A Practical Algorithm for Structure and Motion Recovery from Long Sequence of Images

Miroslav Trajković and Mark Hedley

Department of Electrical Engineering, Sydney University, NSW 2006, Australia

E-mail: {miroslav, hedley}@ee.usyd.edu.au

Abstract. In this paper we present an algorithm for structure and motion (SM) recovery under affine projection from video sequences that is suitable for real time applications. The algorithm tracks the motion of a single structure, be it an object or the entire scene itself, allowing for any type of camera motion. This could be used for example to track the motion of a vehicle in a warehouse (single object, static camera) or for visual navigation from a moving platform (track scene from moving camera). The algorithm requires a set of features to be detected in each frame, and that at least four features are correctly matched between each three consecutive frames. Compared to previous algorithms, this novel algorithm has a lower computational cost, dynamically detects outliers and allows for previously lost features to reappear in the sequence. The algorithm has been tested on real image sequences, and compared to other algorithms we have found that our algorithm has both a smaller error and a lower computational time.

1. Introduction

In any machine vision task the objects of most interest, and the most demanding to handle, are those that are moving. The reason for this is that once we have information about static objects, by virtue of their being static, we do not need to update information about them at the video rate. Moving objects on the other hand should be tracked frame by frame, and these are the objects that the machine vision system will usually need to respond to (for example collision avoidance). A fundamental machine vision task then is to identify and track moving objects in a video sequence, and to determine their structure, which is assumed to be rigid. In this paper we will approach the problem of determining the motion and structure of a single rigid object moving in the field of view of a static camera. The same algorithm can be used if there is a moving camera in a static environment to determine the motion of the camera and the structure of the environment.

One of the most widely used approaches to structure and motion (SM) recovery is the factorization algorithm proposed by Tomasi and Kanade [8]. Image feature points are tracked over several frames and the SM parameters are computed using a singular value decomposition (SVD). This algorithm cannot be used for real time applications as this is a batch method and can only compute the parameters once all measurements have been made. The algorithm is also computationally very expensive (due to the use of the SVD). Morita and Kanade [5] developed a sequential version that has almost the same accuracy but has a much lower computational burden. The issues not addressed in these papers were how to prevent outliers incorrectly biasing the result, and how to handle a set of feature points that changes dynamically.

Held [1] partially extended the Morita and Kanade algorithm to include new features. The video sequence was divided into subsequence, and the features tracked over the subsequence. To track between the subsequences there needed to be common tracked features between consecutive subsequences. The algorithm was verified using synthetic measurements only, with added Gaussian noise. McLauchlan *et al.* [2,3] used the variable state-dimension filter (VSDF) to handle the problem of missing and new features. They posed the structure from motion problem as a parameter estimation problem and solved it by using the Extended Kalman Filter (EKF). They updated the structure and *only the last* motion estimate at each time step and therefore achieved a low computational cost $O(k)$ compared to when the complete motion matrix has been updated. The authors mentioned outlier rejection, but they gave neither a method nor a reference to handle this problem. Another limitation of their algorithm is that features that have been obscured and reappear are treated as new feature points, and are not recognised as a previously tracked feature.

In this paper we present an algorithm that iteratively computes the SM parameters at each time step. It has the same computational cost $O(n+k)$ as [2] but it will update both the structure and complete motion parameters, so it converges to the true solution. It is made possible by noting that structure should remain constant over time (rigid body assumption) and by updating it at each time step. Motivated by work of Reid and Murray [6] we have extended their algorithm so that it can easily include new features and recover the complete structure and motion providing that for each three consecutive frames there are at least four correctly matched feature points. Furthermore, our algorithm detects outliers at each time step. Finally we developed a procedure that checks whether newly appeared features are genuinely new, or old features that have reappeared. The correct classification of these new features both decreases the computational cost of the algorithm and improves its accuracy.

