Combining Two-view Constraints For Motion Estimation

Venu Madhav Govindu
Somewhere in India
venu@narmada.org

Abstract

In this paper we describe two methods for estimating the motion parameters of an image sequence. For a sequence of N images, the global motion can be described by $N-1$ independent motion models. On the other hand, in a sequence there exist as many as $\frac{N(N-1)}{2}$ pairwise relative motion constraints that can be solve for efficiently. In this paper we show how to linearly solve for consistent global motion models using this highly redundant set of constraints. In the first case, our method involves estimating all available pairwise relative motions and linearly fitting a global motion model to these estimates. In the second instance, we exploit the fact that algebraic (ie. epipolar) constraints between various image pairs are all related to each other by the global motion model. This results in an estimation method that directly computes the motion of the sequence by using all possible algebraic constraints. Unlike using reprojection error, our optimisation method does not solve for the structure of points resulting in a reduction of the dimensionality of the search space. Our algorithms are used for both 3D camera motion estimation and camera calibration. We provide real examples of both applications.

1. Introduction

Many of the available camera motion estimation algorithms that use feature correspondences can be roughly categorised into statistical or algebraic methods. The statistical methods typically define the likelihood function for given noise models and involve non-linear optimisation methods. For example, [1] and [7] involve minimisation of the distance between feature points and reprojected points for the estimated motion and structure in a least-squares sense. This optimal method is of course computationally expensive and inherits the problems of initialisation and convergence from the minimisation techniques used. On the other hand, the algebraic methods do have fast, linear solutions for two [11], three [6] and four [3] views. Some other linear methods for multi-frame motion estimation are [9], [8] where structure and motion are solved for simultaneously using a factorisation technique based on using rank constraints. However the estimation accuracy of such methods is significantly reduced by the presence of noise in the data.

In this paper we describe two methods that overcome the above mentioned limitations. Both the techniques we describe make efficient use of the existing redundancy of information in a sequence to derive fast and accurate motion estimates. These methods are based on the observation that an N image sequence is parametrised by $N-1$ independent motions, but there exist *as many as* $\frac{N(N-1)}{2}$ image pairs in this sequence, each of which provides a constraint on the global motion. Consequently, we have an overdetermined system of equations. In the rest of this paper we shall develop methods to exploit this information redundancy for motion estimation.

Our first method can be intuitively described using a three-dimensional rotation model although the same can be applied to $2D$ image motion models as well. For an N frame sequence, the global motion can be described by $N-1$ rotations. In a typical sequence, we can compute many more relative rotations between image pairs (upto a maximum of $\frac{N(N-1)}{2}$ which is typically much larger than N). Due to the presence of noise, the composition constraint on these two-frame motions (ie. $R_{ij} = R_{jk}R_{ki}$) will not be satisfied, ie. individual two frame motion estimates will not be "consistent" with each other. Given these motion estimates, we can now find a best-fit global motion model of $N-1$ rotations. As a result, we have a highly redundant system of equations that can be solved linearly to derive accurate global motion models. The inherent advantage of such a technique is that it effectively "averages" the various two-frame relative motions and thereby reduces the error due to any single pair.
The second method described in this paper directly combines the two stages of the first method into one algorithm. Each image pair provides two-view algebraic (ie. epipolar) equations that can be parametrised by $N-1$ motion models. This enables us to use an optimisation method that combines the redundant set of algebraic constraints across all image pairs and gives a solution with high accuracy. Since the use of epipolar constraints eliminates solving for the three-dimensional structure of the feature points, we greatly reduce the dimensionality of the search space,

resulting in faster algorithms.

In Section 2 we describe our two-stage "linear fitting" method for motion estimation. Section 3 describes the results of applying this method to motion estimation and camera calibration for real image sequences. In Section 4 we describe our second "direct estimation" method that combines the multiple two-view algebraic constraints and also characterise the performance of our methods.

