Consistency Models for Motion and Calibration Estimation

Venu Govindu
venu@davinci.netvista.net *

Abstract

In this paper we introduce a fast, linear method for estimating the motion parameters of an image sequence. For a sequence of images, a redundant set of two frame motion estimates can be computed. Due to the presence of noise, different two frame motion estimates will not be consistent with each other. However, by using the redundant set of pairwise motion estimates, we can enforce consistency and solve for the global motion. Our algorithm flexibly utilises all the information available in the sequence in a linear manner. It is fast and accurate and can be implemented both in a batch or recursive formulation. Our method is applicable to both three-dimensional global motion and two-dimensional image motion models. Examples of results on real data are provided to validate our method for 3D camera motion estimation and 2D image mosaicing. In this paper, we also describe the use of our measure of motion consistency in recovering the calibration (focal length) of an image sequence. Focal length estimation is posed as a one-dimensional search using the quality of fit for the rotation estimates. Real examples of camera calibration are included to illustrate the good performance of our algorithm in recovering the required focal length.

1 Introduction

In this paper we examine the problem of estimating 2D and 3D motion models and focal length for an image sequence. Before introducing our method for motion estimation, we will briefly describe the existing literature that is directly related to our approach. For point features, fast, linear methods exist for estimating motion in the case of two [12], three [6] and four [4] views. Some linear methods for multi-frame motion estimation are [11], [10] where structure and motion are solved for simultaneously using a factorisation technique based on the SVD. However, this depends on being able to track features across the entire sequence.

For point features, the optimal solution (MLE) is defined to be the minimisation of the distance between feature points and reprojected points for the estimated motion and structure in a least-squares sense (eg [2], [9]). This minimisation is carried out using the Levenberg-Marquardt gradient-descent method which is non-linear and is typically very slow. Moreover, an appropriate initialisation is required to ensure convergence. On the other hand, the method described in this paper is fast (since it is linear) and although it is not optimal, it gives reliable results by exploiting all the available redundancy of information in a sequence. Also, as will be clear, our method is not limited to feature correspondences and can be applied in any situation where we can get two-frame estimates.

In this paper we describe a method that uses multiple two-frame motion estimates and computes a linear fit of these estimates to obtain a globally consistent motion description. We develop the intuitive idea of our algorithm using a three-dimensional rotation model. Consider the case of 3 frames shown in Fig. 1 (a). As is obvious, if we start at frame i and return to it via frames j and k, we have the requirement that $R_{ki}R_{jk}R_{ij} = I$, ie. the composition of all the transformations is an identity since we have returned to our original frame of reference. This is the notion of "consistency", ie. the individual pairwise transformations are consistent with each other.

In the general case, we have an N frame sequence, and hence we need to estimate N rotation matrices with respect to a reference frame. However, due to the presence of noise in the data, individual pairwise estimates will be erroneous, therefore the composition constraint on two frame motions (ie. $R_{ik} = R_{jk}R_{ij}$) will not be satisfied. In other words, two frame motion estimates are not "consistent" with each other. However as will be clarified in the next section, each one of these equations provides a constraint on the global motion estimates. Moreover, we note that in a sequence of N frames, there are more than N pairwise rotations that can be estimated (upto a

*This research was conducted while the author was at the NEC Research Institute, Princeton, USA.

maximum of $\frac{N(N-1)}{2}$ in the case where the relative rotation between every pair can be estimated), which provide a redundant system of linear equations for the global motion wrt a reference frame. Since two frame motion estimation is linear (for point features) and the system of equations is also linear, the overall algorithm is very fast compared to the non-linear, optimal method. Also, since we use the inherent redundancy of information in a sequence, the estimation is reliable and accurate. Hence, our method is linear and consistent. In Section 2 we describe the general framework of our approach to motion estimation. In subsections 2.1 and 2.3, we develop the solution for the multi-frame estimation of three-dimensional rotation and translation respectively. Section 3 details the results of tests on real image sequences. In Section 5 we describe the solution for the linear motion models between images (ie. Affine and Projective). In Sec. 6 we provide real image examples to illustrate the improvement in performance achieved by using our method.

We also observe that our measure of consistency can be used to recover camera calibration. The algorithm is based on the idea that for erroneous estimates of camera calibration parameters, the individual pairwise rotation estimates will also be erroneous. This will be reflected in the residual error of the least squares fit for global motion. The larger the error in calibration (say focal length), the worse the fit. In Section 4, we demonstrate the efficacy of our algorithm in recovering the focal length for real image sequences.

