

# FACTORIZATION WITH ERRONEOUS DATA

Henrik Aanæs<sup>a</sup>, Rune Fisker<sup>a,c</sup>, Kalle Åström<sup>b</sup> and Jens Michael Carstensen<sup>a</sup>

<sup>a</sup> Technical University of Denmark

<sup>b</sup> Lund Institute of Technology

<sup>c</sup> 3Shape Inc.

**KEY WORDS:** Robust statistics, feature tracking, Euclidean reconstruction, structure from motion

## ABSTRACT

Factorization algorithms for recovering structure and motion from an image stream have many advantages, but traditionally requires a set of well tracked feature points. This limits the usability since, correctly tracked feature points are not available in general. There is thus a need to make factorization algorithms deal successfully with incorrectly tracked feature points.

We propose a new computationally efficient algorithm for applying an arbitrary error function in the factorization scheme, and thereby enable the use of robust statistical techniques and arbitrary noise models for individual feature points. These techniques and models effectively deal with feature point noise as well as feature mismatch and missing features. Furthermore, the algorithm includes a new method for Euclidean reconstruction that experimentally shows a significant improvement in convergence of the factorization algorithms.

The proposed algorithm has been implemented in the Christy–Horn factorization scheme and the results clearly illustrate a considerable increase in error tolerance.

## 1 INTRODUCTION

Structure and motion estimation of a rigid body from an image sequence, is one of the most widely studied fields within the field of computer vision. A popular set of solutions to the subproblem of estimating the structure and motion from tracked features are the so-called factorization algorithms. They were originally proposed by [Tomasi and Kanade, 1992], and have been developed considerably since their introduction, see e.g. [Christy and Horn, 1996, Costeira and Kanade, 1998, Irani and Anandan, 2000, Kanade and Morita, 1994, Morris and Kanade, 1998, Poelman and Kanade, 1997, Quan and Kanade, 1996, Sturm and Triggs, 1996].

These factorization algorithms work by linearizing the observation model, and give good results fast and without any 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. [Slama, 1984, Triggs et al., 2000].

The factorization algorithms assume that the correspondence or feature tracking problem has been solved. The correspondence problem is, however, one of the difficult fundamental problems within computer vision. No perfect and fully general solution has been presented. For most practical purposes one must abide with erroneous 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 erroneous data. One such badly tracked feature can corrupt the result considerably.

A popular way of addressing the sensitivity of the 2-norm to outliers is by introducing weights on the data, such that

less reliable data is down-weighted. This is commonly referred to as weighted least squares. We here propose a method for doing this in the factorization framework. Hereby the sensitivity to outliers or erroneous data is reduced. In other words we allow for an arbitrary Gaussian noise model on the 2D features, facilitating correlation between the 2D features, directional noise on the individual 2D features in each frame and an arbitrary variance. In this paper we focus on different sizes of the variance on the individual 2D features, in that this in itself can address most of the issues of concern.

In order to down-weight less reliable data these have to be identified. A popular way to do this is by assuming that data with residual over a given threshold are less reliable. This assumption is the basis of most robust statistics, and is typically implemented via Iterative Reweighted Least Squares (IRLS). IRLS allows for arbitrary weighting functions. We demonstrate this by implementing the Huber M-estimator [Huber, 1981] and the truncated quadratic [Black and Rangarajan, 1996].

The proposed approach applies robust statistical methods in conjunction with a factorization algorithm to obtain better result with erroneous data.

There has been other attempts to address the problem of different noise structures in the factorization framework [Irani and Anandan, 2000, Morris and Kanade, 1998]. Irani and Anandan [Irani and Anandan, 2000] 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. As an example, a mismatched feature in one frame does in general not mean that the same feature mismatch occurs

in other frames. For missing features, the noise model of [Irani and Anandan, 2000] is inadequate, as will be seen later. [Morris and Kanade, 1998] proposes 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. It is noted, that [Jacobs, 2001] have proposed a heuristic method for dealing with missing data.