2. Linear Fitting for Motion Estimates

In this section we develop our first solution for multi-frame motion estimation. For N images, there exist $N-1$ motions that we want to estimate. We denote the motion between frame i and the reference frame as M_i, and the relative motion between two frames i and j as M_{ij}. Hence we have the relationship $M_{ij} = M_j {M_i}^{-1}$. This relationship captures the notion of "consistency", ie. the composition of any series of transformations between frames i and j should be identical to M_{ij}. However, due to the presence of noise in our observations, the various transformation estimates would not be consistent with each other. Hence $\hat{M}_{ij} M_i \neq M_j$, where $\hat{M}_{ij}$ is the estimated transformation between frames i and j. However we can rewrite the given relationship as a constraint on the global motion model $\{M_1, M_2, \cdots, M_N\}$ which completely describes the motion. Since in general we have upto $\frac{N(N-1)}{2}$ such constraints, we have an overdetermined system of equations.

$$\hat{M}_{ij} M_i - M_j = 0, \forall i \neq j \tag{1}$$

which can be rewritten as

$$\begin{bmatrix} \cdots & \hat{M}_{ij} & \cdots & -I & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ M_i \\ \vdots \\ M_j \\ \vdots \end{bmatrix} = 0 \tag{2}$$

and solved linearly to give a consistent set of global motion estimates, $\{M_i\}$. Intuitively, we want to estimate $\{M_i\}$ that are most consistent with the measurements $\{M_{ij}\}$ in a least-squares sense. Thus the errors in individual estimates of $\hat{M}_{ij}$ are "averaged" out.

It may be noted that in Eqn. 1, we are not required to use every pairwise constraint to get a solution. For extended sequences, there is seldom any overlap between frames well separated in time, therefore their relative two-frame motions cannot be estimated. However we can still get a consistent solution as long as we have more than $N-1$ relative motions. This is a crucial difference between our algorithm and the factorisation based methods of [9] and [8] where points are required to be tracked throughout the entire sequence or the camera calibration method described by [5] which requires the estimation of the relative motion between every image pair. It may also be emphasised that our linear fitting method requires a set of relative motions and is independent of the method used to obtain these estimates. Therefore in principle we can use any method (ie. feature correspondences, flow or direct methods) for estimating the relative motions. It may be also noted that the individual motion estimates can be computed independently, ie. in parallel.

2.1. Rotation Estimation

In this subsection we describe the linear least squares solution for estimating three-dimensional rotation. Here the consistency relationship for rotations is $R_{ij} = R_j {R_i}^{-1}$. The error in individual relative rotations is modeled by a rotation about an arbitrary axis. This is represented by the matrix R_{error}, hence

$$\hat{R}_{ij} = R_j {R_i}^{-1} R_{error}, \tag{3}$$

where R_{error} represents a rotation of magnitude $\|\omega\|$ about the axis represented by the vector $\frac{\omega}{\|\omega\|}$ ([4]).
The linear solution can be stated as follows :

$$\hat{R}_{ij} R_i - R_j = 0 \tag{4}$$

However, since rotation matrices are constrained to $SO(3)$ whereas any linear method will result in a solution in $\mathcal{R}^9$ we have to rewrite the linear solution using a quaternion representation of rotations ($q = \{q_0, q_1, q_2, q_3\}$). The reader is referred to [4] for details on the quaternion representation. We denote the quaternion corresponding to R_i by q^i and the linear transformation representation of R_{ij} as Q_{ij}, ie. the relationship $R_{ij} R_i = R_j$ can be rewritten as $Q_{ij} q^i = q^j$, where

$$Q = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \tag{5}$$

Hence Equation 4 can be rewritten as $\hat{Q}_{ij} q^i - q^j = 0$ where $\hat{Q}_{ij}$ corresponds to the estimated matrix $\hat{R}_{ij}$. This system of equations can be solved linearly. We can also show that this least squares solution is optimal in the Maximum Likelihood sense.

Lemma 1 *For uniform, Gaussian distributed rotation error, the linear least squares solution for the rotation transformations is the Maximum Likelihood Estimate.*

