

Outlier Correction in Image Sequences for the Affine Camera*

D. Q. Huynh[†] R. Hartley[‡] A. Heyden[◇]

[†]School of Comp. Sc. and Software Eng., The University of Western Australia, AUSTRALIA

[‡]Research School of Information Sciences, The Australian National University, AUSTRALIA

[◇]School of Technology and Society, Malmö University, SWEDEN

du@csse.uwa.edu.au

hartley@syseng.anu.edu.au

heyden@ts.mah.se

Abstract

It is widely known that, for the affine camera model, both shape and motion can be factorized directly from the so-called image measurement matrix constructed from image point coordinates. The ability to extract both shape and motion from this matrix by a single SVD operation makes this shape-from-motion approach attractive; however, it can not deal with missing feature points and, in the presence of outliers, a direct SVD to the matrix would yield highly unreliable shape and motion components. In this paper, we present an outlier correction scheme that iteratively updates the elements of the image measurement matrix. The magnitude and sign of the update to each element is dependent upon the residual robustly estimated in each iteration. The result is that outliers are corrected and retained, giving improved reconstruction and smaller reprojection errors. Our iterative outlier correction scheme has been applied to both synthesized and real video sequences. The results obtained are remarkably good.

1. Introduction

The aim in the shape-from-motion problem is to recover both structure and motion from a number of image measurements, which can be in the form of point coordinates, line coordinates, conics, etc. Many research papers about different approaches to the problem have been reported in the literature. The most general approach involves the use of the perspective camera model. The affine camera model, which was introduced in [15], became popular due to the reduction in the number of unknown parameters involved. When a large number of images, e.g. an image sequence, is available, the use of the affine camera allows one to retrieve both shape and motion directly from image measurements. This procedure, first proposed by Tomasi and Kanade [20], involves first stacking the image point coordinates to form the so-called image measurement matrix and then factorizing the matrix into the shape and motion components via the singular value decomposition.

The problem with missing image point data in the image

*This research was in part supported by a UWA Startup grant and SSF Junior Individual Grant, project 99-241. Huynh thanks Martin Johansson of Lund University for providing the casino video sequence.

measurement matrix is long recognized [20]. More recent works on the treatment of missing image features include those by Jacobs [12], Heyden and Kahl [8], and Hartley and Schaffalitzky [6]. In [12], missing data are dealt with by fitting a low rank matrix to an image measurement matrix with missing elements. In [8], the *closure constraints* are used. In [6], the power method is employed for the factorization, and generalized to cover the cases where there are missing data and where the projection is perspective.

The factorization problem becomes more complicated when the perspective camera model is introduced into the system, as the *relative depth* of each image point must also be determined. Different approaches have been reported to deal with this problem: Sturm and Triggs [19] solve for the relative depths via the fundamental matrix; Heyden [7] and Sparr [18] incorporate the subspace constraint into their iterative methods to estimate the relative depths. Their methods were later extended in [2] to the recursive case in which more images can be incorporated when new tracked feature points become available. Another iterative approach that is similar to [2] but that leads to a different optimization problem is recently reported in [14].

The presence of outliers is inevitable when one has to work on real image data. The RANSAC (Random Sample Consensus) paradigm proposed by Fischler and Bolles [4] detects outlying data by first randomly selecting samples of the minimum number of data items required to estimate a given entity and then looking for consensus of the estimates among the samples. In the estimation of the fundamental matrix and trifocal tensor, this paradigm and the Least Median Squares (LMedS) method have both been applied [21, 22]. The LMedS method can be naturally extended to outlier detection in image sequence data and has been reported in [16, 3, 10]. In all these reported works on outlier detection, the improvement of the estimates of the entities involved, after the elimination of outliers, has been significant, which indicates the importance of treatments of outliers. Once the estimates of these entities have been found, outliers can always be corrected in a postprocess and reintroduced back into the system. In this paper, we present an outlier correction scheme based on the factorization approach reported in [6]. Our scheme is iterative in nature and it handles the correction of outliers and

the factorization into shape and motion components simultaneously. The correction of outlying data in a data set is not new in the literature of computation of robust estimation, as described in Section 3. However, to our knowledge, the correction of outliers has not been attempted in the structure-from-motion problem by many researchers. The only piece of related work is that of Aanæs et al [1] who employ weighting factors to reduce the influence of outliers. In [1], each weighting factor is set to be inversely proportional to the computed residual of each image feature point. Our approach, on the other hand, is based on an estimator proposed by Huber [9] (see also [5]) which bounds the influence of outliers by bringing them toward the majority yet allows them to have maximum influence.

