Noname manuscript No.
(will be inserted by the editor)

Camera Network Calibration and Synchronization from Silhouettes in Archived Video

Sudipta N. Sinha · Marc Pollefeys

Abstract In this paper we present an automatic method for calibrating a network of cameras that works by analyzing only the motion of silhouettes in the multiple video streams. This is particularly useful for automatic reconstruction of a dynamic event using a camera network in a situation where precalibration of the cameras is impractical or even impossible. The key contribution of this work is a RANSAC-based algorithm that simultaneously computes the epipolar geometry and synchronization of a pair of cameras only from the motion of silhouettes in video.

Our approach involves first independently computing the fundamental matrix and synchronization for multiple pairs of cameras in the network. In the next stage the calibration and synchronization for the complete network is recovered from the pairwise information. Finally, a visual-hull algorithm is used to reconstruct the shape of the dynamic object from its silhouettes in video. For unsynchronized video streams with sub-frame temporal offsets, we interpolate silhouettes between successive frames to get more accurate visual hulls. We show the effectiveness of our method by remotely calibrating several different indoor camera networks from archived video streams.

1 Introduction

For over a decade now, researchers in computer vision have been interested in digitizing in 3D, time-varying events that have been recorded by video cameras from multiple viewpoints. Often the events involve performances by human actors. The eventual goal is to allow the viewer to observe the event from any arbitrary viewpoint. This is called *free-viewpoint video* and this has promising applications in 3D tele-immersion, in digitizing rare cultural performances and sports action and in generating content for 3D video-based realistic training and demonstrations for surgery, medicine and other technical fields.

In 1997, Kanade et. al. [16] coined the term *virtualized reality* and reconstructed real life scenes involving humans using a large cluster of cameras in an indoor environment. Since then, various systems have been developed that can digitize human subjects performing various actions [6–8, 10, 11, 21, 25] – these systems

Sudipta N. Sinha
Department of Computer Science,
UNC Chapel Hill, USA
E-mail: ssinha@cs.unc.edu
Current Affiliation: Microsoft Research, Redmond, USA
E-mail: sudipta.sinha@microsoft.com

Marc Pollefeys
Department of Computer Science,
UNC Chapel Hill, USA
E-mail: marc@cs.unc.edu
Current Affiliation: Institute of Computational Science,
ETH Zurich, Switzerland
E-mail: marc.pollefeys@inf.ethz.ch

use an indoor room-sized camera setup that typically consists of 8 to 15 synchronized cameras recording at 15 to 30 frames per second. We will refer to such an arbitrary configuration of cameras as a *camera network*.

Currently in all multi-camera systems [6–8, 10, 11, 21, 25], calibration and synchronization must be done during an offline calibration phase before the actual video is captured. Someone must be physically present in the scene with a specialized calibration object such as a planar calibration grid or a point LED and special calibration data has to be collected. This makes the process of camera deployment and acquisition fairly tedious. Multiple calibration sessions are often required over a longer duration, as there is no easy way to maintain the calibration.

Despite the recent success of structure-from-motion techniques for uncalibrated sequences, they cannot be used to reliably calibrate camera networks as they rely on automatic feature matching which typically fails when camera pairs have a very wide baseline and very few interest point correspondences actually exist between such pairs. Since these cameras observe an event from widely separated viewpoints, the background views in these cameras often barely overlap. However, silhouettes of the same foreground object or objects are observed from these viewpoints and can thus provide the required correspondences.

In this paper, we propose a flexible technique which can recover all the necessary information from silhouettes present in the recorded video streams – thus eliminating the need for an explicit offline calibration phase before the video capture. This is a great benefit for surveillance systems and makes it possible to remotely calibrated camera networks deployed in hazardous environments. Since our approach is based on silhouettes, it is particularly useful for multi-camera shape-from-silhouette systems [17, 6, 21] as visual-hulls can now be reconstructed from uncalibrated and unsynchronized video streams.

At the core of our approach, is a robust RANSAC-based algorithm [2] that computes the epipolar geometry by analyzing the silhouettes of a moving object in a video. The epipole positions are randomly hypothesized at every RANSAC iteration and a model for the epipolar geometry is generated via the epipolar line homography parameterization; this is then efficiently verified using all the available data. Random sampling is used for exploring the 4D space of possible epipole positions as well as for dealing with outliers in the silhouette data. This algorithm is based on the constraints arising from the correspondence of frontier points and epipolar tangents [12, 22, 32] of silhouettes in two views.

We first independently compute the epipolar geometry and temporal offset between various pairs of cameras in the network. Next, the synchronization of the complete network is robustly recovered. The network calibration is recovered in a stratified way – a projective reconstruction is incrementally computed from the epipolar geometry estimates and the two view matches. This is then upgraded to a metric one using self-calibration. Finally, the camera parameters are refined using a standard bundle adjustment step [30]. The effectiveness of our approach is demonstrated by remotely calibrating camera networks from archived multi-view video streams previously acquired by researchers and thereby reconstructing the recorded events using a shape from silhouette approach. Preliminary versions of the proposed approach appeared in [26–28].

2 Related Work

The recovery of camera pose from silhouettes was studied in depth by [15, 22, 31, 32] in the past. Recently there has been some renewed interest in the problem [4, 12, 14]. However, most of these techniques can be applied only in specific settings and have requirements that render them impractical for general camera networks observing an unknown dynamic scene. These include that the observed object be static [15, 12], the use of a specific camera configuration (at least partially circular) [14, 32], the use of an orthographic projection model [12, 31], and a good initialization [4, 33].

In our method, we take advantage of the fact that a camera network observing a dynamic object records many different silhouettes, yielding a large number of epipolar constraints that need to be satisfied by every camera pair. Our algorithm is based on the constraints arising from the correspondence of frontier points and epipolar tangents for silhouettes in two views. This constraint was also used in previous work [12, 22, 24, 32] but either for specific camera motion or restricted camera models or in the situation where a good initialization was already available.

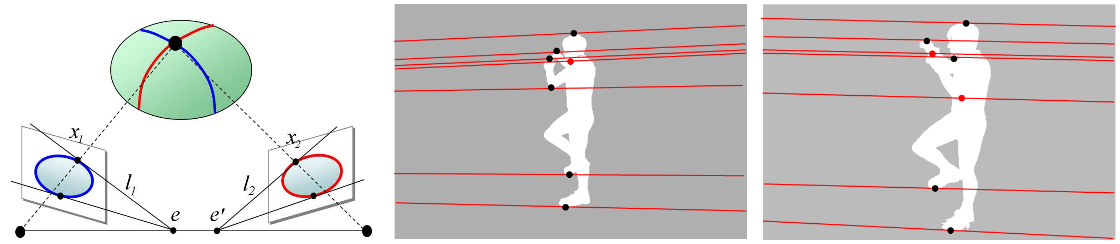


Fig. 1 (a) Frontier points are the true correspondences on silhouettes in two views. A pair of frontier points x_1 and x_2 are shown along with the corresponding epipolar tangents l_1 and l_2 respectively. (b) While many frontier points may exist on the silhouettes of a human, they are hard to detect unless the epipole locations are known.

When a solid object is seen in two views, the only true point correspondences on the apparent contour occur at special locations called *frontier points*. In Figure 1(a), one pair of frontier points is denoted by x_1 and x_2 respectively. Note that the viewing rays that correspond to a matching pair of frontier points such as x_1 and x_2 must intersect at a true surface point in the tangent plane of the surface. The contour generators or rims must also intersect at such a surface point. This point, along with the camera baseline, defines an epipolar plane that must be tangent to the surface. This gives rise to corresponding epipolar lines such as l_1 and l_2, which are tangent to the silhouettes at the frontier points. Frontier point correspondence does not extend to more than two views in general. A convex shape, fully visible in two views, can have exactly two pairs of frontier points. For a non convex shape such as a human figure, there can be several potential frontier points, but many of them will be occluded or will not appear on the silhouette (see Figure 1(b)).

If the location of the epipole in the image plane is known, matching frontier points can be detected by computing tangents to the silhouettes from the epipoles. However, when the epipole locations are unknown, it is difficult to directly recover the frontier points. In [32] Wong and Cipolla searched for outermost epipolar tangents for circular motion. In their case, the existence of fixed entities in the images, such as the horizon and the image of the rotation axis, simplified the search for epipoles. We too use only the *extremal* frontier points and outer-most epipolar tangents because, for fully visible silhouettes, these are never occluded. Also, extremal frontier points must lie on the convex hull of the silhouette as well which can be represented more compactly in general.

Furukawa et.al. [12] directly searched for frontier points on a pair of silhouettes to recover the epipolar geometry. Their approach assumes an orthographic camera model and requires accurate silhouettes. It does not work unless there are at least four unoccluded frontier point matches in general position. Hernandez et.al. [14] generalized the idea of epipolar tangencies to the concept of *silhouette coherence*, which numerically measures how well a solid 3D shape corresponds to a given set of its silhouettes in multiple views. They performed camera calibration from silhouettes by solving an optimization problem where *silhouette coherence* is maximized. However they only dealt with circular turntable sequences, which have fewer unknown parameters, so their optimization technique does not generalize to an arbitrary camera network. Boyer [4] also proposed a criterion that back-projected silhouette cones must satisfy such that the true object is enclosed within all of the cones. They used it to refine the calibration of a camera network but their approach requires good initialization.

We first present the method for recovering epipolar geometry for the case where all the cameras are synchronized. We then show how to extend the algorithm to simultaneously recover the epipolar geometry as well as the temporal offset in the unsynchronized case. The use of RANSAC [2] makes our approach robust to silhouette extraction errors which is important for dealing with real datasets where silhouettes extracted automatically using background segmentation will typically have some errors.

3 Epipolar Geometry from Dynamic Silhouettes

Given non trivial silhouettes in a pair of video streams, such as that of a person (see Figure 1(b)), if we can detect matching frontier points in corresponding frames, we can use the 7-point algorithm to estimate

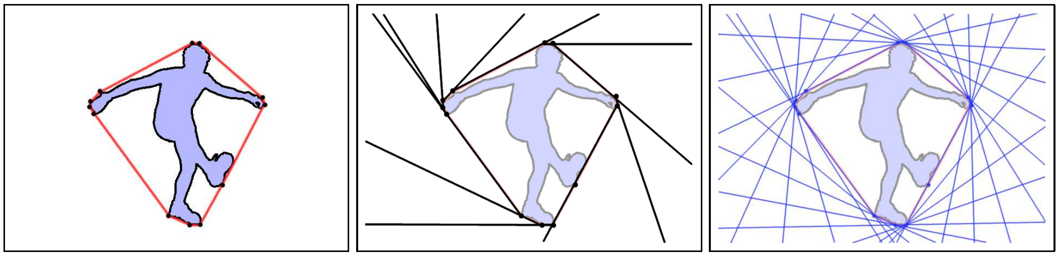


Fig. 2 (a) The Convex Hull of the silhouette in a video frame. (b) The Tangent Table representation (c) The space of all tangents to the convex hull parameterized by θ.

the epipolar geometry by computing the fundamental matrix. When the epipolar geometry is known, pairs of frontier points can be easily detected by computing tangents from the epipoles to the silhouettes. However, it is difficult to directly find matching frontier points without knowing the epipolar geometry. This is a typical "chicken and egg" problem. Since the location of the epipoles is sufficient to determine the frontier points, our approach will consist of randomly guessing the epipoles, thereby generating a hypothesis for the epipolar geometry and then testing it for consensus on the bundle of epipolar tangents to all the silhouettes. This will require an efficient representation for silhouettes that allows fast tangency computations. Our compact representation, described next requires a few hundred bytes per silhouette and allows us to work on thousands of frames from video.

