

ROBUST FACTORIZATION

Henrik Aanæs Rune Fisker Kalle Åström

Jens Michael Carstensen

Abstract

Factorization algorithms for recovering structure and motion from an image stream have many advantages, but they usually require a set of well tracked features. Such a set is in general not available in practical applications. There is thus a need for making factorization algorithms deal effectively with errors in the tracked features.

We propose a new and computationally efficient algorithm for applying an arbitrary error function in the factorization scheme. This algorithm enables the use of robust statistical techniques and arbitrary noise models for the individual features. These techniques and models enable the factorization scheme to deal effectively with mismatched features, missing features and noise on the individual features. The proposed approach further includes a new method for Euclidean reconstruction that significantly improves convergence of the factorization algorithms.

The proposed algorithm has been implemented as a modification of the Christy–Horn factorization scheme, which yields a perspective reconstruction. Based on this implementation a considerable increase in error tolerance is demonstrated on real and synthetic data.

The proposed scheme can however be applied to most other factorization algorithms.

keywords: Robust statistics, feature tracking, perspective reconstruction, Euclidean reconstruction, structure from motion.

1 Introduction

The reconstruction of structure and motion of a rigid object from an image stream is one of the most studied problems within computer vision. A popular way of addressing this problem is to extract and track features through the image sequence and then limit the problem to estimating the structure and motion of these tracked features. A family of effective and popular algorithms for solving this estimation problem are the so called factorization algorithms, see e.g. [6, 7, 16, 17, 18, 20, 22, 28].

These factorization algorithms work by linearizing the camera observation model and give good results rapidly and without an initial guess for the solution. Hence the factorization algorithms are good candidates for solving the structure and motion problem, either as a full solution or as initialization to other algorithms such as bundle adjustment, see e.g. [25, 30].

The factorization algorithms assume that the correspondence or feature tracking problem has been solved. The correspondence problem is, however, one of the most difficult fundamental problems within computer vision. No perfect and truly general solution has yet been presented. For most practical purposes one must deal with erroneously tracked features as input to the factorization algorithm. This fact poses a considerable challenge to factorization algorithms, since they implicitly assume independent identical distributed Gaussian noise on the 2D features (the 2-norm is used as error function on the 2D features). This noise assumption based on the 2-norm is known to perform rather poorly in the presence of outliers induced by such erroneous data. These errors typically arise from mismatching 2D features or from a 2D feature being absent due to occlusion. It is common for badly tracked features to disturb the estimation of structure and motion considerably.

Previous attempts have been made at addressing this problem. Irani and Anandan [16] assumes that the noise is separable in a 3D feature point contribution and a frame contribution. In other

words if a 3D feature point has a relatively high uncertainty in one frame it is assumed that it has a similar high uncertainty in all other frames. However, large differences in the variance of the individual 2D feature points is critical to the implementation of robust statistical techniques that can deal with feature point noise, missing features, and feature mismatch in single frames. Morris and Kanade [20] propose a bilinear minimization method as an improvement on top of a standard factorization. The bilinear minimization incorporates directional uncertainty models in the solution. However, the method does not implement robust statistical techniques. Tomasi and Kanade [28] and Jacobs [17] address the problem of missing data points by the use of heuristics. Attempts at solving similar linear problems, in the presence of missing and erroneous data, has also been made, e.g. [24].

Here we propose a combined approach that deals effectively with missing features and is robust towards errors in the matching of the 2D features in a factorization framework. This is achieved by allowing for an arbitrary noise model on the 2D features – i.e. we are not restricted to a Gaussian model. Hereby the proposed approach is capable of dealing effectively with mismatched features or outliers by the use of robust statistics. Arbitrary noise models also deal with missing 2D features by emulating them as being located arbitrarily in the image with very high noise variance.

The proposed approach is implemented as an improvement to the factorization algorithm of Christy and Horaud [6]. The Christy–Horaud algorithm has the advantage, that it assumes a perspective camera model as opposed to a linearized version, e.g. [28]. The Christy–Horaud algorithm with the proposed approach incorporated deals efficiently with real and simulated data containing large feature errors.

The presentation is organized by giving an overview of the factorization algorithms in Section 2 followed by a discussion of how to deal with erroneous data in Section 3. In Section 4 a new

numerical algorithm for estimating the optimal subspace of a matrix with weighted entries is proposed. In Sections 5 and 6 schemes improving the robustness of the factorization approach are presented, followed by experimental results in Section 7.

2 Factorization Overview

As a courtesy to the reader and to introduce notation a short overview of the factorization algorithm is presented. For a more thorough introduction the reader is referred to [5]. All the factorization methods cited utilize some linearization of the pinhole camera with known intrinsic parameters:

$$s_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_i^T & t_i^x \\ \mathbf{b}_i^T & t_i^y \\ \mathbf{c}_i^T & t_i^z \end{bmatrix} \begin{bmatrix} \mathbf{P}_j \\ 1 \end{bmatrix}, \quad (1)$$

where the 3D feature, \mathbf{P}_j , is projected in frame i as (x_{ij}, y_{ij}) , $\mathbf{t}_i = [t_i^x, t_i^y, t_i^z]^T$ is the appropriate translation vector and $\mathbf{a}_i^T, \mathbf{b}_i^T$ and \mathbf{c}_i^T are the three rows vectors of the rotation matrix. The used/approximated observation model can thus be written as:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \mathbf{M}_i \mathbf{P}_j, \quad (2)$$

where \mathbf{M}_i is the 2×3 'linearized motion' matrix associated with frame i .

When n features have been tracked in k frames, $i \in [1 \dots k]$ and $j \in [1 \dots n]$, the observations from (2) can be combined to:

$$\mathbf{S} = \mathbf{M} \mathbf{P}, \quad (3)$$

where \mathbf{M} is a $2k \times 3$ matrix composed of the \mathbf{M}_i and \mathbf{P} is a $3 \times n$ matrix composed of the \mathbf{P}_j .

Thus the elements of \mathbf{S} are given by:

$$\mathbf{S} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kn} \\ y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{k1} & \cdots & y_{kn} \end{bmatrix} .$$

The solution to this linearized problem is then found as the \mathbf{M} and \mathbf{P} that minimize:

$$\mathbf{N} = \mathbf{S} - \mathbf{MP} , \quad (4)$$

where \mathbf{N} is the residual between model, \mathbf{MP} , and the data, \mathbf{S} . The residuals, \mathbf{N} , are usually minimized using the Frobenius norm. This is equivalent to minimizing the squared Euclidean norm of the reprojection error, i.e. the error between the measured 2D features and the corresponding reprojected 3D feature. Thus the objective function is:

$$\min_{\mathbf{M}, \mathbf{P}} \|\mathbf{S} - \mathbf{MP}\|_F^2 = \min_{\mathbf{M}, \mathbf{P}} \sum_{j=1}^n \|\mathbf{S}_j - \mathbf{MP}_j\|_2^2 , \quad (5)$$

where \mathbf{S}_j and \mathbf{P}_j denote the j^{th} column of \mathbf{S} and \mathbf{P} , respectively. In this case the solution to \mathbf{M} and \mathbf{P} can be found through the singular value decomposition, SVD, of \mathbf{S} .

