Accurate Camera Calibration for Off-line, Video-Based Augmented Reality

Simon Gibson, Jon Cook, Toby Howard, Roger Hubbold
Advanced Interfaces Group,
Department of Computer Science,
University of Manchester, UK.

Dan Oram
Visual Communication Group,
Canon Research Centre Europe Ltd.
United Kingdom.

Abstract

Camera tracking is a fundamental requirement for video-based Augmented Reality applications. The ability to accurately calculate the intrinsic and extrinsic camera parameters for each frame of a video sequence is essential if synthetic objects are to be integrated into the image data in a believable way. In this paper, we present an accurate and reliable approach to camera calibration for off-line video-based Augmented Reality applications.

We first describe an improved feature tracking algorithm, based on the widely used Kanade-Lucas-Tomasi tracker. Estimates of inter-frame camera motion are used to guide tracking, greatly reducing the number of incorrectly tracked features. We then present a robust hierarchical scheme that merges sub-sequences together to form a complete projective reconstruction. Finally, we describe how RANSAC-based random sampling can be applied to the problem of self-calibration, allowing for more reliable upgrades to metric geometry. Results of applying our calibration algorithms are given for both synthetic and real data.

1. Introduction

One of the fundamental components of any Augmented Reality system is the ability to merge synthetically generated objects with images of a real environment. Camera tracking, or match-moving, is an essential step of this process, and the accuracy of the tracking algorithm can significantly affect the perceived accuracy of the augmented environment.

Camera tracking involves estimating, for each frame of an image sequence, the intrinsic and extrinsic parameters of the camera at the time when the frame was captured. The extrinsic parameters describe the position and orientation of the camera, and the intrinsic parameters contain measurements such as focal length, principal point, pixel aspect ratio and skew [14].

For some applications of Augmented Reality, calculation of these parameters must be done in real-time, in order to merge synthetic objects with a live video feed, or to display them in a see-through head-mounted display. Vision-based tracking algorithms used in these situations typically update the estimate of camera motion in a sequential fashion, from one frame to the next (see, for example [1, 25, 4]). These algorithms often make assumptions regarding the nature of the camera motion or scene structure, limiting their applicability when a high degree of accuracy and generality is required.

The sequential nature of these algorithms, along with other non-realtime approaches (e.g. [7, 6]), means that the calibrations produced are susceptible to drift. The overall accuracy often favours the images used to initiate the sequential reconstruction, and later frames of the sequence have a tendency to drift out of the original coordinate frame when awkward or near-degenerate camera motions are encountered. This is caused by an accumulation of error over the sequence, meaning that the calibrations are not always satisfactory when the registration of objects to images must be done as accurately as possible, with no visible drift or jitter.

For off-line Augmented Reality applications, the problems caused by these approaches can be overcome, because there is no need for the calibration to be achieved either sequentially, or in real-time. Merging-based approaches to projective reconstruction (such as [9]) calibrate small sections of the image sequence, and then merge these sections together. This process attempts to distribute any error as evenly over the sequence as possible, thereby reducing drift.

There are many applications for such off-line video-based Augmented Reality – architectural and archeological visualization [23], and post-processing for television or film special effects being the most common. Indeed, it could be argued that due to the prolific nature of film special effects, this off-line style of processing is currently the most in-demand form of Augmented Reality.

An overview of our system for augmented video production is given in Figure 1. This diagram shows the logical connections between components of the system. The main

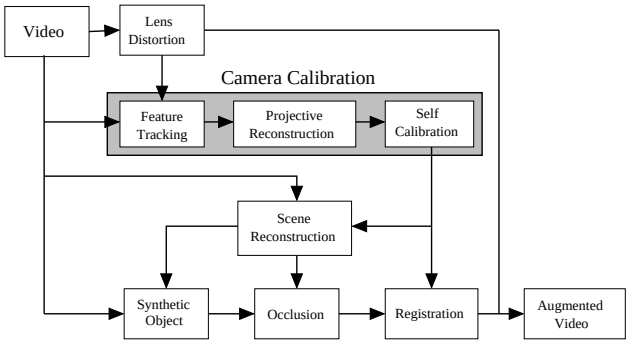

Figure 1. An overview of the video-based Augmented Reality system, showing the logical connections between the components of the system. In this paper, we concentrate on the camera calibration module.

focus of this paper is the camera calibration module shown in the middle of the diagram. The input video is first corrected to remove the effects of geometric lens distortion (see e.g. [26] for further details). The calibration process then begins by automatically identifying and tracking a large number of feature points throughout the sequence. Structure and motion recovery algorithms are then used to reconstruct the position of these features, and the camera projection matrices for each frame. Because this reconstruction is performed in an unknown projective basis, self-calibration is required to upgrade the camera projection matrices to a metric framework.

Additional parts of our system, which are described elsewhere [10, 11], use this calibration data to assist in reconstructing a geometric description of the environment. When augmenting the video sequence with synthetic objects, this scene description helps to resolve occlusions between real and synthetic objects. The camera calibration data is finally used to accurately register the synthetic object with each frame of the video sequence.

The camera tracking algorithm presented in this paper is novel in several ways. Firstly, we describe an improved algorithm for feature tracking, based on the widely used Kanade-Lucas-Tomasi (KLT) feature tracker [27, 24]. We use estimates of inter-frame camera motion to guide the feature tracker, greatly reducing the number of incorrectly tracked features. We also present a reliable approach to projective reconstruction, using carefully selected subsequences of the video data, and hierarchical algorithms to merge sub-sequences into a single reconstruction. Finally, we show how random sampling algorithms can be applied to the problem of self-calibration, allowing for more accurate upgrades from projective to metric geometry. The result of these improvements is an accurate and reliable camera tracking algorithm that can be used for video-based Augmented Reality.

The remainder of this paper presents the camera tracking process in more detail. To begin with, Section 2 describes the details of our automated feature tracking algorithm. Following that, Section 3 discusses the hierarchical merging scheme used to construct a complete projective reconstruction. After details of our random-sampling self-calibration algorithm are given in Section 4, we show results for synthetic and real scenes in Section 5.