Binary foreground silhouettes are computed for every video frame using standard background segmentation techniques. Instead of explicitly storing the complete silhouette $\mathcal{S}$, we compute and store only the convex hull $\mathcal{H}_{\mathcal{S}}$ and its dual representation for every frame (see Figure 2). This compact representation allows us to efficiently compute outer tangents to silhouettes in long sequences containing potentially thousands of different silhouettes. The convex hull $\mathcal{H}_{\mathcal{S}}$ is represented by an ordered list of k 2D points in the image ($v_1 \ldots v_k$ in counter-clockwise order (ccw)). The 2D lines tangent to $\mathcal{H}_{\mathcal{S}}$ are parameterized by the angle $\theta = 0 \ldots 2\pi$ (in radians) that the line subtends with respect to the horizontal direction in the image. For each vertex v_k, an angular interval $[\theta_k^1, \theta_k^2]$ is computed – this set represents all lines that are tangent to $\mathcal{H}_{\mathcal{S}}$ at the vertex v_k. These tangent lines are directed, i.e. they are consistently oriented with respect to the convex hull. Thus for a direction θ, there is always a unique directed tangent $\mathbf{l}_\theta$.

While a fundamental matrix has seven degrees of freedom (*dofs*), our method randomly samples only in the 4D space of epipoles, because once the epipoles positions are fixed, potential frontier point matches can be determined, and from them the remaining three degrees of freedom of the epipolar geometry can be computed via an epipolar line homography [13]. We propose to use RANSAC not only to handle outliers (erroneous silhouettes) but also to efficiently explore the 4D space of epipole locations. The algorithm is described here assuming synchronized video, but will be extended to the unsynchronized case in Section 5.

To generate a hypothesis for the epipolar geometry (the fundamental matrix is denoted by $\mathbf{F}_{\mathrm{ij}}$), we randomly guess the position of epipoles $\mathbf{e}_{\mathrm{ij}}$ and $\mathbf{e}_{\mathrm{ji}}$ in the two views. The pencil of epipolar lines in each view centered on the epipoles forms a 1D projective space [13]. Three pairs of corresponding epipolar lines are sufficient to determine a epipolar line homography $\mathbf{H}_{\mathrm{ij}}^{-T}$, that uniquely determines the transfer of epipolar lines and the fundamental matrix is then given by $\mathbf{F}_{\mathrm{ij}} = [\mathbf{e}_{\mathrm{ij}}]_\times \mathbf{H}_{\mathrm{ij}}$. These three pairs of epipolar lines (epipolar tangents in our case) in the two views will be denoted by $\{\mathbf{l}_i^k\}$ and $\{\mathbf{l}_j^k\}$ in views i and j respectively where $k = 1 \ldots 3$. These are related as follows: $[\mathbf{l}_j^k]_\times \; \mathbf{H}_{\mathrm{ij}} \; \mathbf{l}_i^k \;=\; 0$ where $k = 1 \ldots 3$. Solving for $\mathbf{H}_{\mathrm{ij}}$ generates a hypothesis for the $\mathbf{F}_{\mathrm{ij}}$ using the relation above. We now describe the details of how the epipoles are sampled and how the hypothesis is verified using all the silhouettes.

3.1 Hypothesis Generation

At every RANSAC iteration, we randomly choose a pair of corresponding frames from the two sequences. In each of the two frames, we randomly sample two directions and obtain outer tangents to the silhouettes corresponding to these two directions. The first direction θ_1 is sampled from the uniform distribution $\mathbf{U}(0, 2\pi)$, while the second direction θ_2 is chosen as $\theta_2 = \theta_1 - x$, where x is drawn from the normal

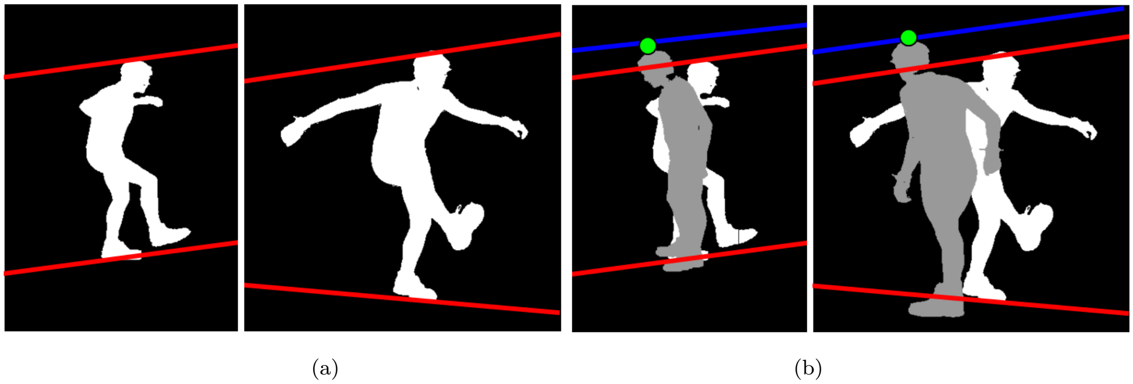


Fig. 3 (a) For corresponding random frames, two random directions are sampled in each image. The intersection of the corresponding epipolar tangents generates the epipole hypothesis. (b) Another pair of corresponding frames is randomly selected and the outermost epipolar tangents to the new silhouette are computed (shown in blue). The three pairs of lines can be used to estimate the epipolar line homography.

distribution $\mathbf{N}(\pi, \frac{\pi}{2})$. The intuition behind this sampling strategy is that epipoles commonly tend to lie far away from the principal point in the image plane resulting in epipolar lines that are often close to parallel.

Note that alternative approaches could also be used for sampling the epipoles. In the calibrated case, a better strategy would be to sample both the epipole directions randomly on a sphere [19]. However, in the uncalibrated case (unknown focal length), this is equivalent to random sampling on an arbitrary ellipsoid and this method would produce results comparable to our approach. Although our epipole sampling is based on the shape of silhouettes in the data, the random selection of silhouettes from video and the variability of silhouette shapes in long sequences neutralizes the bias that would occur if we repeatedly used the same silhouette.

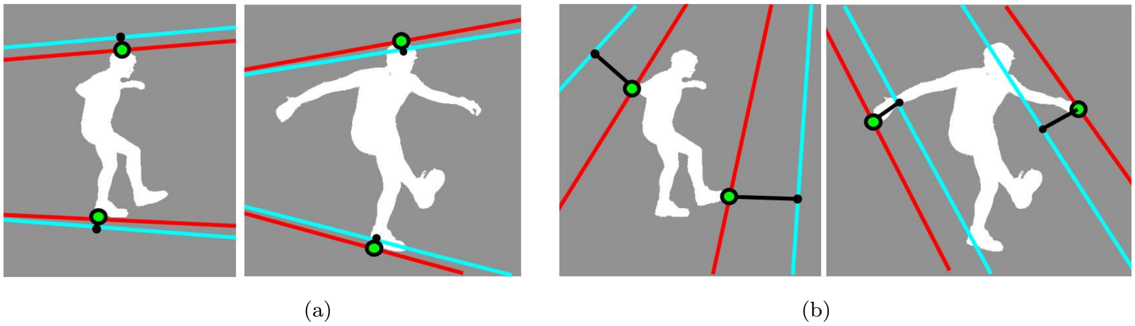


Fig. 4 The epipolar transfer error distribution is computed for the hypothesized epipolar geometry model using all the silhouettes in video. Here only a single pair of frames are shown. The original outer tangents are shown in red while the transferred epipolar lines are shown in blue. (a) For a good hypothesis, the epipolar transfer error is small. (b) The situation is shown for a bad hypothesis.

The convex hull of the silhouette contains a unique directed tangent for each direction that is sampled. The two tangent lines in the first view are denoted by $\mathbf{l}_i^1$ and $\mathbf{l}_i^2$, while those in the second view are denoted by $\mathbf{l}_j^1$ and $\mathbf{l}_j^2$ respectively (these are shown in red in Figure 3(a)). [1] The intersections of the pair of epipolar tangents produce the hypothesized epipoles $\mathbf{e}_{\mathrm{ij}}$ and $\mathbf{e}_{\mathrm{ji}}$ in the two views. We next randomly select another pair of frames and compute outer tangents from the epipoles $\mathbf{e}_{\mathrm{ij}}$ and $\mathbf{e}_{\mathrm{ji}}$ to the silhouettes (actually to their convex hulls) in both views. If there are two pairs of outer tangents, we randomly select one. This third

[1] If silhouettes are clipped, the second pair of tangents is chosen from another frame.