It is noted, that for any invertible 3×3 matrix, \mathbf{A} :

$$\mathbf{MP} = \mathbf{MAA}^{-1}\mathbf{P} = \tilde{\mathbf{M}}\tilde{\mathbf{P}} . \quad (6)$$

Hence the solution is only defined up to an affine transformation. In [6] an Euclidean reconstruction is achieved by estimation of an \mathbf{A} , such that the rotation matrices, $[\mathbf{a}_i \ \mathbf{b}_i \ \mathbf{c}_i]^T$, are as orthonormal as possible. Further details and an improved approach is presented in Section 6.

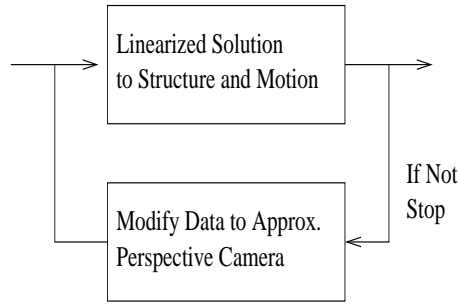


Figure 1: Overview of the Christy–Horaud factorization algorithm. A linearized – e.g. paraperspective – solution is iterated into a perspective.

2.1 Christy–Horaud Factorization

The factorization algorithm that the proposed method extends on is that of Christy and Horaud [6], see Figure 1. The Christy–Horaud algorithm is an extension to the work in [22, 28] partly by incorporating the work of [8]. The algorithm has the advantage of iteratively achieving a solution to the original non–linearized problem, thus achieving perspective reconstruction. This is achieved by iteratively solving a linearized version of the problem, and then modifying the data to approach the perspective camera. The update formula for the data, i.e. \mathbf{S} or x_{ij}, y_{ij} is [6]:

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \left(\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} - \begin{bmatrix} x_{o_i} \\ y_{o_i} \end{bmatrix} \right) (1 - \epsilon_{ij}) , \quad (7)$$

where $(\tilde{x}_{ij}, \tilde{y}_{ij})$ is the updated data, (x_{o_i}, y_{o_i}) is the object origin projected onto frame i and ϵ_{ij} is the scaled depth defined as:

$$\epsilon_{ij} = \frac{\mathbf{c}_i^T \cdot \mathbf{P}_j}{t_i^z} . \quad (8)$$

For further details the reader is referred to [5, 6].

3 Dealing with Imperfect Data

3.1 Types of Errors

A framework for dealing with imperfect data should be geared towards the types of errors expected. Three types of errors have been identified, the first two originating from [31].

- **Bad Feature Locations.** The locations of a 2D feature is distorted. If this distortion is anisotropic and/or the covariance structure varies the Frobenius norm is sub-optimal. This is the problem addressed in [16, 20].
- **False Matches.** There has been a mismatch of 2D features, i.e. two 2D feature originating from *different* 3D features have been matched as originating from the *same* 3D feature.
- **Missing Feature.** The projection of a 3D feature can not be found or matched, e.g. due to occlusion or simply by 'breakdown' of the feature matching algorithm. This is addressed in [17, 28].

Our proposed method deals with all these types of errors by attaching an uncertainty to the individual 2D features. It is implemented by associating a weighting structure to \mathbf{S} . This mainly effects the solution of (5), but in the case of Christy–Horaud it also effects the object frame origin (see Section 5). When weights are introduced (5) becomes:

$$\min_{\mathbf{M}, \mathbf{P}} \sum_{j=1}^n \|\mathbf{V}_j(\mathbf{S}_j - \mathbf{MP}_j)\|_2^2, \quad (9)$$

where \mathbf{V}_j is an $2k \times 2k$ weighting matrix representing the weights of the j^{th} column of \mathbf{S} . The minimization of (9) is the subject of Section 4. The $\mathbf{V}_j^T \mathbf{V}_j$ is seen to be the inverse covariance structure of \mathbf{S}_j and (9) is equivalent to minimizing the Mahalanobis distance. The variance, Σ_j , of the noise on S_j is incorporated by:

$$\Sigma_j^{-1} = \mathbf{V}_j^T \mathbf{V}_j. \quad (10)$$

An approach with this uncertainty formulation is seen to deal with the three types of identified errors:

- **Bad Feature Locations.** Assuming Gaussian noise, the weights can be constructed to incorporate the uncertainty structure of the 2D features as shown above. This approach can even deal with arbitrary Gaussian noise. With the presented formulation of the weights - one V_j matrix per column - covariance between the x and y coordinates can be expressed. An extension to general covariance between all features would require a 3D tensor formulation of the weights.
- **False Matches.** A mismatched 2D feature can be down weighted, even approaching zero, leaving it out of the optimization to any desired extent.
- **Missing Features.** Is equivalent to predicting that the missing 2D feature is located somewhere in the image, but that the uncertainty of the prediction is very high.

The information of missing features should be apparent from the feature matching algorithm and should be directly expressed in Σ_j . If prior knowledge about the distribution of bad feature location is present, this can also be expressed in Σ_j . However this not a requirement. *False Matches* are almost by nature unknown a priori so robust statistical techniques are employed to deal with these errors. Note that in the absence of prior knowledge, the Σ_j is initialized as the identity matrix.

3.2 Adaptive Weighting

As described in [14, 29, 31] among others, false matches give rise to outliers in the data. Here outliers are understood as 2D features where the residual, i.e. the distance between the original 2D feature and it's reprojected counterpart is large. If the percentage of false matches is

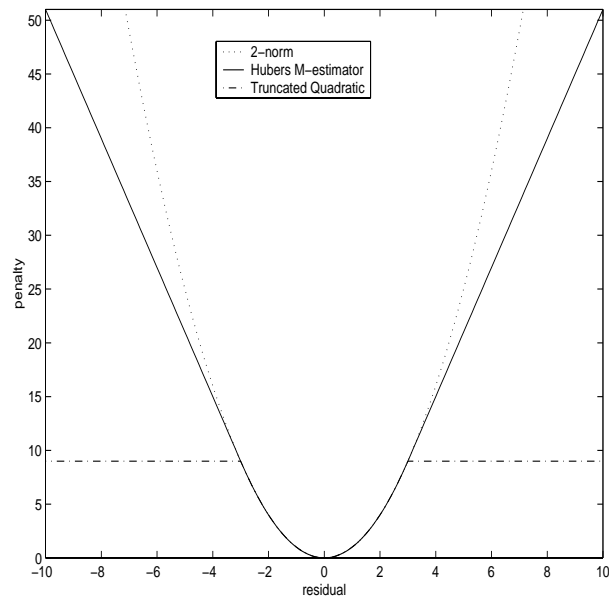


Figure 2: Three popular error functions with $k = 3$ for the truncated quadratic and the Huber's M-estimator.

relatively low (considerably lower than 50%) false matches can be dealt with effectively by diminishing the effect of outliers. In practice, this is done via a robust error function. Popular robust functions are the truncated quadratic and Huber's M-estimator (sometimes called Huber Norm), see Figure 2. For a detailed discussion the reader is referred to [2]. The use of robust error functions efficiently deals with the problem of false matches.