The paper is organized as follows. In Section 2, we give a brief review of the affine factorization method and related work. In Section 3, we first give a brief background on computation of robust estimates, drawing upon a similar but simpler problem, and then describe our iterative outlier correction scheme in detail. In Section 4, experiments on both synthesized and real video image sequences are given. Issues related to the proposed scheme are discussed in Section 5, and conclusions are given in Section 6.

2. Background

2.1. Affine factorization

Given a scene point $\mathbf{X}^j = (X^j, Y^j, Z^j, 1)^\top$ and its corresponding image point $\mathbf{x}^j = (x^j, y^j)^\top$, for some index j . Under the affine projection onto the i^{th} image, they are related by

$$\mathbf{x}^j = P_i \mathbf{X}^j, \quad (1)$$

where $P_i \in \mathbb{R}^{2 \times 4}$. With the availability of m images and n scene points, (1) becomes

$$\begin{bmatrix} \mathbf{x}_1^1 & \cdots & \mathbf{x}_1^n \\ \vdots & \ddots & \vdots \\ \mathbf{x}_m^1 & \cdots & \mathbf{x}_m^n \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix} [\mathbf{X}^1 \cdots \mathbf{X}^n] \quad (2)$$

$$\Leftrightarrow \mathbf{M} = \mathbf{P}\mathbf{X},$$

where matrix $\mathbf{P} \in \mathbb{R}^{2m \times 4}$ is known as the motion matrix, $\mathbf{X} \in \mathbb{R}^{4 \times n}$ the shape matrix, and $\mathbf{M} \in \mathbb{R}^{2m \times n}$ the image measurement matrix.

Given a matrix \mathbf{M} constructed from a number of feature points tracked in a sequence of images, the first problem to solve is factorization to retrieve the shape and motion matrices. Note that most factorization techniques (e.g. [20, 1]) reduce \mathbf{P} to $\mathbb{R}^{2m \times 3}$ and \mathbf{X} to $\mathbb{R}^{3 \times n}$ by setting the origin of each image coordinate system at the centre of mass of the image feature points. However, as these centres of mass cannot be reliably computed due to the presence of outliers, we retain both \mathbf{P} and \mathbf{X} to be rank-4 matrices.

2.2. Related work

In [6] an algorithm is given that takes a random initial estimate of $\mathbf{A}^{(0)}$ or $\mathbf{B}^{(0)}$ and iteratively finds a solution to the factorization problem

$$\mathbf{M} = \mathbf{A}\mathbf{B}^\top,$$

where $\mathbf{M} \in \mathbb{R}^{p \times q}$, $\mathbf{A} \in \mathbb{R}^{p \times r}$, and $\mathbf{B} \in \mathbb{R}^{q \times r}$. If the rank of \mathbf{M} is exactly r then the factorization converges in one iteration; if the rank of \mathbf{M} is greater than r then only a small number of iterations is required. This iterative scheme involves simple updates to the two factor matrices: At the k^{th} iteration, given that $\mathbf{A}^{(k-1)}$ is known,

1. compute $\mathbf{B}^{(k)}$ using

$$\mathbf{B}^{(k)} = \mathbf{M}^\top \mathbf{A}^{(k-1)} \left(\mathbf{A}^{(k-1)\top} \mathbf{A}^{(k-1)} \right)^{-1}, \quad (3)$$

2. compute $\mathbf{A}^{(k)}$ for the next iteration using

$$\mathbf{A}^{(k)} = \mathbf{M}\mathbf{B}^{(k)} \left(\mathbf{B}^{(k)\top} \mathbf{B}^{(k)} \right)^{-1}. \quad (4)$$

One advantage of this simple iterative scheme is that the computation cost, which involves inversions of small matrices (for the affine factorization problem, $r = 4$) and some matrix multiplications, is much smaller than with the SVD approach. This advantage is significant when the given matrix \mathbf{M} is large, which is often the case for the shape-from-motion problem.