2. Feature Tracking

Our automatic feature tracking algorithm is based upon the iterative KLT algorithm [27, 24], with modifications to account for colour images [15] and changes in pixel brightness and contrast [17]. Features of interest are selected using the Harris corner detector [12], applied so that the distribution of features over the image is as even as possible.

As features are tracked, they may be lost due to poor localization by the tracking algorithm, or because they move beyond the frame boundary. When this occurs, we select new features to replace those lost, and continue tracking, always trying to maintain a constant number of features in each frame. One novel aspect of our tracking algorithm is that after the first tracking phase has completed, a second pass moves from the end frame back towards the first, tracking those new features that were introduced as replacements. This *back-tracking* increases the tracking duration for each replacement feature, and improves the overall robustness of the calibration.

The most significant improvement we have made to the KLT tracking algorithm is to employ an estimate of inter-frame camera motion to assist in the tracking of features from one frame into the next. This so-called *guided tracking* has been found to significantly reduce the number of *outlying* features. Reducing the number of these outliers is important if the motion recovery algorithm described in the next Section is to operate reliably. Note that this approach to guided tracking is different from previous *guided matching* algorithms [2] that attempt to match candidate features between images, rather than explicitly track features from one frame to the next.

The general approach taken is to track features through the sequence until a reliable fundamental matrix [14] can be estimated, and then to use this to determine the set of inlying feature tracks, and hence the camera motion relative to the starting frame. If a fundamental matrix cannot be estimated reliably, a planar projective homography is used to guide feature tracking instead. Tracking then moves back to the starting frame, and the motion estimate is used to identify those tracks where a feature position deviates significantly from its estimated motion (i.e. epipolar line, or homography-mapped image location).

More explicitly, guided tracking begins by selecting a set of candidate feature points in frame $i = 0$. These features are then tracked into frames $j = i + 1$, $j = i + 2$, and so on, until one of the following is satisfied:

- The end of the sequence is reached, or 10 frames are processed,
- More than 50% of the features are lost, or
- When a planar homography is robustly estimated from the feature motion between frames i and j, the overall RMS error for the fit exceeds a user-specified threshold.

The third item in this list is the most important: we assume that if a planar homography fits the data poorly, then it is likely that the fundamental matrix can be estimated reliably. This is because the most significant degeneracies during fundamental matrix estimation occur when there is little camera translation, meaning that the feature set fails to uniquely define the epipolar geometry, and the estimation algorithms become numerically ill-conditioned [14]. By looking at the overall quality of fit for a planar homography to the motion of features between frames i and j, we get an indication of when a fundamental matrix estimate is likely to be well conditioned. Typically, we use a threshold of around 4 pixels for homography fitting, as this has been found to provide a balance between accurate epipolar geometry and a large number of feature tracks.

Once a suitable frame, j, has been found, we use a random sampling algorithm to estimate the fundamental matrix between frames i and j [14, 29]. This allows us to identify the set of features that have tracked reliably from frame i to frame j. The tracking algorithm then uses these inlying features to determine the epipolar geometry relating frame i to each frame $i+1, i+2, \ldots, j$. The tracking of features from frame i to these frames is then repeated, with the epipolar geometry between two frames used to identify those features that have tracked incorrectly. This is achieved by testing to see if each feature moves a significant distance from its corresponding epipolar line during the iterative tracking procedure. Tracking for those features that deviate significantly from the motion is halted. Once frame j is reached, a new motion estimate is initiated, and the process repeats. If 10 frames are passed and a fundamental matrix has not been found, planar homographies are estimated from the inlying feature points, and used to map features from one frame into the next, with the transformed locations used as initial estimates for the KLT tracking algorithm.

Although a thorough examination of the benefits of this tracking algorithm is beyond the scope of this paper, we have observed that it consistently increases the number of inlying features, thereby reducing the overall reprojection error, and also improves the stability of the final calibration.

Figure 2. Inlying feature projections obtained using the standard KLT tracker (left), and the guided KLT tracker introduced in this paper (right).

By way of example, Figure 2 shows one frame from a small 100-frame sequence captured from the passenger seat of a car driving along a road. This sequence was tracked using both the standard KLT tracker [24] and the guided algorithm described above [1]. For the standard tracker (targeted to 250 features per-frame), $17,772$ inlying feature projections were found out of a total of $22,115$ (80% inliers). Tracking time was just over 4 minutes on a 1.4GHz Athlon CPU. In contrast, the guided tracking algorithm took 10 minutes, but was able to find $26,871$ inliers out of $28,970$ projections (93% inliers), using the same target of 250 features per-frame. This increase in both the total and inlying numbers of projections is due to the more reliable detection of outlying features during tracking, which causes a larger number of good candidate features to be found.

After calibration using the structure and motion recovery algorithm described in Section 3, the mean reprojection error was 0.33 pixels using the feature tracks obtained with the standard tracking algorithm (standard deviation of 0.23). Using the same calibration algorithms and the features obtained with guided tracking, the mean reprojection error was reduced to 0.28 pixels, with a standard deviation of 0.20. The improved accuracy of the projective reconstruction also produced a more accurate estimate of focal length when self-calibration was applied (see Section 4). Focal lengths after guided tracking varied between 855 and 857 pixels over the length of the sequence (the "true" focal length, measured at the same zoom factor using vanishing points, is approximately 860 pixels). Using standard tracking, the self-calibration algorithm was more error-prone, resulting in focal length estimates varying between 553 and 554 pixels. Similar improvements in accuracy and stability have been obtained for other test sequences.

[1] Feature track animations for this sequence, as well as those presented in Figures 3 and 6 are included in the additional material accompanying this paper.