These error functions are implemented via Iteratively Reweighted Least Squares (IRLS). IRLS works by iteratively fitting the model to the data by minimizing the weighted least squares of the residuals. These weights are then altered such that the residuals are re-weighted according to the desired error function and not the 2-norm. In this approach, the weights, w_{ij} , from the

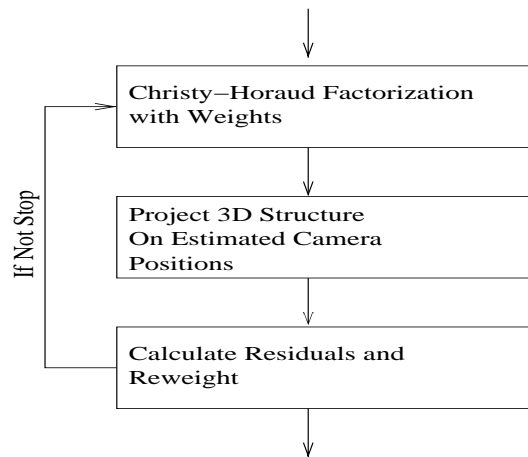


Figure 3: Overview of the proposed reweighting scheme. This is employed to deal effectively with false matches

4 Separation with Weights

The main computational problem in the proposed approach is to determine a solution to the weighted least squares problem (9). The solution to (9) is \mathbf{M} and \mathbf{P} given \mathbf{S} and \mathbf{V}_j . Note that a SVD can not be applied as for (5). To solve (9), a method similar to the idea in the Christy-Horand factorization algorithm [6] is proposed. This method is generally known as surrogate modeling, see e.g. [3]. Surrogate modeling works by applying a computationally 'simpler' model to iteratively approximate the original 'hard' problem.

The best known example of surrogate modeling is probably the Newton optimization method. Here a 2^{nd} order polynomial is approximated to the objective function in each iteration and a temporary optimum is achieved. This temporary optimum is then used to make a new 2^{nd} order approximation, and thus a new temporary optimum. This is continued until convergence is achieved.

Here (5) is used to iteratively approximate (9) getting a temporary optimum, which in turn can be used to make a new approximation. The approximation is performed by modifying the original data, \mathbf{S} , such that the solution to (5) with the modified data, $\tilde{\mathbf{S}}$, is the same as (9) with

the original data. By letting $\tilde{\cdot}$ denoting modified data, the goal is to obtain:

$$\begin{aligned}
\min_{M,P} \sum_{j=1}^n \|\mathbf{V}_j(\mathbf{S}_j - \mathbf{M}\mathbf{P}_j)\|_2^2 &= \\
\min_{M,P} \sum_{j=1}^n \mathbf{N}_j^T \mathbf{V}_j^T \mathbf{V}_j \mathbf{N}_j &\stackrel{def}{=} \\
\min_{M,P} \sum_{j=1}^n \tilde{\mathbf{N}}_j^T \tilde{\mathbf{N}}_j &= \\
\min_{M,P} \sum_{j=1}^n \|\tilde{\mathbf{S}}_j - \mathbf{M}\mathbf{P}_j\|_2^2 , &
\end{aligned} \tag{13}$$

where $\mathbf{N} = [\mathbf{N}_1 \dots \mathbf{N}_n]$ denotes the residuals:

$$\mathbf{N}_j = \mathbf{S}_j - \mathbf{M}\mathbf{P}_j .$$

Hereby the subspace, \mathbf{M} , can be found via SVD and \mathbf{P} via the normal equations once \mathbf{M} is known. Let q denote the iteration number, then the algorithm goes as follows:

1. **Initialize.** $\tilde{\mathbf{S}}^0 = \mathbf{S}$, $q = 1$.
2. **Estimate Model.** Get \mathbf{M}^q by the singular vectors corresponding to the three largest singular values of $\tilde{\mathbf{S}}^{q-1}$, via SVD. Get \mathbf{P}^q from

$$\forall j : \quad \mathbf{P}_j^q = \left[\mathbf{M}^{qT} \mathbf{V}_j^T \mathbf{V}_j \mathbf{M}^q \right]^{-1} \mathbf{M}^{qT} \mathbf{V}_j^T \mathbf{V}_j \cdot \mathbf{S}_j .$$

3. **Calculate Residuals.** $\mathbf{N}^q = \mathbf{S} - \mathbf{M}^q \cdot \mathbf{P}^q$.

4. **Modify Data.**

$$\begin{aligned}
\forall j : \quad \tilde{\mathbf{N}}_j^q &= \mathbf{V}_j \mathbf{N}_j^q \\
\tilde{\mathbf{S}}^q &= \mathbf{M}^q \mathbf{P}^q + \tilde{\mathbf{N}}^q .
\end{aligned}$$

5. **If Not Stop.** $q = q + 1$, goto 2. The stop criteria is

$$\|\mathbf{N}^q - \mathbf{N}^{q-1}\|_\infty < tolerance .$$

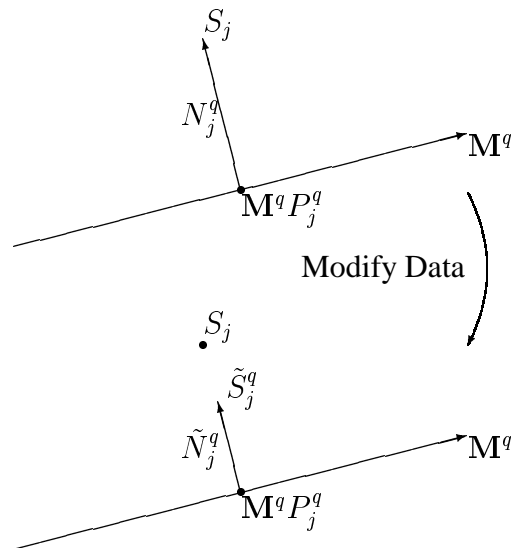


Figure 4: A geometric illustration of how the data is modified in steps 3. and 4. of the proposed algorithm for separation with weights.

As illustrated in Figure 4 the data, S , is modified such that the Frobenius norm of the modified residuals, \tilde{N}_j^q , are equal to norm of the original residuals, N_j^q , in the norm induced by the weights, V_j . The last part of step 2. ensures that the residual, N_j , is orthogonal to M in the induced norm, since $M^q P_j^q$ is the projection of S_j onto M^q in the induced norm.

4.1 Separation Performance with Given Weights

A quasi-Newton method could also be used to solve (9). One of the most effective quasi-Newton methods is the Broyden, Fletcher, Goldfarb & Shanno method (BFGS) [4, 10, 11, 23]. However this is not recommended, since the proposed method is faster and more reliable. Partly because with a 'standard' optimization method the problem is very likely to become ill-conditioned due to the potentially large differences in weights.

