	0.01
	IRLS	7795 (9573)	731 (758)	1.69	–	2.76	0.03

Table 4: Comparison between robust perspective factorization results using EM and IRLS. The figures in paranthesis correspond to the total number of observations (third column) and to the total number of expected 3-D points (fourth column).

	<i>Corner</i>	<i>Arc</i>	<i>Semi-Spherical</i>	<i>Dino</i>	<i>Temple</i>
EM	6%	2%	9%	39%	29%
IRLS	0%	0%	0%	34%	19%

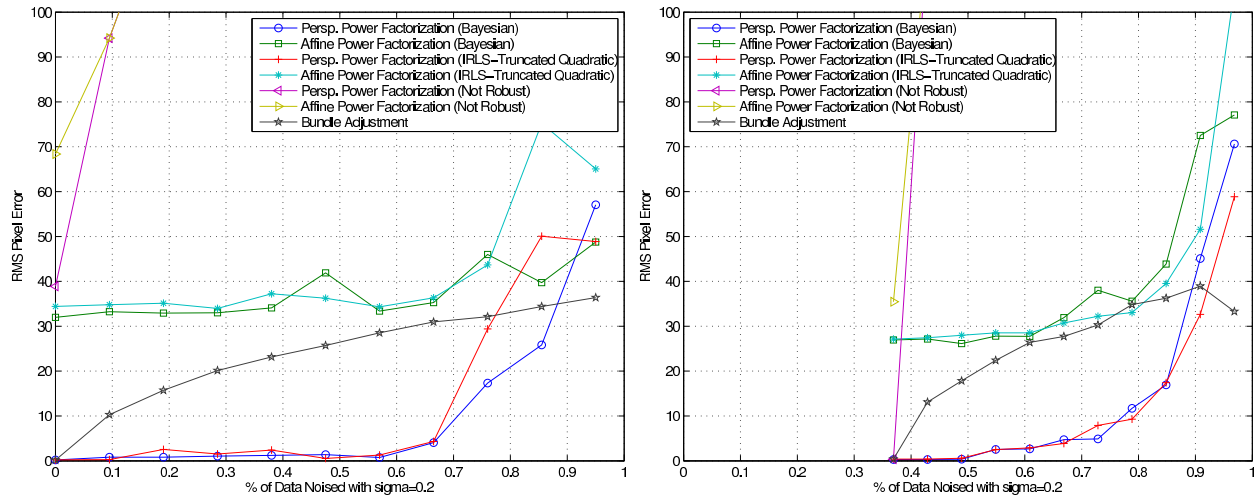
Table 5: Percentage of outliers detected by the two algorithms.

to high-amplitude noise. It generally performs better than the IRLS method and provides a clear advantage against the non-robust methods, which *exit* the graphs as soon as the noise level increases. As it can be observed, in the *Semi-Spherical Case*, the solution deteriorates a lot faster in the presence of noise, due to the lack of the redundancy in the data (128 3-D points and 10 cameras, versus 292 points and 30 cameras in the *Corner Case* and 232 points and 30 cameras in the *Arc Case*).

Figure 11 compares our method (a), with the bundle adjustment method (b), in the *Arc Case* and when 20% of the input data was corrupted by high-amplitude noise ($\sigma = 0.20$ of the image size). On both figures the ground truth is shown in grey and the result of the algorithm is shown in blue (or dark in the absence of color). Notice that with this level of data perturbation,

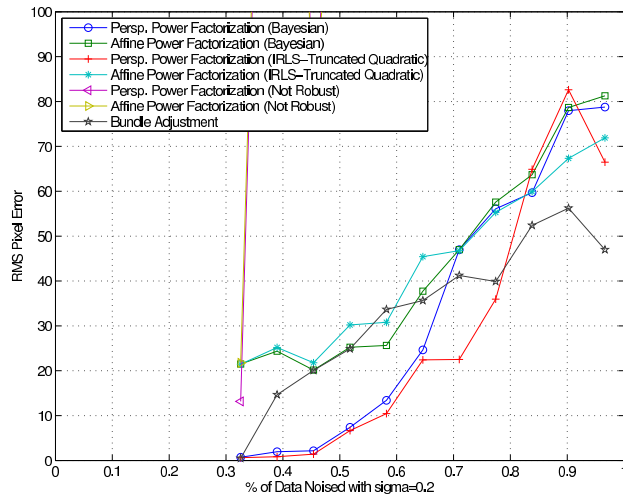
inria-00446987, version 1 - 14 Jan 2010

bundle adjustment completely failed to find the correct solution.



(a) Arc Case

(b) Corner Case



(c) Semi-Spherical Case

Figure 10: Behavior of various robust and non robust algorithms when an increasing percentage of the input data are corrupted by high-amplitude noise, namely $\sigma = 0.2$ of the image size.