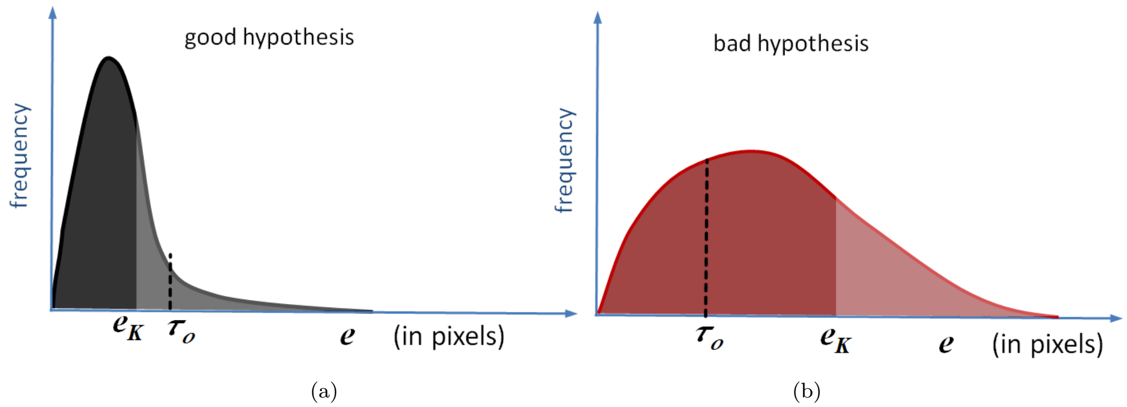


Fig. 5 (a) The error distribution is shown for a good hypothesis. Note that the $\mathbf{K}^{\text{th}}$–quantile e_K is much smaller than $\tau_{\mathbf{o}}$. (b) For a bad hypothesis, the error distribution is much more spread out and the $\mathbf{K}^{\text{th}}$–quantile e_K is greater than $\tau_{\mathbf{o}}$.

pair of lines is denoted by $\mathbf{l}_i^3$ and $\mathbf{l}_j^3$ respectively (these are shown in blue in Figure 3(b)). Now $\mathbf{H}_{\text{ij}}$, the epipolar line homography, is computed from the three corresponding lines [2] $\{\mathbf{l}_i^k \leftrightarrow \mathbf{l}_j^k\}$ where $k = 1 \ldots 3$. The quantities ($\mathbf{e}_{\text{ij}}$, $\mathbf{e}_{\text{ji}}$, $\mathbf{H}_{\text{ij}}$) form the model hypothesis in each iteration of our algorithm.

3.2 Model Verification

Each randomly generated hypothesis for the epipolar geometry is evaluated using all the data available. This is done by computing outer tangents from the hypothesized epipoles to the whole sequence of silhouettes in each of the two views. For unclipped silhouettes, we obtain two tangents per frame, whereas for clipped silhouettes there may be one or even zero tangents. Every epipolar tangent in the first view is transferred through $\mathbf{H}_{\text{ij}}$ to the second view (see Figure 4), and the reprojected epipolar transfer error e is computed based on the shortest distance from the original point of tangency to the transferred line.

$$e = d(\mathbf{x}_{\text{i}}, \mathbf{l}_{\text{i}}^t) + d(\mathbf{x}_{\text{j}}, \mathbf{l}_{\text{j}}^t) \tag{1}$$

where $d(\mathbf{x}, \mathbf{l})$ represents the shortest distance from a 2D point $\mathbf{x}$ to a 2D line $\mathbf{l}$, and $\mathbf{x}_{\text{i}}$ and $\mathbf{x}_{\text{j}}$ represent the point of tangencies in the two images which when transferred to the other view, gives rise to epipolar lines $\mathbf{l}_{\text{j}}^t$ and $\mathbf{l}_{\text{i}}^t$ respectively.

Figure 5 shows the typical symmetric epipolar transfer error distributions. We use an outlier threshold denoted by $\tau_{\mathbf{o}}$ to classify a certain hypothesis as good or bad. The value of $\tau_{\mathbf{o}}$ is automatically computed (described next) and is typically in the range of 0.005–0.02 % of the image width. The $\mathbf{K}^{th}$ quantile of the error distribution denoted by $e_{\mathbf{K}}$ is computed (in all our experiments, $\mathbf{K} = 0.75$, or 75%). If $e_{\mathbf{K}} \leq \tau_{\mathbf{o}}$, then the epipolar geometry model is considered a promising candidate and is recorded.

The points of tangency can remain stationary over successive frames of video. This gives rise to duplicate matches, which must be removed in order to compute a meaningful error distribution. While computing epipolar tangents one frame at a time for the whole sequence, we check the potential frontier point matches for duplicates using spatial hashing for local search in the 2D images.

The RANSAC-based algorithm looks for $\mathbf{n_S}$ promising candidates. These candidates are then ranked based on the inlier count and the best ones are further refined. A stricter threshold $\tau_{\mathbf{in}}$ of 1 pixel is used to determine the tangents which are inliers. While evaluating a hypothesis, we maintain a count of the tangents that exceed the outlier threshold $\tau_{\mathbf{o}}$ and reject a hypothesis early, when a partial outlier count indicates that the total expected outlier count is likely to be exceeded (i.e. with high probability). This

[2] There are two ways to pair $\{\mathbf{l}_i^1, \mathbf{l}_i^2\}$ with $\{\mathbf{l}_j^1, \mathbf{l}_j^2\}$, and we generate and check both hypotheses.

allows us to abort early whenever the model hypothesis is completely inaccurate, avoiding the redundancy of computing outer tangents from epipoles to all the silhouettes for many completely wrong hypotheses.

The best 25% of the promising candidates are then refined using multiple iterations of nonlinear (Levenberg Marquardt) minimization and guided matching. During each nonlinear minimization step, the total symmetric epipolar distance in both images for the set of inlier point correspondences is minimized. During each guided matching step, the epipole positions are recovered from the current estimate of the fundamental matrix. Next, tangents are again recomputed from these epipoles to all the silhouettes. The inlier count steadily increases, since more epipolar tangents are included as the estimate of the fundamental matrix becomes more accurate. The final solution is obtained when the inlier count stabilizes.

In practice, many of the promising candidate solutions for the epipolar geometry from the RANSAC step, when iteratively refined converge to the same solution. Therefore we stop when three promising candidates converge to the same solution. The refined solution with the highest number of inliers is the final one. Comparing the Frobenius norm of the difference of two normalized fundamental matrices is not a suitable measure for comparing two fundamental matrices, so we use the statistical measure proposed by [34]. The complete method for computing the epipolar geometry is summarized in Algorithm 1.

```
Input: Pair of Sequences $\{\mathcal{S}_i\}$ and $\{\mathcal{S}_j\}$ of silhouettes
Output: Fundamental Matrix $\mathbf{F_{ij}}$

$\{\mathcal{H}_{\mathcal{S}_i}\} \leftarrow$ Compute Convex Hull And Tangent-Tables($\{\mathcal{S}_i\}$);
$\{\mathcal{H}_{\mathcal{S}_j}\} \leftarrow$ Compute Convex Hull And Tangent-Tables($\{\mathcal{S}_j\}$);
$\tau_{\mathbf{o}} \leftarrow$ Compute Outlier Threshold        (Section 3.3);
$n_S \leftarrow \max(2\tau_{\mathbf{o}}, 10)$;
candidates $\leftarrow$ { } ;

repeat
    (F, model) $\leftarrow$ Make Hypothesis      (Section 3.1) ;
    Evaluate (F)                         (Section 3.2) ;
    if Promising Solution ;
        candidates $\leftarrow$ candidates $\cup$ (F, model) ;
until ( |candidates| $==$ $n_S$  ||  maximum iterations exceeded)

$C_k \leftarrow$ Rank And Find Best (k, candidates)  using inlier count ;

NonLinear Minimization And Iterative Refinement($\{C_k\}$);

return  Rank And Find Best (1, $C_k$)  using inlier count ;
```

Algorithm 1: An overview of the algorithm for computing the epipolar geometry from silhouettes in a pair of synchronized videos. The hypothesis generation and verification steps are described in Section 3.1 and Section 3.2 respectively. The automatic approach for tuning relevant parameters are described in Section 3.3.

3.3 Automatic Parameter Tuning:

Our algorithm has a few critical parameters, the total number of RANSAC iterations $\mathbf{N}$, the number of promising candidates $\mathbf{n_S}$, and the outlier threshold $\tau_{\mathbf{o}}$. We automatically determine these parameters from the data, making our approach completely automatic and convenient to use. The number of iterations $\mathbf{N}$ depends on the desired promising candidate count denoted by $\mathbf{n_S}$.

In our implementation, $\mathbf{N}$ is chosen as min ($\mathbf{n}$, $\mathbf{N_I}$) where $\mathbf{n}$ is the number of iterations required to find $\mathbf{n_S}$ candidates ($\mathbf{N_I}$ is set to 10^6 in all our experiments). $\mathbf{n_S}$ is determined by the outlier threshold $\tau_{\mathbf{o}}$. A tighter (i.e. lower) outlier threshold can be used to select very promising candidates but such occurences are rare. If the threshold is set higher, promising candidates are obtained more frequently but at the cost of finding a few ambiguous ones as well. When this happens, a larger set of promising candidates must be analyzed. Thus, $\mathbf{n_S}$ is set to max ($2\tau_{\mathbf{o}}$, 10) in our implementation.

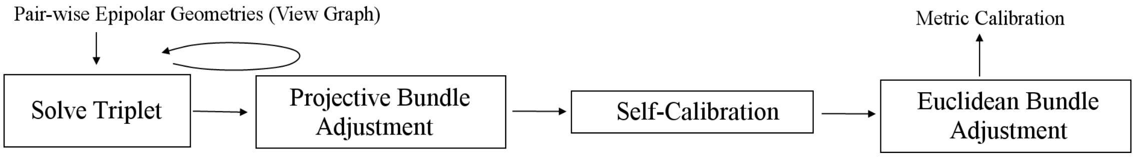

Pair-wise Epipolar Geometries (View Graph)
Solve Triplet
Projective Bundle Adjustment
Self-Calibration
Euclidean Bundle Adjustment
Metric Calibration

Fig. 6 Overview of Camera Network Calibration from Epipolar Geometries.

We also compute $\tau_{\mathbf{o}}$ automatically from the data during a set of preliminary RANSAC iterations. During this stage, the hypothesis and verification iterations proceed as described earlier, but these are only used to compute $\tau_{\mathbf{o}}$, and the promising candidates found at this stage are not used later. We start with a large value of $\tau_{\mathbf{o}}$ (= 50 pixels in our implementation) and iteratively lower it as follows. We compare $\mathbf{e_K}$, the $\mathbf{K}^{\text{th}}$–quantile (K = 75) with the current value of $\tau_{\mathbf{o}}$. If $\mathbf{e_K} < \tau_{\mathbf{o}}$, we simply reset $\tau_{\mathbf{o}}$ to the smaller value $\mathbf{e_K}$. If $\tau_{\mathbf{o}} \leq \mathbf{e_K} \leq (\tau_{\mathbf{o}} + 1)$, then we increment a counter $\mathbf{C}_{\tau_{\mathbf{o}}}$. If $\mathbf{e_K} > (\tau_{\mathbf{o}} + 1)$, then the value of $\tau_{\mathbf{o}}$ is not changed. We reset $\mathbf{C}_{\tau_{\mathbf{o}}}$ to zero whenever the threshold is lowered. If either $\tau_{\mathbf{o}}$ falls below 0.005% of the image width or $\mathbf{C}_{\tau_{\mathbf{o}}}$ becomes equal to 0.005% of the image width, we accept the current estimate of $\tau_{\mathbf{o}}$ as final.

4 Camera Network Calibration

We next consider the problem of recovering full camera calibration from pairwise epipolar geometries. Given a sufficient number of edges in a view graph where each edge represents an estimate of the respective fundamental matrix, our goal is to recover the Euclidean camera matrices for all the cameras in the network. An overview of our approach is described in Figure 6.