To illustrate this, the proposed method and the BFGS were tested against each other, see Table 1. The S matrix was formed by (3) with noise added from a compound Gaussian distribution. The compound distribution consisted of two Gaussian distributions, one with a standard deviation

10 times larger than the other. The fraction of the larger varying Gaussian is the *Noise Level*. It is seen, that the proposed method performs better than BFGS, and that the BFGS approach did not converge for $S = 40 \times 40$ and *Noise Level*=0.5, due to ill-conditioning.

S	Noise	Proposed	BFGS	Flop
$k \times n$	Level	Method		Ratio
20x40	0.02	1.20e+07	2.32e+08	19.33
20x40	0.10	1.58e+07	5.81e+08	36.73
20x40	0.50	5.50e+07	4.22e+08	7.67
40x40	0.02	7.20e+07	1.99e+09	27.58
40x40	0.10	1.15e+08	3.64e+09	31.73
40x40	0.50	3.59e+08	–	–
80x40	0.02	5.17e+08	1.78e+10	34.41
80x40	0.10	8.00e+08	7.08e+10	88.52
80x40	0.50	2.30e+09	8.74e+10	37.93

Table 1: Computational time comparison of the proposed algorithm with MatLab’s BFGS (fminu()), – denotes that the optimization did not converge due to ill-conditioning. The computational time is measured in Flops.

A formal proof of convergence seems to be infeasible. However this is a common problem for most numerical optimization schemes. The convergence has been followed closely during the tests of the algorithm. These empirical results show that the convergence has been very good.

A thorough investigation of the numerical properties of the algorithm is presented in [21]. Here the proposed method for separation is also compared to other alternatives besides BFGS with favorable results. Among these are the types of algorithms where M is estimated given P and vice versa in an iterative manner, like the one presented in [24].

5 The Object Frame Origin

In order to achieve robust factorization, the calculation scheme of the object frame origin, (x_{o_i}, y_{o_i}) applied in the Christy–Horaud algorithm needed to be improved. The object frame origin is the point around which the camera model is linearized, and errors here will propagate to the rest of the estimation. Making the estimation origin robust is essential to deal efficiently with erroneous data. In the Christy–Horaud algorithm, the 2D features corresponding to an arbitrary 3D feature is chosen. This scheme is error prone with erroneously tracked features, e.g. if the chosen feature corresponds to a false feature match.

A natural alternative, would be to iteratively calculate the center of mass from the estimated structure and reproject it onto the estimated camera positions. This however turns out to be *unstable*.

We still choose to use the center of mass, but calculated from the 2D features and the reprojected 3D features in a similar manner as (7). Combining 2D features and the reprojected 3D features allows us to diminish the influence of the 2D features with high uncertainty, while stabilizing the estimation with the reliable 2D features. The formula for the object frame origin of frame i is given by:

$$\begin{bmatrix} x_{o_i} \\ y_{o_i} \end{bmatrix} = \frac{\sum_{j=1}^n \begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} (1 + \epsilon_{ij})}{n + \sum_{j=1}^n \epsilon_{ij}}, \quad (14)$$

where ϵ_{ij} is defined in (8) and $(\tilde{x}_{ij}, \tilde{y}_{ij})$ is a weighted mean between the 2D feature and the reprojected 3D feature defined by.

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \gamma_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + (1 - \gamma_{ij}) \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \end{bmatrix}. \quad (15)$$

Here γ_{ij} denotes the weight. This weighted mean ensures robustness towards outliers, since

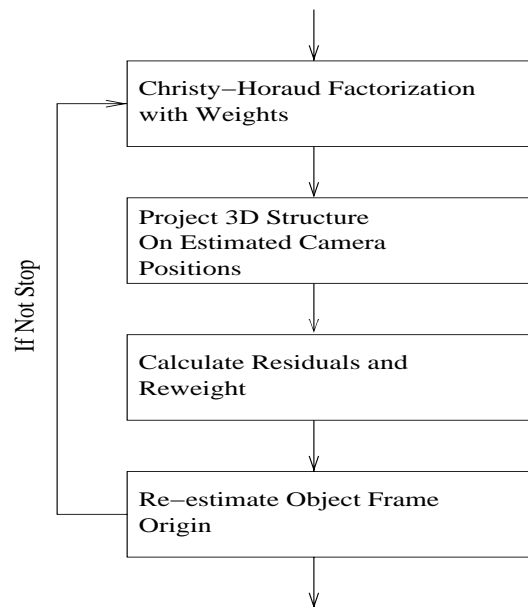


Figure 5: Overview of the proposed approach. This approach also deals with estimating the object frame origin robustly.

outliers are partly replaced by the estimate of the model. If $\mathbf{V}_j^T \mathbf{V}_j$ is *diagonal* then the weight γ_{ij} is given by:

$$\gamma_{ij} = \frac{1}{\pi} \arctan\left(\alpha\left(v_{ij} - \frac{1}{k'}\right)\right) + \frac{1}{2}, \quad (16)$$

where v_{ij} is the i^{th} diagonal element of \mathbf{V}_j . The 'cutoff' value k' serves the same function as k in the truncated quadratic and the Huber's M-estimator. It is seen that (16) is a smooth version of the step function around $\frac{1}{k'}$ where α controls how smooth this approximation is.

If $\mathbf{V}_j^T \mathbf{V}_j$ is *not diagonal* then it's principal components are used. So instead of v_{ij} the eigenvalues of $\mathbf{V}_j^T \mathbf{V}_j$ and a linear combination of (x_{ij}, y_{ij}) , are used in (14) and (15). This linear combination corresponds to the eigenvectors of $\mathbf{V}_j^T \mathbf{V}_j$. With this modification to the approach, the final flow chart is see in Figure 5.

6 Euclidean Reconstruction

The objective of Euclidean reconstruction is to estimate the \mathbf{A} in (6) such that the estimated $\mathbf{a}_i, \mathbf{b}_i$ and \mathbf{c}_i of (1) are as orthonormal as possible. In the paraperspective case [22], which is the linearization used by Christy and Horaud [6], the M_i 's composing \mathbf{M} are given by:

$$\mathbf{M}_i = \frac{1}{t_i^z} \begin{bmatrix} \mathbf{a}_i^T - x_{oi} \mathbf{c}_i \\ \mathbf{b}_i^T - y_{oi} \mathbf{c}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I}_i^T \\ \mathbf{J}_i^T \end{bmatrix},$$

where (x_{oi}, y_{oi}) is the object origin projected in frame i .