2. Problem Background

In this paper we assume an affine camera model which has the form:

$$p = MS + t \quad (1)$$

where S is 3×1 is the world coordinate of a feature point, p is 2×1 image projection of S , M is 2×3 projection matrix and t is 2×1 translation vector. This model is a generalisation of the orthographic camera model [5,8] and is a good approximation of the perspective projection when the change in depth is small compared to the average distance from the optical center. This condition is almost always satisfied for independently moving objects. If the object of interest undergoes rigid motion the image projections will not change if we fix the world coordinates of the object of interest and change camera parameters M and t accordingly. Hence, without loss of generality, we may assume that scene is static and that the camera is moving.

Given k projections (images) of n scene points we want to recover the camera motion parameters $M(j)$, $t(j)$, $j = 1, 2, \dots, k$ and scene structure parameters S_i , $i = 1, 2, \dots, n$ (world vectors of tracked scene points). These parameters are related by the measurement equation:

$$p_i(j) = M(j)S_i + t(j) \quad (2)$$

where $p_i(j)$ is the projection of i^{th} point onto the j^{th} image, and i and j vary from $1 \dots n$ and $1 \dots k$ respectively. To compute the the SM parameters using the measurement equations, we must minimise the cost function:

$$C(k) = \sum_{j=1}^k \sum_{i=1}^n \|p_i(j) - M(j)S_i - t(j)\|^2 \quad (3)$$

For real time implementation the parameters must be computed following the acquisition of each frame. It may be shown that the translational component $t(j)$ at each frame is given by the centroid of the feature locations $t(j) = \overline{p_i(j)}$ and equation (2) can now be rewritten as $w_i(j) = M(j)S_i$ where $w_i(j) = p_i(j) - t(j)$, or in matrix form:

$$W = MS \quad (4)$$

where $W = [w_i(j)]_{k \times n}$, $M = [M(1)^T \dots M(k)^T]^T$ and $S = [S_1 \dots S_n]$, and this is the equation obtained (although in different way) by Tomasi and Kanade [8]. To solve it, they employed singular value decomposition of measurement matrix and showed that $M = U_3 \Sigma_3^{1/2}$ and $S = \Sigma_3^{1/2} V_3^T$, where U_3, Σ_3 and V_3 are submatrices of U, Σ and V corresponding to the three largest singular values of W . However the structure and motion parameters can be determined without computing the full SVD, since

$$W^T W = V \Sigma^2 V^T \text{ and } WW^T = U \Sigma^2 U^T \quad (5)$$

hence the structure and motion parameters are given by the eigenvector decomposition of $W^T W$ and WW^T respectively.

Note that equation (4) does not have unique solution. In fact, if A is an arbitrary invertible 3×3 matrix the matrices MA and $A^{-1}S$ are valid solutions [8]. To ensure a unique solution, the first two rows of M are fixed, and now in equation (3) j will range from $2 \dots k$ [3].

3. Algorithm Details

3.1 Structure and Motion Recovery

Unlike Morita and Kanade who have computed motion and structure by iteratively updating equation (4) and McLauchlan who applied EKF to the measurement equation (2), we employ direct minimisation of the cost function (3). For some initial number of frames (usually 3) we initialise the motion and structure matrices by solving equation (4). Once we have the matched features at time $k+1$, and have computed values M , t and S at time k , we wish to find the updated parameters that minimise

$$C(k+1) = C = \sum_{j=2}^{k+1} \sum_{i=1}^n w_{ij} \|p_i(j) - M(j)S_i - t(j)\|^2 = \sum_{j=2}^{k+1} \sum_{i=1}^n w_{ij} \|v_i(j)\|^2 \quad (6)$$

where w_{ij} is a binary weight which is unity if feature i is present in frame j , and zero otherwise.

