CALIBRATING A NETWORK OF CAMERAS FROM LIVE OR ARCHIVED VIDEO

Sudipta N. Sinha, Marc Pollefeys

{ssinha,marc}@cs.unc.edu
Department of Computer Science,
University of North Carolina at Chapel Hill, USA.

ABSTRACT

We present an automatic approach for calibrating a network of cameras using live video captured from them. Our method requires video sequences containing moving people or objects but does not require any special calibration data. The silhouettes of these moving objects visible in a pair of views, are used to compute the epipolar geometry of that camera pair. The fundamental matrices computed by this method are used to first obtain a projective reconstruction of the complete camera configuration. Self-calibration is then used to upgrade the projective reconstruction into a metric reconstruction. We have extended our approach to deal with unsynchronized video sequences captured at the same frame-rate, by simultaneously recovering the epipolar geometry as well as the temporal offset between a pair of cameras. We use our approach to calibrate and synchronize a four-camera system using archived video containing a moving person. Next, the silhouettes are used to construct the visual hull of the moving person using known Shape-from-Silhouette algorithms. Additional experiments on computing the fundamental matrix of two views from silhouettes are also performed.

1. INTRODUCTION

In surveillance camera networks, live video of a dynamic scene is often captured from multiple views. We aim to recover the complete calibration of such camera networks using only the videos of the observed dynamic events, which will eventually be used for 3D-reconstruction of these events. This will enable us to calibrate camera networks observing large area training activities and cultural events. Different pairs of archived video sequences may have a time-shift between them (assuming all the cameras have the same frame rate) since recording would be triggered by moving objects, with different cameras being activated at different instants in time. Our method simultaneously recovers the synchronization and epipolar geometry of such a camera pair. This method is particularly useful for Shape from Silhouette systems [1, 2, 3] as visual hulls can now be reconstructed from uncalibrated and unsynchronized video of moving objects.

Different existing Structure and Motion approaches using silhouettes [4, 5, 6] either require good initialization or fail for certain camera configurations and most of them require static scenes. Traditionally, calibration objects like checkerboard patterns or LED's have been used for calibrating multi-camera systems [7] but this requires physical access to the observed space. This would be impossible for a remotely deployed camera network. Our method can calibrate such cameras remotely and also handle wide-baselines camera pairs, arbitrary camera configurations and also a lack of photometric calibration.

At the core of our approach is a robust RANSAC [8] based algorithm that computes the epipolar geometry from two video sequences of dynamic objects. This algorithm is based on the constraints arising from the correspondence of frontier points and epipolar tangents [4, 9, 10] of silhouettes in two views. These are points on an objects' surface which project to points on the silhouette in two views. Epipolar lines passing through the images of a frontier point must correspond. Such epipolar lines are also tangent to the silhouettes at the imaged frontier points. Previous work used those constraints to refine an existing epipolar geometry [9, 10]. Here we take advantage of the fact that video sequences of dynamic objects will contain many different silhouettes, yielding many constraints that must be satisfied. We use RANSAC [8] not only to remove outliers in silhouette data but also sample the space of unknown parameters. We first demonstrate how the method works with synchronized video. We then describe how pair-wise fundamental matrices and frontier point can be used to compute a projective reconstruction of the complete camera network, which is then refined to a metric reconstruction. An extension of the RANSAC based algorithm allows us to recover the temporal offset between a pair of unsynchronized video sequences, where both are acquired at the same frame rate. A method to synchronize the whole camera network is then presented.

In Sec. 2 we present the background theory. Sec. 3 describes the algorithm that computes the epipolar geometry from dynamic silhouettes. Full camera network calibration is discussed in Sec. 4 while Sec. 5 describes how we deal with unsynchronized video. Experimental results are presented in different sections of the paper and we conclude with scope for future work in Sec. 6.