8 Conclusions

In this paper we described a robust factorization method based on data clustering and on the EM algorithm. First we recalled the classical maximum-likelihood approach within which all the observations are supposed to be independent and identically distributed. This amounts to classify

inria-00446987, version 1 - 14 Jan 2010

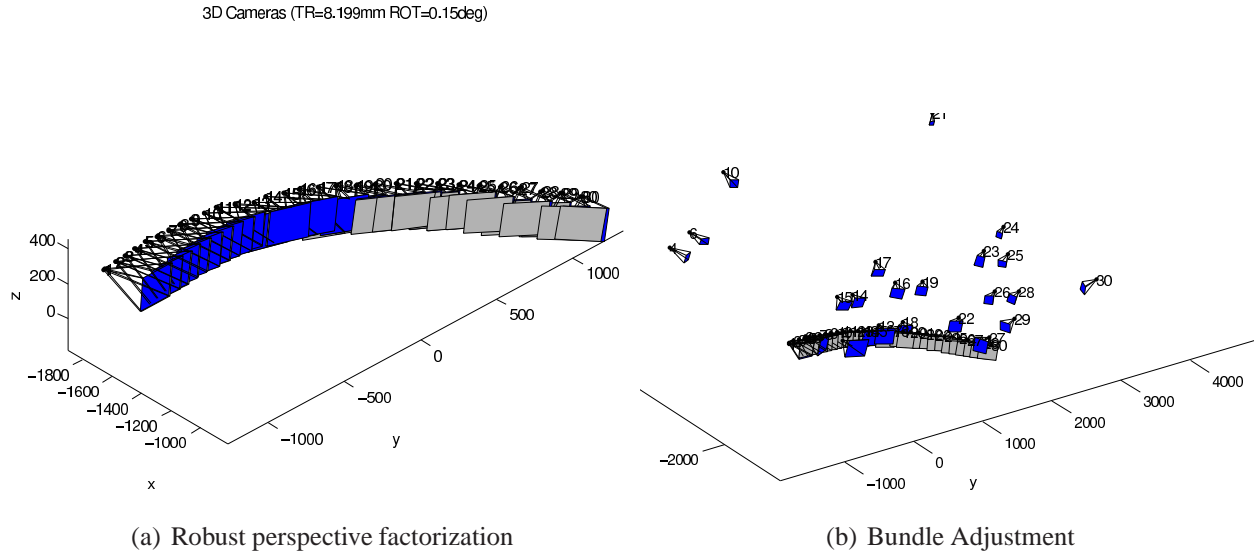


Figure 11: Calibration results in the *Arc Case* for (a) the proposed method and for (b) bundle adjustment method, when 20% of the input data are corrupted with high-amplitude noise, namely $\sigma = 0.2$ of the image size. The 2-D reprojection error is of 0.71 pixels for (a) and 15.84 pixels for (b). The groundtruth is represented in gray.

all the observations in one cluster – inliers. Next we considered a mixture model within which the likelihood of the inlier class has a normal distribution and the likelihood of the outlier class has a uniform distribution. This naturally leads to ML with missing variables which is solved in practice via the Expectation-Maximization algorithm. We formally derived the latter in the specific case of 3-D reconstruction and of a Gaussian/uniform mixture; This allowed us to rely on EM’s convergence properties.

Moreover, we devised two shape and motion algorithms: (i) affine factorization with EM and (ii) robust perspective factorization, the former residing in the inner loop of the latter. These two algorithms are very general since they can accomodate with any affine factorization and with any iterative perspective factorization methods.

We performed extensive experiments with two types of data sets: multiple-camera calibration and 3-D reconstruction. We compared the calibration results of our algorithm with the results obtained using other methods such as the bundle adjustment technique and IRLS. It is interesting to notice that there is almost no noticeable quantitative difference between our algorithm and a non-linear optimization method such as bundle adjustment. The 3-D reconstruction results obtained with a single camera and objects lying on a turntable are also very good. Whenever possible, we

inria-00446987, version 1 - 14 Jan 2010

compared our results with ground-truth data, such as the external camera parameters provided by the Middlebury multi-view stereo data set. In order to further assess the 3-D reconstruction results, we used the output of the robust perspective factorization method, namely a cloud of 3-D points, as input of a mesh-based reconstruction technique.

Our Gaussian/uniform mixture model and its associated EM algorithm may well be viewed as a robust regression method in the spirit of M-estimators. We compared our method with IRLS using a truncated quadratic loss function. The results show that our method performs slightly better, although we believe that these results are only preliminary. A thorough comparison between outlier detection using probability distribution mixture models on one side, and robust loss functions on the other side is a topic in its own right. In the future we plan to extend our method to deal the more difficult problem of multiple-body factorization.