We have chosen to implement the proposed method in conjunction with the factorization algorithm by Christy and Horaud [Christy and Horaud, 1996]. This factorization assumes perspective cameras as opposed to linearized approximations hereof. This yields very satisfactory results, as illustrated in Section 5. In order to solve a practical problem we propose a new method for Euclidean reconstruction in Section 4, as opposed to the one in [Poelman and Kanade, 1997].

## 2 FACTORIZATION OVERVIEW

This is a short overview of factorization algorithm. For a more detailed introduction the reader is referred to [Christy and Horaud, 1994, Christy and Horaud, 1996]. The factorization methods cited all utilize some linearization of the pinhole camera with known intrinsic parameters:

$$\begin{bmatrix} sx_{ij} \\ sy_{ij} \\ s \end{bmatrix} = \begin{bmatrix} \mathbf{a}_i^T & t_i^x \\ \mathbf{b}_i^T & t_i^y \\ \mathbf{c}_i^T & t_i^z \end{bmatrix} \begin{bmatrix} P_j \\ 1 \end{bmatrix} \quad (1)$$

where the 3D feature,  $P_j$ , is projected in frame  $i$  as  $(x_{ij}, y_{ij})$ ,  $t_i$  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:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = M_i P_j \quad (2)$$

where  $M_i$  is the  $2 \times 3$  '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  $M_i$ ,  $\mathbf{P}$  is a  $3 \times n$  matrix composed of the  $P_j$  such that:

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

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

$$\mathbf{N} = \mathbf{S} - \mathbf{M}\mathbf{P} \quad (4)$$

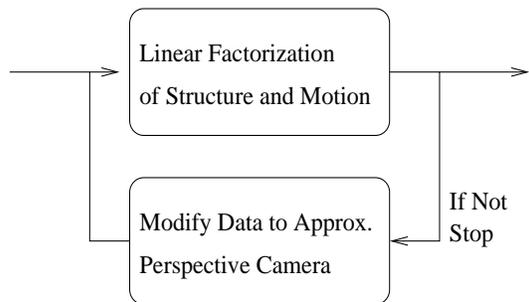


Figure 1: Overview of the Christy–Horaud algorithm.

where  $\mathbf{N}$  is the residual between model,  $\mathbf{M}\mathbf{P}$ , and the data,  $\mathbf{S}$ . The residuals,  $\mathbf{N}$ , are usually minimized in 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 (4) becomes:

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

where  $S_j$  and  $P_j$  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 via the singular value decomposition, SVD, of  $\mathbf{S}$ .

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

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

Hence the solution is only defined up to an affine transformation. In [Christy and Horaud, 1996], a Euclidean reconstruction is achieved by estimation of an  $\mathbf{A}$ , such that the rotation matrices,  $[a_i \ b_i \ c_i]^T$ , are as orthonormal as possible. Further details are given in Section 4.

### 2.1 Christy–Horaud Factorization

The approach we improve on by introducing arbitrary noise models is the approach of Christy and Horaud [Christy and Horaud, 1996]. This approach iteratively achieves a solution to the original non-linearized version of the pinhole camera. These iterations consist of modifying the observations  $x_{ij}, y_{ij}$ , and hence  $\mathbf{S}$ , as if they were observed by an *imaginary* linearized camera, which in turn requires an estimate of the structure and motion, see Figure 1. The update formulae is given by:

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \left( \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} - \begin{bmatrix} x_{o_{ij}} \\ y_{o_{ij}} \end{bmatrix} \right) \left( 1 - \frac{\mathbf{c}_i \cdot P_j}{t_i^z} \right) \quad (7)$$

where  $(\tilde{x}_{ij}, \tilde{y}_{ij})$  is the updated data and  $(x_{o_{ij}}, y_{o_{ij}})$  is the object frame origin.

## 3 SEPARATION WITH WEIGHTS

In order to deal with erroneous data, (5) should be computed with weights:

$$\min_{\mathbf{M}, \mathbf{P}} \sum_{j=1}^n \|V_j(S_j - \mathbf{M}P_j)\|_2^2 \quad (8)$$

where  $V_j$  is an  $2k \times 2k$  weighting matrix representing the weights of the  $j^{th}$  column of  $\mathbf{S}$ . In the case of Gaussian noise  $V_j^T V_j$  is the covariance structure of  $S_j$  and (8) is equivalent to minimizing the Mahalanobis distance.