3. Merging-Based Projective Reconstruction

Once a suitable set of features have been identified, we employ a merging-based projective reconstruction algorithm to estimate the feature locations and camera projection matrices for each frame of the sequence. We have chosen to take a merging approach to reconstruction because sequential algorithms [1, 4] are heavily reliant on a good initial estimate of structure, and are also susceptible to drift over long sequences. Factorization methods (e.g. [16]) suffer less from drift and error accumulation by calculating all camera projection matrices and structure at the same time, but suffer from a lack of robustness, flexibility, and meaningful error criteria. To overcome these problems, in [9] it was proposed to reconstruct small sequences and then merge them hierarchically to create larger sequences. In this section, we present a new method for merging-based projective reconstruction that is both robust, and relatively fast. Indeed, by altering the number of merging passes performed, a simple trade-off can be made between accuracy and calibration time.

3.1. Sub-Sequence Reconstruction

Reconstruction is achieved by first selecting a set of key-frames with which to build small, sub-sequence reconstructions. For each pair of key-frames, a separate projective reconstruction will be built, consisting of the camera projection matrices for each frame between the key-frame pair, and the position of inlying features visible in those frames. In Section 3.2, we will describe how these sub-sequences may be merged together to form a complete projective reconstruction.

Key-frames are selected so that the epipolar geometry between each can be estimated reliably, and so that overlapping sequences can be merged together using structure and frames common to each. The selection process starts by positioning a key-frame at frame 1. All possible pairings of the first frame with consecutive frames in the sequence are then considered. For each pairing (i, j), the following score, $\mathcal{S}_{ij}$ is evaluated:

$$\mathcal{S}_{ij} = w_1(1.0 - \frac{n_{1_{ij}}}{n_{2_{ij}}}) + w_2\frac{1}{e^2_{H_{ij}}} + w_3 e^2_{F_{ij}} \qquad (1)$$

where $n_{2_{ij}}$ is the number of features that were reconstructed in the previous key-frame pair, $n_{1_{ij}}$ is the number of those features that can also be reconstructed in this pair [2], $e_{H_{ij}}$ is the median reprojection error when a planar homography is fitted to the feature data using a random sampling algorithm [29], and $e_{F_{ij}}$ is the median epipolar error when a fundamental matrix is estimated using a similar sampling algorithm. $w_{1,2,3}$ are weights used to alter the relative significance of each score.

The first term in Equation 1 measures the fraction of features that were reconstructed in the previous key-frame pair, but cannot possibly be reconstructed in this pair. The second term includes the homography error, $e_{H_{ij}}$, and is used to test for degeneracy between the two views: it follows that the smaller the value of $\frac{1}{e^2_{H_{ij}}}$, the worse the homography fits the data, indicating that the estimate of epipolar geometry is unlikely to be degenerate. Finally, the third term represents the squared median epipolar error.

Typically, the homography error is small when there is little camera motion between frames. Clearly, by detecting this lack of motion, our calibration algorithm will be able to handle these situations, and this type of degenerate motion will not cause numerical problems. Note that there are other less common situations where a homography can fit feature motion and the camera *does* undergo significant motion, notably where the features all lie on a planar surface. Alternate detection methods have been proposed to deal with these situations [22], and although these techniques could be incorporated into this framework, we will not consider these situations further in this paper.

Assuming that a key-frame has already been placed at frame i, the next key-frame should be chosen so that the weighted sum of these three terms is minimised. This is achieved by evaluating Equation 1 for a pairing of frame i with each frame $j = i + 1, \ldots$. This is continued until less than 50% of the features that are tracked from frame i remain in frame j. The frame with the lowest score is then marked as the next key-frame, and the process continues to the end of the sequence, searching for the best pairing with this key-frame. Typically, we use relative weights of $w_1 = 3$, $w_2 = 10$, and $w_3 = 1$, giving more weight towards the homography error and number of common features than to the epipolar error.

Figure 3 shows a typical set of key-frames selected by our algorithm, and illustrates how the key-frame selection process is used to initiate the projective reconstruction of camera motion. At the bottom of the figure are sample images from a 200 frame video sequence, captured using a hand-held digital video camera. The graph directly above these images plots frame number on the horizontal axis, versus the magnitude of inter-frame camera motion (obtained using the calibration algorithms described below) on the vertical axis. Above the graph, a time-line indicates the position of key-frames in the sequence [3].

It is important to notice that the middle third of the se-

[2] A feature can be reconstructed in pair (i, j) if there are projections into at least two frames $i \leq k_1, k_2 \leq j$. For the first pair, $n_{2_{ij}}$ is simply taken to be the total number of possible features.

[3] Note that each key-frame is marked with a vertical line, but for clarity, only a sub-set of the key-frames are numbered. The un-numbered key-frames are marked using short vertical lines (e.g. there are 4 additional key-frames between frame 1 and frame 21).