It is convenient to group the motion parameters M and t into a single vector defined as $m = [M_{11} M_{12} M_{13} t_1 M_{21} M_{22} M_{23} t_2]^T$. To determine the structure and motion parameters that minimise the cost function we must solve:

$$\begin{aligned} \frac{\partial C}{\partial m(j)} &= \sum_{i=1}^n w_{ij} D_i^T v_i(j) = 0, \quad j = 2, 3, \dots, k+1 \\ \frac{\partial C}{\partial S_i} &= \sum_{j=2}^{k+1} w_{ij} E^T(j) v_i(j) = 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (7)$$

where $D_i = -\partial v_i / \partial m$ is a simple function of S_i , and $E = -\partial v_i / \partial S_i = M$. For simplicity the time parameter j has been dropped. Equation (7) forms a non-linear set of equations to be solved. There are three possible approaches to doing this:

1. Use the previously determined values of M and S_i to calculate D_i and E , then linearise the equations using

$$v_i(j) = \hat{v}_{ij} + \frac{\partial v_i}{\partial m} \Delta m(j) + \frac{\partial v_i}{\partial S_i} \Delta S_i = \hat{v}_{ij} - D_i \Delta m(j) - E^T(j) \Delta S_i \quad (8)$$

From this we can rewrite the system as a linear set of equations to be solved for $\Delta m(j)$ and ΔS_i . This is the set of equations obtained in [3], and the algorithm has linear convergence.

2. Linearise all terms in equation (7), rather than assuming that the derivatives are constant. From this a different set of linear equations is determined that are similar to those in the previous technique, but this has quadratic convergence.
3. The simplest method is to solve for $m(j)$ using the previous values for the derivatives and S_i . Then use the updated parameters to solve for S_i . This is alternated until convergence. This method has linear convergence, but the lowest computational cost.

We used the second method because it has quadratic converge, and usually found three to five iterations sufficient. For the initial estimates of the parameters at each stage of the algorithm we used the parameters determined in the previous stage.

3.2 Matching and Outlier Rejection

Before we can recover the structure and motion parameters we need to have matched feature points (here corners) between consecutive images. This usually involves matching two sets of corners using a cross correlation of a window about each corner [9]. This procedure will often give some small number of incorrect matches (outliers), which can significantly bias the structure and motion estimates. These outliers should be detected so they can be ignored. For this we developed a technique based on robust statistics, using the Least Median of Squares (LMedS) estimator [4], which is described in this section.

Matching between two images.

Having matched the corners between two consecutive images, we randomly select four matched corners to obtain a sample matrix $W_S = [\mathbf{x}_1 \ \mathbf{y}_1 \ \mathbf{x}_2 \ \mathbf{y}_2]'$ where $\mathbf{x}_i$ is a vector containing the x coordinates of the four selected corners in image i , and $\mathbf{y}_i$ is likewise defined. We can decompose this 4×4 matrix using the SVD:

$$W_S = M_S S_S + t_S .$$

where M_S and t_S can be found using equation (5), then we find the structure vector S_R for the remaining set of matched corner points C_R by solving the equation

$$W_R = M_S S_R + t_S ,$$

where W_R is the measurement matrix corresponding to C_R . The position error from the model (M_S, t_S) is given by

$$D = W_R - (M_S S_R + t_S)$$

The standard variation of the error vector $\varepsilon = [d_i]$ is computed using the equation [4]

$$\sigma^0 = 1.4826 \left(1 + \frac{5}{n-p} \right) \text{median}(\varepsilon_i)$$

where n is number of points in C_R and p is the number of sample points (4 in our case). The number of samples of four sets of matched corners which have to be taken to assure that at least one consists only of inliers with probability higher than γ is given by

$$N_S = \frac{\log(1-\gamma)}{\log(1-(1-\pi)^p)}$$

where π is expected percentage of outliers in the sample. We find the sample with the lowest standard variance, and use the corresponding vector ε .