### 3.1 Separation with Weights

The solution to (8) 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 (8), a method similar to the idea in the Christy-Horand factorization algorithm [Christy and Horand, 1996] is proposed. This method is generally known as surrogate modeling, see e.g. [Booker et al., 1999]. 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 (8) 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 (8) with the original data. By letting  $\tilde{\cdot}$  denoting modified data, the goal is to obtain:

$$\begin{aligned} \min_{M,P} \sum_{j=1}^n \|\mathbf{V}_j(\mathbf{S}_j - \mathbf{M}\mathbf{P}_j)\|_2^2 &= & (9) \\ \min_{M,P} \sum_{j=1}^n \mathbf{N}_j^T \mathbf{V}_j^T \mathbf{V}_j \mathbf{N}_j &\stackrel{def}{=} \\ \min_{M,P} \sum_{j=1}^n \tilde{\mathbf{N}}_j^T \tilde{\mathbf{N}}_j &= \\ \min_{M,P} \sum_{j=1}^n \|\tilde{\mathbf{S}}_j - \mathbf{M}\mathbf{P}_j\|_2^2 \end{aligned}$$

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

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

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

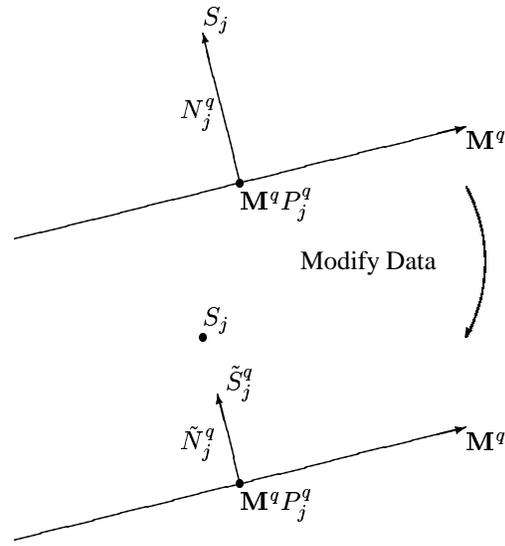


Figure 2: A geometric illustration of how the data is modified in steps 3. and 4. of the proposed algorithm for separation with weights.

#### 4. Modify Data

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

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

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

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

The reason this approach is used, and not a quasi-Newton method, e.g. BFGS [Fletcher, 1987] on (8), is that faster and more reliable results are obtained. In part because that with 'standard' optimization methods the problem is very likely to become ill-conditioned due to the potentially large differences in weights.

To illustrate this, some test runs were made, comparing the computation time needed to solve some 'typical' problems, see Table 1. The  $\mathbf{S}$  matrix was formed by (3) where to noise was added from a compound Gaussian distribution. The compound distribution was formed by two Gaussian distributions, one with a standard deviation 10 times larger than the other. The frequency of the larger varying Gaussian is the *Noise Ratio*. It is seen, that the proposed method performs better than BFGS, and that the BFGS approach did not converge for  $\mathbf{S} = 40 \times 40$  and *Noise Ratio*=0.5.

### 3.2 Arbitrary Error Functions

When dealing with erroneous data, robust statistical norms or error functions become interesting, see e.g. [Black and

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

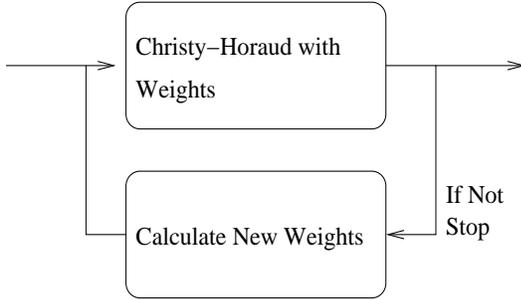


Figure 3: Overview of the proposed approach for arbitrary error functions.