2 Consistency of Motion Estimates

In this section we develop our solution for multi-frame motion estimation. As mentioned in Section 1, for N images, there exist N motions (wrt a reference frame) that we want to estimate. We denote the motion between frame i [1] and the reference frame as M_i, and the relative motion between two frames i and j as M_{ij} (See Fig. 1 (b)). Hence we have the relationship

$$M_{ij} = M_j {M_i}^{-1} \tag{1}$$

Due to the presence of noise in our observations, the transformation estimates would not be consistent. 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 Eqn. 1 in the form of a constraint on the global motion model [2] , ie. $\hat{M}_{ij} M_i - M_j = 0$. In general there are upto $\frac{N(N-1)}{2}$ such constraints on the N motion models. Hence we have an overdetermined system of equations

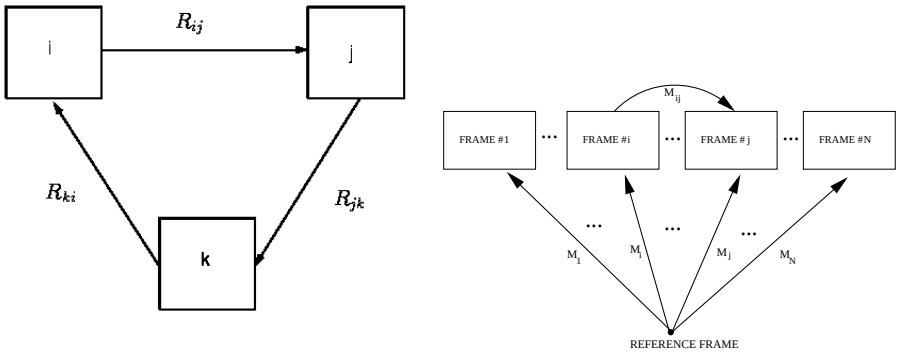
$$\hat{M}_{ij} M_i - M_j = 0, \forall i \neq j \tag{2}$$

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 in such a linear system of equations. Such an averaging ameliorates the situation when certain individual pairwise estimates have higher amounts of error. In our case, since we are deriving a least squares fit for a redundant system of equations, the errors will be forced to redistribute in a manner such that the higher error in this equation will be corrected to a significant extent. In contrast, the traditional method cascades pairwise transformations between adjacent pairs, ie. when transforming to the reference frame we have $M_k = M_{(k-1)k} \cdots M_{23} M_{21}$. Such a composition of the transformation uses the minimal set of constraints and does not exploit the redundancy of information that is available in a sequence. Consequently, any individual error will affect all subsequent estimates.

We would like to emphasise that we are not required to use every pairwise constraint to get a solution. For extended sequences, there is seldom overlap between frames well separated in time, therefore their relative two-frame motions cannot be estimated. However we can still get a consistent solution by utilising all the pairwise motions that can be estimated in such a long sequence (typically there would be overlap between images within some distance to each other). This is a crucial difference between our algorithm and the factorisation based methods of [11] and [10]. In their case, points are required to be tracked throughout the entire sequence (which is seldom possible for extended sequences), but we are not constrained by such a requirement since enough pairwise estimates are computable in a long sequence resulting in a redundant system. In other words, our method can utilise all available redundancy to give reliable results. However, obviously in Eqn. 2, the estimate accuracy depends on the amount of redundancy used, ie. the number of constraint equations employed to obtain a global solution.

[1] Henceforth, we shall use the term "frame" to denote a coordinate frame attached to a given image

[2] The set of estimates $\{M_1, M_2, \cdots, M_N\}$ will be referred to as the global motion model



(a) (b)

Figure 1: (a) The individual transformations are "consistent" if $R_{ki}R_{jk}R_{ij} = I$. (b) The relative motions M_{ij} are estimated from the data. The global motion $\{M_1 \cdots M_N\}$ wrt a reference frame is estimated by least squares fitting of the relative motions.

Moreover, to compute the motion between frame i and j, we do not require that the points be visible (and tracked) throughout the subsequence between frames i and j. As long as we can find enough matched features (tracked or otherwise) between images, we can compute the required two-frame motion estimate. Therefore we are not constrained to using only sequences where the motion between adjacent frames is small to enable tracking and can solve for motion parameters of sets of images acquired from arbitrary viewpoints. Moreover, in the most general case, we are not even constrained to using point (or discrete) features. The input to our algorithm is a redundant set of pairwise motion estimates. Each of these estimates can potentially be obtained using any method (optical flow, direct methods etc.)! [3] In this context, it must be emphasised that our method is orders of magnitude faster than the optimisation solutions like the Levenberg-Marquardt minimisation. Moreover our linear method optimises in the parameters space instead of the image space (as is the case with the optimal solution) and hence the computations can be carried out independent of the three-dimensional structure of the scene being view. In contrast, most non-linear techniques have to optimise in both structure and motion resulting in high computational costs since we have the additional burden of estimating the three-dimensional structure of the visible scene.