In the tracking of image features, it often happens that some features become missing in the tracking process due to occlusion and/or change of lighting conditions. Another advantage of this iterative scheme is that it can handle missing entries in the given matrix. Thus, instead of having a separate post-processing step to fill in the missing entries, the iterative scheme allows one to carry out the factorization of a matrix that has *holes*.

For the affine factorization problem, since \mathbf{X} has the form that its last row is 1s, by rewriting (2) as

$$\mathbf{M} = \begin{bmatrix} \tilde{\mathbf{P}} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}} \\ \mathbf{1}^\top \end{bmatrix}, \quad (5)$$

where $\tilde{\mathbf{P}} \in \mathbb{R}^{2m \times 3}$, $\mathbf{t} \in \mathbb{R}^{2m}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{3 \times n}$, and $\mathbf{1}$ is an n -vector that contains 1s, it can be verified that the update given in (3) becomes (see also [6])

$$\tilde{\mathbf{X}}^{(k)\top} = \left(\mathbf{M}^\top - \mathbf{t}\mathbf{t}^\top \right) \tilde{\mathbf{P}}^{(k-1)} \left(\tilde{\mathbf{P}}^{(k-1)\top} \tilde{\mathbf{P}}^{(k-1)} \right)^{-1}. \quad (6)$$

In this paper, we modify this update scheme to our affine factorization problem where outlying coordinates are present in the image measurement matrix. In our iterative outlier correction scheme, in addition to the update of the two factor matrices, the matrix \mathbf{M} itself is also refined so as to correct the outlying coordinates.

3. Correcting the outlying coordinates

3.1. Background on computation of robust estimates

Many practical applications come in the form where the method of least squares can be applied.

Given a linear system, with an unknown vector $\mathbf{x} = (x_1, \dots, x_q)^\top \in \mathbb{R}^q$, of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (7)$$

where $\mathbf{A} \in \mathbb{R}^{p \times q}$ with $p > q$, and $\mathbf{y} = (y_1, \dots, y_p)^\top \in \mathbb{R}^p$. Suppose that our aim is to get a least squares fit on the observations y_i to yield the fitted values \hat{y}_i and residuals $r_i = y_i - \hat{y}_i$, for $i = 1, \dots, p$. The setting of (7) to

$$(\mathbf{y} - \mathbf{A}\mathbf{x})^\top (\mathbf{y} - \mathbf{A}\mathbf{x}) \rightarrow \min \quad (8)$$

obviously yields the estimate $\hat{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{y}$ and the estimate $\hat{\mathbf{y}} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{y}$. Define

$$\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top. \quad (9)$$

Then clearly $\mathbf{H}\mathbf{y} = \hat{\mathbf{y}}$. Because of this mapping from \mathbf{y} to $\hat{\mathbf{y}}$, \mathbf{H} is often called the *hat matrix*. This matrix has the property of being idempotent, i.e., $\mathbf{H}^2 = \mathbf{H}$, and $\text{tr}(\mathbf{H}) = q$. It is also known to contain useful information about leverage points. Indeed, many studies in the literature (see e.g. Huber [9]) report that leverage points show up with large h_i values, where $(h_1, \dots, h_p)^\top = \text{diag}(\mathbf{H})$. Unfortunately, if we have a gross error in y_i rather than in \mathbf{A} then the gross error may not show up in r_i . Instead, it may show up elsewhere, say in r_j , if H_{ij} happens to be large. Nevertheless, \mathbf{H} still plays a significant role in the estimation of the standard error of each observation, as discussed below. An alternative scheme, which we found more suitable to our outlier correction problem and which has been reported in the literature of robust estimation, is: If the observations y_i are noisy then the influence of outliers can in fact be deflated by iteratively replacing the observations y_i by some pseudo-observations y_i^* , via the following criteria (see Huber [9])