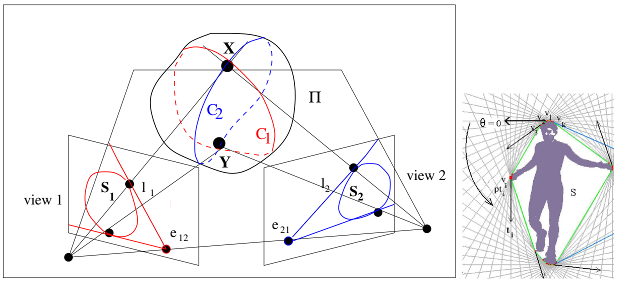


Figure 1: (a)Frontier Points and Epipolar Tangents.(b) The Tangent Envelope.

2. BACKGROUND AND PREVIOUS WORK

Our algorithm exploits the constraints arising from the correspondence of frontier points and epipolar tangents [4, 9]. Frontier points on an objects' surface are 3D points which project to points on the silhouette in the two views. In Fig. 1(a), X and Y are frontier points on the apparent contours C_1 and C_2, which project to points on the silhouettes S_1 and S_2 respectively. 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 images of frontier points, X and Y can correspond. Morever, the image of the frontier points corresponding to the outer-most epipolar tangents [4] must lie on the convex hull of the silhouette. The silhouettes are stored in a compact data structure called the tangent envelope, [11] (see Fig. 1(b)).

Video of dynamic objects contain many different silhouettes, yielding many constraints that are satisfied by the true epipolar geometry. Unlike [12] who search for all possible frontier points and epipolar tangents on a single silhouette, we only search for the outermost frontier points and epipolar tangents, but from multiple silhouettes. Sufficient motion of the object within the 3D observed space gives rise to a good spatial distribution of frontier points and increases the accuracy of the fundamental matrix.

3. COMPUTING THE EPIPOLAR GEOMETRY

The RANSAC-based algorithm takes two sequences as input, where the j^{th} frame in sequence i is denoted by S_i^j and the corresponding tangent envelope by $T(S_i^j)$. F_{ij} is the fundamental matrix between view i and view j, (transfers points in view i to epipolar lines in view j) and e_{ij}, the epipole in view j of camera center i. While a fundamental matrix has 7 dof's, we only randomly sample in a $4D$ space because if the epipoles are known, the frontier points can be determined, and the remaining degrees of freedom of the epipolar geometry can be derived

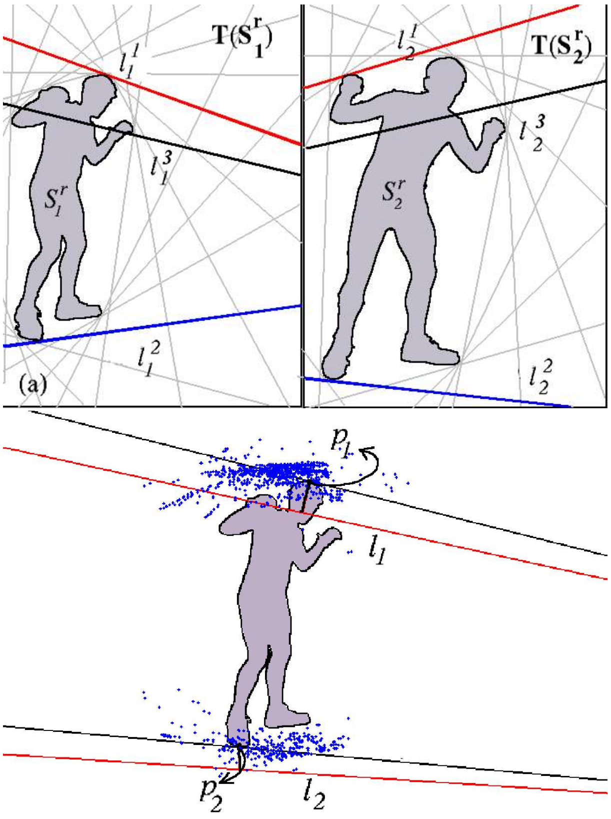


Figure 2: (a) The 4D hypothesis of the epipoles (not in picture). (b) All frontier points for a specific hypothesis and a pair of transferred epipolar lines l_1, l_2.

from them. The pencil of epipolar lines in each view centered on the epipoles, is considered as a $1D$ projective space [13] [Ch.8, p.227]. The epipolar line homography between two such $1D$ projective spaces is a $2D$ homography. Knowing the epipoles e_{ij}, e_{ji} and the epipolar line homography fixes F_{ij}. Three pairs of corresponding epipolar lines are 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 $F_{ij} = [e_{ij}]_\times H_{ij}$.