Proof: We assume a uniform, Gaussian distribution for the rotation error, ω. This implies that the rotation errors are about axes that are randomly oriented and that the magnitude of the rotation error angle has a Gaussian distribution. For such a noise model, the optimal (Maximum Likelihood Estimate) solution is $\arg\ \min_{\mathcal{R}} \sum_{i,j} \|\omega_{ij}\|^2$ where $\mathcal{R}$ represents the consistent motion estimate, $\{R_1, R_2, \cdots, R_N\}$. Using Equation 5, the linear system of equations can be rewritten as

$$Q_{ij} Q_{error} q^i - q^j = \epsilon_{ij} \tag{6}$$

where ϵ_{ij}'s are the residuals of the fit.
Since the estimation error is modeled by a small rotation, using a Taylor series expansion, we have $R_{error} \approx I + [\omega]_\times$, where ω represents the error in the estimate (here $[.]_\times$ denotes the cross-product matrix, ie. $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}$). The equivalent quaternion representation is $q = [1, \omega_1, \omega_2, \omega_3]$. Hence, we have

$$Q_{error} = \begin{pmatrix} 1 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 1 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 1 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 1 \end{pmatrix} \tag{7}$$

Therefore we can rewrite Eqn. 6 as,

$$Q_{ij} \begin{pmatrix} 1 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 1 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 1 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 1 \end{pmatrix} q^i - q^j \quad = \quad \epsilon_{ij} \tag{8}$$

Now since $Q_{ij} q^i - q^j = 0$, we can remove the corresponding terms in Eqn. 8. This results in the relationship

$$Q_{ij} \begin{pmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 0 \end{pmatrix} q^i \quad = \quad \epsilon_{ij} \tag{9}$$

Since the norm of a quaternion is 1, by carrying out the multiplication in 9, we have $\sum_{i,j} \|\epsilon_{ij}\|^2 = \sum_{i,j} \|\omega_{ij}\|^2$. Hence the least squares error of Equation 6 is the Maximum Likelihood Estimate.

2.2. Translation Estimation

In the case of translation estimation, the consistency equations will be of the form $T_{ij} = T_j - R_{ij} T_i$. However for two frames the inter-frame translation estimates are known only upto a scale factor (ie. we know only the translation direction, t_{ij}). Hence we have equations of the form $t_{ij} = \lambda_{ij}(T_j - R_{ij} T_i)$ where λ_{ij}'s are unknown scale factors. However we can utilise the cross-product relationship, $t_{ij} \times (T_j - R_{ij} T_i) = 0$. This cross-product constraint can also be described as $[t_{ij}]_\times (T_j - R_{ij} T_i) = 0$. Hence we have a linear system of equations that can be solved to estimate the translations between different frames and the reference frame. An interesting outcome of such a method is that the linear system of equations enables us to recover three-dimensional translations from only the heading directions!

We model the error in translation direction estimation by a small rotation of the true translation direction, ie. $\hat{t}_{ij} = R_{error} t_{ij}$ where R_{error} is a small rotation represented by ω. Here the rotation axis represented by ω has to lie in the subspace orthogonal to t_{ij}. Similar to our assumption for rotation error, we assume that the error in translation direction is modeled by rotation vector ω that is uniform, Gaussian distributed in the subspace orthogonal to t_{ij}.

Using a first-order Taylor approximation for rotation $R_{error} \approx I + [\omega_{ij}]_\times$, each linear equation can be written as

$$\begin{aligned} [\hat{t_{ij}}]_\times (T_j - R_{ij} T_i) &= 0 \\ \Rightarrow [(I + [\omega_{ij}]_\times) t_{ij}]_\times (T_j - R_{ij} T_i) &= 0 \end{aligned} \tag{10}$$

Now we note that $t_{ij} \times t_{ij} = 0$ and $\|(\omega_{ij} \times t_{ij}) \times t_{ij}\| = \|\omega_{ij}\|$, since $\omega_{ij} \perp t_{ij}$, which implies that $(\omega_{ij} \times t_{ij}) \perp t_{ij}$. Therefore, for the residual error in each equation of 10, we have

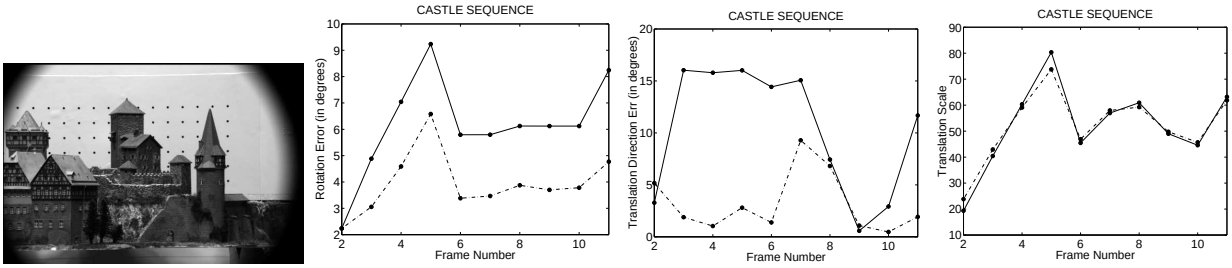
$$\frac{1}{\lambda_{ij}} (\omega_{ij} \times t_{ij}) \times t_{ij} = \epsilon_{ij} \tag{11}$$