The next step is outlier rejection. For each point we determine the initial weight w_i .

$$w_i = \begin{cases} 1 & \text{if } \varepsilon_i \leq 2.5\sigma^0 \\ 0 & \text{if } \varepsilon_i > 2.5\sigma^0 \end{cases}$$

Then we compute the *robust* standard deviation estimate as

$$\sigma = \sqrt{\sum_i w_i \varepsilon_i^2 / \left(\sum_i w_i - p \right)}$$

Finally, outliers are found as points whose residual error is outside the confidence interval 2.5σ .

Matching between three images

By matching feature points between three images we can make use of the rigidity constraint. For each set of three frames, the corners are matched between the consecutive frames and the outliers rejected. Then, given the position of a feature in

any two images, and the motion parameters M and t , its position in the third image is uniquely determined. Hence, by using this procedure, we can detect those outliers which are “correctly” matched over each pair of consecutive images (satisfy the epipolar constraint) but do not satisfy the rigidity constraint over three frames.

3.3 New Feature Points

New feature points will be included in the measurement matrix if they appear and are matched in three consecutive frames. Consider frames $n-2$, $n-1$, n and $n+1$, and let C_1 denote a set of features which are matched over frames $n-2$ to n , while C_2 denotes the set of features which have been matched *only* over frames $n-1$ to $n+1$. Let P be an arbitrary feature from C_2 and let $p^{(n)}$ denote its image projection in the n^{th} frame. Since the motion (and structure) parameters are estimated over the first n frames, the structure parameter S of this point is found by solving the set of equations,

$$\begin{bmatrix} p^{(n-1)} \\ p^{(n)} \end{bmatrix} = \begin{bmatrix} M^{(n-1)} \\ M^{(n)} \end{bmatrix} S + \begin{bmatrix} t^{(n-1)} \\ t^{(n)} \end{bmatrix}.$$

Hence the initial parameters for the algorithm in Section 3.1 have been found.

Before we add the point from C_2 to the measurement matrix, and perform an update, we first check whether this point is genuinely new or whether it is an old feature that has reappeared. Let P_i be a previous feature that had disappeared and P_l be an arbitrary feature from C_2 . We can say that P_i and P_l represent the same feature if the world distance between them is small enough, *i.e.* if

$$d_{il} = \|\Delta S\| = \|S_i - S_l\| < \alpha \quad (9)$$

where α is an unknown threshold to be determined.

While we do not know what is *small* in the structure space (especially because it is only unique up to an affine transformation), we can say what is small in the image space (typically three to five pixels). If the features are close in structure space then the distance between their projections in the image plane has to be small but the reverse need not be true. We use the criteria that the two points are the same one if the distance between their image projections is less than threshold β for all camera locations. Mathematically, this condition can be expressed as

$$\max_{\Delta S, j} (\|p_i(j) - p_l(j)\|) \leq \beta. \quad (10)$$

The difference between image projections is given by $\Delta p = p_i - p_l = M\Delta S$ where the index j has been dropped for clarity. By applying the Euclidean norm to both sides we obtain $\|\Delta p\|_E \leq \|M\|_S \|\Delta S\|_E$ where the index S denotes the spectral norm, which is defined as a square root of the maximum eigenvalue of the matrix $M^T M$, *i.e.* $\|M\|_S = \sqrt{\max \lambda(M^T M)}$. Since the spectral norm is subordinate to Euclidean norm there exists at least one ΔS such that equality holds, and from equation (10) we get $\|\Delta p\|_E = \|M\|_S \|\Delta S\|_E \leq \beta$. If $\lambda_j = \|M(j)\|_S$, then by combining (9) and (10) the threshold $\alpha = \beta/\lambda$ where λ is the largest value of λ_j .

4. Experimental Results

4.1 POV image sequence

This artificially generated sequence consists of 90 frames of an object rotating around the fixed axes (see Fig. 1 a–c). The rotation is a constant four degrees per frame. The corners are detected and matched using an algorithm similar to [9]. This is a difficult sequence, because no single corner appears in all the frames, and because the number of points is small – it varies from 6 to 13 per frame. For this reason, both the [8] and [5] are not applicable to this sequence as the number of features is not constant. The VSDF algorithm may be used, but in its original form (without the reappearance of features) this will have a growing number of features as old features reappear, this will lead to slower computation and greater errors in the calculated parameters.