If for every frame we use a fixed number of previous frames to compute the two-frame relative motions, the linear system of equations can be recast in a recursive formulation allowing for real-time computation of motion. Eqn. 2 can be rewritten as

$$P_n m_n = 0 \tag{3}$$

where P_n is a matrix containing the terms M_{ij} and m_n is a column vector denoting the global motion parameters being estimated, ie. $\{M_1, M_2, \cdots, M_n\}$ for n frames. Now if the maximum distance between overlapping frames is k, ie. $|i - j| \leq k$, for computing the motion for frame $n + 1(M_{n+1})$, we can hold all previous motion estimates fixed and recompute the least squares solution. In this case, the new system of equations to be solved is $P_{n+1}m_{n+1} = 0$ where the only unknown is M_{n+1}. Since, we are solving for the least squared value of the residual error, we have a quadratic equation in M_{n+1}. Such a recursive scheme would have a fixed computational load per frame.

2.1 Rotation Estimation

In this section we describe the linear least squares solution for three-dimensional rotation. We represent the relative rotation between frames i and j as R_{ij}. Therefore the consistency relationship is $R_{ij} = R_j {R_i}^{-1}$. The error in rotation estimates can be modeled by a rotation about an arbitrary axis. This is represented by the matrix R_{error}, hence

[3] However, the choice of algorithm for the two-frame case will affect the total computation time.

$$\hat{R}_{ij} = R_j {R_i}^{-1} R_{error}, \tag{4}$$

where R_{error} represents a rotation of magnitude $\|\omega\|$ about the axis represented by the orientation of the three-dimensional vector $\frac{\omega}{\|\omega\|}$ ([5]). Thus the linear solution can be stated as follows :

$$\hat{R}_{ij} R_i - R_j = 0 \tag{5}$$

Clearly, the 9 elements of the rotation matrix have only 3 degrees of freedom (ie. $R \epsilon SO(3)$). However, if we were to directly solve for the least squares solution, our solution would be an element of $\mathcal{R}^9$. Hence the linear solution cannot be directly computed using the row or column ordered representation of the rotation matrix. However, the correct linear solution can be computed using the quaternion representation.

2.2 Quaternion Representation

Any three-dimensional rotation transformation can be uniquely represented by a four-dimensional quaternion $\mathbf{q} = \{q_0, q_1, q_2, q_3\}$, where $\mathbf{q} \epsilon S^3$, ie. it is constrained to have a norm equal to 1. Therefore ${q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 = 1$ [5].

The quaternion representation of a product of two rotation matrices is a linear transformation of the elements of the quaternion representations of the two matrices. If 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 $R_{ij} R_i = R_j$ is represented as $Q_{ij} q_i = q_j$, where

$$Q = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \tag{6}$$

Hence Equation 5 can be rewritten as $\hat{Q} ij q_i - q_j = 0$ where $\hat{Q}_{ij}$ is the matrix in Equation 6 corresponding to $\hat{R}_{ij}$. This system of equations can be solved linearly. We can also show that this linear 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 3D rotation represented by the Euler angles, 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 (within the range $[0, \pi)$). For such a noise model, the optimal (Maximum Likelihood Estimate) solution is

$$\arg \min_M \sum_{i,j} \|\omega_{ij}\|^2 \tag{7}$$

where M represents the consistent motion estimate, $\{R_1, R_2, \cdots, R_N\}$.

Using Equation 6, the linear system of equations can be rewritten as

$$Q_{ij} Q_{error} q^i - q^j = \epsilon_{ij}, \tag{8}$$

where Q_{ij} and Q_{error} are the linear transformations associated with rotations R_{ij} and R_{error}. Since the estimation error is modelled 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 $q = [1, \omega_1, \omega_2, \omega_3]$. Therefore, we have

$$Q_{error} = \begin{pmatrix} 1 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 1 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 1 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 1 \end{pmatrix} \tag{9}$$

where ϵ_{ij}'s are the residuals of the fit. Now since $Q_{ij} q_i - qj = 0$, we can remove the corresponding terms in Eqn. 9. Therefore, 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 8 is equal to the quantity minimised in Equation 7. Therefore the linear solution is the Maximum Likelihood Estimate.