$$\Rightarrow \sum_{i,j} \|\epsilon_{ij}\|^2 = \sum_{i,j} \|\frac{1}{\lambda_{ij}} \omega_{ij}\|^2 \tag{12}$$

Therefore, the least squares solution results in unequal weighting of the error terms. While this solution may be sufficiently accurate, we can further refine the solution by an iterative, weighted least squares method as described below.

For notational convenience, we will drop the subscripts ij. λ^n indicates the weights at iteration n.

- Initialise scalar weights $\lambda^0 = 1$
- At step n, solve $[t_{ij} \lambda^{n-1}]_\times (T_j - R_{ij} T_i) = 0$



(a) One frame (b) Rotation Error (c) Translation Dir Error (d) Translation Scale

Figure 1: (a) shows one frame from the Castle sequence. (b) and (c) show the error in rotation and translation direction. Solid line indicates the baseline algorithm, dotted line indicates our linear fitting method. (d) shows the recovered translation scale by our linear fitting method (indicated by dotted line). Ground truth is shown in solid line.

- Update $\lambda^n = \frac{1}{\|T_j - R_{ij} T_i\|}$
- Repeat till convergence

For the above iterative scheme we have empirically observed that convergence is achieved in about $3-4$ iterations. Hence the additional computational load is insignificant. Also at step n in the iterative scheme defined above, the least squares error is

$$E = \sum_{ij} \| \frac{\lambda^n}{\lambda^{n-1}} \omega_{ij} \|^2 \tag{13}$$

At the minimum of the objective function after convergence, we have the condition $\lambda^n = \lambda^{n-1}$. This implies that $\sum_{i,j} \| \frac{\lambda^n}{\lambda^{n-1}} \omega_{ij} \|^2 = \sum_{i,j} \|\omega_{ij}\|^2$. Therefore, the least square error is identical to the error attained by the optimal solution.

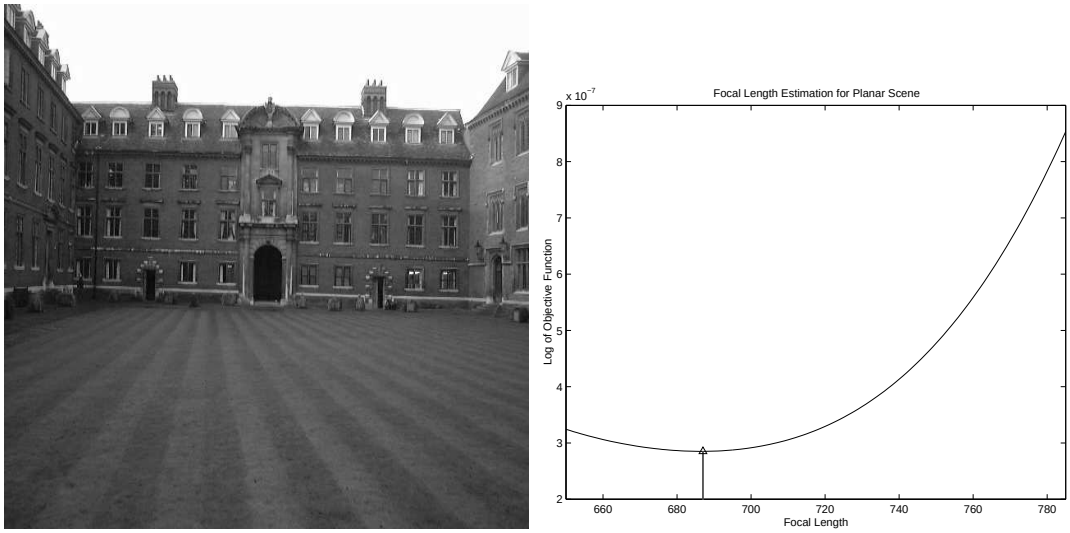
3. Applications

The method described in the preceding sections is quite flexible and can be applied to a wide variety of scenarios. Since we are able to compute 3D rotation and translation, our techniques can be applied to computing 3D motion, camera calibration and also estimating the structure of the scene being viewed. In this section we provide real examples of motion estimation and camera calibration to illustrate the performance of our method.