$$y_i^* = \begin{cases} y_i & \text{if } |r_i| \leq cs_i, \\ \hat{y}_i - cs_i & \text{if } r_i < -cs_i \\ \hat{y}_i + cs_i & \text{if } r_i > cs_i, \end{cases} \quad (10)$$

where the constant c regulates the amount of robustness. Good values for c are in the interval $[1, 2]$, e.g. $c = 1.5$. In (10), the entity $\mathbf{s} = (s_1, \dots, s_p)^\top$ is the standard error vector, in which each s_i is some estimate of the standard error of y_i given by

$$s_i = \sqrt{1 - h_i} \hat{\sigma}. \quad (11)$$

The entity $\hat{\sigma}^2$ above is the classical estimate of the variance of a single observation: $\hat{\sigma}^2 = \sum_{i=1}^p r_i^2 / (p - d)$, with d

being the number of unknown parameters to be estimated. In the example above, $d = q$.

The method described above treats both inliers and outliers uniformly in the data set. That is, there are no weighting factors involved (c.f. [1]). Outlying observations, which have large residual magnitudes $|r_i|$, would not fulfil the criterion $|r_i| \leq cs_i$ and would be updated in the iterations. The idea here is to *clean* the data by iteratively pulling the observations y_i toward the fitted observations \hat{y}_i until convergence is achieved. As demonstrated in both synthesized and real video sequences (see Sections 4 and 5), convergence is guaranteed and the resultant, refined joint image measurement matrix also gives a much smaller mean and standard deviation of reconstruction error.

From the steps above, we see that only those elements of y_i that have large residuals are updated (see (10)) and that the signs and magnitudes of the updates are, respectively, dependent upon the signs of the residuals and the magnitudes of the standard errors given in (11). In the formula, $r_i = y_i - \hat{y}_i$, for estimating residuals, the magnitude of r_i remains rather large even after a number of iterations. An alternative formula (which appeared to give better results for our problem) is to define r_i as

$$r_i = y_i^* - \hat{y}_i. \quad (12)$$

However, this requires the estimation of $\hat{\sigma}^2$ given above to be replaced by $\hat{\sigma}^2 = \sum_{i=1}^p r_i^2 / ((p - q)(v/p)^2)$, where v is the number of *unmodified* observations.

3.2. Our iterative outlier correction scheme

By relating the above formulation with the iterative scheme for affine factorization reported in [6], it is not difficult to see that the two problems are similar and that the above update criteria are applicable since outliers are present in the image measurement matrix \mathbf{M} rather than the \mathbf{P} (or \mathbf{X}) matrix. Our problem of factorizing a noisy image measurement matrix is an extension of the above least squares problem in that

- instead of having an unknown vector \mathbf{x} we now have an unknown matrix (either the motion matrix \mathbf{P} or the shape matrix \mathbf{X}) to compute in each iteration;
- instead of having a residual vector we now have a residual matrix \mathbf{R} ; correspondingly, a matrix \mathbf{S} is required to store the estimate of the standard error of each element of \mathbf{M} .

The combination of the above criteria for updating the image measurement matrix \mathbf{M} and the iterative estimation of the motion and shape matrices produces the following iterative outlier correction scheme:

Generate a random matrix $\tilde{\mathbf{X}}^{(0)} \in \mathbb{R}^{3 \times n}$. Set $\mathbf{M}^{(0)} = \mathbf{M}$. Initialize $\mathbf{v} = (v_1, \dots, v_n)^\top$ to $(2m, \dots, 2m)^\top$, where v_j stores

the number of unmodified elements for the j^{th} column of M . For iteration $k = 1, 2, \dots$