2.3 Translation Estimation

In the case of translation estimation, the consistency equations will be of the form $T_{ij} = T_j - T_i$. 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

$$t_{ij} = \lambda_{ij} (T_j - T_i) \tag{10}$$

where λ_{ij}'s are unknown scale factors. But we can utilise the cross-product relationship, $t_{ij} \times (T_j - T_i) = 0$. This cross-product can also be described as

$$[t_{ij}]_\times (T_j - T_i) = 0 \tag{11}$$

Hence we have a linear system of equations that can be solved to estimate the translations between different frames and the reference frame. It may be noted that such a linear system of equations enables us to recover three-dimensional translations from only the translation directions [4].

Since the translation estimates are corrupted due to the presence of noise in the observations, we have to model the effect of noise on the translation direction estimates. 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}.

Since $R_{error} \approx I + [\omega_{ij}]_\times$, each linear equation can be written as

$$[\hat{t_{ij}}]_\times (T_j - T_i) = 0 \tag{12}$$
$$\Rightarrow [(I + [\omega_{ij}]_\times) t_{ij}]_\times (T_j - T_i) = 0$$

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 12, we have

$$\frac{1}{\lambda_{ij}} (\omega_{ij} \times t_{ij}) \times t_{ij} = \epsilon_{ij} \tag{13}$$
$$\Rightarrow \sum_{i,j} \|\epsilon_{ij}\|^2 = \sum_{i,j} \|\frac{1}{\lambda_{ij}} \omega_{ij}\|^2 \tag{14}$$

Therefore, the least squares solution for Equation 11 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, compute the least squares solution of

$$[t_{ij} \lambda^{n-1}]_\times (T_j - T_i) = 0 \tag{15}$$

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

The above iteration scheme is reminiscent of the EM algorithm [13, 14]. At each step, the new scales better approximate the appropriate scaling parameters and therefore move the system of equations closer to a least squares solution (where all equations have the same "weightage"). We have empirically observed that convergence is achieved in about $3 - 4$ iterations. Therefore, the additional computational load is insignificant. Also at step n in the iterative scheme defined above, the least squares error is

$$E = \sum_{ij} \|\frac{\lambda^n}{\lambda^{n-1}} \omega_{ij}\|^2 \tag{16}$$

At the minimum of the objective function after convergence has been achieved, we have the condition $\lambda^n = \lambda^{n-1}$. This implies that

$$\sum_{i,j} \|\frac{\lambda^n}{\lambda^{n-1}} \omega_{ij}\|^2 = \sum_{i,j} \|\omega_{ij}\|^2 \tag{17}$$

Therefore, the least square error is identical to the error attained by the optimal solution.

3 Experimental Evaluation of 3D Motion Estimation

In this Section, we describe real experiments that were used to test and validate our algorithms for estimating three-dimensional motion as described in the preceding sections. We are unable to include an empirical evaluation of our algorithm here due to space constraints. We evaluate the performance of our algorithm on a real sequence for which ground truth data is available. The well-known Castle Sequence consists of 11 frames taken with a camera that primarily translates and zooms. A frame from this sequence is shown in Figure 2 (a). The errors in rotation and translation direction estimation are shown in subfigures (b) and (c). We can see a significant gain in performance of our method (indicated in dotted line) over the baseline [5] method (indicated in solid line). The comparison of the recovered translation scale with the ground truth is shown in (d). As can be seen, there is very good agreement of

[4] It should be mentioned that since the translation directions are computed in different frames of reference, they need to be rotated to conform to measurements in the reference frame. This is done using the rotation estimate R_i obtained using the method described in the previous subsection.

[5] The baseline is established by computing the transformation between every frame and the reference frame.

the recovered scales with the ground truth value. It may be noted that our method automatically recovers global translation scale (upto a single factor) from only translation direction estimates.

We would also like to point out that this is a particularly difficult sequence for linear methods to work on. It is a well known fact that the Eight Point algorithm biases its estimate of translation direction towards the viewing direction. In this case, the true translation is most of the time orthogonal to the viewing direction (ie. in the x direction). Hence the biases can be significant as seen in the error values for the baseline case. However, our method performs quite well inspite of the fact that its input values are estimates that are significantly biased.

The computational load of our method is also very low since it is a purely linear approach. The average run time of our method for 6 frames is about 0.4 seconds for a MATLAB implementation. Hence our method, while being suboptimal is preferable in many situations and as discussed in Section 2 is amenable to a recursive implementation.