3.1. 3D Motion Estimation

In essence our algorithm uses the same information that a standard two-frame method uses but combines the many possible constraints to improve estimation. To characterise the improvement in performance obtained, we use data sets for which ground truth is available so as to be able to quantitatively characterise the improvement obtained. An empirical comparison of our method is shown in Section 4.1. Here we illustrate the results on the familiar CASTLE sequence for which ground truth data is available. To solve for the inter-frame rotation and translation directions, we compute the essential matrices between different images in this sequence using the eight-point algorithm [2] and decompose these matrices into the corresponding rotations and translation directions. Thereafter the global motion models are computed by linear fitting as described in the preceding sections. The errors in rotation and translation direction estimation are shown in subfigures (b) and (c) of Figure 1. We compare the performance of our algorithm with a baseline method which computes the relative motion between each frame and the reference frame (ie. frame 1). Thus any improvement in performance over the baseline method is directly attributable to the exploitation of information redundancy by our method. We can see a significant gain in performance of our method (indicated in dotted line) over the baseline method (indicated in solid line). The comparison of the recovered translation scale with the ground truth is shown in (d) and can be seen to be quite accurate (It is not possible to compute the translation scale in the baseline case).

In this example we can observe that our algorithm significantly improves performance by efficiently exploiting the available data redundancy. We would also like to point out that this is a particularly difficult sequence for linear methods to work on given the large errors in the individual two-frame estimates (as can be seen from the baseline errors). However, our method affords significant improvement in performance inspite of the fact that its input values are estimates that are significantly erroneous.



(a) Building Facade

(b) Focal Length Estimation

Figure 2: (a) shows one frame from the set of 4 images used. (b) shows the objective function obtained for camera focal length estimation. The estimated focal length (687) is indicated by the arrow.

Intuitively, the accuracy of our method can be explained by the fact that the individual errors between image pairs have different orientations. Since our methods combines these estimates, these individual errors are "averaged" in the parameter space resulting in greater accuracy.

3.2. Camera Calibration

The above analysis had assumed a calibrated camera. However we can use our rotation fitting method to also compute the camera calibration parameters. We will illustrate our method using homographies although it can apply equally well to the case of epipolar geometry described by the essential/fundamental matrices. When the scene being viewed is planar, the relationship between feature correspondences across images can be described by a collineation [5], $\mathbf{p}_i \propto \mathbf{G}_{ij}\mathbf{p}_j$, where $\mathbf{p}_i$ and $\mathbf{p}_j$ represents the projections of feature points on images i and j and $\mathbf{G}_{ij}$ represents the collineation between the two images. Further if the camera calibration is represented by a matrix $\mathbf{K}$, the collineation is related to the inter-image homography by the relationship, $\mathbf{G}_{ij} \propto \mathbf{K}\mathbf{H}_{ij}\mathbf{K}^{-1}$. Now for a given calibration matrix $\mathbf{K}$, we can extract the various homographies. In turn these homographies can be decomposed into rotation-translation pairs. We use the method described by [10] for this purpose. In general the decomposition results in two possible motion representations which are mutually indistinguishable. However given the many image pairs, it is possible to select the correct rotation-translation decompositions by looking for the best possible consistency over various subsequences. Thus the two-fold ambiguity due to the homography decomposition can also be easily resolved. Consequently, we can extract the various inter-frame rotations $\mathbf{R}_{ij}$ in this manner.