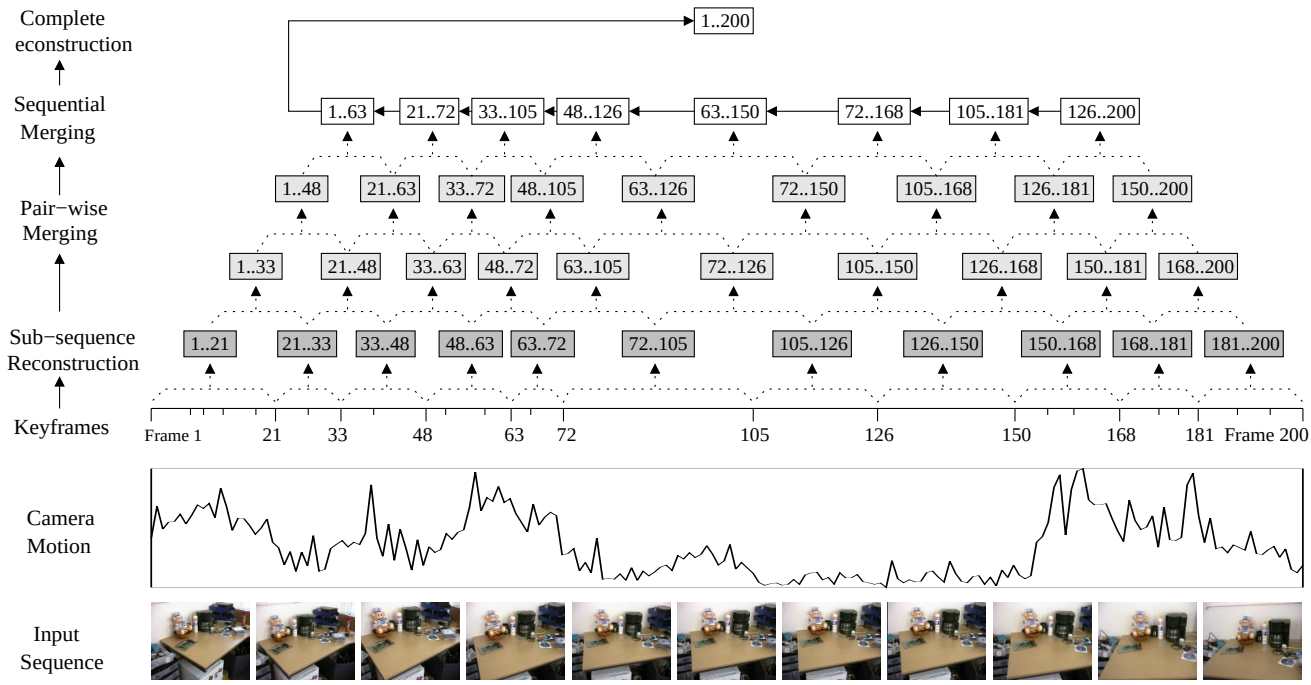

Figure 3. Key-frame selection, sub-sequence reconstruction and hierarchical merging for a 200 frame sequence (see text for details).

quence contains relatively little camera motion. If key-frames were selected uniformly along the sequence, the camera motion between key-frames in this central region would not necessarily be well-defined, causing numerical problems when estimating the epipolar geometry. The key-frame selection algorithm described above, however, has detected this lack of motion, and key-frames have been positioned sparsely throughout this region.

Once a set of key-frames has been identified, two-view projective reconstructions are built for each pair of key-frames, using random sampling algorithms [29] to estimate the epipolar geometry and projective structure. The reconstruction of a sub-sequence is then completed by estimating the camera projection matrices for non-key frames between each pair, using the inlying features and a resectioning algorithm [14]. Levenberg-Marquardt bundle adjustment [13] is then applied to each sub-sequence, minimizing reprojection errors and distributing the overall residual error evenly across the sub-sequence.

Note that this method of key-frame selection is different to the approach described in [19, 18], where feature matching between frames occurred after, rather than before key-frame selection. Furthermore, in [19, 18], no explicit attempt was made to avoid degeneracy, and key-frame selection was based on image sharpness criteria rather than reconstruction error.

3.2. Hierarchical Merging Scheme

At this stage of the reconstruction process, we have obtained projective reconstructions for sub-sequences of the entire video sequence. In order to obtain a projective reconstruction for the entire video sequence, these sub-sequences must be merged together. To reduce error, a robust hierarchical merging technique will be introduced. The benefits of hierarchical merging schemes such as ours is that they are able to distribute error more evenly over the entire sequence, thereby reducing drift and increasing the apparent accuracy of object registration. Results will be shown later comparing calibration error for the merging scheme against a simple sequential algorithm.

Figure 3 illustrates the stages of our hierarchical merging algorithm. Starting with the sub-sequence reconstructions (first row above the key-frame line), pairs of sub-sequences are robustly merged together using techniques described in the next section. This *pair-wise merging* continues for a user-specified number of passes through all the sequences, with each pass reducing the total number of sub-sequences by one. In Figure 3, three passes are shown.

After the user-specified number of pair-wise merging passes have been performed, merging continues in a sequential fashion (top row in Figure 3), merging all sub-sequences into the first. Clearly, sequential merging can quickly produce a completed projective reconstruction con-

taining all frames. This speed comes at the expense of accuracy, however, when compared to merging adjacent pairs of sub-sequences. By altering the number of rounds of pair-wise merging, the user can easily trade speed against accuracy of reconstruction. Calibrations generated with varying numbers of pair-wise merging rounds will be compared in Section 5.

3.2.1. Projectivity Estimation

Although two overlapping sub-sequences will contain a large number of common features, the projective basis in which each is represented will be different. Because of this, it is necessary to use corresponding scene structure to compute the change in projective basis that maps one sub-sequence onto another. This change in basis takes the form of a projectivity, H in $\mathcal{P}^3$. Suppose that a point j has matching structure represented by $\mathbf{X}_j$ in the first sub-sequence, and $\mathbf{X}'_j$ in the second sub-sequence. It follows that

$$\begin{array}{c} \mathbf{X}_j \simeq H\mathbf{X}'_j \\ P_i \simeq P'_i H^{-1} \end{array} \tag{2}$$

where P_i and P'_i are the camera projection matrices represented in each projective basis, and $\simeq$ indicates equality up to an arbitrary scale factor. We determine H by minimising the following term:

$$\sum_{ij} d^2(P_i H\mathbf{X}'_j, P_i\mathbf{X}_j) \tag{3}$$

for all structure j, and all frames i in sub-sequence 1 in which this structure is observed. This measure states that, for a common feature j, we wish to transform its location $\mathbf{X}'_j$ in sub-sequence 2 into sub-sequence 1, and minimise the re-projection error in all frames of sub-sequence 1 in which the feature is visible, compared to the projections of the feature position $\mathbf{X}_j$. The distance function $d(\mathbf{X},\mathbf{Y})$ may represent either Euclidean or algebraic distance between the image locations $\mathbf{X}$ and $\mathbf{Y}$:

$$d_E^2 = \sum_{k=1}^{3}\left(\frac{X_k}{X_4} - \frac{Y_k}{Y_4}\right)^2 \quad d_A^2 = \sum_{k=1}^{3}(X_kY_4 - Y_kX_4)^2 \tag{4}$$