1. Construct $X^{(k-1)\top} = [\tilde{X}^{(k-1)\top} \mathbf{1}]$.
2. Compute $P^{(k)} = M^{(k-1)} X^{(k-1)\top} (X^{(k-1)} X^{(k-1)\top})^{-1}$.
3. Let $[\tilde{P}^{(k)} \mathbf{t}^{(k)}] = P^{(k)}$ and the rank-3 hat matrix $H \in \mathbb{R}^{2m \times 2m}$ be $\tilde{P}^{(k)} (\tilde{P}^{(k)\top} \tilde{P}^{(k)})^{-1} \tilde{P}^{(k)\top}$. Then the estimate \hat{M} and residual matrix R are

$$\hat{M} = H (M^{(k-1)} - \mathbf{t} \mathbf{1}^\top) + \mathbf{t} \mathbf{1}^\top \quad (13)$$

$$R = M^{(k-1)} - \hat{M}. \quad (14)$$

4. Let $\hat{\sigma}^2 = (\hat{\sigma}_1^2, \dots, \hat{\sigma}_n^2)$ be the vector that stores the estimates of variances of single observations. Each element of $\hat{\sigma}^2$ is given by

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{2m} R_{ij}^2}{(2m-3)(v_j/2m)^2}, \quad \text{for } j = 1, \dots, n, \quad (15)$$

where v_j is the number of unmodified M elements on the j^{th} column. The denominator in (15) above involves the term $2m-3$ since, for each $\hat{\sigma}_j^2$, a column vector of length 3 of $\tilde{X}^{(k)}$ is estimated.

5. Let $S \in \mathbb{R}^{2m \times n}$ be the matrix that stores the standard errors of all the estimates in matrix \hat{M} . Then each element of S is defined as $S_{ij} = \sqrt{1 - h_i} \hat{\sigma}_j$.
6. Update each element, M_{ij} , of M for the next iteration as follows:

$$M_{ij}^{(k)} = \begin{cases} M_{ij}^{(k-1)} & \text{if } |R_{ij}| \leq c S_{ij} \\ \hat{M}_{ij} - c S_{ij} & \text{if } R_{ij} < -c S_{ij} \\ \hat{M}_{ij} + c S_{ij} & \text{if } R_{ij} > c S_{ij}, \end{cases} \quad (16)$$

where c was kept constant at 1.5 in all our experiments. Update v_j , for $j = 1, \dots, n$, and use them in the following iteration.

7. Compute $X^{(k)}$ for the next iteration as follows:

$$\tilde{X}^{(k)\top} = (M^{(k)\top} - \mathbf{1} \mathbf{t}^{(k)\top}) \tilde{P}^{(k)} (\tilde{P}^{(k)\top} \tilde{P}^{(k)})^{-1}. \quad (17)$$

8. Terminate the iteration when the largest update to the elements of M is less than a given threshold, i.e. when, for some small fixed $\varepsilon > 0$,

$$\max_{i,j} |M_{ij}^{(k)} - M_{ij}^{(k-1)}| < \varepsilon. \quad (18)$$

Otherwise, go to Step 1.

9. Set $P = P^{(k)}$, $X = X^{(k)}$, and $M = M^{(k)} = P^{(k)} X^{(k)}$.

In our experiments, we found that (14) (c.f. (12)) gave better results than the formula $R = M - \hat{M}$ in that the reconstruction and reprojection errors were smaller while no extra iterations are required. We have experimented with

various modified versions of the algorithm given above. For instance, the update given in (16) can be deferred until after the first two iterations, since in our outlier correction problem, P is also an unknown matrix to be estimated and that the hat matrix may not be well defined in the first iteration. For the same reason, if an update to M is desired in the first iteration, one may set H to $\text{diag}(\frac{3}{2m}, \dots, \frac{3}{2m})$ so that all columns of $M^{(0)}$ would be equally weighted when the estimate of \tilde{P} is not entirely certain. In the experiments described in the next section, we only update the image measurement matrix from the second iteration onward. An alternative scheme is to reverse the role of P and X and refine M^\top instead. We have experimented with this alternative scheme and found no advantages over the scheme described above.