The set of inter-frame rotations extracted forms the basis for our calibration method. It may be observed that using an incorrect calibration matrix will introduce errors in the homographies estimated from the collineations (since $\mathbf{H} = \mathbf{K}^{-1}\mathbf{G}_{ij}\mathbf{K}$). This in turn implies that the rotation matrices extracted will also be erroneous. Consequently, the set of erroneous rotation matrices will result in a bad global fit, ie. a large residual error for the Maximum Likelihood Estimate described in Section 2.1 ($\sum_{i,j}\|\omega_{ij}\|^2$). Thus the residual error for rotation fitting can be used as an objective function for camera calibration estimation.

$$\mathbf{H}_{ij} \propto \mathbf{K}^{-1}\mathbf{G}_{ij}\mathbf{K}$$
$$\mathbf{H}_{ij} \; decomposes \; to \; \{\mathbf{R}_{ij}\} \Rightarrow \{\mathbf{Q}_{ij}\}$$
$$Objective \; function : \min_{\mathbf{K}} \sum_{i,j} \|\mathbf{Q}_{ij}\mathbf{q}^i - \mathbf{q}^j\|^2$$

We summarise the above as follows. For a given estimate of camera calibration, the homography matrices are obtained

and the rotation matrices are obtained and disambiguated. Thereafter a global fitting of the rotation matrices is carried out and the residual error for this fit is the objective function. Using this objective function the camera parameters are estimated by means of standard minimisation routines.

In our experiments we used the data set described in [5] (See Figure 2). Point correspondences obtained from 4 images of a building facade (planar) are used to compute the camera parameters. Since the location of the principal point does not have a significant impact on the reconstruction accuracy, we focus on estimating the focal length for this set of images. We fix the principal point at the image center and carry out an optimisation for the focal length. We obtain a focal length estimate of 687 which is in excellent agreement with the result of 678 pixels reported in [5] for the same data set. Optimising over 3 parameters, ie. focal length and the x and y co-ordinates of the principal point results in very similar results. The objective function for camera focal length is shown in Figure 2 (b) and the solution obtained is indicated by an arrow. The accuracy of the estimated focal length using only 4 images illustrates the efficacy of our method in estimating camera calibration and the three-dimensional motion of the sequence.

4. Direct Estimation of Motion Parameters

In the algorithms described above we have used a two-stage process. Firstly, image data is used to compute the motion parameters and subsequently these motion parameters are "averaged", ie. the process of averaging the various estimates takes place in the parameter space. In this section we develop an alternative method which exploits the available information redundancy by directly computing the motion parameters while using the multiple algebraic constraints provided by the different image pairs available in a sequence.

If we consider the image pair (i, j) the algebraic (epipolar) constraint is given by

$$\mathbf{p}_j{}^T[\mathbf{t}_{ij}]_\times \mathbf{R}_{ij}\mathbf{p}_i = 0 \tag{14}$$

However we note that the relative rotation and translation between images i and j can be parametrised in terms of the global motion model. In other words, we can rewrite the motion parameters as

$$\begin{aligned} \mathbf{R}_{ij} &= \mathbf{R}_j\mathbf{R}_i{}^{-1} \\ \mathbf{t}_{ij} &\propto \mathbf{T}_j - \mathbf{R}_j\mathbf{R}_i{}^{-1}\mathbf{T}_i \end{aligned} \tag{15}$$

By combining Eqn. 14 and 15, we can derive an objective function for global motion estimation that exploits all possible epipolar constraints. This can be rewritten as

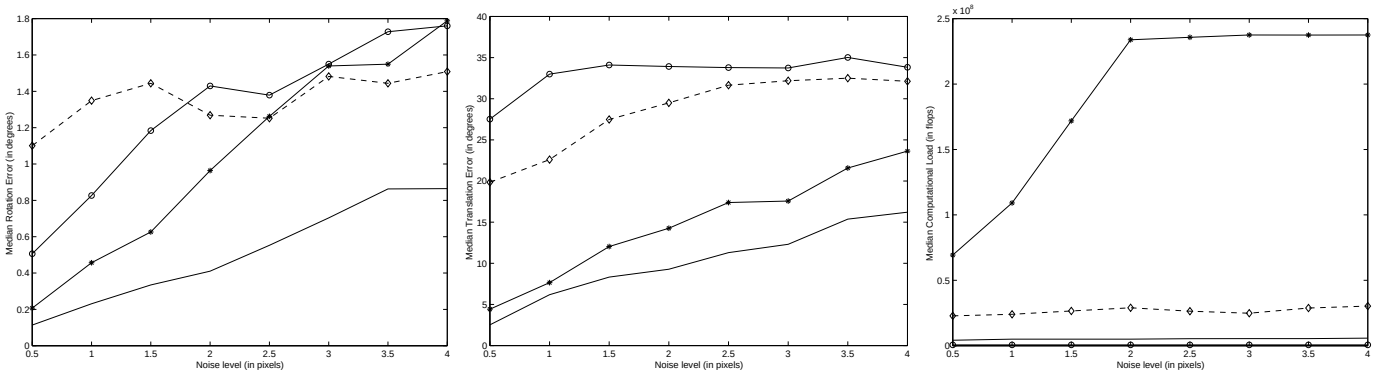
$$\arg \min_{\{\mathcal{R},\mathcal{T}\}} \mathbf{p}_j{}^T[\mathbf{T}_j - \mathbf{R}_j\mathbf{R}_i{}^{-1}\mathbf{T}_i]_\times \mathbf{R}_{ij}\mathbf{p}_i, \forall i \neq j \tag{16}$$