4 Focal Length Estimation

The above analysis assumes that the feature correspondences are available in normalised coordinates (ie. the co-ordinate system of an ideal pinhole camera). As we will demonstrate in this Section, we can use our measure of consistency of the two-frame estimates to accurately estimate the calibration of the camera. In general, camera calibration is a hard problem and is computationally intensive [17]. However in recent times, researchers have focused on the relatively easier problem of estimating camera focal length (by assuming the rest of the camera parameters to be known) [18, 19]. This is a reasonable assumption to make since except for focal length, all other camera parameters like, image center, pixel aspect ratio, skew etc. do not change and can be easily obtained. Our algorithm is based on the intuitive notion that assuming a wrong focal length will result in erroneous estimates of the three-dimensional rotation between different image pairs in the sequence. As a result, when we perform the least-squares estimation (Eqn. 5), the residual error $\sum \|\omega_{ij}\|^2$ will be large. We consider this error value.

Let the fundamental matrix between images i and j be F_{ij}, and let the true calibration matrix be K_0. Therefore the corresponding Essential matrix will be ${E^0}_{ij} = {K_0}^T F_{ij} K_0$. The equivalent rotation matrix R_{ij} can be extracted from ${E^0}_{ij}$. We now denote the least squares error for fitting the set of rotation matrices $\mathcal{R} = \{R_{ij}\}$ to be $FIT(\mathcal{R}(f))$, where f denotes the assumed focallength.

Now if we assume the calibration is $\hat{K}(f)$ instead of the correct value K_0, we have

$\hat{E}_{ij} = \hat{K}(f)^T F_{ij} \hat{K}(f)$. Therefore, for large errors in calibration, the essential matrix $\hat{E}$ will move away further on the essential manifold from E^0. This will result in large errors in rotation estimates. Therefore, the focal length estimate is

$$\min_f FIT(\mathcal{R}(f)) \tag{18}$$

which is carried out by means of a one-dimensional search. To evaluate the performance of our algorithm, we performed focal length estimation on two image sequences. In both cases, we assumed the camera center to be fixed and located at the image center and that the pixel aspect ratio was 1. For the first experiment we used correspondences from a sequence of 8 images of 1536×1024 pixels. See Figure 4 of [16] for details about this sequence. The global minimum of our energy function is clearly located and occurs at a focal length of 1560 pixels which is equivalent to an estimated field of view of $52.42^\circ \times 36.34^\circ$. The correct field of view (FOV) for this sequence was $51^\circ \times 38^\circ$. Thus there is excellent agreement between the estimated calibration and the ground truth.

The second experiment we carried out was with the well known PUMA sequence. In this experiment, we used 32 feature correspondences that were tracked over 16 frames. The true field of view in this case was 40° and the estimated field of view was 44.3°. Some of the discrepancy in the result can be attributed to the limitation of linear estimation of the fundamental matrices. Experimentally, we have observed that using better fundamental matrix estimates (obtained using non-linear minimisations) does result in improved performance. However, this improvement comes at the expense of increased computational time due to the non-linear minimisations used for estimating the fundamental matrices. Given the simplicity of using linear estimates for both rotation estimation and obtaining a global fit, we believe that our algorithm is a good way of estimating the focal length of an image sequence. Due to space constraints, we are unable to discuss the performance of our method under noise. However, we would like to emphasise

that for moderate amounts of noise, our method gives accurate estimates.
We believe that the above experiments adequately demonstrate that our algorithm is effective in computing three-dimensional motion and camera calibration. It is also important to reemphasise that our algorithm runs very fast compared to nonlinear optimisation schemes.

5 Image Motion Models

When we are interested in computing a two-dimensional image motion model, a linear image transformation (Projective or Affine) sufficiently captures the relative motion between pairs of images. In this case, the linear relationship can be described as

$$\begin{pmatrix} x^{'} \\ y^{'} \\ 1 \end{pmatrix} = \lambda P \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{19}$$

where (x, y) and $(x^{'}, y^{'})$ are the co-ordinates of corresponding points in the first and second images respectively. Since the 3×3 transformation P is projective, λ is an unknown scalar except when P takes on the form of an Affine matrix (ie. when its third row is of the form $\{0, 0, 1\}$) in which case $\lambda = 1$. Hence the linear system that solves for a consistent system of projective transformations is