A Derivation of equation (19)

We recall eq. (17):

$$Q_{ML} = \frac{1}{2} \sum_{i,j} \left((\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top \mathbf{C}^{-1} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) + \log(\det \mathbf{C}) \right)$$

Taking the derivative with respect to the entries of the 2×2 matrix \mathbf{C} we obtain:

$$\frac{\partial Q_{ML}}{\partial \mathbf{C}} = \frac{1}{2} \sum_{i,j} \left(-\mathbf{C}^{-\top} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top \mathbf{C}^{-\top} + \mathbf{C}^{-\top} \right) \quad (38)$$

$$= -\mathbf{C}^{-\top} \left(\frac{1}{2} \sum_{i,j} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top \right) \mathbf{C}^{-\top} + \frac{m}{2} \mathbf{C}^{-\top} \quad (39)$$

where $m = k \times n$. By setting the derivative to zero we obtain eq. (19):

$$\mathbf{C} = \frac{1}{m} \sum_{i,j} (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})) (\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}))^\top$$

B Derivation of equation (21)

When considering isotropic covariance, $\mathbf{C} = \sigma^2 \mathbf{I}$, hence $\det \mathbf{C} = \sigma^4$, and the equation becomes:

$$Q_{ML} = \frac{1}{2} \sum_{i,j} \left(\frac{1}{\sigma^2} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2 + 2 \log(\sigma^2) \right) \quad (40)$$

By taking the derivative with respect to σ^2 , we obtain:

$$\frac{\partial Q_{ML}}{\partial \sigma^2} = \frac{1}{2} \sum_{i,j} \left(-\frac{1}{(\sigma^2)^2} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2 + 2 \frac{1}{\sigma^2} \right) \quad (41)$$

$$= \frac{1}{2} \sum_{i,j} \left(\frac{2\sigma^2 - \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta})\|^2}{\sigma^2} \right) \quad (42)$$

By setting the derivative to zero, we obtain eq. (21):

$$\sigma^2 = \frac{1}{2m} \sum_{i,j} \|\mathbf{s}_{ij} - \hat{\mathbf{s}}_{ij}(\boldsymbol{\theta}^*)\|^2$$

where $m = k \times n$.

References

- [1] H. Aanaes, R. Fisker, and K. Astrom. Robust factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1215–1225, Sept. 2002.
- [2] P. Anandan and M. Irani. Factorization with uncertainty. *International Journal of Computer Vision*, 49(2/3):101–116, 2002.
- [3] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [4] J-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker - description of the algorithm. Technical report, Intel Corporation, 2001.
- [5] S. Brant. Closed-form solutions for affine reconstruction under missing data. In *Proceedings of Statistical Methods for Video Processing (ECCV '02 Workshop)*, pages 109–114, 2002.