4. Experiments

We tested the iterative outlier correction scheme described above on both synthesized and real image sequences. To closely monitor the updates in each iteration, our synthesized image sequences contained 5 random, affinely projected images and 30 random scene points, giving a M matrix of dimensions 10×30 . Small uniform random noise (up to ± 0.5 pixels) was first added to each of the elements of M , outliers were then added to a percentage (varied from 5 to 45%) of columns of M . The errors were up to ± 10 pixels. For each outlying column in M , up to 4 (corresponding to 2 images or 40% of the number of image frames) of the 10 image coordinates could be contaminated by outliers. For each noisy image measurement matrix M , the iterative scheme described in Section 3.2 was carried out to refine M , P and X ; the true 3D points were used to upgrade the structure stored in X to Euclidean and the reconstruction and reprojection errors were evaluated. The value of ε in (18) was set to 10^{-3} in all experiments.

Table 1 shows the results of 100 runs of the scheme for each level of contamination. In each run, different noise and outliers were added, so a different M matrix was used. In the Table, the means ($E_{\mathbf{x}}$) and standard deviations ($\sigma_{\mathbf{x}}$) of reprojection errors (in pixels) and the means ($E_{\mathbf{z}}$) and standard deviations ($\sigma_{\mathbf{z}}$) of reconstruction errors obtained from the original noisy M matrix and the refined M matrix are compared. The improvement in both reconstruction and reprojection errors is evident. The means (E_k) and standard deviations (σ_k) of the numbers of iterations are also given in the Table. As the percentage of outliers increases, both the reconstruction and reprojection errors increase accordingly, and more iterations are required. For each level of noise, we also ran our scheme 100 times on the same M matrix with different $X^{(0)}$ s. It was confirmed from our experiments that the performance of the scheme is immune from the initial estimate of \tilde{X} .

The images and output of our real experiments are shown

% outliers	using the original noisy \hat{M}				using the refined \hat{M}					
	E_x	σ_x	E_x	σ_x	E_x	σ_x	E_x	σ_x	E_k	σ_k
5%	0.46	0.17	0.70	1.28	0.39	0.10	0.18	0.31	98.05	48.70
10%	0.62	0.33	1.34	1.87	0.45	0.18	0.26	0.48	128.08	58.05
15%	0.73	0.40	2.42	22.18	0.50	0.23	0.35	0.64	134.65	50.19
20%	0.89	0.51	4.01	41.31	0.56	0.30	0.47	0.79	152.47	66.63
25%	1.00	0.58	5.73	49.61	0.59	0.34	0.52	0.91	165.24	68.09
30%	1.13	0.67	10.44	186.13	0.67	0.42	0.65	0.99	161.73	51.18
35%	1.18	0.71	9.63	119.80	0.67	0.41	0.66	1.03	171.65	71.80
40%	1.30	0.79	8.94	161.57	0.74	0.47	0.79	1.10	183.87	63.71
45%	1.40	0.86	10.85	50.70	0.76	0.48	0.79	1.06	181.97	65.96

Table 1: Reprojection and reconstruction errors from the synthesized image sequences.

in Figs. 1 and 2. To improve the visibility of the corrected feature point coordinates, all the images have been significantly brightened. In both experiments, we used the KLT feature tracker [13, 17] to track feature points frame by frame. In our first experiment, a sequence of 131 images of a desk scene was taken and 162 feature points were tracked. In order to compare the performances of our scheme with our previous LMedS technique [10], 7 frames at 20 frames apart were extracted from the sequence, giving a $M \in \mathbb{R}^{14 \times 162}$ matrix. Because of the large separation of frame numbers, the accumulation of erroneous tracking gave outliers (shown as red line segments in Fig. 1) as large as 45 pixels in frame 120 (see the tails of red line segments in frame 100). The output from our scheme is shown in Fig. 1(b). A total of 135 iterations were required. This experiment demonstrates that the scheme is capable of correcting large outlying errors.

In our second experiment, we extracted frames 0 to 200 from a long video sequence of over a thousand images of a wall and ceiling of a casino in Madrid. 252 feature points were found in this shorter sequence, giving a $M \in \mathbb{R}^{402 \times 252}$. Our scheme was applied to this large M matrix and converged after 106 iterations. Three images from this sequence are shown in Fig. 2(b), superimposed onto each image are the corrected image coordinates. The same image sequence was fed to our LMedS technique and the detected, but uncorrected, outliers are shown in Fig. 2(a). The coordinates of all clusters of outliers were successfully corrected (see Fig. 2(b)), e.g. the rightmost cluster of outliers in frame 80 (see frame 60); the centred cluster of outliers in frame 60 (see frame 40).