$$\hat{P}_{ij} P_i - \lambda_{ij} P_j = 0 \tag{20}$$

In [7] and [8], consistency frameworks are adopted to solve for global image motion models. In [7], images are aligned to a global mosaic and subsequently a simultaneous bundle adjustment of the orientations of all the images is carried out to correct for errors in positioning. In [8], the global motion model is computed such that the sum of the image intensity variance at each pixel in the constructed mosaic is minimised. This measure of the minimisation of the variance of the aligned image pixels is equivalent to solving for the global motion model using a consistency measure where we minimise the total sum of pairwise residual sum of squared difference (SSD) between two images. However, both of these methods involve a non-linear optimisation. In [1], the author proposes a system similar to Eqn. 20 to solve for a consistent set of transformations to align images in a mosaic. However, in that case, the unknown factor λ_{ij} is omitted which would only be appropriate when the image transformation model is an affine transformation and not projective. In the case of projective transformations, the scaling of different elements is not the same. The third column consists of the translation components that have a much larger scale than the rest of the entries (these values can be of the order of the size of the image itself, say 256). On the other hand, the third row is typically close to $[0, 0, 1]$. This unequal scaling implies an uneven weighting in the least squares fitting of the estimates and can lead to numerical instabilities. Therefore, it is important to apply a "whitening" transformation to estimates $\hat{P}_{ij}$ before solving for a consistent solution. The need for a whitening process has been noted earlier by Hartley in the case of the Eight Point algorithm [3]. In our method, we solve for the different relative transformations P_{ij} and apply a "whitening transformation" to give equal weightage to different terms. Since we also need to incorporate the estimation of the unknown scaling factors (λ), we use the following iterative scheme to solve for a consistent set of transformations.

- For the kth row of all P_{ij},$(k = 1, 2, 3)$, compute the average value (s_k) of the corresponding elements
- Compute the scaling transformation $S = diag([\frac{1}{s_1}, \frac{1}{s_2}, \frac{1}{s_3}])$
- Whiten individual transformations as $P^{'}{}_{ij} \leftarrow S P_{ij} S^{-1}$

Then we apply the following iterative scheme :

* Set all $\lambda_{ij} = 1$
* Solve the linear system $\hat{P}^{'}_{ij} P_i^{'} - \lambda_{ij} P_j^{'} = 0$
* Update $\lambda_{ij} = \frac{\|P^{'}{}_{ij} P^{'}{}_i\|}{\|P^{'}{}_j\|}$
* Repeat till convergence

- Unwhiten the individual global motion models as $P_i \leftarrow S^{-1} P^{'}{}_i S$

6 Evaluation of 2D Image Model Estimation

To evaluate the performance of this method, we study the ability of our algorithm to construct image mosaics with noisy data. In Fig. 3 (a), we show the mosaic constructed using frames from a long sequence. To solve for the relative pairwise transformations, we extract 10 tracks over the entire sequence using the KLT tracker [15].

In this experiment we evaluate the performance of our algorithm with different amounts of noise in the point correspondences. Here the 2D affine model is used. Our objective is to study the improvement in performance (under noise) with increased use of the redundant information available in the sequence. For the input to our algorithm, we compute the relative affine motion between every frame i and j, such that $|i-j| \leq k$ where k is a parameter we choose. The parameter k determines the amount of redundancy we utilise. For example, if $k = 3$, then we compute the pairwise affine model for every pair of images that are within a distance of 3 in the sequence. Thus in an N frame sequence, when $k = 1$, there is no redundancy utilised, ie. we simply compute the transformations between adjacent frames, therefore the set of pairwise transformations computed is, $(i,j) = \{(1,2),(2,3),\cdots,(N-1,N)\}$. In the case where $k \geq N-1$, we compute the pairwise transformation between all possible pairs in this set of images.

Although we use feature point correspondences, our measure of performance is based on image intensity instead since all pixels that correspond to the same point on the mosaic should have the same intensity. Thus, we define the following figure of merit