At every iteration, we randomly choose the rth frames from each of the two sequences. As shown in Fig. 2(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

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

produces the epipole hypothesis (e_{12} , e_{21}). 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 2(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. 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. 2(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. 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 1.25 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.

3.1. Results

Experiments were performed with two different 2-view video sequences, each with a moving person in an indoor environment and about 2 minutes long captured at 30 fps. See Fig. 3(a),(c) for two corresponding frames with epipolar lines corresponding to the fundamental matrices we compute for the two datasets respectively. Manually

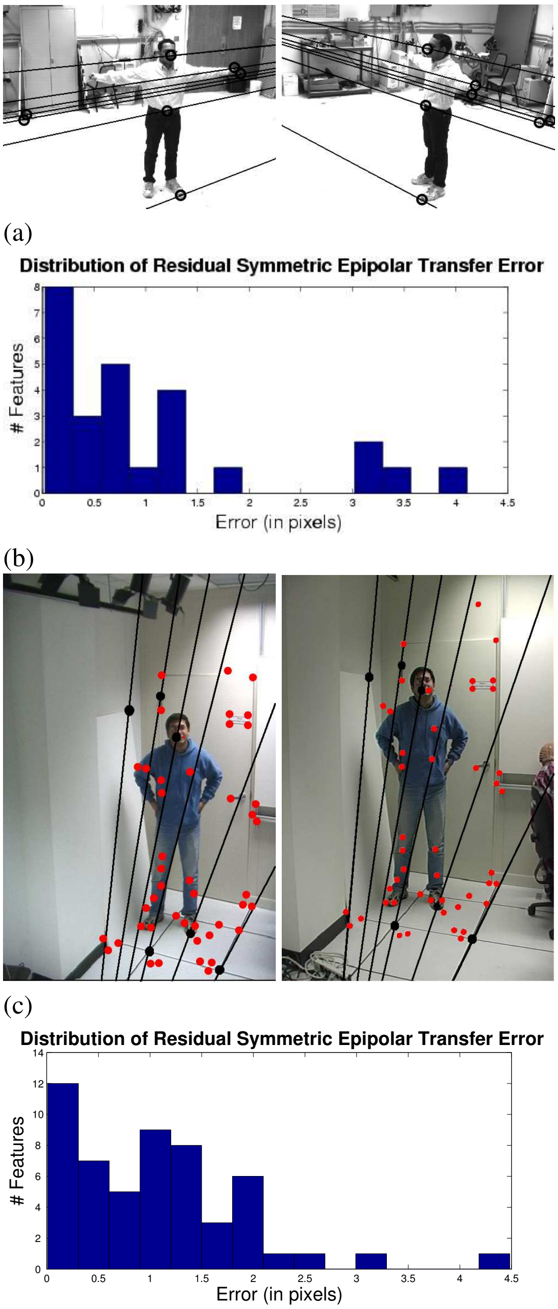


Figure 3: (a) Corresponding frames from dataset 1 (wide-baseline camera pair with textureless floor and few reliable common features) showing corresponding epipolar lines that we computed. (b) The distribution of the symmetric epipolar transfer error for the computed fundamental matrix F, for 26 manually clicked points for dataset 1. (Root mean square residual = 1.31 pixels). (c) Corresponding frames from dataset 2 (vertically oriented cameras) showing corresponding epipolar lines that we computed. (d) The distribution of the symmetric epipolar transfer error for the computed F, for 54 manually clicked points. (Root mean square residual = 1.38 pixels).

[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.

clicked corresponding points in the 2 views were used to test the accuracy of the computed F, as shown in Fig. 3(b),(d). The root mean square reprojection error for the two datasets were 1.31 and 1.38 pixels respectively (Note that these errors will be reduced further after the bundle adjustment step).