Here the minimisation is over the global motion parameters, $\mathcal{R} = \{\mathbf{R}_1, \cdots, \mathbf{R}_N\}$ and $\mathcal{T} = \{\mathbf{T}_1, \cdots, \mathbf{T}_N\}$ [1]. Thus by combining the epipolar constraints from all possible image pairs, we avoid preferring a reference frame. In contrast to two-frame methods, our method exploits a high degree of redundancy since we have upto $\frac{N(N-1)}{2}$ constraints for estimating $N-1$ motion models. This averaging of the errors in the various epipolar constraints resulting in a more accurate motion estimate. In contrast with the use of reprojection error for solving for structure-from-motion (henceforth referred to as the reprojection method), the epipolar constraints are obtained by eliminating structure. This results in a large reduction of the dimensionality of the search space. Eg., if we use 5 images and 20 feature correspondences, our direct optimisation method uses a search space of 24 dimensions, since we solve for $(5 - 1 =)$ 4 global motion models that consist of 3 rotation and 3 translation parameters each. In contrast, the reprojection methods uses $(20 \times 3 =)$ 60 additional unknowns to represent the three-dimensional structure of the points resulting in an 84 dimension space over which the objective function is optimised. While our method is obviously sub-optimal (by dint of the elimination of the structure variables), it affords the advantage of a reduced dimensional search space thus allowing for faster algorithms. This is especially important in the case of a large number of feature correspondences being used.

4.1. Experimental Evaluation

In this subsection we describe the experiments used to empirically evaluate our two methods described above and compare them with the reprojection method. In each trial run, we generate a sequence of 5 images of 20 points each. The scene consists of 20 points randomly located within a 60° field-of-view. The camera is subjected to random rotations and translations. White Gaussian noise of different levels is added to the projected image points. In our experiment we are interested in comparing the performance of both our algorithms with that of solving for structure-from-motion using reprojection error as an optimisation metric. Since our methods do not use the three-dimensional structure in solving for the motion as opposed to the reprojection method we define a new baseline for comparison. For a baseline comparison, we solve for the motion of the

[1] For notational convenience, we shall call this method the direct optimisation method.



(a) Rotation Error (b) Translation Direction Error (c) Computational Load (in fbps)

Figure 3: Median rotation and translation direction errors over 50 trials for different noise levels (equivalent to pixels in a 256×256 image). The errors are shown in degrees. The baseline performance is shown in solid line, the linear fitting of individual estimates is shown in circles and lines. Our direct optimisation method is shown in diamonds and dashes and the reprojection method is shown using stars and lines.

camera using reprojection error *but* use the known three-dimensional structure instead of solving for it as an additional unknown. In other words we use the optimal reprojection error metric but solve only for the camera motion using a known structure. As a result, this baseline method establishes a lower bound since neither our algorithms nor the reprojection method can perform better than this bound [2].

Figure 3 shows the performance of the three methods at different noise levels. We represent the root mean square value of the rotation error, error in translation direction and the computational load for each method. All values shown are the median values for 50 trials. We choose to show the computational load of each method alongwith its performance since this would constitute a more accurate characterisation of each method instead of just comparing accuracy. An accurate estimate that is computationally expensive is not always desirable compared to a faster, sub-optimal one. Both optimisations (ie. our direct optimisation method and the reprojection method) are implemented using MATLAB's implementation of Levenberg-Marquardt least-squares minimisation. While one can improve the performance of these optimisation routines by exploiting their special structure, we believe that the current computational values are a fair reflection of the comparative performance of these methods.