$$score = E[\sigma^2(I_i(\mathbf{p}, P_i))], \tag{21}$$

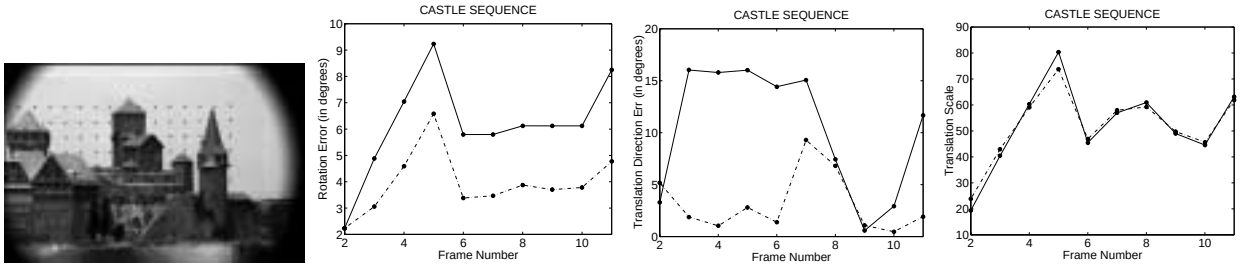
where $E[.]$ denotes the expectation operator. In other words, we estimate the required transformations and warp all images to the reference frame (The resultant average of the images is shown in Fig. 3(a). Subsequently, we compute the variance of the image intensity at each pixel in the reference frame.

The performance of our algorithm is shown in Fig. 3 (b). For each experiment, we choose a value of k. Different amounts of Gaussian noise is added to all the point correspondences and the mosaic is estimated by averaging all warped images. The resultant figure of merit is plotted for different amounts of noise. At low levels of noise, the difference in performance with increasing redundancy is negligible. This implies that the original data set is good enough to generate accurate mosaics and using more information cannot result in improved performance. However as we increase the noise level, it can be easily noted that the quality of reconstruction degrades. In the case where $k = 1$, ie. when the transformations between adjacent pairs are cascaded, the performance is quite poor. However, we can observe that an increasing use in the redundant information does significantly improve performance. In fact, when $k = 4$, there is almost a 40% improvement in performance compared with the "baseline" case of $k = 1$. Of course, this improvement does entail an increased computational load. However, the total computational load is not very significant. In conclusion, this experiment is a simple and powerful illustration of the argument that within computational constraints, we should utilise as much of the available redundancy as possible.

Finally, we demonstrate the usefulness of our notion of linear consistency in constructing a mosaic. In Fig. 4 we show mosaics constructed with the same set of images. In this example, the camera has imaged the scene by taking 6 consecutive overlapping images along 4 horizontal strips resulting in 24 images. In Figure 4 (a), the mosaic is constructed by computing pairwise projective transformations between adjacent images (using matched feature points) along the horizontal direction and then aligning images in each horizontal strip with respect to the reference frame by computing the transformation between the rightmost image in each strip and the reference frame. Thus for each image, we are able to compute a product of projective transformations that will align it with respect to the reference frame. As a result, there is good alignment along a horizontal strip. However, as can be clearly seen, there is gross misalignment between different strips. This results in significant errors along a vertical direction (some error regions are indicated in circles). This is due to the fact that the errors in individual horizontal strips accumulate as we move right to left along a strip. But the errors in individual strips are independent, resulting in gross misalignment along the vertical direction. In Figure 4 (b), in addition to the relative transformations used in Figure 4 (a) we computed some of the relative transformations between adjacent pairs along the vertical direction. These transformations were then used to compute a consistent set of transformations as described above. As can be observed by comparing the areas marked by the circles in the two images, for our method there is good alignment over the entire mosaic and none of the anomalies of Figure 4(a) are present.

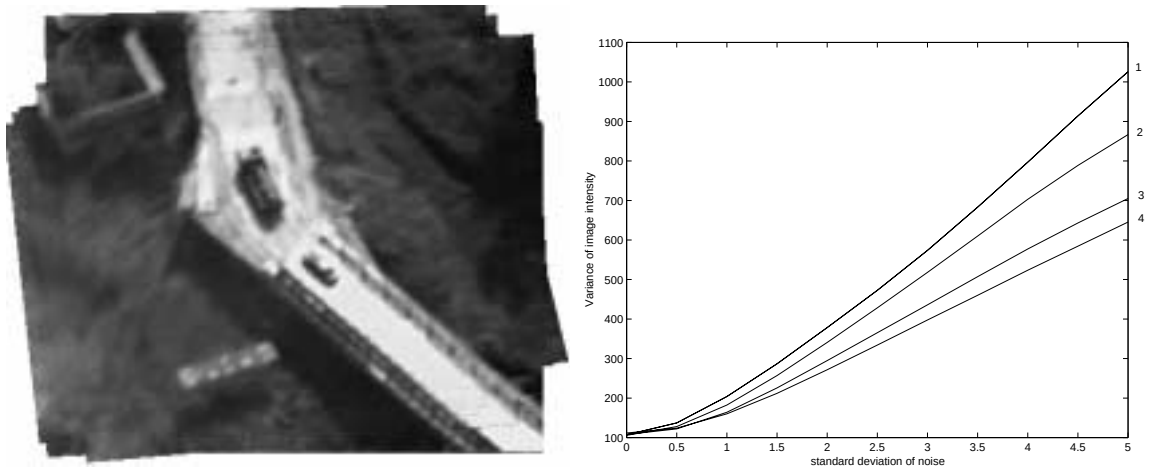
7 Conclusion

In this paper we have introduced a linear method for computing the motion between images of a sequence. This method efficiently exploits the redundancy of pairwise motions estimates to give accurate estimates for the motion of the entire sequence. In comparision with non-linear methods, our method is