4. CAMERA NETWORK CALIBRATION FROM PAIRWISE EPIPOLAR GEOMETRY

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 only 2-view correspondences. Levi and Werman [14] have recently shown how this could be achieved given a subset of all possible fundamental matrices between N views with special emphasis on the solvability of various camera networks. Here we briefly describe our iterative 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 colinear centers for which the 3 fundamental matrices F_{12}, F_{13} and F_{23} have been computed (Fig. 4(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) [13], choosing P_1 and P_2 as follows :

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

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} and $\overline{F}_{23}$.

Using the camera triplet as a building block, we could handle our N-view camera network by the method of induction. The projective reconstruction of a triplet (as described above) initialises the projective reconstruction of the whole network. At every step a new view that has edges to any two views within the set of cameras reconstructed so far forms a new triplet which is resolved in identical fashion. This process is repeated until all the cameras have been handled.

This projective calibration is first refined using a projective bundle adjustment which minimizes the reprojection error of the pairwise frontier point matches. Next, we use the linear self-calibration algorithm [15] to estimate the rectifying transform for each of the projective cameras. We rectify these projective cameras into metric cameras, and use them to initialize the Euclidean bundle adjustment [16]. The Euclidean bundle adjustment step produces the final calibration of the full camera network.

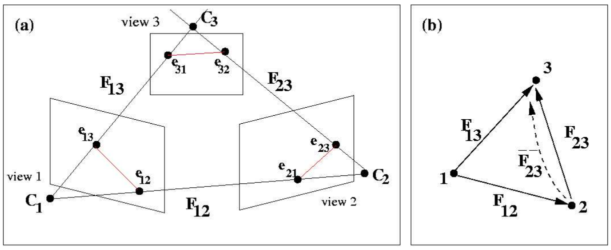


Figure 4: (a) Three non-degenerate views for which we estimate all F matrices. (b) The three-view case. $\overline{F}_{23}$ is the closest approximation of F_{23} we compute. (c)&(d) The induction steps used to resolve larger graphs using our method.

4.1. Results

Here we present results from full calibration of the 4-view video dataset which was 4 minutes long and captured at 30 fps [3] (see 5). We computed the projective cameras from the fundamental matrices $F_{12}, F_{13}, F_{23}, F_{14}, F_{24}$. On an average, we obtained one correct solution, one which converged to a global minimum after non-linear refinement for every 5000 hypothesis[3]. This took approximately 15 seconds of computation time on a 3.0 GHz PC 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.

F_{23} and F_{24} had to be adjusted by the method described in Section 4, which actually improved our initial estimates. The projective camera estimates were then refined 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 5(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 5(b)). The silhouettes are completely filled, except for fast moving bodyparts where the reprojected visual hull is sometimes a few pixels smaller on one

[3] For all 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.