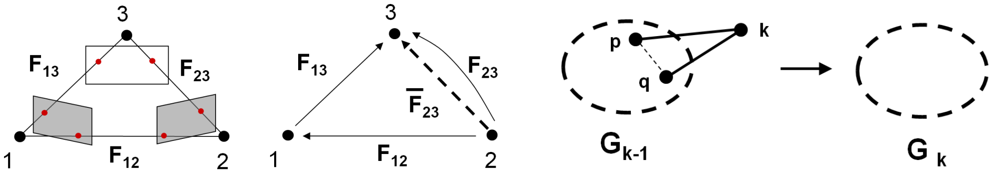
An important step in this approach is to compute an accurate projective reconstruction of the camera network from epipolar geometries and two view matches. We start by first recovering a triplet of projective cameras from the fundamental matrices between the three views. Using an incremental approach, we add a new camera to the calibrated network by resolving a different triplet of cameras each time. Each time a new camera is added, all the parameters corresponding to the cameras and 3D points are refined using *projective bundle adjustment*. Finally when a full projective reconstruction is available, standard techniques for self-calibration and Euclidean (metric) bundle adjustment is used to compute the final metric camera calibration.

In our silhouette-based calibration work, frontier point correspondences do not generalize to more than two views. In a three-view case, the frontier points in the first two views do not correspond to those in the last two views. Although three-view correspondences, called *triple points*, do exist on the silhouette as reported by [10,18], they are hard to extract from uncalibrated images. Thus, we are restricted to only two-view correspondences over different pairs in our camera network and so cannot directly adopt an approach like that of [23].

Instead, we incrementally compute a full projective reconstruction of a camera network from these two-view correspondences and the corresponding fundamental matrices. Levi and Werman [20] studied the following problem. Given only a subset of all possible fundamental matrices in a camera network, when is it possible to recover all the missing fundamental matrices? They were mainly concerned with theoretical analysis, and their algorithm is not suited for the practical implementation of computing projective reconstructions from sets of two-view matches in the presence of noise.

4.1 Resolving Camera Triplets

Given any two fundamental matrices between three views, it is not possible to compute three consistent projective cameras. The two fundamental matrices can be used to generate canonical projective camera pairs $\{\mathbf{P}_1, \mathbf{P}_2\}$ and $\{\mathbf{P}_1, \mathbf{P}_3\}$, respectively. However these do not correspond to the same projective frame. $\mathbf{P}_3$ must be chosen in the same projective frame as $\mathbf{P}_2$, and the third fundamental matrix is required to enforce this. These independently estimated fundamental matrices are denoted by $\mathbf{F}_{12}$, $\mathbf{F}_{13}$, and $\mathbf{F}_{23}$,



(a) (b) (c)

Fig. 7 (a) Three nondegenerate views for which the fundamental matrices have been estimated independently. (b) Family of solutions for the third fundamental matrix ($\overline{\mathbf{F}}_{23}$), compatible with the other two ($\mathbf{F}_{12}$ and $\mathbf{F}_{13}$). We look for a compatible solution closest to the measured $\mathbf{F}_{23}$. (c) New camera k incrementally linked to a calibrated network by resolving a triplet involving two other cameras in G_{k-1}.

while the unknown projective cameras will be denoted by $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$, respectively (see Figure 7). The three fundamental matrices are said to be compatible when they satisfy the following constraint.

$$\mathbf{e}_{23}^{\mathrm{T}}\ \mathbf{F}_{12}\ \mathbf{e}_{13} = \mathbf{e}_{31}^{\mathrm{T}}\ \mathbf{F}_{13}\ \mathbf{e}_{21} = \mathbf{e}_{32}^{\mathrm{T}}\ \mathbf{F}_{23}\ \mathbf{e}_{12} = 0 \tag{2}$$

The three fundamental matrices available in our case are not compatible because they were independently estimated from two-view correspondences. A linear approach for computing $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$ from three compatible fundamental matrices is described in [13]. However, it is not suitable when the fundamental matrices are not compatible, as in our case. We now describe our linear approach to compute a consistent triplet of projective cameras. As described in [13], given $\mathbf{F}_{12}$, canonical projective cameras, $\mathbf{P}_1$ and $\mathbf{P}_2$ as well as $\mathbf{P}_3$ can be chosen as follows:

$$\begin{gathered}\mathbf{P}_1 = [\mathbf{I}|0] \qquad \mathbf{P}_2 = [[\mathbf{e}_{21}]_\times \mathbf{F}_{12}|\mathbf{e}_{21}] \\ \mathbf{P}_3 = [[\mathbf{e}_{31}]_\times \mathbf{F}_{13}|0] + \mathbf{e}_{31}\mathbf{v}^{\mathrm{T}}\end{gathered} \tag{3}$$

$\mathbf{P}_3$ has been defined up to an unknown 4-vector $\mathbf{v}$ (Eq. 3). By expressing $\mathbf{F}_{23}$ as a function of $\mathbf{P}_2$ and $\mathbf{P}_3$ we obtain the following.

$$\overline{\mathbf{F}}_{23} = [[\mathbf{e}_{32}]_\times \mathbf{P}_3 \mathbf{P}_2^+ \tag{4}$$

The expression for $\overline{\mathbf{F}}_{23}$ is linear in $\mathbf{v}$. Hence, all possible solutions for $\overline{\mathbf{F}}_{23}$ span a 4D subspace of $\mathbb{P}^8$ [20]. We solve for $\mathbf{v}$, which produces the solution closest to the measured $\mathbf{F}_{23}$ in the 4D subspace. $\mathbf{P}_3$ can now be computed by substituting this value of $\mathbf{v}$ into Equation 3. The resulting $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$ are fully consistent with $\mathbf{F}_{12}$, $\mathbf{F}_{13}$, and the matrix $\overline{\mathbf{F}}_{23}$ computed above.

In order to choose $\mathbf{F}_{12}$, $\mathbf{F}_{13}$, and $\mathbf{F}_{23}$ for this approach, we must rank the three fundamental matrices based on an accuracy measure, the least accurate one is assigned to be $\mathbf{F}_{23}$ while the choice of the other two does not matter. To rank the fundamental matrices based on the accuracy of their estimates, their inlier spread score s_{ij} is computed as follows:

$$s_{ij} = \sum_{(u,v)\in \mathrm{P}_i} |u-v|_2 + \sum_{(u,v)\in \mathrm{P}_j} |u-v|_2$$

Here P_i and P_j represent the set of 2D point correspondences in views i and j that forms the set of inliers for the corresponding fundamental matrix $\mathbf{F}_{ij}$. A higher inlier spread score indicates that $\mathbf{F}_{ij}$ is stable and accurate. The score is proportional to the inlier count, but also captures the spatial distribution of the 2D inliers.

Our method works only when the camera centers for the three cameras are not collinear. This degenerate configuration can be detected by analyzing the location of the six epipoles (when all three camera centers are collinear, $\mathbf{e}_{ij} = \mathbf{e}_{ik}$ for various permutations of the three views). In our method, when a degenerate triplet is detected, we reject it and look for the next best possibility. For most camera networks (all the datasets used in our experiments), cameras were deployed around the subject and collinearity of camera centers was never a problem.

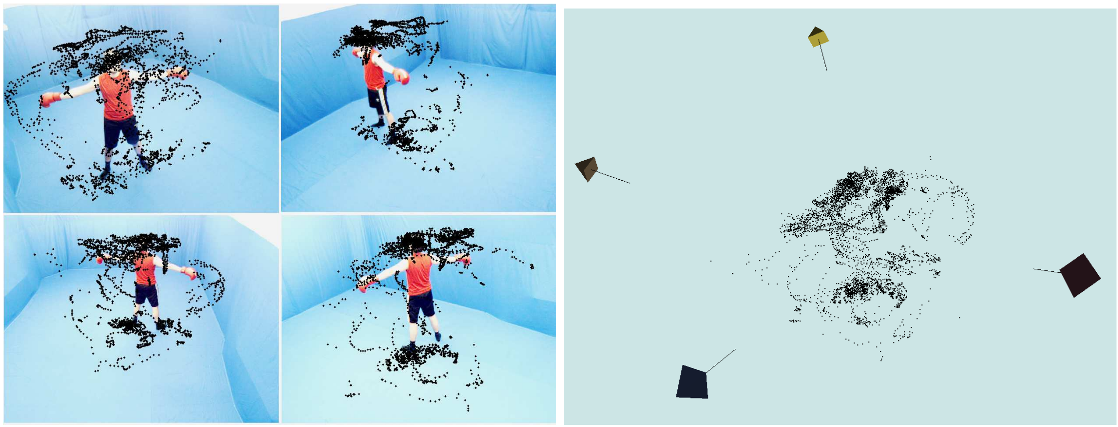


Fig. 8 The final metric reconstruction of the camera network and the 3D point cloud corresponding to all the frontier points recovered from the four view video streams [1].

4.2 Incremental Construction

Our incremental approach to projective reconstruction starts by greedily choosing a set of three views for which the fundamental matrices are, relatively the most accurate. As described in the previous section, this triplet is resolved, resulting in a partial projective reconstruction of three cameras. Next, cameras are added one at a time using the approach described next. The process stops when either all cameras have been added or no more cameras can be added to the network because of insufficient links (fundamental matrices).

Given G_{k-1}, a calibrated camera network with $(k-1)$ cameras, we first need to choose the camera that will be added next to this calibrated network. To do this, we inspect the links (epipolar geometries) between cameras that belong to G_{k-1} and those that have not been reconstructed yet. The camera chosen for reconstruction, is denoted by k, and the two cameras within G_{k-1} corresponding to the two links are denoted by p and q, respectively. Thus for cameras p and q in G_{k-1} and k, the new view, we now reconstruct a triplet of consistent projective cameras from $\mathbf{F}_{pk}$, $\mathbf{F}_{qk}$, and $\mathbf{F}_{pq}$ (here $\mathbf{P}_k$ plays the role of $\mathbf{P}_3$). Since the fundamental matrix corresponding to any pair within G_{k-1} can be computed, the choice of p and q are irrelevant, because all projective cameras are known. Finally, the computed projective camera $\mathbf{P}_k$ is transformed into the projective frame of G_{k-1}. This produces a complete projective reconstruction of G_k, the camera network with the added new camera.

For a network with N cameras in general position, this method will work if a sufficient number of links are present in the camera network graph. The various solvable cases are discussed in [20]. In our case, resolving the initial triplet requires three links, and every subsequent view that is added requires at least two links. Thus, the minimum number of unique links that must be present in the graph is $3 + 2(N-3) = 2N-3$. When more links are available in the graph, our ranking procedure chooses the best ones and the less accurate links may never be used.