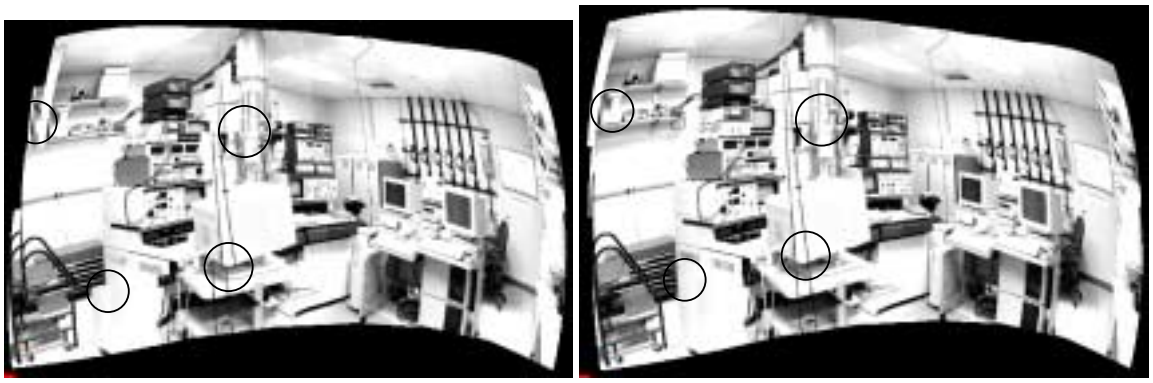
(a) One frame from sequence (b) Rotation Error (c) Translation Dir Error (d) Translation Scale

Figure 2: (b) and (c) show the error in rotation and translation direction for the Castle Sequence. The solid line indicates the baseline algorithm and the dotted line indicates our method. (d) shows the recovered translation scale by our method (indicated by dotted line). Ground truth is shown in solid line.



(a) Generated Mosaic (b) Performance vs Noise Level

Figure 3: (a) shows the mosaic created by our method. (b) shows the performance of our algorithm for different amounts of noise. Each curve represents the amount of redundancy used. The maximum overlap distance used (k) is shown alongside each curve (1 - 4).



(a) (b)

Figure 4: (a) Mosaic created by aligning images along horizontal strips. Gross vertical errors can be seen at the boundaries of strips (some are indicated by circles). These errors disappear in (b) created by our method that uses all available information.

extremely fast. It also gives focal length estimates to a good degree of accuracy.

References

[1] Davis, J., "Mosaicing of Scenes with Moving Objects", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Press, pp. 354-360, 1998.

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

[3] 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.

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

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

[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] Shum, H., and Szeliski, R., "Construction and refinement of panoramic mosaics with global and local alignment," *International Conference on Computer Vision*, pp. 953-958, 1998.

[8] Sawhney, H., Hsu, S. and Kumar, R., "Robust Video Mosaicing through Topology Inference and Local to Global Alignment," *European Conference on Computer Vision*, pp. 103-119, 1998.

[9] 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.

[10] 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.

[11] 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.

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

[13] Dempster, A. P. and Laird, N. M. and Rubin, D. B., "Maximum Likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, B(39):1-38, 1977.

[14] McLachlan, G. J. and Krishnan, T., *The EM Algorithm and Extensions*, Wiley 1996.

[15] Birchfield, S., "Derivation of the Kanade-Lucas-Tomasi tracking equation", *http://robotics.stanford.edu/ birch/klt*, 1996.

[16] Oliensis, J., "Fast and accurate self-calibration", *Proceedings of International Conference on Computer Vision*, pp. 745-752, 1999.

[17] Faugeras, O., and Luong, Q. T., and Maybank, S. J., "Camera self-calibration : Theory and experiments", *Proceedings of European Conference on Computer Vision*, pp. 321-334, 1992.

[18] Pollefeys, M., and Koch, R., and VanGool, L., "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters", *International Journal of Computer Vision*, 32(1):7-25, August 1999.

[19] Triggs, B., "Autocalibration and the absolute quadric", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 609-614, 1997.