As can be seen in Fig. 3 (a), the rotation estimates obtained show that while the reprojection method is somewhat better for low noise levels than our algorithms, all three of them are virtually indistinguishable for moderate or high noise values. On the other hand, one significant difference that is observed in the case of translation estimation (Fig. 3 (b)) is that the reprojection method is substantially better than either of our methods. This difference is attributable to the fact that both of our methods work by eliminating structure from the projection equations and solve for camera motion only. This process of elimination results in biased estimation of the camera translation directions. While this is a limitation of our method in comparison to the global optimisation method using reprojection error, we note the significant difference in computational load as observed in Fig. 3 (c). Our linear estimation method of Section 2 has a fixed computational load and is orders of magnitude lower than than of the reprojection method (on the average its less than 1% of the load of the reprojection method). Similarly, the computational load of our direct optimisation method is about 10 times smaller than the reprojection method. This difference in computational load would be more pronounced if one were to use more images and more feature points. It may also be noted that for moderate and high noise levels, the computational load for the reprojection method is bounded by the maximum limit that we set for the optimisation routine, otherwise it would take even longer to converge. In other words we should use our algorithms to solve for the camera motion when we have limited computational time available, ie. in the case of real-time systems where there are hard time constraints. In fact when the time constraints are severe, our linear method should be used since

[2] It must be noted that this baseline method is defined for comparison purposes only since in practice we will seldom have knowledge of the real three-dimensional structure.

it guarantees a solution in a fixed, *known* amount of time.

5. Conclusion

In this paper we have introduced two methods for computing the motion between images of a sequence. Both of our methods efficiently exploit the information redundancy of pairwise motions to give accurate estimates for the motion of the entire sequence. In comparison with non-linear methods, our methods are extremely fast and result in accurate motion estimates. Algorithms similar in spirit have also been developed for 2D image motion models and show improved accuracy but are not described here due to space constraints.

6. Acknowledgements

The author would like to thank John Oliensis and David Jacobs of NECI for their support and one of the anonymous reviewers for useful comments regarding Section 4. The image data used in Fig. 2 was provided by Ezio Malis.

References

[1] Hartley, R., "Euclidean Reconstruction from Uncalibrated Views", *Proceedings of the DARPA-ESPIRIT Workshop on Applications of Invariance in Computer Vision*, pp. 187-202, 1993.

[2] Hartley, R., "In Defence of the 8-point algorithm", *Proceedings of the 5th International Conference on Computer Vision*, IEEE Computer Society Press, pp. 1064-1070, 1995.

[3] Hartley, R., "Computation of the Quadrifocal Tensor", *Proceedings of the 5th European Conference on Computer Vision*. pp.20-35, 1998.

[4] Kanatani, K., *Group-Theoretical Methods in Image Understanding*, Springer-Verlag, 1990.

[5] Malis, E., and Cipolla, R., "Multi-view constraints between collineations : application to self-calibration from unknown planar structures," *European Conference on Computer Vision*, 2000.

[6] Shashua, A., " Trilinear Tensor: The Fundamental Construct of Multiple-view Geometry and its Applications," *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC97)*, 1997.

[7] Szeliski R. and Kang S. B., "Recovering 3D Shape and Motion from Image Streams using Nonlinear Least Squares," *Journal of Visual Communication and Image Representation*, vol. 5, pp. 10–28, 1994.

[8] Sturm P., and Triggs, B., "A Factorization Based Algorithm for Multi-Frame Projective Structure and Motion", *Proceedings of the 4th European Conference on Computer Vision*, pp 709-720, 1996.

[9] Tomasi, C. and Kanade, T., "Shape and Motion from image streams under orthography : A factorization method," *International Journal of Computer Vision*, vol. 9(2), pp. 137-154, 1992.

[10] Triggs, B., "Autocalibration from planar scenes," *Proceedings of the 5th European Conference on Computer Vision*, pp 89-105, 1998.

[11] Zhang Z., "Determining the Epipolar Geometry and its Uncertainty: A Review", *INRIA Research Report No. 2927*, July 1996.