4.3 Computing the Metric Reconstruction

Every time a new camera is added, a projective bundle adjustment is done to refine the calibration of all cameras in the partial network. This prevents error accumulation during the incremental construction. Camera networks are typically small, containsing 4 to 12 cameras, therefore, performing the projective bundle adjustment after adding each camera is not a computational burden. Once a full projective reconstruction of the camera network has been computed, a linear self-calibration algorithm [23] is used to upgrade from a projective reconstruction to a metric reconstruction.

Finally, a Euclidean bundle adjustment minimizes the overall reprojection error of a point cloud corresponding to the frontier points matched in two views while parameterizing the cameras in terms of the intrinsic and extrinsic parameters (see Figure 8). In all cases, we constrain the camera *skew* to be zero but impose no other parameter constraints. Depending on the exact scenario, other constraints could be enforced at this step for higher accuracy, for example enforcing a fixed aspect ratio of pixels and enforcing the principal point to be at the center of the image. For higher accuracy, radial distortion in the images should also be modeled in the Euclidean bundle adjustment which typically further reduces the final reprojection error. However, estimation of radial distortion was not done in our current work and this will be addressed in the future.

5 Dealing with unsynchronized video streams

When the recorded video sequences are unsynchronized, the epipolar tangent constraints which form the basis of the proposed approach still exist – but up to an unknown parameter, the temporal offset. We assume that the video frame rate is constant and known a priori, which is a reasonable assumption for most camera networks.

5.1 Pairwise Synchronization and Epipolar Geometry Estimation

We now describe how the algorithm proposed earlier can be extended to simultaneously recover both the temporal offset as well as the epipolar geometry from a pair of video streams. The main idea is to modify the hypothesis step by sampling an extra dimension – a possible range of temporal offsets, in addition to the $4D$ space of epipoles. This algorithm typically requires more hypotheses than the synchronized case before a stable solution can be found, but a multi-resolution approach for computing the temporal offset speeds it up considerably. The details are now described.

Directly finding the true temporal offset within a large search range will require many hypotheses because the probability of randomly selecting the correct temporal offset is quite low. We therefore adopt a coarse-to-fine strategy for this search. In video containing human subjects, the frontier points and epipolar tangents tend to remain stationary over a range of successive frames. Although such frames are not suitable for accurate synchronization, they could be used for an initial coarse alignment of the two sequences. We will refer to these as *slow* frames.

Without knowing the position of the epipoles, it is impossible to select the *slow* keyframes accurately. Therefore the list of keyframes are computed heuristically using hypothetical epipoles at the corners of the image. Based on such hypothetical epipoles, we analyze the potential motion of frontier points in each sequence independently. This is used to build up list of *slow* keyframes from the original sequences. As the RANSAC-based algorithm searches for promising epipole locations, this information could be used in a feedback loop to choose the hypothetical epipoles and generate better keyframes but at the cost of an extra prior step for the algorithm.

The algorithm proceeds in multiple stages. In the first stage, only the *slow* keyframes are used. A 5D random hypothesis is generated. The epipoles are sampled in the way described earlier. For the random guess for the temporal offset, a large search range is coarsely sampled at this stage. The model verification step analyzes the error distribution in the same way as described in Section 3.2. See Figure 9(a) for a distribution of the candidate solutions for the temporal offsets. The uncertainty of the estimate is also computed from such a distribution.

It is possible that this stage estimates the epipolar geometry quite poorly however it helps to narrow down the search for the temporal offset. For every 40 promising candidates, a 99% confidence interval for the sample mean of the temporal offset is computed and this becomes the new search interval for the temporal offset. The process is continued until the search range becomes smaller than 20 frames.

In the second stage, all the frames are used and the RANSAC-based algorithm samples the temporal offset from the smaller search range recovered from the previous stage. During this stage, all the frames are used to estimate the synchronization and epipolar geometry simultaneously. The offset is now sampled from a small interval of +/- 10 frames around the estimated offset from the previous stage. The distribution of promising epipoles obtained from the previous stage is used to bias the random sampling in the 4D space of epipoles. This allows us to find an accurate solution much more quickly. Although this version

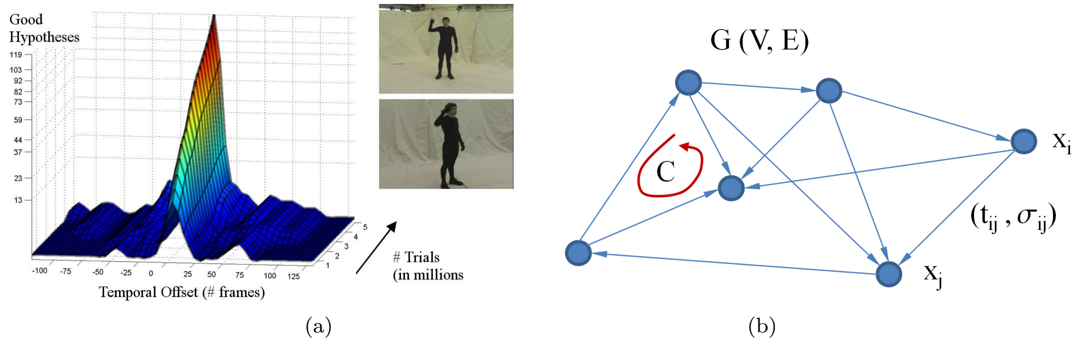


Fig. 9 (a) The distribution of candidate solutions for the temporal offset for a camera pair in the MIT sequence. A strong peak was observed at the true solution while some periodicity in the sequence gave rise to some secondary solutions. (b) The camera network graph where the edges represents pairwise offset measurements.

of the algorithm requires many more RANSAC iterations, the first stage is considerably faster as only a smaller set of keyframes are used. The stratified approach also allows us to sample epipoles from a more accurate prior distribution which helps us find promising candidates more quickly in the final stage.

5.2 Camera Network Synchronization

The camera network synchronization problem is an instance of the general sensor synchronization problem in a network. In our case, every camera can be thought to have an independent timer and the time differences can be measured in frame alignment offsets, since we assume that all cameras are operating at a constant, known frame rate.

We represent the sensor network by a directed graph $G(V, E)$, as shown in Figure 9(b). There are N sensors and each node $v_i \in V$ has a timer denoted by x_i. A directed edge in this network, $e_{ij} \in E$ represents an independent measurement of the time difference $x_j - x_i$ between the two timers. Each estimate t_{ij} has an associated uncertainty represented by the standard deviation σ_{ij} which is inversely proportional to the uncertainty.

When G represents a tree i.e. it is fully connected and has $N-1$ edges, it is possible to synchronize the whole network. When additional edges are available, each of those edges provides a further constraint, which leads to an overdetermined system of linear equations. Each edge contributes a linear constraint of the form $x_i - x_j = t_{ij}$. Stacking these equations produces a $|E| \times N$ system of linear equations. Assuming that each measurement is corrupted by independent Gaussian noise, the maximum likelihood estimate of the N timers is obtained by computing the weighted least squares solution of the linear system (each equation is multiplied by the factor $\frac{1}{\sigma_{ij}}$). The timer estimates (the first camera is fixed at zero) are optimal provided no outliers are present in the edges being considered.

It is fairly easy to detect outlier edges in the network. A consistent network should satisfy the constraint $(\sum_{e \in C} e) = 0 \; \forall$ cycles $C \in G$. For every edge $e \in E$, we check the sum of edges for cycles of length 3 that also contain the edge e. An outlier edge will have a significantly large number of non-zero sums and could be easily detected and removed. This method will produce very robust estimates for complete graphs because $\frac{N(N-1)}{2}$ linear constraints are available for N unknowns. In the minimal case, a fully connected graph with at least N-1 edges is still sufficient to synchronize the whole network although the estimates in this case will be less reliable.

5.3 Silhouette Interpolation for Visual Hull Reconstruction

Typically visual hull methods treat the temporal offset between the multiple video streams as an integer and ignore sub-frame synchronization. Given a specific frame from one video stream, the closest frame in other 30Hz video streams could be as far of as $\frac{1}{60}$ seconds away in time. While this might seems small at first, it can be significant for a fast moving person. This problem is illustrated later in Figure 15(d) where the visual hull was reconstructed from the closest original frames in the sequence. The gray area in the figure represents what is inside the visual hull reconstruction, and the white area corresponds to the reprojection error (points inside the silhouette in one view carved away from other views). Subframe offsets need to be considered to perfectly synchronize the motion of the arms and the legs.

To deal with this problem, we propose temporal silhouette interpolation. Given two adjacent frames i and $i+1$ from a video stream, we compute the signed distance map in each image such that the boundary of the silhouette represents the zero level set in each case. Let us denote these distance maps by $d_i(x)$ and $d_{i+1}(x)$, respectively. Then, for a subframe temporal offset $\Delta \in [0,1]$, we compute an interpolated distance map denoted by $S(x) = (1-\Delta)d_i(x) - \Delta d_{i+1}(x)$. Computing the zero level set of $S(x)$ produces the interpolated silhouette. This simple scheme, motivated by [9] robustly implements linear interpolation between two silhouettes without explicit point-to-point correspondence. However it is approximate and does not preserve shape. Thus, it can be applied only when the inter-frame motion in the video streams is small.

6 Experimental Results

Table 1 summarizes information about the various camera network datasets that we have collected and processed. These multi-view video streams were acquired by various researchers in different indoor scenes. The camera calibration was originally recovered by them using traditional offline calibration grid based techniques [3]. Most of the subjects were humans as these multi-camera networks were designed for capturing virtual models of actors using vision-based markerless motion capture [1,7].

Silhouettes had been extracted for all these sequences using state of the art methods. Although silhouette extraction in the general case is a hard problem, fairly robust and accurate methods are now known for dealing with static backgrounds. For all the datasets described in Table 1 it was possible to extract reasonably good silhouettes which were then used for modeling dynamic scenes using a variant of shape from silhouette and model-based techniques. Using our method, these same silhouettes were used to also recover the camera calibration and synchronization.

6.1 Epipolar Geometry Estimation

We now show results from the silhouette-based estimation of epipolar geometry for a few camera pairs for some of the datasets. In the next section, we present detailed results on the calibration of the full camera network which is derived from the pairwise epipolar geometry estimates.

We tested our method on a 25-view synthetic Kung-fu dataset created by the researchers at MPI–Saarbrucken [7]. The results for a particular camera pair is shown in Figure 10(a). The top row shows the corresponding epipolar lines based on the estimated fundamental matrix while in the bottom half of the image, all the frontier point matches are displayed. The pairwise epipolar geometry for all images with respect to the first view is shown in Figure 10(b) and out of all the 300 pairs, the epipolar geometry for 268 pairs was estimated accurately. Our method cannot handle cameras facing each other which results in epipoles lying somewhere close to the center of the image. In this case, the epipoles often fall inside the silhouette's convex hull and epipolar tangent constraints do not exist.