Since the paraperspective approximation is obtained by linearizing $\frac{1}{t_i^z} \mathbf{c}_i^T \cdot \mathbf{P}_j$ the orthonormal constraints are restricted to \mathbf{a}_i and \mathbf{b}_i . With $\mathbf{Q} = \mathbf{A}\mathbf{A}^T$ these constraints can be formulated as [6, 22]:

$$\begin{aligned} \forall i \quad \mathbf{a}_i^T \mathbf{Q} \mathbf{a}_i &= \mathbf{b}_i^T \mathbf{Q} \mathbf{b}_i \Rightarrow \\ \forall i \quad \frac{\mathbf{I}_i^T \mathbf{Q} \mathbf{I}_i}{1 + x_{oi}^2} - \frac{\mathbf{J}_i^T \mathbf{Q} \mathbf{J}_i}{1 + y_{oi}^2} &= 0 \\ \forall i \quad \mathbf{a}_i^T \mathbf{Q} \mathbf{b}_i &= 0 \Rightarrow \\ \forall i \quad \mathbf{I}_i^T \mathbf{Q} \mathbf{J}_i - \frac{x_{oi} y_{oi} (\mathbf{I}_i^T \mathbf{Q} \mathbf{I}_i)}{2(1 + x_{oi}^2)} - \frac{x_{oi} y_{oi} (\mathbf{J}_i^T \mathbf{Q} \mathbf{J}_i)}{2(1 + y_{oi}^2)} &= 0 \end{aligned}$$

With noise, this cannot be achieved for all i and a least squares solution is sought. In order to avoid the trivial null-solution the constraint $\mathbf{a}_1^T \mathbf{Q} \mathbf{a}_1 = \mathbf{b}_1^T \mathbf{Q} \mathbf{b}_1 = 1$ is added [6, 22] and the problem is linear in the elements of \mathbf{Q} .

Unfortunately it is impossible to reconstruct \mathbf{A} if \mathbf{Q} has negative eigenvalues. This problem indicates that an unmodeled distortion has overwhelmed the third singular value of \mathbf{S} [22]. This is a fundamental problem when the factorization method is used on erroneous data.

To solve this problem we propose to parameterize, \mathbf{Q} as:

$$\mathbf{Q}(\mathbf{e}, \lambda) = \mathbf{R}(\mathbf{e}) \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \mathbf{R}(\mathbf{e})^T, \quad (17)$$

where $\mathbf{R}(\mathbf{e})$ is a rotation matrix with the three Euler angles denoted by \mathbf{e} . The term $\mathbf{a}_1^T \mathbf{Q} \mathbf{a}_1 = \mathbf{b}_1^T \mathbf{Q} \mathbf{b}_1 = 1$ is replaced by $\det(\mathbf{A}) = 1$, such that the overall scale of \mathbf{A} is much more robust and less sensitive to the noise in a particular frame.

Hence the estimation of \mathbf{Q} is a nonlinear optimization problem in six variables, with a guaranteed symmetric positive definite \mathbf{Q} . Our experience shows that this approach to the problem is well behaved with a quasi-Newton optimization method.

7 Experimental Results

In order to illustrate the proposed algorithms ability to deal with the errors identified, it has been applied to a set of real images. To provide a more systematic test, it was then applied to a set of simulated data, where arbitrary errors could be induced.

7.1 Real Data

The proposed approach was run on three sequences, demonstrating different properties. The first sequence was the hotel sequence [19] with accompanying features, see Figure 6. Some of these features were mismatched. To illustrate the handling of mismatched features, notice the different position of feature 64 in Figures 7 and 8. The lines in the images denotes the residual between the tracked 2D features and the back projected 3D estimates. The correct position of a feature is determined by the location with the greatest support.



Figure 6: A sample frame from the Hotel sequence with 197 tracked features, some of these are mismatched.

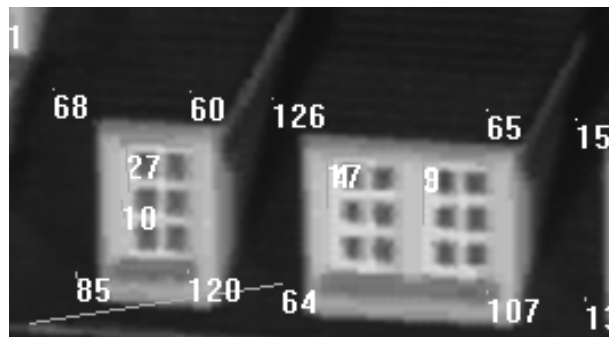


Figure 7: A section of the Hotel sequence illustrating where feature 64 is mismatched. The correct position of feature 64 is at the end of the residual vector (bottom left of the image).

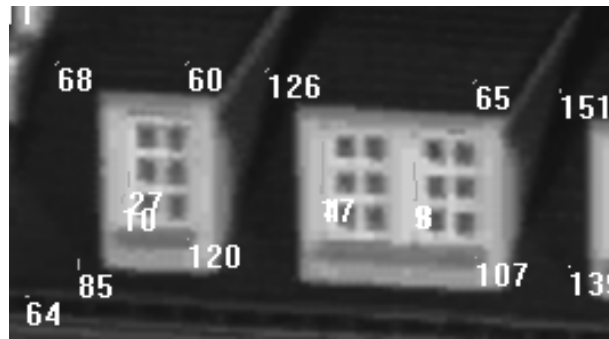


Figure 8: A section of the Hotel sequence where feature 64 is not erroneous. Notice how little effect the false match error has, with the proposed method.

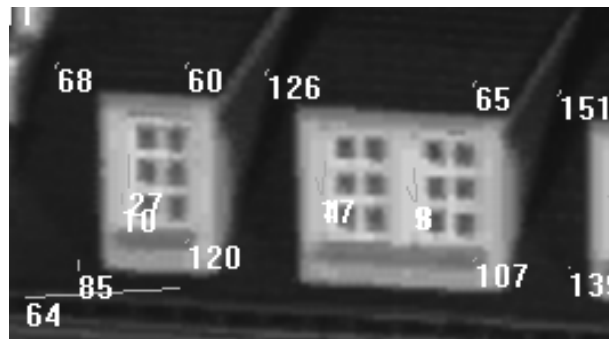


Figure 9: Same section and frame as Figure 8, but with the Christy-Horand method. Notice the increased effect of the errors.

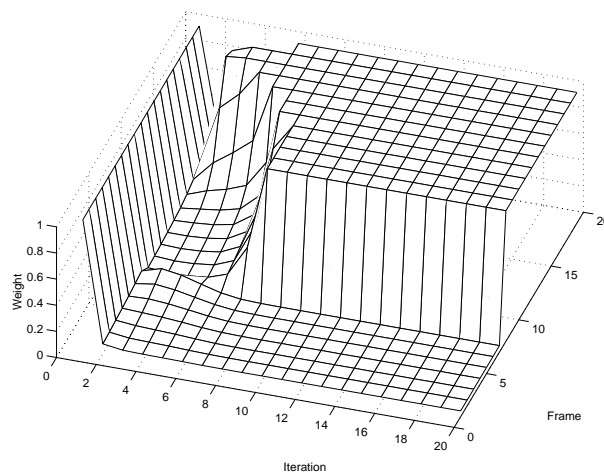


Figure 10: Weight evolution for feature 64 plotted as a function of frame and iteration.

The proposed approach with the truncated quadratic error function was applied to the hotel sequence ($k = 3, k' = 2, \alpha = 20$). The result is shown in Figures 7 and 8. Two things should be noted here. First, that the mismatched feature 64 does not effect the estimate of the other features. Second, that the back projections of the 3D estimate of feature 64 are located correctly in all images (including 7 where the 2D feature itself is mismatched).