Fig. 1. POV sequence: 1st (a), 30th (b) and 60th (c) image in the sequence of 90 images. d) consistent features and recovered structure (the lines are shown for visualisation only).

Using our algorithm we obtained a total of 22 features, and by disregarding all of them which appeared in less than 20 frames we were left with total of 17 features which is quite close to the 16 corners on the object. Corners that have been effectively tracked over all the frames are shown on Fig 1d. As we can see they are all close to the true locations, including those which are currently occluded or not detected. Some distortion is present due to imperfection of the affine model, but the structure has been correctly recovered.

4.2 Hotel Sequence

This sequence consists of 197 points tracked over 181 frames. Corners were detected in the first image and their location in consequent images were determined using an optical flow technique [7] (see Fig. 2 a,b). The SM parameters were calculated using the SVD algorithm and our algorithm, which produced the same results (without outlier rejection), but the former took about an order of magnitude longer to compute. As in [3], we have found that updating motion parameters over all previous frames in each step is unnecessary. A very important step here is the outlier detection. Since we have matches over all frames, the rigidity constraint was checked not over last three frames only, but over the whole sequence. The comparison of mean squared error with, and without outlier removal is given in Fig. 2c.

5. Conclusion

This paper addresses the problem of structure and motion recovery. The first problem is speed and we presented a fast recursive algorithm that updates the *complete* motion and structure after each time step and have shown that it gives the same results as in

[8]. Furthermore, we have developed an outlier detection technique that decreased the error even further. The paper is also one of the first that deals the problem of feature dropout and reappearance. We have developed a procedure that that allows previously lost features to reappear in the sequence and demonstrated that it works in practice.

Fig.2 1st (a) and 181st (b) image from the hotel image sequence; c) Plots of the mean square error (cost function) with (lower curve) and without (upper curve) outlier rejection

Acknowledgments

Financial support for this project was provided by the Australian Research Council. The hotel image sequence was provided by the "Modeling by Videotaping group" in the Robotics Institute, Carnegie Mellon University.

REFERENCES

- [1] Held A., "Piecewise Shape Reconstruction by incremental factorization", *Proc. 7th British Machine Vision Conference*, vol. I, UK, 1996, pp. 333-342.
- [2] McLauchlan P. et al. "Recursive affine structure and motion from image sequences" *Proc. 3rd ECCV*, vol. I, Stockholm, pp. 217-224, 1994.
- [3] McLauchlan P. and Murray D.. "A unifying framework for structure and motion recovery from image sequences" *Proc. 5th ICCV*, MIT, pp. 314-320, 1995.
- [4] Meer P. et al. "Robust regression methods for computer vision: a review", *Intl. Journal Computer Vision* vol. 6., pp. 59-70, 1991.
- [5] Morita T. and Kanade T., "A sequential factorization method for recovering shape and motion from image streams", *Proc. 1994 ARPA Image Understanding Workshop*, vol. II, pp. 1177-1188, Monterey, CA1994.
- [6] Reid I. D. and Murray D. W., "Active Tracking of Foveated Feature Clusters Using Affine Structure", *Intl. J. Computer Vision*, vol. 18, pp. 41-60, 1996.
- [7] Shi J. and Tomasi C., "Good Features to Track". *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593-600.
- [8] Tomasi C. and Kanade T., "Shape and motion from image streams under orthography: A factorization approach", *Intl. J. Comp. Vision*, vol. 9, pp. 137-154, 1991.
- [9] Trajković M. and Hedley M., "Fast feature detection and matching for machine vision", *Proc. 7th British Machine Vision Conference*, vol. I, Edinburgh, UK, 1996, pp. 91-100.