Figure 11(a) shows results on the four view MIT dataset used originally for capturing deformable 3D human shapes from silhouettes [25]. Originally a co-located motion capture system was used for the calibration and synchronization. The video streams are approximately 4 minutes long and captured at 30 frames per second. The human subject is moving around in the scene; occasionally his silhouette gets clipped in the camera's field of view, esp. his feet. However this is handled robustly in our implementation.

Using the proposed approach, we computed the epipolar geometry for all pairs. The results from two pairs are shown in Figure 11(a). The epipolar geometry for one of the six pairs was unstable because in both the views, the feet of the person was consistently clipped in most of the video. Since the person walks around the frontier points near the head of the person are almost planar which is a degenerate configuration for epipolar geometry estimation. Instead of using all 7000 frames that were available we chose every 5th

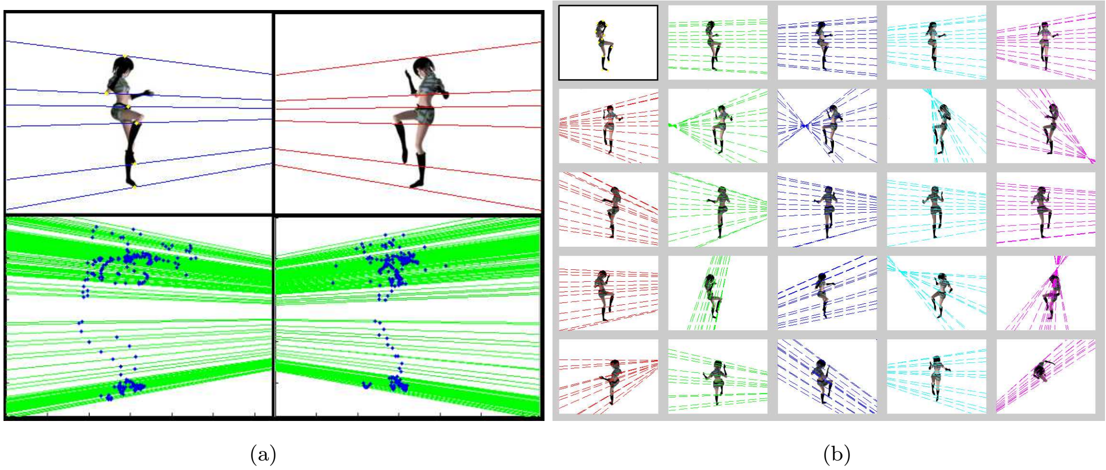


Fig. 10 (a) The estimated epipolar geometry for one of the pairs in the synthetic 25-view Kung-fu sequence. The extracted frontier points and the epipolar tangents are also shown here (b) The estimated epipolar geometry between the first camera and all other 24 cameras are shown.

frame and worked with about 1400 frames from video. Estimating the epipolar geometry took about three minutes on an average for the six pairs. On an average, the RANSAC-based algorithm produced a good solution in about 25000 iterations but for higher reliability, multiple solutions were recovered and checked for consensus.

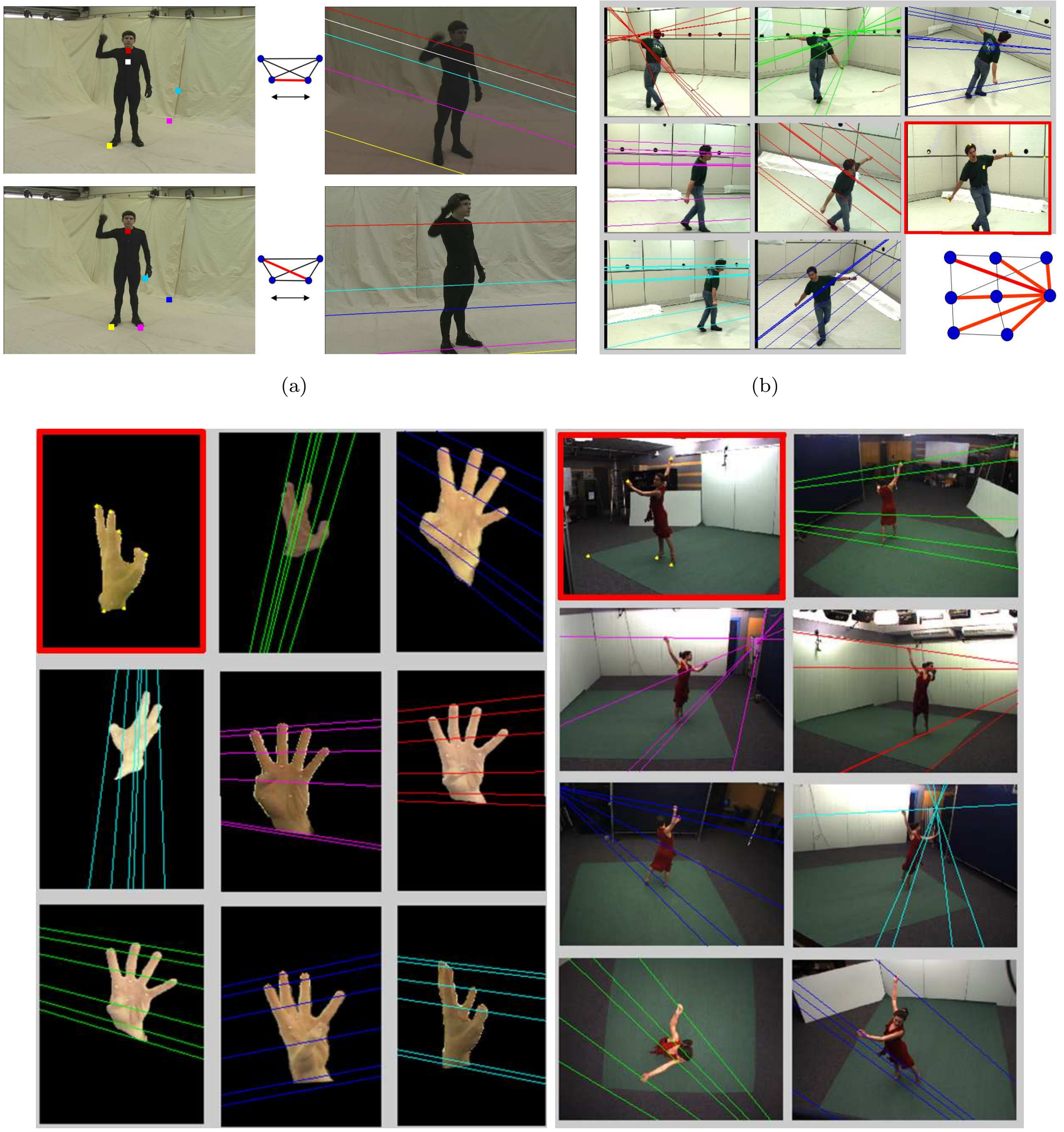
The results from the CMU 3D room sequence is shown in Figure 11(b). Note that the epipoles for some of the pairs coincide with the image of the camera in the respective views. This indicates the accuracy of the epipole estimates for these pairs.

6.1.1 Evaluation

The mean residual error given by $\frac{1}{N}\sum\frac{e}{2}$ where e is defined in Equation 1 is reported for all estimates of the fundamental matrix. The synthetic *Kung-fu* sequence reported a residual error of 0.12 pixels on average while estimates for real datasets had a residual error of 0.25 pixels on an average with a range of 0.2 – 0.31 pixels. Our algorithm for epipolar geometry estimation was evaluated in two cases. First, one of the camera pair from the *Boxer* dataset was tested. Figure 12 (a–b) shows the estimated epipolar geometry using about 1000 frames of video. Subsequently, a checkerboard image pair (not used in our estimation process) was used for evaluation (shown in Figure 12 (c–d)). The user manually clicked 50 corresponding points and the mean symmetric residual error for these points was calculated. Our fundamental matrix estimate had an rms error of 1.21 pixels while the error for the ground truth (derived from the checkerboard based calibration [3]) was 0.78 pixels. The relatively high residuals seem to be due to the error introduced by the user while clicking points. The evaluation was done for another sequence (see Figure 12 (e–f)). This time the mean symmetric residual error was 1.38 pixels. A distribution of the error is shown for the manually specified points (corresponding corner features on both the foreground as well as the background were manually specified).

6.1.2 Discussion

Our proposed algorithm applies RANSAC [2] in an unconventional way. Rather than using it only for robust estimation and handling outliers, we use it to also explore a low dimensional bounded parameters space - the 4D space of epipoles parameterized by the tangent envelope of the silhouettes. RANSAC automatically adjusts its budget of how many iterations to devote to detecting outliers and how many to exploring the parameter space. An alternative approach would involve performing a deterministic, multi-resolution search in the space of epipoles but to use RANSAC only to sample the video frames (in the traditional sense to only deal with outliers). However this has two disadvantages –



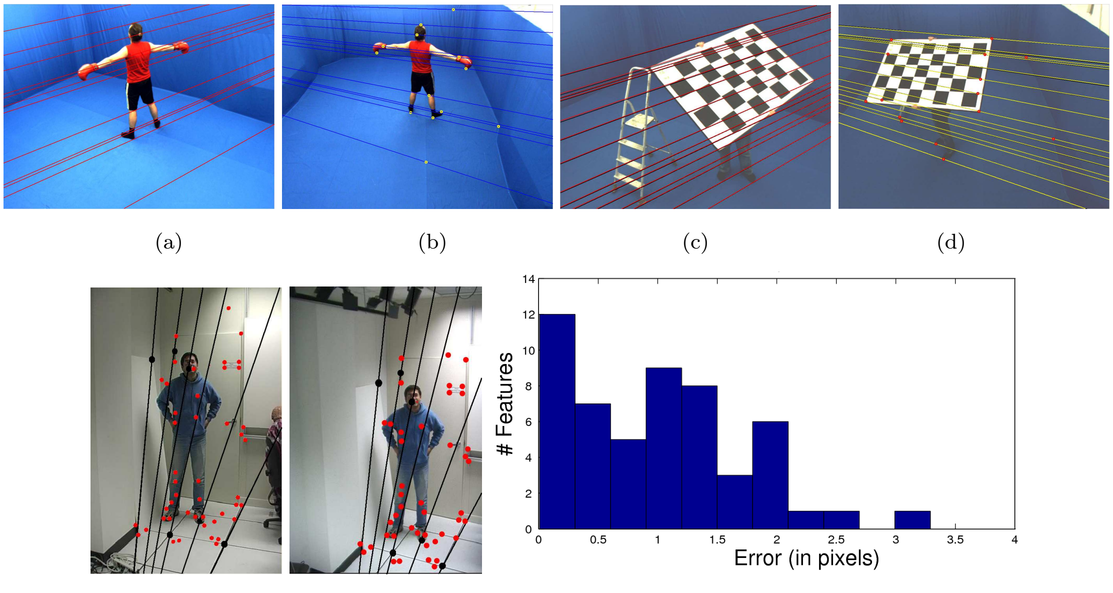
(a) (b)

(c) (d)

Fig. 11 (a) The estimated epipolar geometry for 2 of the 6 pairs in the 4-view MIT dataset. Estimated epipolar geometries for the (b) CMU 3D room dataset (c) Finger sequence [5] and (d) Dancer1 sequence.

