Camera Network Calibration from Dynamic Silhouettes

Sudipta N. Sinha Marc Pollefeys Leonard McMillan.
Dept. of Computer Science, University of North Carolina at Chapel Hill.
{ssinha, marc, mcmillan}@cs.unc.edu

Abstract

In this paper we present an automatic method for calibrating a network of cameras from only silhouettes. This is particularly useful for shape-from-silhouette or visual-hull systems, as no additional data is needed for calibration. The key novel contribution of this work is an algorithm to robustly compute the epipolar geometry from dynamic silhouettes. We use the fundamental matrices computed by this method to determine the projective reconstruction of the complete camera configuration. This is refined into a metric reconstruction using self-calibration. We validate our approach by calibrating a four camera visual-hull system from archive data where the dynamic object is a moving person. Once the calibration parameters have been computed, we use a visual-hull algorithm to reconstruct the dynamic object from its silhouettes.

1 Introduction

Shape-from-Silhouette initially proposed by [3], has recently received a lot of attention and various algorithms for recovering the shape of objects have been proposed [5, 8, 15, 10, 20]. Many Shape-from-Silhouette methods attempt to compute the visual hull [11] of an object, which is the maximal shape that produces the same set of silhouettes seen from multiple views. For a fully calibrated camera, the rays through the camera center and points on the silhouette define a viewing cone [17]. Intersecting viewing cones backprojected from silhouettes in multiple views produces the visual hull of the object. Shape-from-Silhouette implementations are relatively simple and real-time model acquisition techniques exist [5, 15]. However with a few cameras, the visual hull can only coarsely approximate the shape of the real object. For more accurate shape estimates, more silhouettes images are needed. This could be achieved by increasing the number of cameras or by trying to align visual hulls over time, when the scene exhibits rigid motion [8]. Sand et al. [20] use silhouettes to estimate shape of dynamic objects and is able to get good estimates by assuming a parameterized model of human figures.

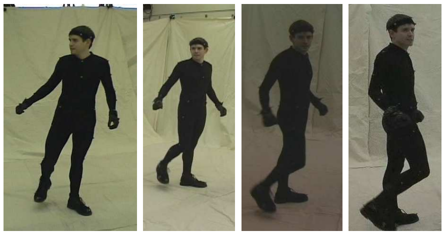

Figure 1: Multi-view Uncalibrated Video Sequence

Most multi-camera Shape-from-Silhouette systems assume that the calibration and pose of the cameras has been precomputed offline via a specific calibration procedure. Typically, the calibration data is obtained by moving a planar pattern [26] or a LED in the field of view of the cameras. This has the significant disadvantage that physical access to the observed space is necessary and it precludes reconfiguration of cameras during operation (at least without inserting an additional calibration session). Some approaches for structure-from-motion for silhouettes have been proposed, but most of these have limitations rendering them impractical for arbitrary unknown camera configurations, which we call a *camera network*. These limitations include : requiring the observed object to be static [7], requiring a specific camera configuration (i.e. at least partially circular) [23], using an orthographic projection model [22], and requiring a good initialization [24].

In this paper we address the problem of calibrating a camera network and constructing the visual hull from the video sequences of a dynamic object using only silhouette information. Our approach is based on a novel algorithm to robustly compute the epipolar geometry from two silhouette sequences. This algorithm is based on the constraints arising from the correspondence of frontier points and epipolar tangents [23, 19, 1, 2]. These are points on an objects' surface which project to points on the silhouette in two views. Epipolar lines which pass through the images of a frontier point must correspond. Such epipolar lines are also tangent to the respective silhouettes at these points. Previous work used those constraints to refine an existing epipolar geometry [19, 1, 2]. Here we take advantage of the fact that a camera network observing a dynamic object will record many different silhouettes, yielding a large number of constraints that need to be satisfied. We devise a RANSAC [4] based approach to extract such matching epipolar tangents in the video sequence. The epipole positions are hypothesized, an epipolar line homography is computed and verified at every RANSAC iteration. Random sampling is used both for exploring the 4D space of possible epipole positions as well as dealing with outliers in the silhouette data. A subsequent non-linear minimization stage computes a more accurate estimate of the epipolar geometry and also provides matching frontier points in the video-sequence. These point matches are used later in a bundle adjustment to improve calibration. Once some of the fundamental matrices are known, a projective reconstruction of the N Cameras can be recovered. This is first refined using a projective bundle adjustment. Next, using self-calibration methods and a Euclidean bundle adjustment, we are able to compute a set of optimal Euclidean cameras. Finally, the metric visual hull of the observed dynamic object is reconstructed for the sequence. Other reconstruction approaches such as multi-baseline stereo or voxel coloring, could also be used with the computed calibration.