Rangarajan, 1996]. This is achieved in the presented setup via Iterative Reweighted Least Squares (IRLS). Where IRLS works by iteratively solving the ‘weighted’ least squares problem and then adjusting the weights, such that it corresponds to the preferred error function, see Figure 3. A typical robust error function is the truncated quadratic:

$$w_{ij} = \begin{cases} 1 & \|N_{ij}\|_{\Sigma_{ij}} < k \\ \sqrt{\frac{k^2}{N_{ij}^2}} & \|N_{ij}\|_{\Sigma_{ij}} > k \end{cases} \quad (10)$$

where  $N_{ij}$  is the residual on datum  $ij$ ,  $w_{ij}$  is the corresponding weight and  $k$  is a user defined constant relating to the image noise. If an a priori Gaussian noise structure,  $\Sigma_{ij}$ , is known for the 2D features, the size of the residuals  $N_{ij}$  is evaluated in the induced Mahalanobis distance, otherwise the 2-norm is used. In the case of a priori known Gaussian noise,  $\Sigma_j$ , it is combined with the truncated quadratic by  $V_j^T V_j = w_j \Sigma_j^{-1} w_j$ , otherwise  $V_j^T V_j = w_j w_j$ .

#### 4 EUCLIDEAN RECONSTRUCTION

The objective of Euclidean reconstruction is to estimate the  $\mathbf{A}$  in (6), such that the  $a_i, b_i$  and  $c_i$  of (1) are as orthonormal as possible. In the paraperspective case [Poelman and Kanade, 1997], which is the linearization used in Christy and Horand [Christy and Horand, 1996], the  $M_i$ ’s composing  $\mathbf{M}$  are given by:

$$M_i = \frac{1}{t_i^z} \begin{bmatrix} a_i^T - x_{oi} c_i \\ b_i^T - y_{oi} c_i \end{bmatrix} = \begin{bmatrix} I_i^T \\ J_i^T \end{bmatrix}$$



Figure 4: A sample frame from the Eremitage sequence.



Figure 5: A sample frame from the Court sequence. The test set was generated by hand tracking 20 features in this sequence of 8 images.

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

Since the paraperspective approximation is obtained by linearizing  $\frac{1}{t_i^z} c_i^T \cdot P_j$  the orthonormal constraints are restricted to  $a_i$  and  $b_i$ . With  $\mathbf{Q} = \mathbf{A}\mathbf{A}^T$  these constraints can be formulated as [Christy and Horand, 1996, Poelman and Kanade, 1997]:

$$\begin{aligned} \forall i \quad a_i^T \mathbf{Q} a_i &= b_i^T \mathbf{Q} b_i \Rightarrow \\ \forall i \quad \frac{I_i^T \mathbf{Q} I_i}{1 + x_{oi}^2} - \frac{J_i^T \mathbf{Q} J_i}{1 + y_{oi}^2} &= 0 \\ \forall i \quad a_i^T \mathbf{Q} b_i &= 0 \Rightarrow \\ \forall i \quad I_i^T \mathbf{Q} J_i - \frac{x_{oi} y_{oi} (I_i^T \mathbf{Q} I_i)}{2(1 + x_{oi}^2)} - \frac{x_{oi} y_{oi} (J_i^T \mathbf{Q} J_i)}{2(1 + y_{oi}^2)} &= 0 \end{aligned}$$

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  $a_1^T \mathbf{Q} a_1 = b_1^T \mathbf{Q} b_1 = 1$  is added [Christy and Horand, 1996, Poelman and Kanade, 1997] and the problem is linear in the elements of  $\mathbf{Q}$ .

This approach has the disadvantage, that if  $\mathbf{Q}$  has negative eigenvalues, it is not possible to reconstruct  $\mathbf{A}$ . This problem indicates that an unmodeled distortion has overwhelmed the third singular value of  $\mathbf{S}$  [Poelman and Kanade, 1997]. This is a fundamental problem when the factorization method is used on erroneous data.



Figure 6: Section of the Eremitage sequence showing the tracked features.



Figure 7: Section of the Eremitage sequence showing the tracked features. It is the frame following the frame shown in Figure 6. Note the change in feature locations.