In all situations where merging is required, our scheme ensures that at least one frame is common to both sub-sequences. Because of this, the projection matrices of the common frame, P_0 and P'_0 may be aligned with the standard projective basis by multiplying with their pseudo inverses, P_0^+ and $P_0^{+\prime}$, in each sub-sequence, resulting in $P_0P_0^+ = P_0^{+\prime}P'_0 = [\mathbf{I}_{3x3}|0]$. This effectively removes 11 of the 15 degrees of freedom in the unknown projectivity [9]. The projectivity H that minimises Equation 3 will now belong to the 4-parameter family of homographies:

$$H = P_0^+ A P'_0, \text{where } A = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0} \\ a_{1,2,3} & a_4 \end{bmatrix} \tag{5}$$

where $\mathbf{a} = (a_{1,2,3,4})$ represent the 4 unknown parameters. We will denote the projection matrices and structure in the first sequence, after alignment with the standard projective basis as follows:

$$\begin{array}{c} \hat{P}_i = P_iP_0^+ \\ \hat{\mathbf{X}}_j = P_0\mathbf{X}_j \end{array}$$

Combining Equations 5 and 3 with an algebraic error measure, and substituting $\mathbf{u}_{ij} = \hat{P}_i\hat{\mathbf{X}}_j$ yields two linear equations in terms of $\mathbf{a}$ for each projected point:

$$(\hat{\mathbf{p}}_n - \mathbf{u}_{ij}\hat{\mathbf{p}}_3)A\hat{\mathbf{X}}'_j = 0 \tag{6}$$

where $\hat{\mathbf{p}}_n$ represents the n-th row of $\hat{P}_i$, and $n \in (1,2,3)$.

Equation 6 was first presented by Fitzgibbon and Zissermann in [9]. Note that it is based on an algebraic distance, which is not a geometrically or statistically meaningful quantity. It has been demonstrated that minimizing algebraic distance may produce solutions different from those that are expected [3, 14]. Significant improvements can be made to this linear approximation by relating Euclidean and algebraic error measures to Equation 6. A Euclidean error measure gives:

$$d_e = \frac{\hat{\mathbf{p}}_nA\hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3A\hat{\mathbf{X}}'} - \mathbf{u}_{ij} \tag{7}$$

whereas algebraic error simply gives:

$$d_A = \hat{\mathbf{p}}_nA\hat{\mathbf{X}}' - \mathbf{u}_{ij}\hat{\mathbf{p}}_3A\hat{\mathbf{X}}' \tag{8}$$

It can be easily verified that Equation 7 can be obtained by dividing Equation 8 by $\hat{\mathbf{p}}_3A\hat{\mathbf{X}}'$.

From Equation 2, we know that $\hat{\mathbf{X}} = \lambda A\hat{\mathbf{X}}'$ for an unknown scale factor λ. Examining the structure of A, however, reveals that the first 3 rows represent an identity mapping. This scale factor λ can therefore be determined from $\hat{\mathbf{X}}_n = \lambda A\hat{\mathbf{X}}'_n$ for $n \in (1,2,3)$. It has been found in practice that λ is best calculated using a least-squares solution and the 3 available constraints to give [20]:

$$\lambda = \sqrt{\frac{1}{3}\left[\left(\frac{\hat{\mathbf{X}}_1}{\hat{\mathbf{X}}'_1}\right)^2 + \left(\frac{\hat{\mathbf{X}}_2}{\hat{\mathbf{X}}'_2}\right)^2 + \left(\frac{\hat{\mathbf{X}}_3}{\hat{\mathbf{X}}'_3}\right)^2\right]^2} \tag{9}$$

Since λ is now known, we can approximate the division of Equation 8 by $\hat{\mathbf{p}}_3A\hat{\mathbf{X}}'$, by a term involving the projection of the equivalent structure from the other sub-sequence, $\hat{\mathbf{p}}_3\lambda\hat{\mathbf{X}}$. This allows us to construct a new linear algorithm to estimate the projectivity H that uses a close approximation to Euclidean, rather than algebraic error:

$$\frac{\hat{\mathbf{p}}_nA\hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3\lambda\hat{\mathbf{X}}} - \frac{\mathbf{u}_{ij}\hat{\mathbf{p}}_3A\hat{\mathbf{X}}'}{\hat{\mathbf{p}}_3\lambda\hat{\mathbf{X}}} = d_{E_a} \tag{10}$$

3.2.2. Robust Merging

We apply the MSAC random sampling algorithm [29] to the problem of estimating the projectivity relating two sub-sequences, as this is an approach to parameter estimation that is capable of dealing with data that may contain outlying samples. The basis of all random sampling methods is to take a large number of minimal (or almost minimal) samples of data, and estimate the model (i.e. the 4 unknown parameters of the projectivity) using each minimal sample set. Equation 10 defines two linear constraints on the 5 unknown parameters of A for each projected point. To ensure the linear system is suitably over-constrained, we choose sets of 4 features common to each sub-sequence, and use QR decomposition to solve for the projectivity.

It is hoped that one or more of these minimal sets will not contain any outlying data, and so will produce a valid estimate. The projectivity that minimizes a robust Huber function of the residuals given by Equation 10 is identified [4]. In general, random sampling algorithms are very fast due to the small size of the feature sets used. In order to further accelerate the estimation process for longer sub-sequences, only residuals associated with key-frames are counted in the Huber function. Although this is not essential, it has been found to significantly decrease running time, without any significant impact on overall error. Once the projectivity that minimizes the residuals has been found, it is iteratively refined using the Levenberg-Marquardt algorithm, minimizing the same Huber function.