As our calibration approach relies on silhouettes, it depends on a robust background segmentation approach. Our RANSAC algorithm, however, allows a reasonable ratio of bad silhouettes. It is also important that the frontier points cover a sufficient part of the image and depth range to yield satisfactory results. This requires sufficient motion of the observed object over the space observed by the cameras. Advantages of our method are that it does not rely on feature matching and wide-baselines between camera pairs are handled well. Our approach is particularly well suited for systems that rely on silhouette extraction for reconstruction, as in this case no additional data needs to be extracted for calibration. We cannot directly compute the epipolar geometry of camera configurations where the epipole is located within the convex hull of the silhouette, but we can often handle this case as the projective reconstruction stage only requires a subset of the fundamental matrices. The remainder of this paper is organized as follows. Section 2 presents the background theory and terminology. The details of our algorithm are presented in Section 3. Section 4 shows our results on a real dataset and we finally conclude with discussions in Section 5.

2 Background and notation

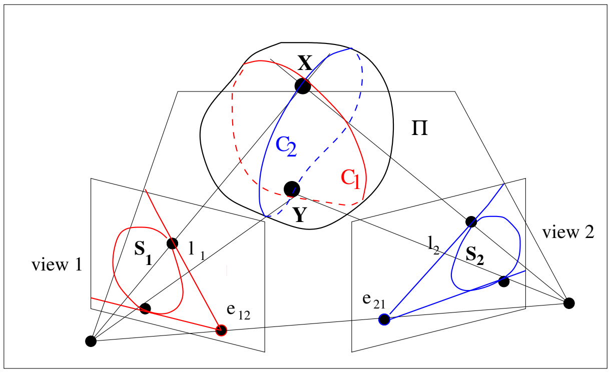

Figure 2: The frontier points and epipolar tangents for two views.

The signifance of epipolar tangencies and frontier points has been extensively studied in computer vision [19, 17, 23, 13]. Frontier points are points on the object's surface which project to points on the silhouettes in two views. In Fig. 2, X and Y are frontier points which project to points on the silhouettes S_1 and S_2 respectively. They both lie on the intersection of the apparent contours, C_1 and C_2 which give rise to these two silhouettes. The projection of Π, the epipolar plane tangent to X gives rise to corresponding epipolar lines l_1 and l_2 which are tangent to S_1 and S_2 at the images of X in the two images respectively. No other point on S_1 and S_2 other than the projected frontier points, X and Y are guaranteed to correspond. Unfortunately, frontier point constraints do not, in general exist over more than two views. In a three-view case generally, the frontier points in the first and second view do not correspond to those in the second and third view. As we show later, this has important implications for the recovery of the projective camera network configuration. For a complicated non-convex polytope object such as a human figure, there could be many potential frontier points. However it is hard to find all of them in uncalibrated sequences since the position of the epipoles are unknown [19] a priori. In [23] Wong et. al searches for outer-most 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 simplify the search for epipoles. We also look for the two outer epipolar tangents and make the key observation that the image of the frontier points corresponding to these outer-most epipolar tangents must lie on the convex hull of the silhouette. We apply a RANSAC-based approach to search for the epipoles and compute the epipolar line homography which satisfies the epipolar geometry as well as retrieve the corresponding frontier points in the whole seqeunce.

We shall denote the Fundamental Matrix between view i and view j by F_{ij} (transfers points in view i to epipolar lines in view j) and the epipole in view j of camera center i as e_{ij}. The pencil of epipolar lines in each view centered on the epipoles, is considered as a $1D$ projective space [9] [Ch.8 p.227]. The epipolar line homography between two such $1D$ projective spaces is a $2D$ homography. Knowing the position of the epipoles e_{ij}, e_{ji} (2 dof each) and the epipolar line homography (3 dof) fixes F_{ij} which has 7 dof. Three pairs of corresponding epipolar lines are

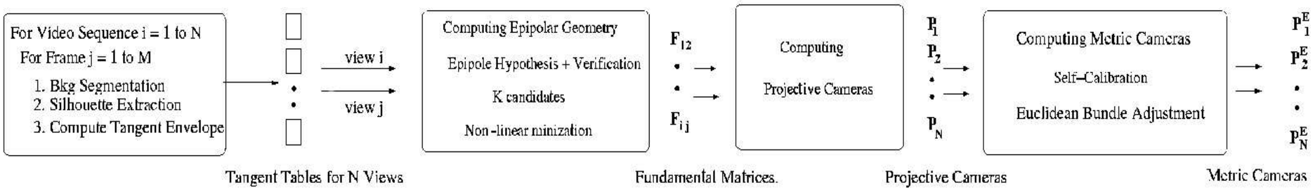

Figure 3: The Calibration Procedure.

sufficient to determine the epipolar line homography $H_{ij}^{-\top}$ so that it uniquely determines the transfer of epipolar lines (note that $H_{ij}^{-\top}$ is only determined up to 3 remaining degrees of freedom, but those do not affect the transfer of epipolar lines). The fundamental matrix is then given by $F_{ij} = [e_{ij}]_\times H_{ij}$.