Comparing with the Christy–Horaud approach it is seen that the mismatch error in Figure 7 effects the overall estimation of feature 64’s 3D position. To give a more quantitative evaluation, the residuals of the non erroneous features were summed up – it is assumed that there were no more than 5% errors. It is seen from Table 2 that due to the capability to implement a robust error function, the fit to the non–erroneous data is improved significantly.

Error Function:	$\frac{1}{n} \sum_i Res _i$
Truncated Qudratic (robust)	1.94 <i>pixels</i> ²
2–Norm (non–robust)	3.40 <i>pixels</i> ²

Table 2: Comparison of the 95% smallest residuals. It is noted, that without sub–pixel feature location this number is highly unlikely to be lower than 1.

To illustrate the reweighting process, the evolution of the weights off the mismatched feature 64 is depicted in Figure 10. After the first iteration all the 2D features are down weighted, since only an erroneous 3D estimate exists due to the mismatch features and the uniform weighting. In the following iterations the correctly located features obtains increasing weights whereas the weights of the mismatch feature decrease toward zero. The plot directly shows how the proposed method can be used to detect mismatched features.

To demonstrate the proposed approach’s ability to deal with missing features, it was run on the kitchen sequence [19], see Figure 11. Here some of the accompanying features were missing, e.g. feature 5 in figure 13. Note that the back projection of feature 5 in figure 13 is located



Figure 11: A sample image from the kitchen sequence.

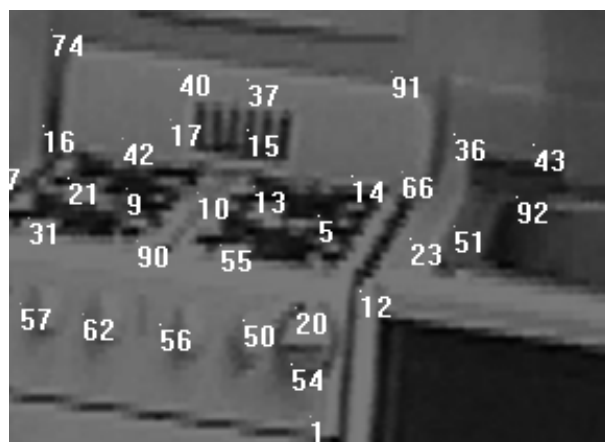


Figure 12: Close up on the kitchen sequence. Notice the location of feature 5.

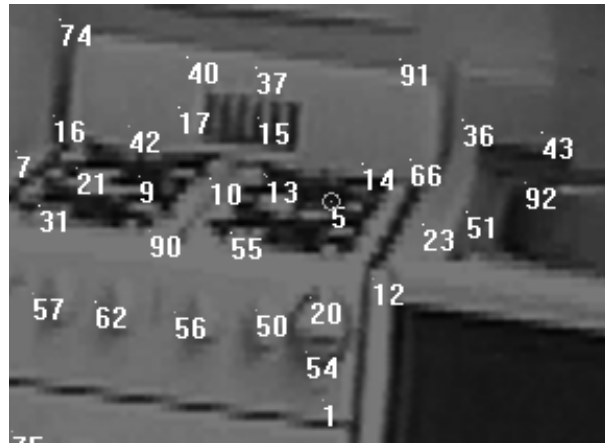


Figure 13: Closeup on the kitchen sequence. A circle around a feature denotes it is the estimated location. The reason being that the respective feature is missing in this frame. Not that the estimated position of feature 5 correspond to it's position in figure 12.



Figure 14: A sample frame from an image sequence of Thorvaldsens Museum in Copenhagen with 20 hand tracked features through 8 frames.

correctly, indicating that the 3D estimate is correct. From the depicted residuals, which are hardly visible, it is seen that the missing features does not disrupt the structure and motion estimation. Thus giving the desired result in dealing with missing features.

The proposed modification to the Christy–Horaud algorithm should not considerably decrease it's ability to deal with perspective reconstruction. To validate this, both approaches were applied to a sequence with considerable depth, see Figure 14. The 2D features were carefully hand–tracked to ensure that the algorithms were directly comparable. The results were evaluated by comparing the two estimated structures via the standard shape distance measure from

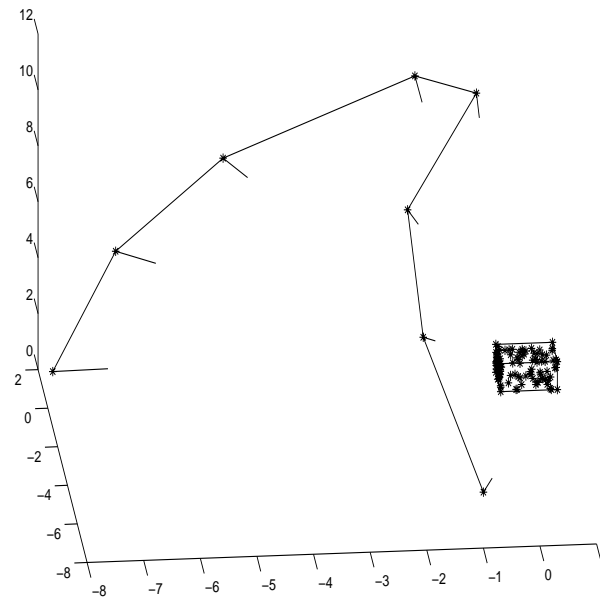


Figure 15: The setup for the simulated data. A box with 100 features is 'photographed' from 8 views. These views are marked on the trajectory curve.

statistical shape analysis, the Procrustes distance [9, 12]. The Procrustes distance is obtained by normalizing the two structures and the applying the similarity transform [14] such that the mean squared error is minimized. The remaining mean squared error is then the Procrustes distance. The resulting Procrustes distance was 0.03. This implies that the Christy–Horaud method with the proposed enhancement maintains it's ability to deal with perspective data.

7.2 Simulated Data

To perform a more systematic test of the proposed algorithms ability to deal with errors in the tracked data a simulated data set was created, see Figure 15. Several kinds of errors were introduced into this data set, hereby testing the approach with respect to the three identified types of errors. The Hubers M–estimator and truncated quadratic error functions were applied

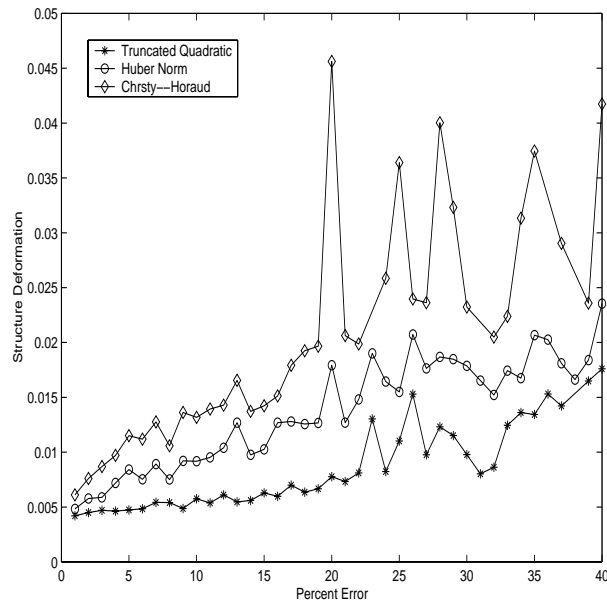


Figure 16: Errors for the simulated data corrupted with a compound distribution of two Gaussians. The abscissa denotes the likelihood, $[0 \dots 1]$, that samples are obtained from the higher varying Gaussian.

with the parameters settings $k = 0.02$ ¹ and $k' = 2$ and we set $\alpha = 20$. The results were evaluated by the Procrustes distance [9, 12]. In the extreme cases where an estimation process did not converge, this is denoted by a missing measurement.