The final estimate of the projectivity can then be used to merge the second sub-sequence into the first. For each piece of common structure, we now have two candidate positions: $\mathbf{X}_j$ and $H\mathbf{X}'_j$, and the one with the smallest RMS re-projection error is kept. Similarly, for overlapping frames, we have two candidate projection matrices: P_i and $P'_i H^{-1}$. Again, we keep the matrix giving the smallest RMS re-projection error for all visible, inlying features. All other features and projection matrices that are present in the second sub-sequence but not in the first are transformed using Equation 2. Features and projection matrices that are present in the first sub-sequence, but not the second, remain unchanged. After merging, outlying features are re-identified, and bundle adjustment is applied to distribute any error evenly throughout the new sequence. Merging continues, as described above, until the user-specified number of pair-wise merging passes have been performed. The same projectivity estimation algorithm is also used during the final sequential merging pass.

[4] Note that the residuals for *all* points are measured, not just for the minimal sample set used to estimate the projectivity.

4. Self-Calibration

The final stage of video sequence calibration involves upgrading the reconstruction from a projective to a metric framework. This is necessary because the projective framework in which the sequence is reconstructed does not preserve important quantities such as distances and angles [14]. The relationship between projective and metric geometry is generally unknown, and so this upgrade must be achieved using *self-calibration* algorithms. Typically, these involve estimating the position of the absolute conic or its dual [30, 21]. For long sequences, an accumulation of error in the projective reconstruction means that the position of the conic often varies slightly along the sequence. Using previous methods for self-calibration, we have found that this variation often causes the conic estimation algorithm to fail, due to the non-positive semi-definite nature of its intermediate matrix representation (note that we have found that this still occurs when the robust merging scheme described in the previous section is used).

We take a novel approach to self-calibration, using a RANSAC-based random sampling algorithm to estimate the absolute dual quadric (ADQ). Using this method, we select random sets of projection matrices from the completed sequence with which to apply the linear algorithm proposed in [21, 22] and estimate the ADQ. We automatically reject matrix sets that cause ADQ estimation to fail.

In addition, we are also able to include other contraints on the camera parameters within the random sampling framework. Our algorithm is capable of selecting a conic that minimizes camera skew and deviations from a known aspect ratio or principal point, and also of applying constraints on constant or known focal length. More specifically, for each candidate set $i = 1 \dots M$ of projection matrices, we estimate the ADQ, and then determine the calibration parameters for each frame j of the whole sequence [21, 22]:

$$\mathbf{K}_j = \begin{bmatrix} f_{j_x} & s_j & c_{j_x} \\ 0 & f_{j_y} & c_{j_y} \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Assuming that the aspect ratio and principal point are known, and the projective reconstruction has been transformed so that they are 1 and $(0, 0)$ respectively, we evaluate the following residual for the candidate ADQ:

$$r_i = \sum_{j=0}^{N} \omega_1 s_j^2 + \omega_2 (1 - \frac{f_{j_y}}{f_{j_x}})^2 + \omega_3 (c_{j_x}^2 + c_{j_y}^2) \tag{12}$$

where the first term measures deviation from zero skew, the second term from unit aspect ratio, and the third from a principal point of $(0, 0)$, and with $\omega_{1,2,3}$ used to weight the in-

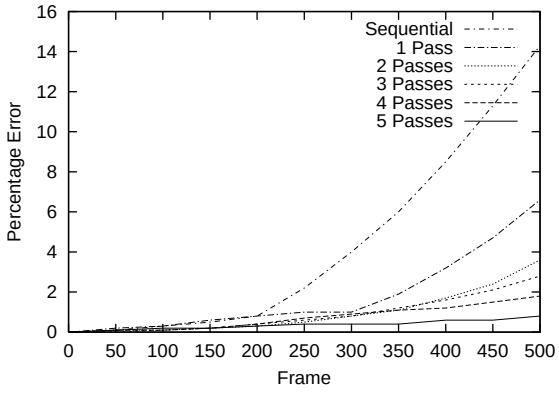

Figure 4. Error in camera position for a 500 frame synthetic test environment.

dividual terms [5]. In situations where the focal length over the sequence is known to be constant, we may calculate the average focal length $\bar{f}_i = \sum_{j=0}^{N} f_{j_x}$, and add an additional term $\omega_4(\bar{f}_i - f_{j_x})^2$ into the summation of Equation 12 that measures the deviation from this average value. Similarly, if the focal length is known to vary, but it has been measured for one or more frames of the sequence (e.g. using vanishing points [5]) then extra terms that account for these deviations can be included in Equation 12 as well. After identifying the ADQ that minimizes Equation 12, the same residual measure is used in a non-linear optimization algorithm to improve the estimate of the ADQ.

We have found that applying random sampling techniques to the problem of self-calibration significantly increases the usefulness and applicability of the basic linear algorithm, especially for longer video sequences. In the next section we present results for both real and synthetic environments that demonstrate the benefits of both the self-calibration algorithm and our hierarchical merging scheme.

5. Results

To test the accuracy of the projective reconstruction algorithm outlined above, we have measured the amount of error in camera position over the course of a simple 500 frame synthetic test sequence. The synthetic camera path was created using a spline, to mimic a hand-held camera moving in approximately a straight line. Every 25 frames, the camera position was randomly perturbed from this line. The camera focal length was set at a constant value of 900 pixels. Camera viewing direction was also controlled using a spline, initially set to look 30 degrees away from the camera path, rotating towards 90 degrees at the end of the sequence, and randomly perturbed at 25 frame intervals.

Feature tracks were generated by positioning points randomly in front of cameras in the sequence. Starting from frame i in front of which a point was positioned, a feature track was generated by projecting the point back into frames $i+1, i+2, \ldots$, and adding Gaussian noise of standard deviation 0.5 pixels. Additionally, 15% of the feature projections were made outlying by adding Gaussian noise of standard deviation 10 pixels. The feature track was extended backwards from frame i to $i-1, i-2, \ldots$ in a similar fashion. In both directions, the feature track was terminated at random 5% of the time, or when it moved out of the image boundary. Feature tracks were generated in this fashion until 300 projections were present in each frame of the sequence. In total, around 5000 features were created for each sequence, with an average track length of just over 30 frames.