The metric camera network configuration of a system with N cameras is described by a set of Euclidean camera projection matrices $P_i^E = K_i[R_i^T \; -R_i^T t_i]$; $i = 1 \ldots N$ where K_i represent the camera intrinsics, R_i and t_i describe the rotation and translation of the camera center of the ith camera w.r.t the world coordinate frame. The set of projective camera matrices will be denoted by P_i; $i = 1 \ldots N$. Those camera matrices are related by a projective transformation T_E^P so that $P_i = T_E^P P_i^E; i = 1 \ldots N$.

3. Our approach

Fig. 3 describes the step by step procedure we follow. We have N fixed cameras placed around an object. The input to the system is N synchronized video sequences of M frames each. We denote the set of silhouettes in the jth set of frames by S_j^i; $i = 1 \ldots N$. Our goal is to compute the Euclidean camera projection matrices P_i^E corresponding to the camera network configuration.

3.1 Silhouette Tangent Envelopes

For every frame in each sequence, a binary segmentation of the object is computed using background segmentation. Noisy patches are cleaned up using a hole-filling algorithm that uses an area threshold in pixels to distinguish noisy blobs from the object's silhouette blob. Instead of explicitly storing every silhouette S, we directly compute and store its tangent envelope $T(S)$, which is a more compact representation. The tangent envelope of S, (see Fig. 4(a)) consists of its convex hull $CH(S)$, stored as an ordered list of k vertices ($v_1 \ldots v_k$ in counter-clockwise order (CCW)) and a table of directed tangents parameterized by the angle $\theta = 0^o \ldots 360^o$, where for every tangent t_i we store p_i, the point of tangency on the convex hull $CH(S)$. A 1^o sampling interval is chosen for the tangent tables. The tangent orientation defines a consistent direction of the tangent with respect to $CH(S)$ such that there is only one tangent t_θ in a direction $\overline{d}_\theta$ and not two. This simplifies tangency computations later on.

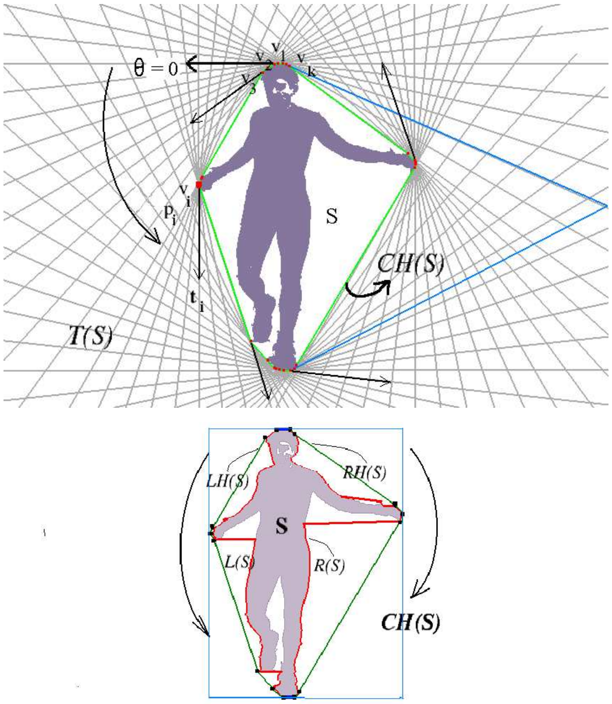

Figure 4: (a) The Tangent Envelope T(S) for silhouette S (only 1 in 6 tangents is shown for clarity). We sample for the epipoles in this tangent space parameterized by θ. (b) Computing the Silhouette Convex Hull CH(S)

To compute the Tangent Envelope $T(S)$ for a silhouette S, we first construct its convex hull $CH(S)$ using Melkman's $O(n)$ on-line convex hull algorithm [16]. This requires a simple path traversing all points in the point-set, which can be computed in $O(nlogn)$ time. In our case, a top-down scan of the bounding box of S implicitly gives us 2 simple paths in $O(n)$ time, a left extreme boundary $L(S)$ and a right extreme boundary $R(S)$ of S (see Fig. 4(b)). In a single pass, we obtain ordered lists of vertices for the left hull $LH(S)$, in CCW order and right hull $RH(S)$ in CW order. A union of $LH(S)$ and $RH(S)$ produces $CH(S)$ which is stored in CCW order. If the silhouettes are clipped at image boundaries, we store the convex hull as a single ordered list instead of multiple connected segments. We introduce new vertices where the silhouettes are clipped and store flags to indicate the segments which lie inside the images. The next step is computing the tangency points p_θ for $\theta = 0^o \ldots 360^o$. We start by determining p_0 and then rotate the tangent t_θ (incrementing θ) allowing it to switch to the next point in $CH(S)$ when required. This step takes $O(\theta)$ time. $T(S)$ is an extremely compact representation and allows us to compute tangents to $CH(S)$ from any external point in $O(log\theta)$ time. $CH(S)$ typically had 25-35 vertices for the image resolution of our datasets. A single frame required only about 500 bytes of storage. Therefore the tangent tables for several minutes of multi-camera video would easily fit into memory. This would allow us to efficiently access thousands of video frames without any memory bottlenecks. Efficient tangent computation is key to the feasibility of our algorithm as we see in 3.2.2. Computing a pencil of tangents to a sequence of silhouettes is further optimized by using temporal coherence between silhouettes.