The first experiment consists of corrupting the 2D features of the simulated data by a compound distribution of two Gaussians one with a standard deviation of 0.005 and the other 10 times larger. Two different error functions were applied; the Huber M-estimator and the truncated quadratic. For comparison the Christy–Horand algorithm was also applied. From Figure 16 it is seen, that the choice of error functions has a considerable effect on the result, and that the proposed approach is capable of implementing them.

In the second experiment, an increasing number of the 2D features were removed from the data

¹It should be noted, that k depends on the image noise and hence also on the image size. In this simulated data, the image size was unnaturally small (0.25×0.25), and as such $k = 0.02$ should not be seen as a guide line.

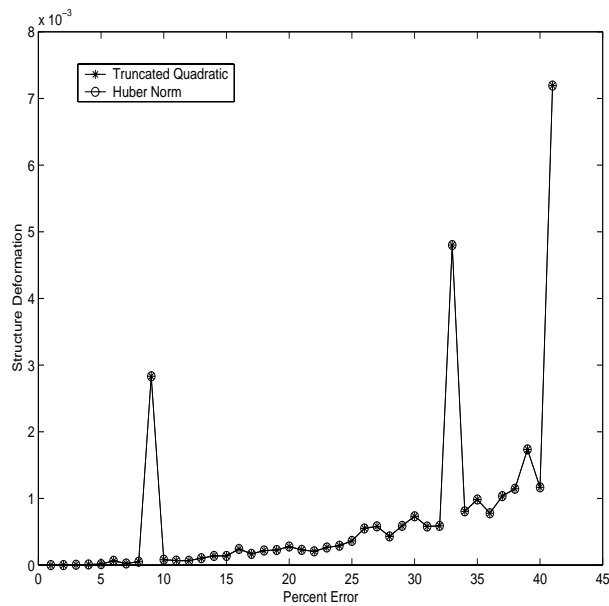


Figure 17: Errors for corrupting the simulated data by removing the 2D features at random. The abscissa indicates how many percent of the data has been removed. No difference is seen between the two error functions.

set. These missing features were emulated as being located in the center of the frame in question with a weight equal to 10^{-6} of the weight of the normal 2D features. From the results, it is seen that the proposed approach is highly robust towards this type of error, since up to 40% missing features corrupts the estimated structure by less than 10^{-2} , see Figure 17. The approach of Christy and Horaud could not work on this data and was not included.

To test the tolerance to mismatched features, we emulated these by swapping 2D features of the simulated data. The results in Figure 18 illustrate that the proposed approach also deals efficiently with mismatched features and has been shown to be robust toward the identified types of errors.

To challenge the proposed approach all the experiments on the simulated data are performed with up to 35–40% errors. It should be noted, that the algorithm works very well with up to 10–20% errors. This is the amount of errors that the proposed approach is expected to work on.

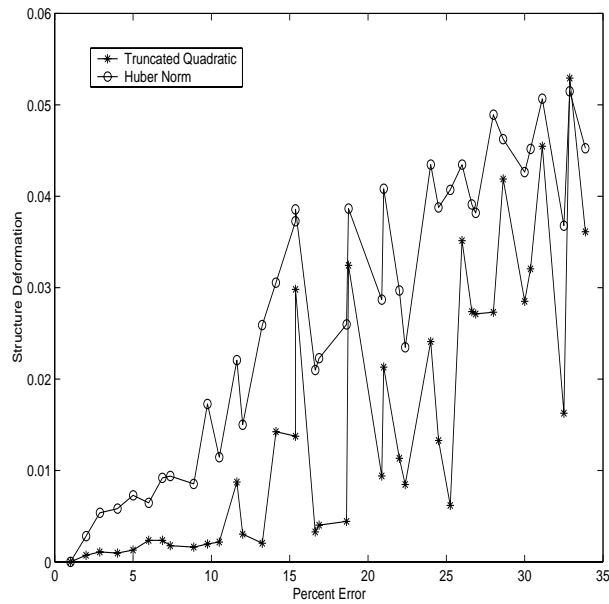


Figure 18: Errors for the simulated data with swapped features emulating mismatched features. The data set upon which features were swapped was the original uncorrupted. The abscissa denoted the percentage of altered features.

To evaluate the effect of the proposed approach for estimating the object frame origin noise was added exclusively to the feature that Christy–Horaud uses as object frame origin. The evaluation were performed on the simulated data and the noise was Gaussian with a standard deviation of 0.05. Five experiments were conducted comparing the Christy–Horaud approach to the proposed approach. The results are shown in Table 3. It is seen that significant improvement is obtained with respect to error as well as convergence. For validation purposes the same experiment was made, but this time noise was added to a different feature then the one used as object origin by Christy-Horaud. In this case the two approaches gave similar results. Note also, that the original approach for estimation of the object frame origin could be problematic to use with missing features, since the 2D feature chosen as object frame origin is likely to be missing in at least one frame.

In order to demonstrate the benefits of the proposed method for Euclidean reconstruction, the

Factorization Approach	Percent Convergence	Mean Procrustes Distance
Christy–Heraud	60%	0.0153
Proposed Approach	100%	0.0055

Table 3: Results with noise exclusively on the feature Christy–Heraud uses as object frame origin. This amounts to 1% of the data being erroneous.