– this would require prior knowledge of the size of the attraction basin (i.e. convergence region) especially when the search is performed at a coarse level. This is not needed in our approach. For synthetic uncorrupted data (Kung-fu sequence), we found a promising candidate in 1 in 6000 trials on an average. This seems to indicate that selecting the first direction in each image in approximately $\sqrt{6000} = 77$ random directions allows us to sample within the attraction basin of the true solution at least once. For higher reliability, we recover multiple solutions and then look for consensus amongst at least three. This approach was used for all the datasets in our experiments.

– The number of iterations needed in the deterministic strategy would be orders of magnitude higher. Suppose w % of the frames have corrupt silhouettes. For a correct choice of epipoles, a good model can



(a) (b) (c) (d)

(e) (f)

Fig. 12 (a–b) The epipolar geometry recovered for a particular camera pair in the *Boxer* dataset. (c–d) The checkerboard image pair used for evaluation. Ground truth epipolar lines are shown in black. Epipolar lines for our fundamental matrix estimate are shown in red and yellow. The image resolution is 1000 × 800 pixels. (e–f) For another camera pair, the symmetric epipolar transfer error is computed. (g–h) more results.

be computed if two pairs of good silhouettes are chosen. This has a probability of w^4. Thus the number of trials required to ensure that a good model was generated with p % confidence, is $k = \frac{log(1-p)}{log(1-w^4)}$. For choices of $p = 0.95$ and $w = 0.75$ (75% good silhouettes), 8 trials would be required. If epipoles were sampled 4^o apart, the number of trials would be $90^4 \times 8 = 10^8$. Our RANSAC scheme automatically decides how many trials to allocate for handling outliers and how much for exploring the parameter space and requires fewer iterations in practice.

6.2 Camera Network Calibration

Figure 13 shows the camera network reconstructions from various real datasets. Corresponding input video frames are shown along with the visual hull computed using the recovered calibration. The 3D geometry of the camera network is also shown. By reconstructing the visual hull, we show the accuracy of the recovered camera calibration and illusstrate that such dynamic scenes can now be reconstructed from uncalibrated footage. Our calibration approach is particularly well suited for reconstructing visual hulls, as the method is designed to reduce the overall reprojection errors of silhouettes (or frontier points to be more specific). This tends to preserve sharp extremities on the visual hull causing them to be less eroded than what could be expected with an offline calibration method that does not utilize silhouette information. The calibration recovered by our technique could potentially be further refined using the approach proposed in [4].

We evaluated our method for camera network calibration on the Kung-fu sequence as ground truth calibration is known (see Figure 14). Since the metric reconstruction of the camera network obtained by our method is in an arbitrary coordinate system, it first needs to be scaled and robustly aligned to the ground truth coordinate frame. Our method, after the final bundle adjustment produced an overall average reprojection error in the 25 images of 0.11 pixels and the reconstructed visual hull of the Kung-fu character is visually as accurate as that computed from ground truth.

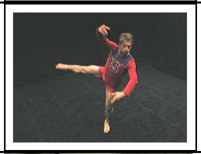
	Name	Cameras	Frames	Pairs	Reprojection Error (final)
	Kung-Fu [7]	25	200	268 / 300 pairs	0.11 pixels
	Ballet [7]	8	468	24 / 28 pairs	0.19 pixels
	MIT [25]	4	7000	5 / 6 pairs	0.26 pixels
	Dancer1 (INRIA)	8	200	20 / 28 pairs	0.25 pixels
	Man (INRIA)	5	1000	10 / 10 pairs	0.22 pixels
	Dancer2 [29]	6	250	11 / 15 pairs	0.23 pixels
	Boxer [1]	4	1000	6 / 6 pairs	0.22 pixels

Table 1 These datasets were previously acquired by various researchers in computer vision. These were calibrated using our proposed approach. The second-last column shows the number of camera pairs for which the epipolar geometry was correctly estimated. The reprojection error after the Euclidean bundle adjustment is listed in the final column.

6.3 Camera Network Synchronization

Figure 15(a) shows the metric 3D reconstruction for the four view MIT sequence. To test the accuracy of the recovered calibration and synchronization, we projected the visual hull back into the images. Inaccurate calibration, poor segmentation or lack of perfect synchronization could give rise to empty regions (white pixels) in the silhouettes. We found that the silhouettes were mostly filled, except for fast-moving parts where the reprojected visual hull was sometimes a few pixels smaller (see Figure 15(a)). This arises mostly when sub-frame synchronization offsets are ignored or due to motion blur or shadows.

For higher accuracy, we computed visual hulls from interpolated silhouettes as described in Section 5.3 The silhouette interpolation was performed using the sub-frame synchronization offsets computed earlier for this sequence. An example is shown in Figure 15(b). Given three consecutive frames, we generated the

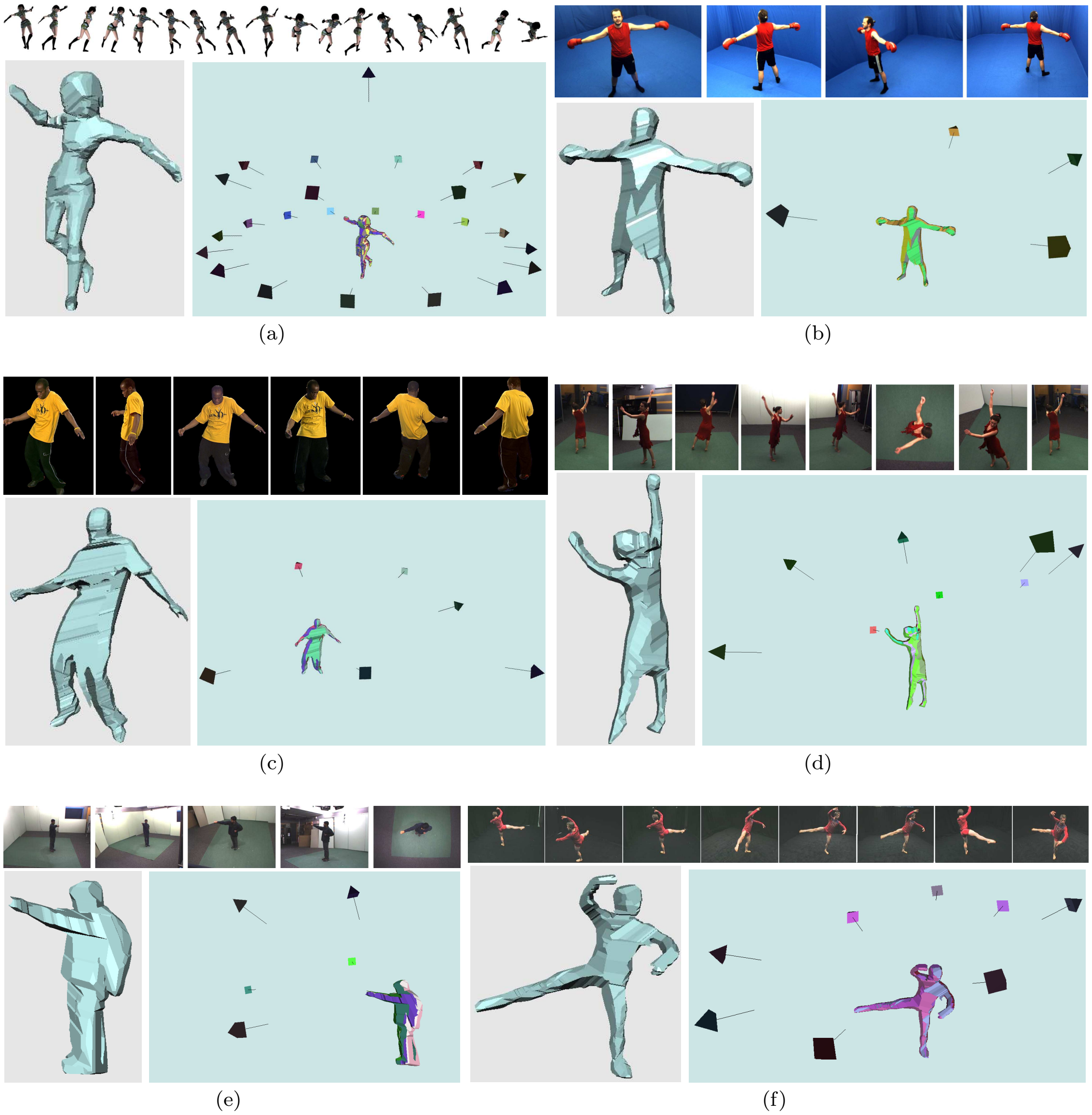


Fig. 13 Metric 3D reconstructions from six different datasets – (a) Kung-fu, (b) Boxer, (c) Dancer2, (d) Dancer1, (e) Man and (f) Ballet. The recovered camera network is visualized along with a visual hull reconstruction (computed using the approach of [10]) of the subject in each case.

middle one by interpolating between the first and the third and compared it to the actual second frame. Our interpolation approach works reasonably for small motion, as would be expected in video captured at 30 frames per second. In Figure 15(c), the visual hull reprojection error is shown with and without sub-frame silhouette interpolation. In the two cases, the reprojection error decreased from 10.5% to 3.4% and from 2.9% to 1.3% of the pixels inside the silhouettes in the four views.