3.2. Computing the Epipolar Geometry

Given non-trivial silhouette shapes, we cannot compute the epipolar geometry linearly from corresponding silhouettes because the location of the frontier points depend on the position of the epipoles. Given an approximate solution, it is possible to refine it using an optimization approach [19, 1]. Since we recover calibration of arbitrary camera configurations using only silhouettes, an initial solution is not available to us. Therefore, we need to explore the full space of possible solutions. While a fundamental matrix has 7 dof's, we only have to randomly sample in a 4D space because once the position of the epipoles are known, the frontier points can be determined, and from them the remaining degrees of freedom of the epipolar geometry can be computed. Here we propose a RANSAC-based approach that in a single step, allows us to efficiently explore this 4D space as well as robustly deal with incorrect silhouettes.

In Section 2 we discuss the parameterization of F_{ij} in terms of the epipole positions e_{ij}, e_{ji} and the homography H_{ij}. The basic step of our algorithm makes a hypothesis on the position of e_{ij} and e_{ji} in the two views. This fixes 4 $dof's$ and leaves us with 3 $dof's$ which can be determined if we have a solution for H_{ij}. To compute H_{ij} we need to pick three pairs of corresponding lines in the two views ($l_i^k \leftrightarrow l_j^k; k = 1 \ldots 3$). Every H_{ij} satisfying the system of equations $[l_j^k]_\times H_{ij}^{-\top} l_i^k = 0; k = 1 \ldots 3$ is a valid solution. Note that these equations are linear in $H_{ij}^{-\top}$.

3.2.1 Epipole Hypothesis and Computing H

At every iteration, we randomly choose the rth frames from each of the two sequences. As shown in Fig. 5(a), we then, randomly sample independent directions l_1^1 from $T(S_1^r)$ and l_2^1 from $T(S_2^r)$ for the first pair of tangents in the two views. We choose a second pair of directions l_1^2 from $T(S_1^r)$ and l_2^2 from $T(S_2^r)$ such that $l_i^2 = l_i^1 - x$ for $i = 1, 2$ where x is drawn from the normal distribution, $N(180, \sigma)^1$. The intersections of the two pair of tangents produces the epipole hypothesis (e_{12} , e_{21}). An alternative approach consists of sampling both epipole directions randomly on a sphere [13], which in the uncalibrated case is equivalent to random sampling on an ellipsoid and yields comparable results. We next randomly pick another pair of frames q, and compute either the first pair of tangents or the second pair. Let us denote this third pair of lines by l_1^3 tangent to $CH(S_1^q)$ and l_2^3 tangent to $CH(S_2^q)$ (see Fig. 5(a)). H_{ij} is computed from ($l_i^k \leftrightarrow l_j^k; k = 1 \ldots 3)^2$. The entities ($e_{ij}$,e_{ji} ,H_{ij}) form the model hypothesis for every iteration of our algorithm.

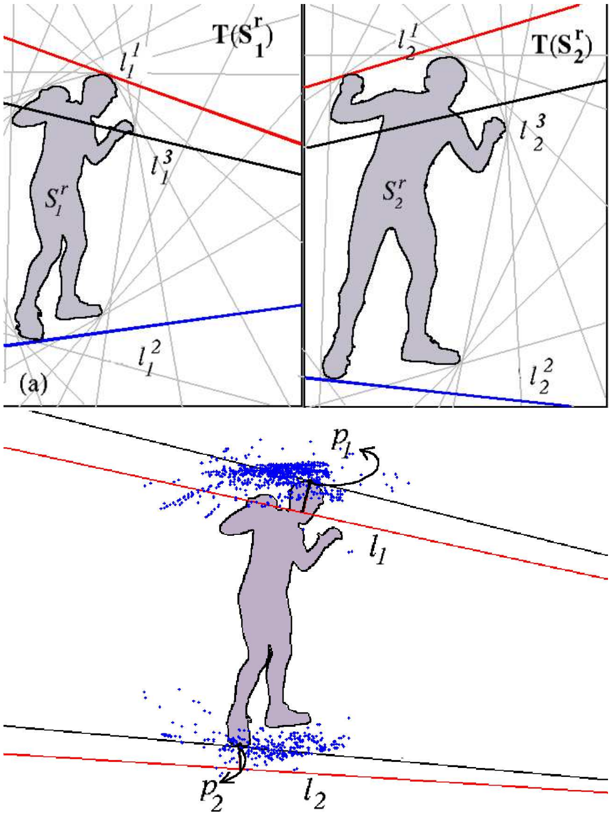

Figure 5: (a) The 4D hypothesis of the epipoles (not in picture). (b) Complete collection of frontier points for one specific epipole hypothesis and one pair of transferred epipolar lines l_1, l_2 (with large residual transfer error).

[1] In case silhouettes are clipped in this frame, the second pair of directions could be chosen from another frame.

[2] For simplicity we assume that the first epipolar tangent pair corresponds as well as the second pair of tangents. This limitations could be easily removed by verifying both hypotheses for every random sample.