5. Discussions

It is encouraging that outliers that are often found in feature tracking in video images can be corrected and retained by this simple iterative scheme. One of our interests during the implementation was to find the pathological cases of the scheme. Since the scheme is immune from the initial guess $X^{(0)}$, this leaves the patterns of outliers that appear in M as the only possible source of problems. In the search for outlier patterns, we discovered the following two extreme

patterns from our synthesized data:

- If more than 50% of the elements in a particular row, say the i^{th} row (or two consecutive rows that correspond to the i^{th} image), of M were outlying coordinates then the scheme would give a large mean reprojection error. The culprit here was the poor estimate of the P_i matrix, which inflated the mean reprojection error. This situation may occur in image capturing in practice using a hand-held video camera when a small number of frames might become jerky due to the movement or walking pace of the cameraman. Fortunately, good 3D reconstruction was always guaranteed if the majority of the affine projection matrices were good.
- If more than 50% of the elements in a particular column, say the j^{th} column, of M were outlying coordinates then the scheme would not be able to correct them, thus giving a wrong estimate of the j^{th} scene point and large reprojection errors for the j^{th} column. However, due to the independent treatment of columns of M in the scheme (c.f. Sections 3.1 and 3.2), the 3D reconstruction errors and reprojection errors for other columns were not affected. While the proportion of wrong coordinates in an outlying column is not an issue in outlier removal schemes, more good coordinates than bad ones are essential in outlier correction schemes.

Being a least-squares based approach, the scheme works only if there are both positive and negative outlying coordinates (from their true values) in the data set. This allows a reasonably good \hat{M} matrix to be attained in early iterations. The subsequent iterations would then simply involve pulling the outlying coordinates toward their true ones. While the errors involved in outliers (especially if they fall onto image edges) may exhibit some correlation between their x - and y -coordinates, as discussed in [11], so long as there are both positive and negative errors in the data set, the scheme would give very good performance.

The number of iterations shown in our experiments may be considered to be large. We found empirically that the scheme converged quickly in early iterations to give good estimates of the M matrix in all the synthesized and real experiments and that in later iterations the scheme gave refinement to digits on the third decimal place onward. One possibility for speeding up the convergence is to increase the value of c slightly in later iterations. However, care must be taken to ensure that such a change would not cause the convergence to oscillate.

6. Conclusions

We have employed an existing well-founded statistical technique to the problem of outlier correction in image sequences for the affine camera. The outlier correction

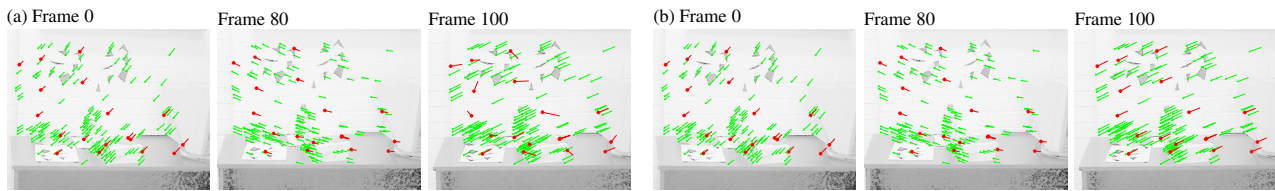


Figure 1: Three images of a video sequence of a desk scene. (a) The detected in- and outliers from our previous LMEDS technique. (b) The corrected image coordinates obtained from our iterative outlier correction scheme.