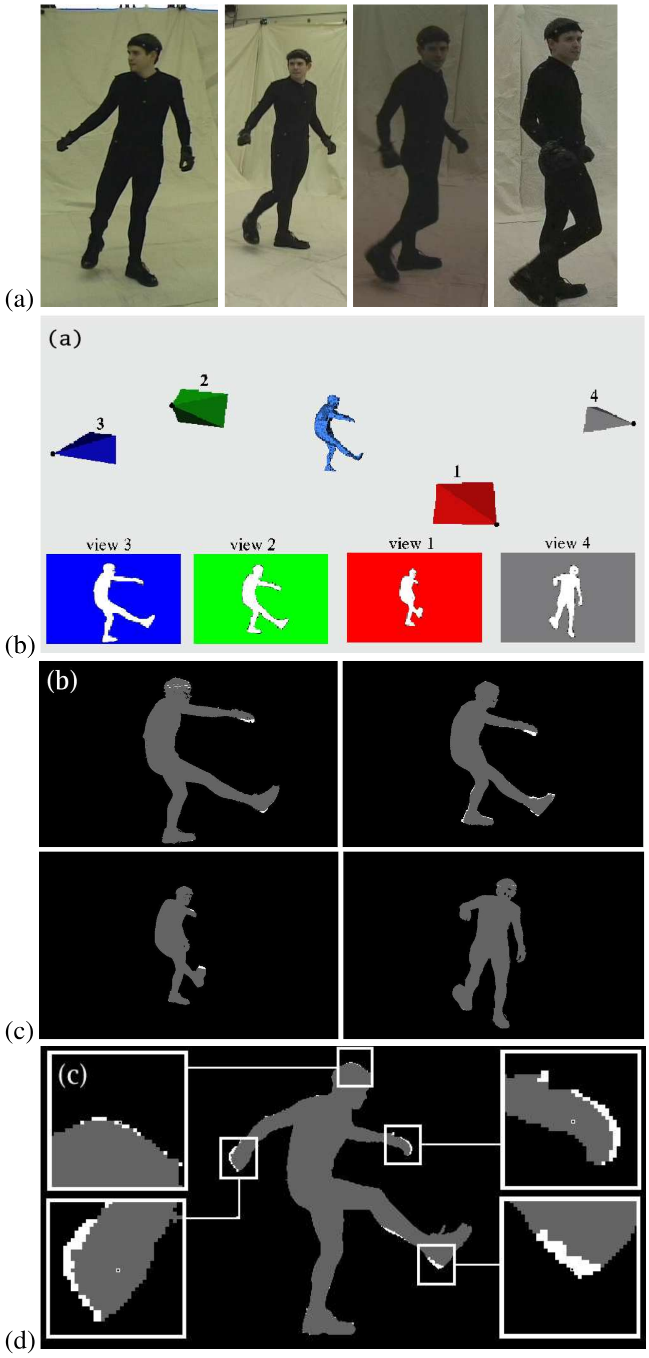


Figure 5: (a) A 4-view Uncalibrated Video Sequence. (b) Recovered camera configuration and visual-hull reconstruction of person. (c) The visual hull reprojected back into the four corresponding images. The silhouettes are completely filled except for fast-moving body parts. (d) Another frame in one of the views shows the effect of ignoring sub-frame synchronization.

side of a silhouette (see Figure 5(c)). This is due to non-perfect synchronization (subframe offsets were ignored) or poor segmentation due to motion blur or shadows.

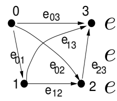
In typical video, outermost frontier points and epipolar tangents often remain stationary over a long time. Such static frames are redundant and representative keyframes must be chosen to make the algorithm faster. We do this by considering hypothetical epipoles (at the 4 image corners), pre-computing tangents to all the silhouettes in the whole video and binning them and picking representative keyframes such that at least one from each bin is selected. For the 4-view dataset, we ended up with 600-700 out of 7500 frames.

5. CAMERA NETWORK SYNCHRONIZATION

To deal with unsynchronized video, we modify our algorithm for computing the epipolar geometry of camera pairs as follows (see [17] for details). At the hypothesis step, in addition to making a random hypothesis for the two epipoles in the $4D$ space of the pair of epipoles, we also randomly pick a temporal offset. The verification step of the RANSAC based algorithm now considers the hypothesized temporal offset for matching frames in the two views throughout the video sequence. To make the algorithm efficient we select keyframes differently, to allow a temporal offset search within a large range. Since the frames containing slow moving and static silhouettes allow a rough alignment, the tangents accumulated in the angular bins during keyframe selection are sorted by angular speed. While selecting representative keyframes we select the ones with static or slowly moving silhouettes. Once a rough alignment is known, a more exhaustive set of keyframes are used to recover the exact temporal offset within a small search range and its variance along with the true epipolar geometry.