3.2.2 Model Verification

Once a model for the epipolar geometry is available, we verify its accuracy. We do this by computing 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 tangent in the pencil of the first view is transferred through $H_{ij}^{-\top}$ to the second view (see Fig. 5(b)) and the reprojection error of the transferred line from the point of tangency in that particular frame is computed. We count the outliers that exceed a reprojection error threshold (we choose this to be 5 pixels) and throw away our hypothesis if the outlier count exceeds a certain fraction of the total expected inlier count. This allows us to abort early whenever the model hypothesis is completely inaccurate (an approach inspired by [6]). Thus tangents to all the silhouettes S_i^j, j ϵ 1 ... M in view i, $i = 1, 2$ would be computed only for a promising hypothesis. For all such promising hypotheses an inlier count is maintained using a lower threshold (we choose this to be 1.5 pixels).

After a solution with a sufficiently high inlier fraction has been found, or a preset maximum number of iterations has been exhausted, we select the solution with the most inliers and improve our estimate of F for this hypothesis through an iterative process of non-linear Levenberg-Marcquardt minimization while continuing to search for additional inliers. Thus, at every iteration of the minimization, we recompute the pencil of tangents for the whole silhouettes sequence S_i^j, j ϵ 1 ... M in view i, $i = 1, 2$ until the inlier count converges. The cost function minimized is the symmetric epipolar distance measure in both images. At this stage we also recover the frontier point correspondences (the points of tangency) for the full sequence of silhouettes in the two views.

3.3. Computing Projective Cameras

Typical approaches for computing projective structure and motion recovery require correspondences over at least 3 views. However, it is also possible to compute them based on two-view correspondences. Levi and Werman [14] have recently described how this could be achieved given a subset of all possible fundamental matrices between N views. They were mainly concerned with theoretical analysis and their proposed algorithm is not suited for practical implementation in the presence of noise. Here we briefly describe our approach which provides a projective reconstruction of the camera network.

The basic building block that we first resolve is a set of 3 cameras with non-collinear centers for which the 3 fundamental matrices F_{12}, F_{13}, F_{23} have been computed (Fig. 6(a),(b)). Given those, we use linear methods to find a consistent set of projective cameras P_1, P_2 and P_3 (see Eq.1) [9], choosing P_1 and P_2 as follows :

$$\begin{gathered} P_1 = [I|0] \quad P_2 = [[e_{21}]_\times F_{12}|e_{21}] \\ P_3 = [[e_{31}]_\times F_{13}|0] + e_{31}v^T \end{gathered} \tag{1}$$

P_3 is determined upto an unknown 4-vector v (Eq. 1). Expressing F_{23} as a function of P_2 and P_3 we obtain :

$$\overline{F}_{23} = [[e_{32}]_\times P_3 P_2^+ \tag{2}$$

which is linear in v, such that all possible solutions for F_{23} span a 4D subspace of P^8 [14]. We solve for v which yields $\overline{F}_{23}$, the closest appromixation to F_{23} in the subspace. P_3 is obtained from the value of v from Eq. 1. The resulting P_1, P_2, P_3 are fully consistent with $F_{12}, F_{13}, \overline{F}_{23}$.

Using the camera triplet as a building block, we could handle our N-view camera network using two different induction steps. The first induction step is as follows. Given a consistent set of cameras for the $(k-1)$-view camera network G_{k-1} and the F matrices, F_{pk}, F_{qk} and F_{pq} for $p, q \,\epsilon\, G_{k-1}$ and k, a new view, we can build G_k using the same linear algorithm used to resolve the 3-view case. We show this induction step in Fig. 6(c). An estimate of F_{pq} is available if the epipolar geometry of view p and view q was computed in the first phase of our algorithm. Otherwise, we could derive F_{pq} since consistent projective cameras for G_{k-1} are already known. The second induction step (as shown in Fig. 6(d)) is applied when independent sets of cameras for camera networks, G_p and G_q, which have the view k in common are available. Consider the triplet of views p, q, k, $p \,\epsilon\, G_p$ and $q \,\epsilon\, G_q$. Based on Eq. 1, cameras P_k and P_q can be chosen as,

$$P_k = [I|0] \quad P_q = [[e_{qk}]_\times F_{kq}|e_{qk}]$$

and $\overline{F}_{pq}$ can be estimated similar to $\overline{F}_{23}$ and this uniquely connects G_p and G_q. F_{pk} and F_{qk} could be derived indirectly if they are not already available from the calibration procedure. This method works with N cameras in general position if one can robustly compute the epipolar geometry for at least $(2N-3)$-view pairs. We use the view triplet as the fundamental building block since the N-view camera network we solve for, can always be decomposed into a single triangle strip. A single triangle strip with N vertices must have $2N-3$ edges by Euler's relation. Using this approach, more general graphs of fundamental matrices can also easily be dealt with. For a detailed discussion of all solvable cases the reader is referred to [14].