For each synthetic sequence, a projective reconstruction was built, and then upgraded from a projective to metric framework. Because self-calibration would introduce additional errors, this upgrade was performed using known projections of ground-truth data into two frames of the sequence (see [14], page 259). This allows us to examine the error in the reconstruction, without it being affected by incorrect self-calibration.

After metric structure and camera motion was obtained, the error in camera position was measured as a percentage of the total length of the camera path, when compared to the original synthetic data. Figure 4 shows results for increasing numbers of pair-wise reconstruction passes, at 50 frame intervals, averaged over 5 different sequences. Also shown are results for a purely sequential merging scheme (no merging passes), where sub-sequence reconstructions were merged into the first using the same robust projectivity estimation algorithm as for the hierarchical scheme. Sequential merging took 4 minutes, with the hierarchical scheme requiring an additional 1, 4, 10, 17 or 27 minutes as the number of passes increased.

As the figure clearly shows, sequential merging introduces a significant amount of drift over the course of the sequence. Hierarchical merging, on the other hand, is capable of producing a calibration where the amount of observed drift is drastically reduced. Even after only 2 pair-wise merging passes, drift is reduced by a factor of 4. As the number of passes increases, drift falls to less than 1.0%, but computation time begins to increase rapidly, requiring a total of 31 minutes for 5 merging passes. This extra computation time is due almost entirely to the increased complexity of bundle adjustment.

[5] Typically, we use weights of 4, 2 and 1 respectively. If the aspect ratio or principal point are not known, their constraints can simply be ignored by setting their respective weights to zero

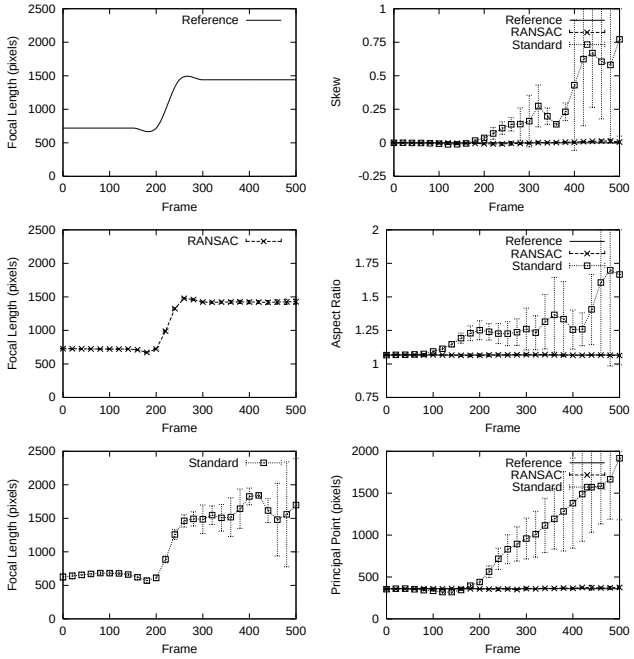

Figure 5. Comparison of our RANSAC-based self-calibration algorithm against a standard linear method.

Figure 6. Examples of augmented video sequences (see text for details).

5.1. Self-calibration

To test the performance of our random-sampling approach to self-calibration, similar 500 frame synthetic sequences were generated, with an additional change in focal length modelled using a spline to simulating a camera zoom. For each sequence, a projective reconstruction was generated using the 5-pass merging scheme, and then the self-calibration algorithm described in Section 4 was applied. We also applied the standard linear algorithm given in [22], using all available projection matrices. Figure 5 shows the results in graph form for each intrinsic parameter. The left-hand column shows graphs for the reference focal length (top), the recovered focal length obtained with the RANSAC algorithm (middle), and the focal length with the standard algorithm (bottom). Mean and standard deviation are shown. In the right-hand column, similar graphs are shown for camera skew (top), aspect ratio (middle), and the X component of the principal point (bottom). Reference values for skew, aspect ratio and principal point were 0, 1.066, and 360 respectively. The accuracy benefits of applying random sampling techniques to the problem of self-calibration can be clearly seen, with all parameters exhibiting serious deviations from the reference values when standard linear estimation was applied.

5.2. Example Augmented Sequences

Figure 6 shows frames from two video sequences that have been calibrated using the techniques described in this paper, and then augmented with synthetic objects. In each case, three original frames are shown in a row, with the same three augmented frames beneath. All sequences were captured at PAL resolution using a Canon-MV1 digital video camera, and are included in the material accompanying this paper [6]. Synthetic objects were rendered and composited using OpenGL.

The top two rows in Figure 6 show a 420 frame sequence containing a significant amount of camera zoom. As can be seen in the accompanying video, our calibration algorithm has been able to model this zoom correctly, and the synthetic object remains accurately registered to the scene throughout the sequence. Also notice that the relative size of the synthetic object to its immediate surrounds remains constant, indicating that the change in camera focal length has been accurately recovered. Feature selection and guided tracking for this sequence took approximately 25 minutes, and 5-pass hierarchical merging took an additional 28 minutes.

The bottom two rows in Figure 6 show frames from a panning sequence also captured by a hand-held camera.

[6] These sequences are also available for download from *http://aig.cs.man.ac.uk/icarus/ismar02*

This sequence was calibrated using a modified version of the algorithm presented in this paper, whereby a projective reconstruction was built using estimates of the planar homography relating each frame, rather than the epipolar geometry. Note that we are able to easily discount feature tracks on the moving vehicles because they do not satisfy the planar homography describing inter-frame camera motion. The estimate of the absolute conic in our RANSAC calibration algorithm was achieved using the techniques described in [8]. The modifications required to support this type of camera motion were modest, showing the flexibility of our calibration system. Tracking time for this 435 frame sequence was 21 minutes, and calibration took an additional 7 minutes. Note that all results presented are those produced directly by our algorithm, without additional bundle adjustment.