7 Conclusions

To conclude, in this paper we have presented a complete approach to recover the full metric calibration of an unsynchronized camera network by analyzing silhouettes in video. At the core of the proposed

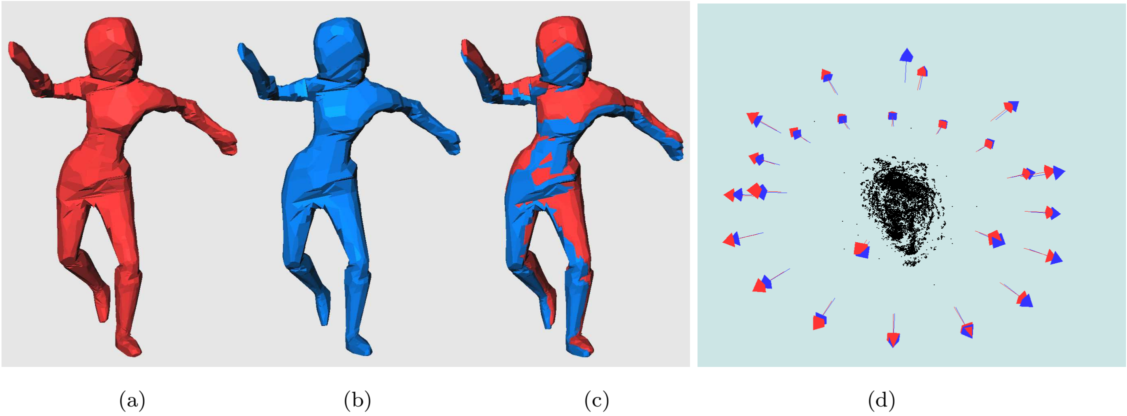
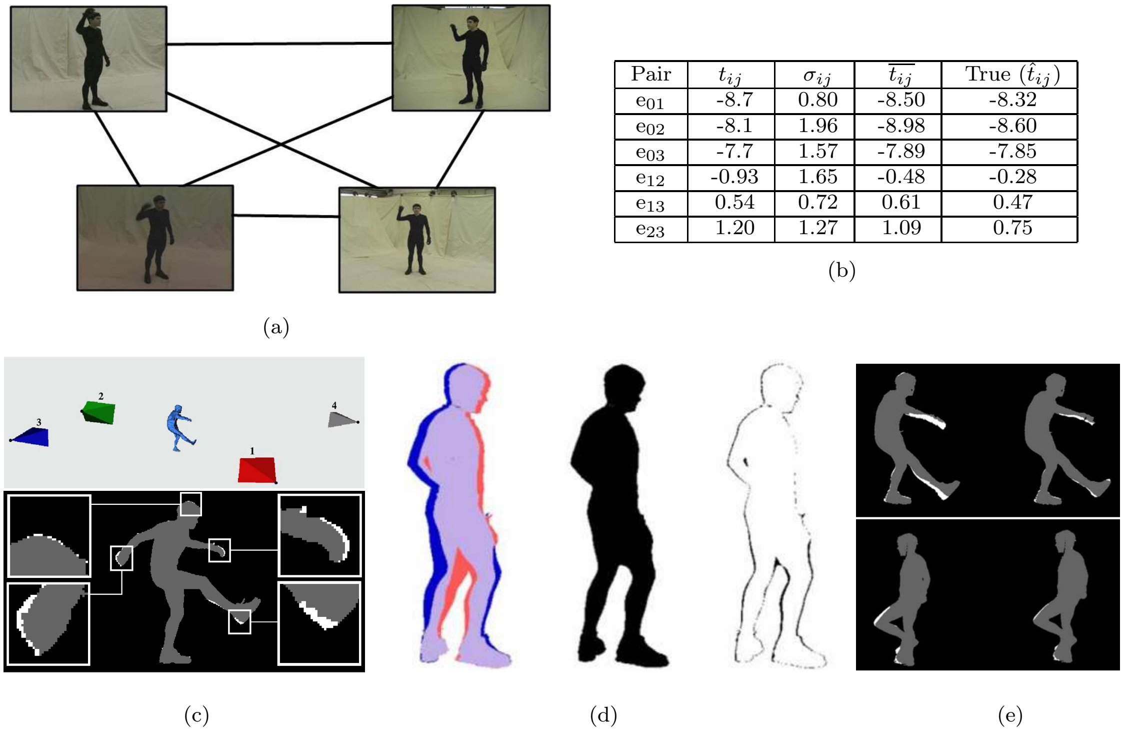


Fig. 14 (Best seen in color) For the Kung-fu sequence, ground truth is available. Models computed using (a) ground truth calibration and (b) the calibration recovered by our method. (c) The registered 3d models. (d) The camera network registered to the coordinate frame of the ground truth data.



Pair	t_{ij}	σ_{ij}	$\overline{t_{ij}}$	True ($\hat{t}_{ij}$)
e_{01}	-8.7	0.80	-8.50	-8.32
e_{02}	-8.1	1.96	-8.98	-8.60
e_{03}	-7.7	1.57	-7.89	-7.85
e_{12}	-0.93	1.65	-0.48	-0.28
e_{13}	0.54	0.72	0.61	0.47
e_{23}	1.20	1.27	1.09	0.75

Fig. 15 (a) The camera network graph for the MIT sequence. (b) Table of pairwise offsets and uncertainties (t_{ij} and σ_{ij}) and final estimates ($\overline{t_{ij}}$). These are within $\frac{1}{3}$ of a frame (i.e. $\frac{1}{100}^{th}$ of a second within the ground truth ($\hat{t}_{ij}$)). (c) Metric 3D reconstructions of the MIT sequences. The visual hull is reprojected into the images to verify the accuracy. (d)(e) Silhouette interpolation using the sub-frame synchronization reduces such reprojection errors.

method, is a RANSAC-based algorithm to efficiently compute the synchronization and epipolar geometry of a pair of cameras. The proposed method will allow more flexibility in camera network calibration and synchronization and will make it possible to digitize events in 3D even from archived video streams.

Our approach begins by independently computing the epipolar geometry and temporal offset for various pairs of cameras in the network. In the next stage, the calibration and synchronization of the complete network is recovered. The effectiveness of our approach is demonstrated by remotely calibrating many archived multi-view video streams previously acquired by researchers in the community. It can easily deal with widely separated views, textureless scenes and is robust to noisy silhouettes caused by poor background segmentation or motion blur and does not require radiometric calibration between the cameras.

In future, we will try to solve the relative pose estimation problem using only silhouettes. This will be useful for the specific case when the camera intrinsics are known in advance. We would also like to explore the possibility of using silhouettes in a similar way as proposed here, to calibrate heterogenous networks comprising of conventional cameras, depth cameras and IR sensors.

Acknowledgment

We gratefully acknowledge the support of David and Lucille Packard Foundation Fellowship, NSF Career award IIS-0237533 and the European Research Council Grant – for their financial support on this project. We would also like to thank many researchers who have shared valuable datasets with us – Peter Sand, Seth Teller, Christian Theobalt, Marcus Magnor, Gabriel Brostow, German Cheung, Luca Ballan, Jonathan Stark, Adrian Hilton and the PERCEPTION research group at INRIA Rhone-Alpes.

References

1. Luca Ballan and Guido Maria Cortelazzo. Multimodal 3d shape recovery from texture, silhouette and shadow information. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 924–930, Washington, DC, USA, 2006. IEEE Computer Society.
2. R. C. Bolles and M. A. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *Proc. of the 7th IJCAI*, pages 637–643, Vancouver, Canada, 1981.
3. J. Bouguet. Matlab camera calibration toolbox, 2000.
4. Edmond Boyer. On using silhouettes for camera calibration. In *ACCV (1)*, pages 1–10, 2006.
5. Gabriel J. Brostow, Irfan Essa, Drew Steedly, and Vivek Kwatra. Novel skeletal representation for articulated creatures. In *ECCV04*, pages Vol III: 66–78, 2004.
6. Chris Buehler, Steven J. Gortler, Michael F. Cohen, and Leonard McMillan. Minimal surfaces for stereo. In *ECCV (3)*, pages 885–899, 2002.
7. Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 569–577, New York, NY, USA, 2003. ACM.
8. G.K.M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, pages I: 77–84, 2003.
9. Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, New York, NY, USA, 1996. ACM.
10. Jean-Sébastien Franco and Edmond Boyer. Exact polyhedral visual hulls. In *Proceedings of the Fourteenth British Machine Vision Conference*, pages 329–338, September 2003. Norwich, UK.
11. Jean-Sébastien Franco, Marc Lapierre, and Edmond Boyer. Visual shapes of silhouette sets. In *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill (USA)*, 2006.
12. Yasutaka Furukawa, Amit Sethi, Jean Ponce, and David Kriegman. Robust structure and motion from outlines of smooth curved surfaces. *PAMI*, 28(2):302–315, February 2006.
13. Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*, volume 23. Cambridge University Press, New York, NY, USA, 2005.
14. Carlos Hernández, Francis Schmitt, and Roberto Cipolla. Silhouette coherence for camera calibration under circular motion. *PAMI*, 29(2):343–349, February 2007.
15. Tanuja Joshi, Narendra Ahuja, and Jean Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. In *ICCV*, pages 290–295, 1995.
16. Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, – 1997.
17. A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, February 1994.
18. Svetlana Lazebnik, Edmund Boyer, and Jean Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *CVPR*, pages I:156–161, 2001.
19. Svetlana Lazebnik, Amit Sethi, Cordelia Schmid, David J. Kriegman, Jean Ponce, and Martial Hebert. On pencils of tangent planes and the recognition of smooth 3d shapes from silhouettes. In *ECCV (3)*, pages 651–665, 2002.
20. Noam Levi and Michael Werman. The viewing graph. *CVPR*, 01:518–522, 2003.

21. Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
22. Paulo R. S. Mendonça, Kwan-Yee K. Wong, and Roberto Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):604–616, 2001.
23. Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207–232, 2004.
24. John Porrill and Stephen Pollard. Curve matching and stereo calibration. *Image Vision Comput.*, 9(1):45–50, 1991.
25. Peter Sand, Leonard McMillan, and Jovan Popović. Continuous capture of skin deformation. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 578–586, New York, NY, USA, 2003. ACM Press.
26. Sudipta N. Sinha and Marc Pollefeys. Synchronization and calibration of camera networks from silhouettes. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 116–119, Washington, DC, USA, 2004. IEEE Computer Society.
27. Sudipta N. Sinha and Marc Pollefeys. Visual-hull reconstruction from uncalibrated and unsynchronized video streams. *3dpvt*, 0:349–356, 2004.
28. Sudipta N. Sinha, Marc Pollefeys, and Leonard McMillan. Camera network calibration from dynamic silhouettes. *cvpr*, 01:195–202, 2004.
29. Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.
30. Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.
31. B. Vijayakumar, D. J. Kriegman, and J. Ponce. Structure and motion of curved 3d objects from monocular silhouettes. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 327, Washington, DC, USA, 1996. IEEE Computer Society.
32. K.Y.K. Wong and R. Cipolla. Structure and motion from silhouettes. In *ICCV*, pages II: 217–222, 2001.
33. Anthony J. Yezzi and Stefano Soatto. Structure from motion for scenes without features. In *CVPR (1)*, pages 525–532, 2003.
34. Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.