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

$$\mathbf{Q}(\mathbf{e}, \lambda) = \mathbf{R}(\mathbf{e}) \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \mathbf{R}(\mathbf{e})^T \quad (11)$$

where  $\mathbf{R}(\mathbf{e})$  is a rotation matrix with the three Euler angles denoted by  $\mathbf{e}$ . The term  $a_1^T \mathbf{Q} a_1 = b_1^T \mathbf{Q} b_1 = 1$  is replaced by  $\det(\mathbf{A}) = 1$ , such that the over all 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.

## 5 EXPERIMENTAL RESULTS

We illustrate the capabilities of the proposed algorithm via three sets of experiments. The first demonstrate the capability of using different error functions. This is followed by a more systematic test of the tolerance for different kinds of possible errors. Finally we show an example of why the proposed method for Euclidian reconstruction is to be preferred.

### 5.1 Different Error Functions

To demonstrate the capability of using different error functions, we used an image sequence of the Eremitage castle, see Figure 4. The 2D features were extracted via the Harris corner-detector [Harris and Stephens, 1988], whereupon the epipolar geometry was used for regularization via RANSAC/MLESAC [Torr, 2000] followed by a non-linear optimization [Hartley and Zisserman, 2000].



Figure 8: Section of the Eremitage sequence showing the tracked features and residuals from the roof using the truncated quadratic with the proposed method. The residuals are denoted by the dark lines.

This enforcement of the epipolar geometry enhanced the quality of the data, but it did not yield perfectly tracked data. There are two main reasons for this.

First, the trees around the castle yield a myriad of potential matches since the branches look pretty much alike. The restriction of correspondances to the epipolar line is not sufficient to amend the situation as is shown in Figures 6 and 7.

Second, when filming a castle, one moves approximately in a plane – both feet on the ground. This plane is parallel to many of the repeating structures in the image, e.g. windows are usually located at the same horizontal level. Hence the epipolar lines are approximately located ‘along’ these repeating structures and errors here cannot be corrected by enforcing the epipolar geometry. In general the sequence contains plenty of missing features, mismatched features and noise.

The truncated quadratic [Black and Rangarajan, 1996], the Hubers M-estimator [Huber, 1981] and the 2-norm were tested as error functions. The reason the proposed method was used with the 2-norm and not the original Christy-Horaud method is, that there were missing features in the data-set. These missing features are incompatible with the Christy-Horaud approach, but the approach presented here deal with them by modeling them as located in the middle of the image with a weight  $10^6$  times smaller than the ‘normal’ data. It is noted that this approach for dealing with missing features can not be expressed in the framework of [Irani and Anandan, 2000].

In order to evaluate the performance the residuals between the 2D features and the corresponding reprojected 3D features were calculated. The desired effect is that the residuals of the ‘good’ 2D features should be small, hereby indicating that they were not ‘disturbed’ by the erroneous

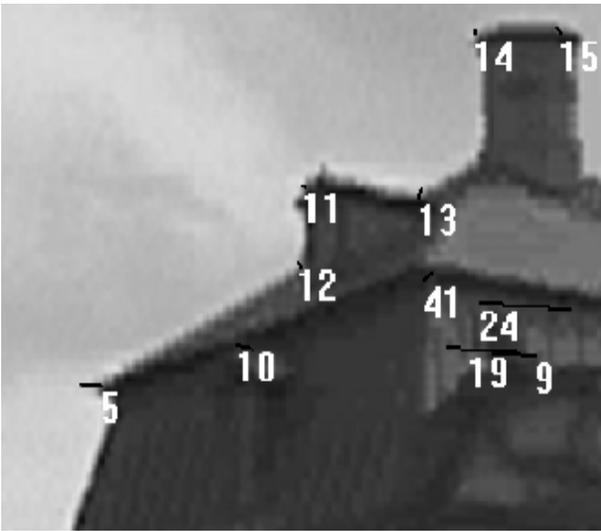


Figure 9: Same section as in Figure 8, but this time with the 2–norm instead of the truncated quadratic.

data. An example of the advantage obtained by the truncated quadratic is illustrated in Figures 8 and 9. In order to further investigate the capability and effect of using different error functions we compared the 80% percent smallest residuals of the three used error functions, see Table 2. The underlying assumption is that no more than 20% of the data is erroneous. It is seen, that via this assumption there is a considerable improvement in choosing other error functions then the 2–norm.