6. Conclusion

In this paper we have described a novel and accurate approach to the problem of calibrating a moving video camera, with the aim of augmenting a video sequence with synthetic objects. We have discussed modifications to a standard feature tracking algorithm that uses an estimate of camera motion to remove incorrectly tracked features, and a hierarchical merging scheme for producing a complete projective reconstruction of the feature locations and camera motion. Finally, we presented a new RANSAC-based approach to the problem of self-calibration.

Much work still remains to be done to produce a completely reliable camera calibration algorithm. Most notably, improvements to the key-frame selection algorithm could be made, using techniques similar to [28]. Similarly, methods of avoiding situations where the scene structure is dominated by a single plane should also be investigated [22]. The number of pair-wise merging passes is currently chosen by the user, but automatic methods that attempt to determine when further merging passes are unnecessary should be feasible. One clear disadvantage of using feature tracking algorithms (as opposed to feature matching as in [9]) is that tracking will fail in instances where there is a large amount of inter-frame camera motion. In this case, feature matching algorithms should be employed in an attempt to allow tracking to continue. Finally, the pair-wise merging scheme needs to be compared against the hierarchical triplet-based approach proposed by Fitzgibbon and Zissermann [9]. Although a similar key-frame selection process to the one described in this paper could clearly be applied to triplet-based merging schemes, the benefits of the complex trifocal tensor estimation algorithms need to be properly evaluated.

In addition to the problem of accurate camera registration, there are many other aspects of the environment that need to be modelled if synthetic objects are to be composited in a believable way. Most importantly, the significant sources of illumination in the scene must be identified so that the objects may be shaded correctly and cast believable shadows. Similarly, the blurring and noise of the camera lens in the video sequence should be reproduced, so that similar effects can be added to the synthetic objects.

A demonstration version of the ICARUS software, used to produce all the results shown in this paper, will shortly be available for download from *http://aig.cs.man.ac.uk/icarus*

Acknowledgements

The authors would like to thank their colleagues in the Advanced Interfaces Group for helpful discussions and comments on this paper. The buddha and dragon models shown in Figure 6 were obtained from the Stanford 3D Scanning Repository. This work was supported by EPSRC grant number GR/M14531, entitled "REVEAL: Reconstruction from Video of Environments with Accurate Lighting" and by EC grant IST-28707 (IST IV-4-2) entitled "ARIS: Augmented Reality Image Synthesis".

References

[1] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. on Pattern Matching and Machine Intelligence*, 17(6):562–575, June 1995.
[2] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequences. In *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695, 1996.
[3] F. Bookstein. Fitting conic sections to scattered data. *Computer Vision, Graphics and Image Processing*, 9:56–71, 1979.
[4] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Motion and structure causally integrated over time. *IEEE Trans. on Pattern Matching and Machine Intelligence*, 24(4):523–535, April 2002.
[5] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. *Proc. British Machine Vision Conference, Nottingham*, 2:382–391, September 1999.
[6] K. Cornelis, M. Pollefeys, and L. V. Gool. Tracking based structure and motion recovery for augmented video productions. In *Proc. ACM Symposium on Virtual Reality and Software Technology (VRST)*, Alberta, Canada, November 2001.
[7] K. Cornelis, M. Pollefeys, M. Vergauwen, and L. V. Gool. Augmented reality from uncalibrated video sequences. In *3D Structure from Images - SMILE 2000, Lecture Notes in Computer Science*, volume 2018, pages 144–160. Springer Verlag, 2001.
[8] L. de Agapito, R. I. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. In *Proc. Computer Vision and Pattern Recognition*, June 1999.
[9] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, pages 311–326. Springer-Verlag, June 1998.
[10] S. Gibson, J. Cook, R. J. Hubbold, and T. L. J. Howard. ICARUS: Interactive reconstruction from uncalibrated image sequences. In *ACM SIGGRAPH 2002 Sketches and Applications Programme*, San Antontio, Texas, USA, July 2002.
[11] S. Gibson, R. J. Hubbold, J. Cook, and T. L. J. Howard. Interactive reconstruction of virtual environments from video sequences. Submitted for Publication, April 2002.
[12] C. Harris and M. Stephens. A combined corner and edge detector. pages 147–151, 1988.
[13] R. Hartley. Euclidean reconstruction from uncalibrated views. *Applications of Invariance in Computer Vision, LNCS-Series*, 825:237–256, 1994.
[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, UK, 2000.
[15] B. Heigl, D. Paulus, and H. Niemann. Tracking points in sequences of color images. In *Proc. 5th Open German-Russian Workshop on Pattern Recognition and Image Understanding (Lecture Notes in Computer Science)*, 1998.
[16] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, pages 981–991, 1999.
[17] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Proc. International Conference on Computer Vision*, July 2001.
[18] D. Nistér. Frame decimation for structure from motion. In *Proc. SMILE 2000*, Dublin, Ireland, July 2000.
[19] D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proc. ECCV 2000*, pages 649–663, Dublin, Ireland, June 2000.
[20] D. Oram. *Projective Reconstruction and Metric Models from Video Sequences*. PhD thesis, Dept. of Computer Science, University of Manchester, September 2001.
[21] M. Pollefeys, R. Koch, and L. van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.
[22] M. Pollefeys, F. Verbiest, and L. V. Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proc. ECCV 2002*, Copenhagen, Denmark, May 2002.
[23] M. Pollefeys, M. Vergauwen, K. Cornelis, and L. V. Gool. 3d acquisition of archaeological heritage from images. In *Proc. CIPA conference, International Archive of Photogrammetry and Remote Sensing*, 2001.
[24] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
[25] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings of the International Symposium on Augmented Reality (ISAR'2000)*, 2000.
[26] R. Swaminathan and S. K. Nayar. Non-metric calibration of wide angle lenses and polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), October 2000.
[27] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
[28] P. H. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, August 1999.
[29] P. H. Torr and A. Zisserman. MLESAC: A new robust estimator with application to existing image geometry. *Computer Vision and Image Understanding*, pages 138–156, April 2000.
[30] W. Triggs. Auto-calibration and the absolute conic. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, 1997.