3.4. Computing Metric Cameras

In this section we briefly describe how the projective calibration obtained by the method described in 3.3 can be upgraded to a metric calibration. First, we use the linear self-calibration algorithm [18], to estimate the transformation

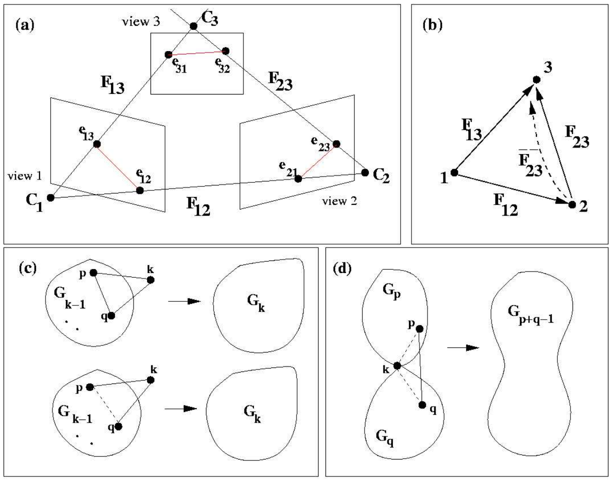

Figure 6: (a) Three non-degenerate views for which we estimate all F matrices. (b) The three-view case. $\overline{F}_{23}$ is the closest approximation of F_{23} we compute. (c)&(d) The induction steps used to resolve larger graphs using our method.

T_E^P, for each of the projective cameras. Both the camera matrices and the frontier points are transformed using T_E^P and these are used to initialize the Euclidean bundle adjustment [21]. At this stage we could extend our camera model to include radial distortion. The Euclidean bundle produces the final calibration of the full camera network.

4. Experimental Results

We applied our techniques to an archived 4-view video footage that was 4 mins. long, captured at 30 fps and was synchronized within a frame [20]. Fig. 1 shows four corresponding frames each from a different camera. The subject is moving within the overlapping view frustum of these 4 views. Occasionally the subjects's silhouette is clipped in some of the views. All background images were available.

We selected a set of keyframes from the long 30 fps video sequences to reduce redundancy in our datasets. This is preferable because in a typical video sequence, the frontier points and the epipolar tangents remain static over long subsequences. Often the motion is periodic and examining a longer sequence does not necessarily provide more information. To deal with this issue, we selected frames that yielded new information for a limited set of epipole hypotheses (we used the 4 image corners in our implementation). From these hypothetical epipoles, a pencil of tangents are computed to the convex hull of all the silhouettes for each pair of sequences. Each of these tangents are inserted into a high-resolution angular bin of size 0.2 degrees each. We compute a minimal subset of frames that covers the set of angular bins in the valid range of angles. We ended up with upto 700 out of 7500 frames from our sequences.

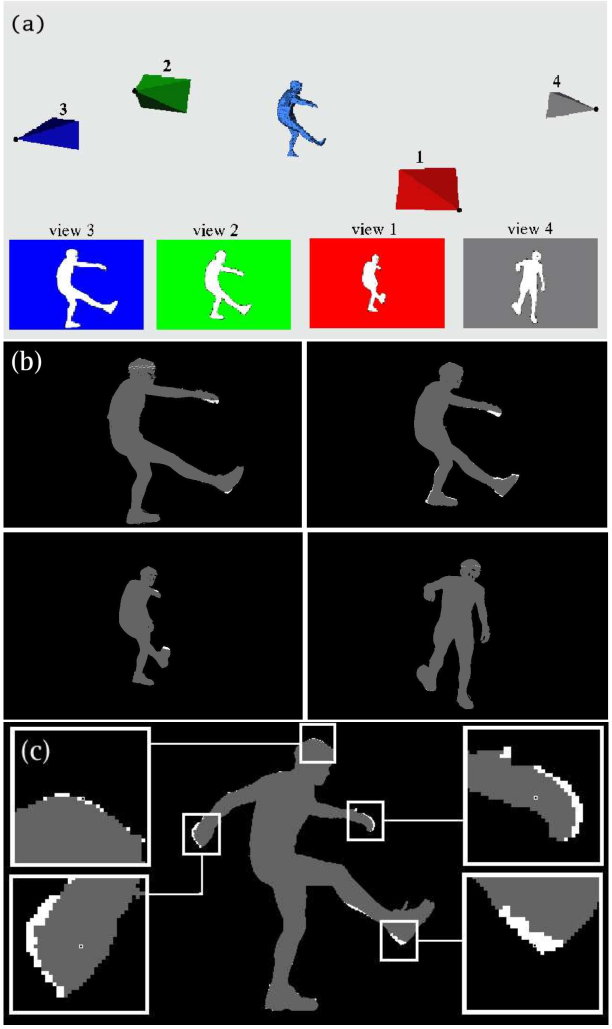

Figure 7: (a) Recovered camera configuration and visual-hull reconstruction of person. (b) The visual hull reprojected back into the four corresponding images. The silhouettes are completely filled except for fast-moving body parts. (c) Another frame in one of the views shows the effect of ignoring sub-frame synchronization.