Error Function:	$\frac{1}{n} \sum_i  Res _i$
2–Norm	5.39 pixels
Hubers M-estimator	4.12 pixels
Truncated Quadratic	2.34 pixels

Table 2: Comparison of the 80% smallest residuals

## 5.2 Error Tolerance

To provide a rigorous experimental validation of the proposed method, a set of experiments were made by taking a ‘good’ data set and gradually degrading it with large errors at random. The used data set was the Court sequence, see Figure 5. Twenty features were traced by hand through the eight frames. The proposed algorithm with the truncated quadratic was compared to the method of Christy–Heraud, hereby accessing the improvements achieved. Experiments were performed with three types of errors, namely large Gaussian noise, missing features and swapping features.

In the first experiment an increasing number of the 2D features were corrupted by large gaussian noise. As a quality measure the mean error between the original non–corrupted data and the reprojected 3D structure was calculated, see Figure 10. In some cases the algorithm did not converge or in the Christy–Heraud case the Euclidean reconstruction faulted. This is illustrated in the Figures by not drawing a bar. It is seen that the effect of this corruption is considerably diminished with the truncated quadratic compared to the original method.

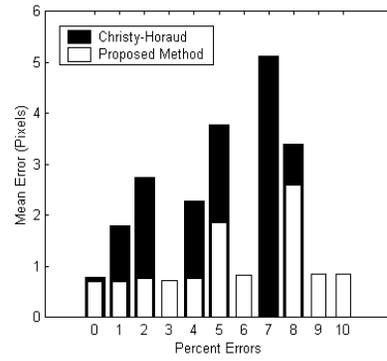


Figure 10: Percentage of 2D features corrupted by Gaussian noise with variance of 400 pixels, and the corresponding mean error of the reconstruction.

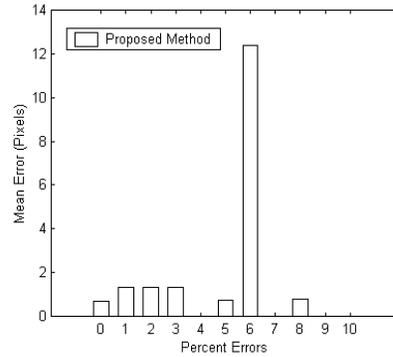


Figure 11: Percentage of 2D features removed, and the corresponding mean error of the reconstruction.

In the next experiment an increasing number of 2D features were removed, see Figure 11. It is seen that the proposed approach converges and that the effect on the reconstruction is negligible. It is impossible to deal with missing data in the original method.

In the last experiment an increasing number of 2D features were swapped within a given frame, Figure 12. The swapping of features is a good emulation of mismatched features. Again a considerable improvement is observed.

It is seen that the proposed approach, used to implement the truncated quadratic as an error function, yields considerably better results. It is also noted that the higher degree of convergence in the proposed approach is due to the proposed approach to Euclidean reconstruction.

## 5.3 Euclidean Reconstruction

To further illustrate the benefits of the proposed method for Euclidean reconstruction a simulated data set was constructed. Here features were swapped by the same scheme as in Figure 12, and the number of non–converging runs were counted in bins of 5, see Figure 13. The result clearly demonstrates the advantages of the proposed method.

## 6 DISCUSSION

Factorization algorithms represent an important set of tools for recovering structure and motion from an image stream. They can be used as a full solution to the problem or as the very important initialization step in non-linear minimization methods [Slama, 1984, Triggs et al., 2000].

We have presented a computationally efficient algorithm for applying arbitrary error functions in the factorization scheme for structure and motion. It has also been demonstrated on real typical data and via rigorous tests, that this scheme deals well with erroneous data.

It is noted, that the particular choice of error function is up to the fancy of the user. For a further survey of the benefits of different error functions the reader is referred to [Black and Rangarajan, 1996].