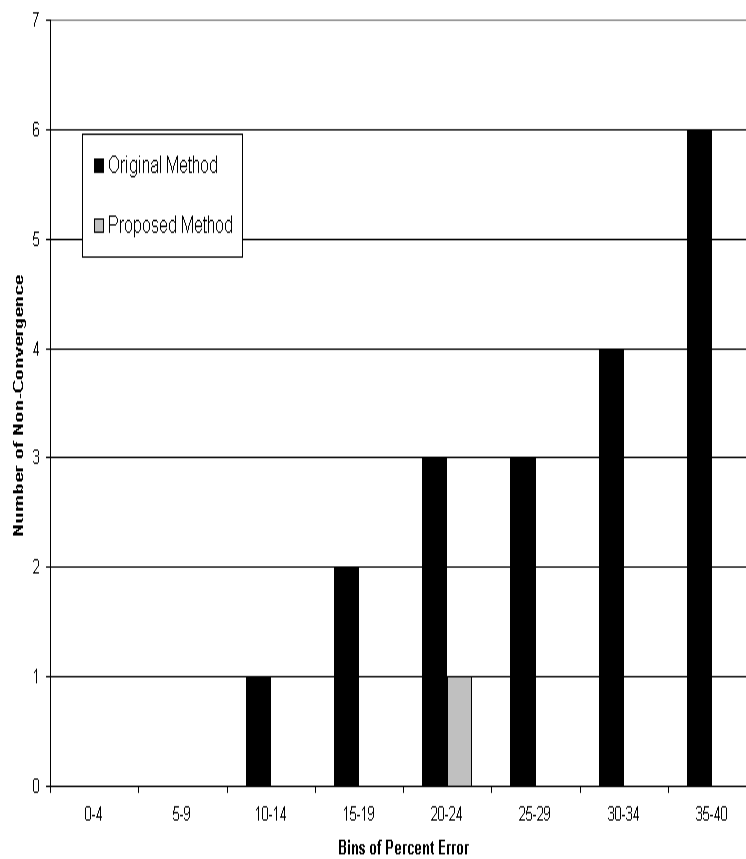


Figure 19: The number of non-converging runs on the experiment of Figure 18, using the Huber M-estimator as error function. The experiment was made with and without the proposed method for Euclidean reconstruction. The runs are pooled in bins of 5 to give a better overview.

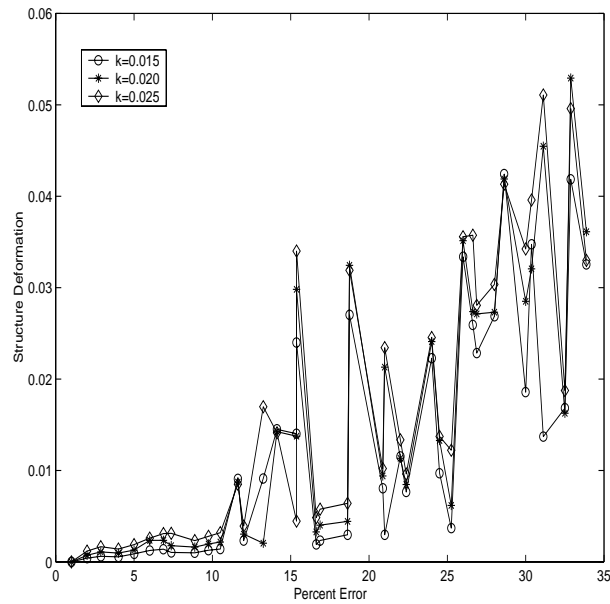


Figure 20: The experiment of Figure 18 repeated with different parameter settings for k in the truncated quadratic error function.

experiments with mismatched features was repeated *without* the proposed approach. Instead the original method for Euclidean reconstruction proposed in [6, 22] was applied. The number of runs that did not converge have been summed up with and without the proposed method. The change is dramatic.

To illustrate, that the proposed method is not overly sensitive to the choice of error function and the involved parameters, these were varied. This was done with the truncated quadratic error function in the experiments with mismatched features, see Figure 18. The k parameter was altered with $\pm 50\%$. From the results in Figure 20 it is seen that the proposed approach is not overly sensitive to the choice of parameters in the error function.

8 Conclusion and Discussion

An approach for applying arbitrary error functions in the factorization algorithms for structure and motion has been presented. This ability has been exploited to implement robust statistical techniques, whereby errors in the 2D features have been dealt with effectively. The algorithm has been implemented as an extension to the Christy–Horaud factorization algorithm [6], whereby perspective reconstruction is achieved. The core of the approach is however so general, that it can also be applied to most other factorization schemes like [1, 15, 26, 27, 22, 28]. This approach has been applied to both simulated and real data, where it is demonstrated that the algorithm deals well with all the expected types of errors. Thus making the proposed approach a robust factorization method.

To further investigate the possibilities of the proposed approach, we aim at implementing it with other factorization algorithms, e.g. [15], hence making a robust factorization algorithm with uncalibrated cameras. The proposed method also allows for more elaborate noise structures incorporating covariance within and between images. This covariance structure has been implemented with the algorithm of Section 4 and sparsely tested, but should be thoroughly tested on real image data, in order to evaluate its robustness. This is the goal of ongoing research.

9 Acknowledgements

We would like to thank the Modeling by Videotaping group at the Robotics Institute, Carnegie Mellon University, for supplying the hotel and kitchen data sequences, and the accompanying tracked features [19]. We would also like to thank Radu Horaud and Claus Gramkow, for valuable comments to this work. And lastly Hans Bruun Nielsen for comments on the numerical aspects.

References

- [1] R. Berthilsson and K. Åström. Extension of affine shape. *Journal of Mathematical Imaging and Vision*, 11(2):119–136, 1999.
- [2] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.
- [3] A.J. Booker, J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, 1999.
- [4] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. ii. the new algorithm. *Journal of the Institute of Mathematics and Its Applications*, 6(3):222–31, 1970.
- [5] S. Christy and R. Horaud. Euclidian shape and motion from multiple perspective views by affine iterations. Technical Report 2421, INRIA, December 1994.
- [6] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iteration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, November 1996.
- [7] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *Int'l J. Computer Vision'98*, 29(3):159–179, 1998.
- [8] D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *Int'l J. Computer Vision'94*, 15(1–2):123–141, 1995.
- [9] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, 1998.

- [10] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13(3):317–22, 1970.
- [11] D. Goldfarb. A family of variable metric methods derived by variational means. *Maths. Comp.*, 24:23–26, 1970.
- [12] J.C. Gower. Generalized Procrustes analysis. *Psychometrika*, 40:33–50, 1975.
- [13] F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley & Sons, 1986.
- [14] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2000.
- [15] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13):981–991, 1999.
- [16] M. Irani and P. Anandan. Factorization with uncertainty. In *European Conf, Computer Vision'2000*, pages 539–553, 2000.
- [17] D.W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82(1):57–81, 2001.
- [18] T. Kanade and D. Morris. Factorization methods for structure from motion. *Phil. Trans. R. Soc. Lond.*, A(356):1153–1173, 1998.
- [19] The modeling by videotaping group at the robotics institute, carnegie mellon university. <http://www.ius.cs.cmu.edu/IUS/mbvc0/www/modeling.html>.

- [20] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Int'l Conf. Computer Vision'98*, pages 696–702, January 1998.
- [21] H. B. Nielsen and H. Aanæs. Separation of structure and motion by data modification. Technical report, IMM, DTU, to appear.
- [22] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997.
- [23] D.F. Shanno. Conditioning of quasi-newton methods for function minimization. *Maths. Comp.*, 24:647–656, 1970.
- [24] H.-Y. 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):854–67, 1995.
- [25] C.C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, 4:th edition, 1984.
- [26] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Int'l Conf. Pattern Recognition'96*, pages 328 – 333, 1996.
- [27] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf, Computer Vision'96*, volume 2, pages 709–720, 1996.
- [28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision'92*, 9(2):137–154, November 1992.

- [29] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int'l J. Computer Vision*, 24(3):271–300, 1997.
- [30] B Triggs, P F McLauchlan, R I Hartley, and A W Fitzgibbon. Special sessions - bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, 1883:298–372, 2000.
- [31] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI Jnl*, 78(1):87–119, 1995.