Using the approach described in Section 3.2 we compute the epipolar geometry for all viewpairs. For the epipole hypotheses, a random epipolar tangent was selected in view i, $i = 1, 2$ at angle θ_i and a second one was selected at $N(\theta_i + 180^o, \sigma)$ (we chose σ to be 30^o). On an average, we obtained one correct solution (converged to global minimum after non-linear refinement) for every 5000 hypothesis[3]. This took approximately 15 seconds of computa-

[3] For the different camera pairs we get respectively one in 5555, 4412, 4168, 3409, 9375 and 5357. The frequency was computed over a total of 150,000 hypothesis for each viewpair.

tion time on a 3.0 GHz PIV with 1 GB RAM. Assuming a Poisson distribution, 15,000 hypothesis would yield approximately 95% probability of finding the correct solution and 50,000 hypothesis would yield 99.99% probability.

We computed the projective camera matrices for the four cameras used in this experiment from the fundamental matrices $F_{12}, F_{13}, F_{23}, F_{14}, F_{24}$. F_{23} and F_{24} were adjusted so that they were consistent with the other fundemental matrices. The projective camera estimates were then improved through a projective bundle adjustment (reducing the reprojection error from 4.6 pixels to 0.44 pixels). The final reprojection error after self-calibration and metric bundle adjustment was 0.73 pixels. Using these projection matrices the visual-hull was constructed as seen in Figure 7(a). To test the accuracy of our obtained calibration, we projected the reconstructed visual hull back into the images. For a perfect system the silhouettes would be filled completely. Mis-calibration would give rise to empty regions in the silhouettes. These tests gave consistent results on our 4-view dataset (see Figure 7(b)). The silhouettes are completely filled, except for fast moving bodyparts where the reprojected visual hull is sometimes a few pixels smaller on one side of a silhouette (see Figure 7(c)). This is due to nonperfect synchronization (subframe offsets were ignored) or poor segmentation due to motion blur or shadows.

Additional experiments were performed with a 2-view dataset that was about 1.5 mins. long, and captured at 30 fps. Fig. 8 shows two corresponding frames with a few epipolar lines corresponding to the fundamental matrix F, that we compute. The reference F was computed by manually picking 50 corresponding features using the method described in [9][Ch.10, p.275]. Our computed F was used to transfer these 50 features from the first view to the second and vice-versa. Fig. 8(c) shows the distribution of the symmetric epipolar transfer error. These results are comparable with the results prior to bundle adjustment for the first set of experiments.

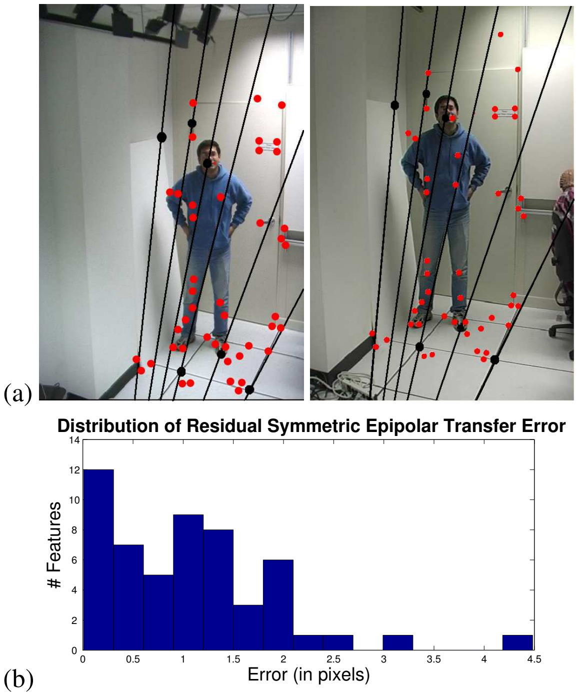

Figure 8: (a) Two corresponding frames from a second dataset with corresponding epipolar lines computed by our algorithm. (b) The distribution of the symmetric epipolar transfer error for the fundamental matrix we compute, corresponding to 50 manually clicked points. The root mean square residual was 1.38 pixels (this is prior to bundle adjustment.

5. Summary and Conclusions

In this paper we have presented a complete approach to obtain the full metric calibration of a camera network from silhouettes. The core of the proposed method is a RANSAC-based algorithm to efficiently compute the fundamental matrix. The proposed method is both robust and accurate. An important advantage of our approach is that it allows calibrating camera networks without the need for the acquisition of specific calibration data. This can be particularly relevant when physical access to the observed space is impractical and when reconfiguration of an active camera network is required during operations, making it suitable for surveillance camera networks. Our approach is intrinsically well suited for dealing with widely separated views, typical of surveillance camera networks and the robust algorithm we utilize, allows us to deal with noisy silhouettes caused by poor background segmentation or motion blur. Another advantage of our method is that it would work well in the absence of texture and is insensitive to poor photometric calibration. At this point we require reasonably well synchronized cameras, but in the future we intend to explore an extension of our approach to deal with unsynchronized cameras and reconstruct visual hulls from unsynchronized video footage. Conceptually, this could be achieved by sampling over one additional dimension. We also intend to study more in detail the possibilities of maintaining the calibration of active camera networks based on silhouettes.