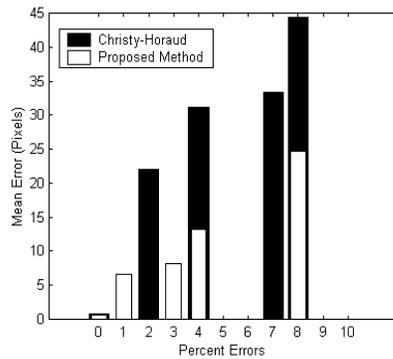


Figure 12: Percentage of 2D features swapped to emulated errors in the correspondence algorithm, and the corresponding mean error of the reconstruction.

## REFERENCES

Black, M. and Rangarajan, A., 1996. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision* 19(1), pp. 57–91.

Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V. and Trosset, M., 1999. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17(1), pp. 1–13.

Christy, S. and Horaud, R., 1994. Euclidian shape and motion from multiple perspective views by affine iterations. Technical Report 2421, INRIA.

Christy, S. and Horaud, R., 1996. Euclidean shape and motion from multiple perspective views by affine iteration. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18(11), pp. 1098–1104.

Costeira, J. and Kanade, T., 1998. A multibody factorization method for independently moving objects. *Int'l J. Computer Vision '98* 29(3), pp. 159–179.

Fletcher, R., 1987. *Practical Methods of Optimizations*. John Wiley & Sons.

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. In: *Proc. Alvey Conf.*, pp. 189–192.

Hartley, R. and Zisserman, A., 2000. *Multiple View Geometry*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK.

Huber, P., 1981. *Robust Statistics*. John Wiley, New York.

Irani, M. and Anandan, P., 2000. Factorization with uncertainty. In: *European Conf, Computer Vision'2000*, pp. 539–553.

Jacobs, D., 2001. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding* 82(1), pp. 57–81.

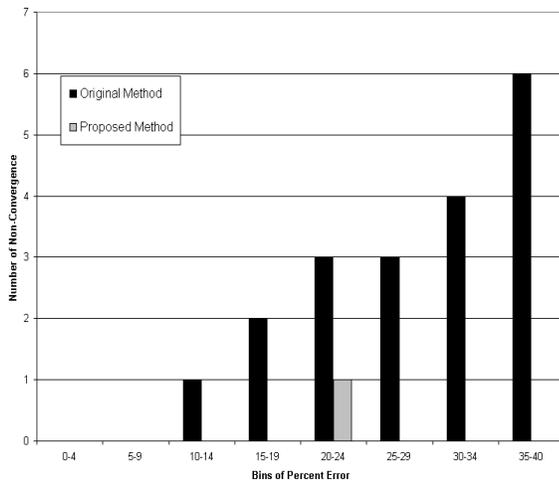


Figure 13: The number of non-converging runs with increasing number of errors, with and without the proposed method for Euclidian reconstruction. The runs are pooled in bins of 5.

Kanade, T. and Morita, T., 1994. A sequential factorization method for recovering shape and motion from image streams. In: Proc. Image Understanding Workshop, Vol. 2, pp. 1177–1187.

Morris, D. and Kanade, T., 1998. A unified factorization algorithm for points, line segments and planes with uncertainty models. In: Int'l Conf. Computer Vision'98, pp. 696–702.

Poelman, C. and Kanade, T., 1997. A paraperspective factorization method for shape and motion recovery. IEEE Trans. Pattern Analysis and Machine Intelligence 19(3), pp. 206–218.

Quan, L. and Kanade, T., 1996. A factorization method for affine structure from line correspondences. In: IEEE Conf. Computer Vision and Pattern Recognition'96, pp. 803–808.

Slama, C. (ed.), 1984. Manual of Photogrammetry. 4:th edn, American Society of Photogrammetry, Falls Church, VA.

Sturm, P. and Triggs, B., 1996. A factorization based algorithm for multi-image projective structure and motion. In: European Conf, Computer Vision'96, Vol. 2, pp. 709–720.

Tomasi, C. and Kanade, T., 1992. Shape and motion from image streams under orthography: a factorization method. Int'l J. Computer Vision'92 9(2), pp. 137–154.

Torr, P. H. S., 2000. Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78(1), pp. 138–156.

Triggs, B., McLauchlan, P. F., Hartley, R. I. and Fitzgibbon, A. W., 2000. Special sessions - bundle adjustment - a modern synthesis. Lecture Notes in Computer Science 1883, pp. 298–372.