inria-00446987, version 1 - 14 Jan 2010

- 1
2
3
4
5
6 [6] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views
7 by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,
8 18(11):1098–1104, November 1996.
9
10
11 [7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood estimation from incom-
12 plete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society,*
13 *Series B*, 39:1–38, 1977.
14
15
16 [8] O. Faugeras. *Three Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press,
17 1993.
18
19
20
21
22 [9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting
23 with applications to image analysis and automated cartography. *Communications of the*
24 *ACM*, 24(6):381–395, June 1981.
25
26
27
28 [10] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density
29 estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.
30
31
32 [11] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press,
33 Baltimore, 1989.
34
35
36 [12] A. Gruber and Y. Weiss. Factorization with uncertainty and missing data: exploring tempo-
37 ral coherence. In *Proceedings Neural Information Processing Systems (NIPS'2003)*, 2003.
38
39
40 [13] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using
41 the em algorithm. In *Proceedings Conference on Computer Vision and Pattern Recognition*,
42 pages 707–714, 2004.
43
44
45 [14] L. Hajder and D. Chetverikov. Robust structure from motion under weak perspective. In
46 *Proc. of the second International Symposium on 3D Data Processing, Visualization, and*
47 *Transmission*, September 2004.
48
49
50 [15] R. Hartley and F. Schaffalitzky. Powerfactorization: 3d reconstruction with missing or
51 uncertain data. Technical report, Australian National University, 2003.
52
53
54 [16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge
55 University Press, Cambridge, UK, 2000.
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4
5
6 [17] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt.*
7 *Soc. Amer. A.*, 4(4):629–642, 1987.
8
9
10 [18] D. Q. Huynh, R. Hartley, and A. Heyden. Outlier correction in image sequences for the
11 affine camera. In *Proceedings ICCV*, volume 1, pages 585–590, 2003.
12
13
14 [19] D. Q. Huynh and A. Heyden. Robust factorization for the affine camera: Analysis and
15 comparison. In *Proc. Seventh International Conference on Control, Automation, Robotics,*
16 *and Vision*, Singapore, December 2002.
17
18
19
20 [20] Kanatani K. Geometric information criterion for model selection. *International Journal of*
21 *Computer Vision*, 26(3):171–189, 1998.
22
23
24 [21] Kanatani K. Uncertainty modeling and model selection for geometric inference. *IEEE*
25 *Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1307–1319, October
26 2004.
27
28
29
30 [22] Q. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences
31 and fundamental matrices. *International Journal of Computer Vision*, 1:5–40, 1997.
32
33
34 [23] Q.-T. Luong and O. D. Faugeras. *The Geometry of Multiple Images*. MIT Press, Boston,
35 2001.
36
37
38 [24] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In
39 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*
40 *'00)*, volume 2, pages 430 – 437, June 2000.
41
42
43 [25] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods
44 for projective structure from motion. In *IEEE Conf. on Computer Vision and Pattern Recog-*
45 *niton (CVPR)*, 2001.
46
47
48 [26] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New-York,
49 1997.
50
51
52 [27] P. Meer. Robust techniques for computer vision. In *Emerging Topics in Computer Vision*.
53 Prentice Hall, 2004.
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4
5
6 [28] D. J. Miller and J. Browning. A mixture model and em-based algorithm for class discovery, robust classification and outlier rejection in mixed labeled/unlabeled data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1468–1483, November 2003.
- 7
8
9
10
11
12
13
14 [29] I. Miyagawa and K. Arakawa. Motion and shape recovery based on iterative stabilization for modest deviation from planar motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1176–1181, July 2006.
- 15
16
17
18
19
20
21 [30] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proceedings of International Conference of Computer Vision*, pages 696–702, 1998.
- 22
23
24
25
26
27 [31] J. Oliensis and R. Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217–2233, December 2007.
- 28
29
30
31
32
33 [32] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.
- 34
35
36
37 [33] P. J. Rousseeuw and S. Van Aelst. Positive-breakdown robust methods in computer vision. In Berk and Pourahmadi, editors, *Computing Science and Statistics*, volume 31, pages 451–460. Interface Foundation of North America, 1999.
- 38
39
40
41
42
43 [34] S. Roweis. EM algorithm for PCA and SPCA. *Proceedings NIPS*, 10:626–632, 1997.
- 44
45
46
47 [35] H. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):855–867, 1995.
- 48
49
50
51
52
53 [36] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.
- 54
55
56
57 [37] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of LNCS, pages 709–720, April 1996.
- 58
59
60
61
62
63
64
65

- 1
2
3
4
5
6 [38] J. Ph. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix
7 factorization with application to structure-from-motion. In *Proc. of IEEE Conference on*
8 *Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007.
9
10
11 [39] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a
12 factorization method. *International Journal of Computer Vision*, 9(2):137–154, November
13 1992.
14
15
16 [40] M. Trajkovic and M Hedley. Robust recursive structure and motion recovery under affine
17 projection. In *Proc. of the British Machine Vision Conference*, September 1997.
18
19
20 [41] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern
21 synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–375. LNCS, 2000.
22
23 [42] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization
24 and gpca. In *Proceedings CVPR*, volume 2, pages 310–316, 2004.
25
26
27 [43] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-d location parameters using dual
28 number quaternions. *CGVIP-Image Understanding*, 54(3):358–367, November 1991.
29
30
31 [44] T. Wiberg. Computation of principal components when data are missing. In *Proceedings*
32 *Symposium of Computational Statistics*, pages 229–326, 1976.
33
34
35 [45] A. Zaharescu, E. Boyer, and R. P. Horaud. Transformesh: a topology-adaptive mesh-based
36 approach to surface evolution. In *In Proceedings of the Eighth Asian Conference on Com-*
37 *puter Vision*, LNCS, Tokyo, Japan, November 2007. Springer.
38
39
40 [46] A. Zaharescu, R. Horaud, R. Ronfard, and L. Lefort. Multiple camera calibration using
41 robust perspective factorization. In *Proceedings of the 3rd International Symposium on*
42 *3D Data Processing, Visualization and Transmission, Chapel Hill (USA)*. IEEE Computer
43 Society Press, 2006.
44
45
46 [47] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern*
47 *Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
48
49
50 [48] Z. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern*
51 *Analysis and Machine Intelligence*, 26(7):892–899, 2004.
52
53
54
55
56
57
58
59
60
61
62
63
64
65