Acknowledgements

We would like to thank Peter Sand [20] for providing us the 4-view dataset. This work was partially supported by the NSF Career award IIS 0237533 and the DARPA/DOI award NBCH 1030015.

References

[1] K. Astrom, R. Cipolla and P. J. Giblin, *Generalised Epipolar Constraints,* In Proc. of ECCV, 1996, pages 97 - 108.

[2] K. Astrom, R. Cipolla and P. J. Giblin, *Motion from the Frontier of Curved Surfaces,* In Proc. IEEE Int. Conf. on COmputer VIsion, 1995, pages 269 - 275.

[3] B. G. Baumgart, *Geometric Modeling for Computer Vision,* Ph.D. Dissertation, Stanford University, August 1974.

[4] R. C. Bolles and M. A. Fischler. *A RANSAC-based approach to model fitting and its application to finding cylinders in range data.* In Proceedings, IJCAI, pages 637–643, 1981.

[5] C. Buehler, W. Matusik,and L. Mcmillan, *Polyhedral Visual Hulls for real-time rendering,* In Proceedings of Eurographics Workshop on Rendering, 2001.

[6] O. Chum and J. Matas, *Randomized RANSAC with Td,d test* In Proc. BMVC, pp. 448-457, 2002.

[7] T. Joshi,N. Ahuja, and J. Ponce, *Structure and motion estimation from dynamic silhouettes under perspective projection,* In Proc. of ICCV '95 (1995).

[8] G. K. M. Cheung, S. Baker,T. Kanade, *Visual hull alignment and refinement across time: a 3D reconstruction algorithm combining shape-from-silhouette with stereo,* In Proc. of CVPR'03, Vol 2, pages 375-382, June'03.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry In Computer Vision,* Cambridge University Press, 2000.

[10] K. N. Kutulakos and S. M. Seitz, *A Theory of Shape by Space Carving,* IJCV, Vol. 38(3), pages 199-218, July'00.

[11] A. Laurentini, *Visual hull concept for silhouette-based image understanding,* IEEE Trans. on PAMI, 16(2):150–162,Feb'94.

[12] S. Lazebnik, E. Boyer, J. Ponce, *On Computing Exact Visual Hulls of Solids Bounded by Smooth Surfaces.* In Proc. of CVPR'01.

[13] S. Lazebnik, A. Sethi, C. Schmid, D. Kriegman, J. Ponce, M. Hebert, *On Pencils of Tangent Planes and the Recognition of Smooth 3D Shapes from Silhouettes.* In Proc of ECCV'02, Vol. 3, pp. 651-665.

[14] Levi N., Werman M., *The Viewing Graph.* In Proc of CVPR'03, June 2003.

[15] W. Matusik, C. Buehler, R. Raskar, L. McMillan, S. Gortler, *Image-Based Visual Hulls,* In Proc. of SIGGRAPH 2000.

[16] A. Melkman, *On-line Construction of the Convex Hull of a Simple Polygon,* Information Proc. Letters 25, p.11, 1987.

[17] P.R.S. Mendonca, K.-Y.K. Wong and R. Cipolla, *Epipolar Geometry from profiles under circular motion,* IEEE Trans. on PAMI 23 (6) (2001) 604-616.

[18] M. Pollefeys, F. Verbiest, L. Van Gool, *Surviving dominant planes in uncalibrated structure and motion recovery*, A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.) Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Lecture Notes in Computer Science, Vol.2351, pp. 837-851.

[19] J. Porrill and S. Pollard, *Curve matching and stereo calibration,* Image and Vision Computing 9, pp. 45–50, 1991.

[20] P. Sand, L. Mcmillan, J. Popovic, *Continuous Capture of Skin Deformation,* SIGGRAPH 2003, July 2003.

[21] B. Triggs, P. McLauchlan, R. Hartley, A. Fiztgibbon, *Bundle Adjustment: A Modern Synthesis*, In B. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, LNCS Vol.1883, pp.298-372, Springer-Verlag, 2000.

[22] B. Vijayakumar, D. Kriegman, and J. Ponce, *Structure and motion of curved 3D objects from monocular silhouettes,* In Proc. of CVPR'96, pages 327–334, June 1996.

[23] K.-Y. K. Wong and R. Cipolla, *Structure and motion from silhouettes,* In Proc. of ICCV'01, June '01

[24] A.J.Yezzi and S. Soatto, *Structure from motion for scenes without features,* In Proc. of CVPR'03, pp:I-171-I-178,vol.1.

[25] Z. Zhang, *Determining the epipolar geometry and its uncertainty: A review.* IJCV, 27(2): 161–195, 1998.

[26] Z. Zhang. *Flexible camera calibration by viewing a plane from unknown orientations*. In Proc. of ICCV99, pages 666–673, Corfu, Greece, Sep'99.