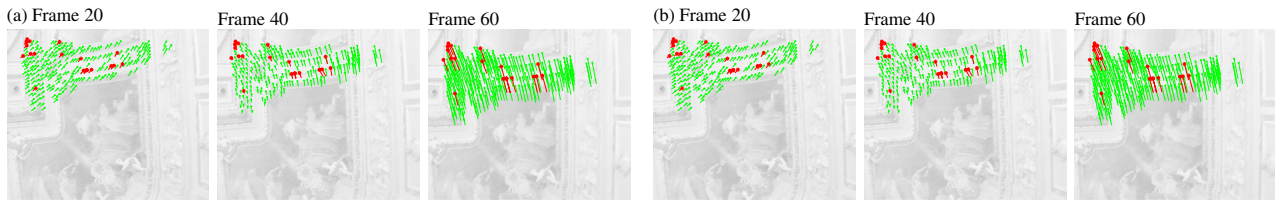


Figure 2: Three images of a video sequence of a casino scene. (a) The detected in- and outliers from our previous LMEDS technique. (b) The corrected image coordinates obtained from our iterative outlier correction scheme.

scheme corrects outliers and factorizes the image measurement matrix simultaneously. The proposed method has been successfully tested on both synthetic and real video data. It would be interesting to compare our scheme with other outlier detection/correction methods (e.g. [1]) and to investigate the extension of this scheme to the problem of shape reconstruction for the perspective camera and when the image measurement matrix has missing elements. We intend to do this in a future publication.

References

- [1] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Robust Factorization. *IEEE Trans. PAMI*, 24(9):1215–1225, 2002.
- [2] R. Berthilsson, A. Heyden, and G. Sparr. Recursive Structure and Motion from Image Sequences using Shape and Depth Spaces. In *Proc. CVPR*, pages 444–449, 1997.
- [3] G. Q. Chen. Robust Point Feature Matching in Projective Space. In *Proc. CVPR*, vol 1, pages 717–722, 2001.
- [4] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *CACM*, 24(6):381–395, 1981.
- [5] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics – The Approach Based on Influence Functions*. John Wiley & Sons, 1986.
- [6] R. Hartley and F. Schaffalitzky. PowerFactorization: 3D Factorization with Missing or Uncertain Data. Technical report, ANU, 2003. Available from <http://www.syseng.anu.edu.au/~hartley>.
- [7] A. Heyden. Projective Structure and Motion from Image Sequences using Subspace Methods. In *Scandinavian Conference on Image Analysis*, pages 1058–1063, 1997.
- [8] A. Heyden and F. Kahl. Reconstruction from Affine Cameras using Closure Constraints. In *Proc. ICPR*, vol 1, pages 47–50, 1998.
- [9] P. J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.
- [10] D. Q. Huynh and A. Heyden. Outlier Detection in Video Sequences under Affine Projection. In *Proc. CVPR*, vol 1, pages 695–701, 2001.
- [11] M. Irani and P. Anadan. Factorization with Uncertainty. In *Proc. ECCV*, vol 1, pages 539–553, 2000.
- [12] D. Jacobs. Linear Fitting with Missing Data: Applications to Structure-from-Motion and its Characterizing Intensity Images. In *Proc. CVPR*, pages 206–212, 1997.
- [13] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, pages 674–679, 1981.
- [14] S. Mohamud and M. Hebert. Iterative Projective Reconstruction from Multiple Views. In *Proc. CVPR*, pages 430–437, 2000.
- [15] J. L. Mundy and A. Zisserman. Appendix - Projective Geometry for Machine Vision. In *Geometric Invariance in Computer Vision*, pages 463–534. The MIT Press, 1992.
- [16] F. Schaffalitzky, A. Zisserman, R. I. Hartley, and P. H. S. Torr. A Six Point Solution for Structure and Motion. In *Proc. ECCV*, vol 1, pages 632–648, 2000.
- [17] J. Shi and C. Tomasi. Good Features to Track. In *Proc. CVPR*, pages 593–600, 1994.
- [18] G. Sparr. Simultaneous Reconstruction of Scene Structure and Camera Locations from Uncalibrated Image Sequences. In *Proc. ICPR*, vol 1, pages 328–333, 1996.
- [19] P. Sturm and B. Triggs. A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In *Proc. ECCV*, pages 709–720, Apr 1996.
- [20] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: a Factorization Method. *IJCV*, 9(2):137–154, 1992.
- [21] P. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, Department of Engineering Science, University of Oxford, 1995.
- [22] 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. *Artificial Intelligence*, 75(1-2):87–120, 1995.