A N-view camera network with pairwise temporal offsets, can be represented as a directed graph where each vertex represents a camera and its own clock and an edge represents an estimate of the temporal offset between the two vertices it connects. Our method in general will not produce a fully consistent graph, where the sum of temporal offsets over all cycles is zero. Each edge in the graph contributes a single constraint: $t_{ij} = x_i - x_j$ where t_{ij} is the temporal offset and x_i and x_j are the unknown camera clocks. To recover a Maximum Likelihood Estimate of all the camera clocks, we set up a system of equations from constraints provided by all the edges and use Weighted Linear Least Squares (each edge estimate is inversely weighted by its variance) to obtain the optimal camera clock offsets. An outlier edge would have only significantly non-zero cycles and could be easily detected and removed before solving the above mentioned system

(a)	e_{ij}	$range$	t_{ij}	σ_{ij}	$\bar{t}_{ij}$	$\hat{t}_{ij}$
	e_{01}	*[-13,-3]*	*-8.7*	*0.80*	*-8.50*	*-8.32*
	e_{02}	*[-11,-1]*	*-8.1*	*1.96*	*-8.98*	*-8.60*
	e_{03}	*[-12,-2]*	*-7.7*	*1.57*	*-7.89*	*-7.85*
	e_{12}	*[-5,5]*	*-0.93*	*1.65*	*-0.48*	*-0.28*
4-view	e_{13}	*[-5,5]*	*0.54*	*0.72*	*0.61*	*0.47*
network graph	e_{23}	*[-6,4]*	*1.20*	*1.27*	*1.09*	*0.75*



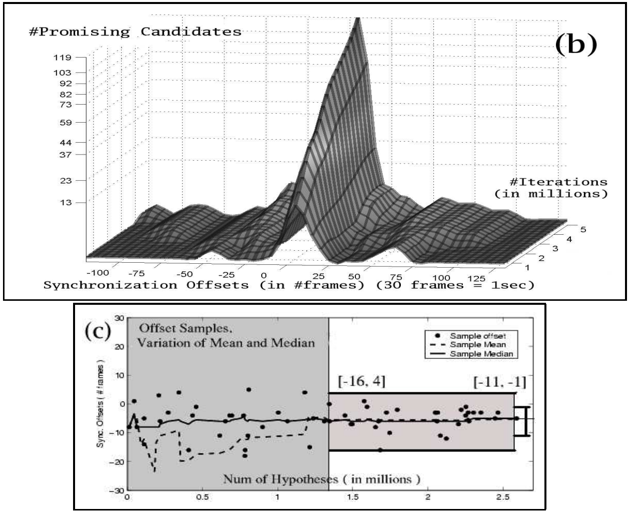


Figure 6: (a) Results of camera network synchronization. (b) Typical sync. offset distribution. (c) Sample offset distribution for rough alignment phase.

of equations. This method will produce very robust estimates for complete graphs but will work as long as a fully connected graph with at least N-1 edges is available.

5.1. Results

We tried our approach on the same 4-view video dataset that was manually synchronized earlier (see Fig. 5). All six view-pairs were synchronized within a search range of 500 frames (a time-shift of 16.6 secs). The sub-frame synchronization offsets from the 1st to the 2nd, 3rd and 4th sequences were found to be 8.50, 8.98, 7.89 frames respectively, the corresponding ground truth offsets being 8.32, 8.60, 7.85 frames. The computed offsets we compute are within 1/75 seconds of the true temporal offsets. Fig. 6(a) tabulates for each view-pair, the +/-5 interval computed from initial rough alignment, the estimates (t_{ij},σ_{ij}) computed by searching within that interval, the Maximum Likelihood Estimate of the consistent offset $\bar{t}_{ij}$, and the ground truth $\hat{t}_{ij}$. Rough alignment required 1.3-2.9 million hypotheses, and 60-120 seconds on a 3 GHz PC with 1 GB RAM.

For the pair of views, 1 & 2, Fig. 6(b) shows the offset distribution within +/-125 frames of the true offset for hypotheses ranging between 1 to 5 million in count. The peak in the range [-5,5] represents the true offset. Smaller peaks indicate the presence of some periodic motion in parts of the sequence. Fig. 6(c) shows a typical distribution of offsets obtained during a particular run and shows the converging search intervals. Fig. 7 shows the effect of synchronizing the 4 video sequences which allows the final calibration and reconstruction, shown in Fig. 5(a).

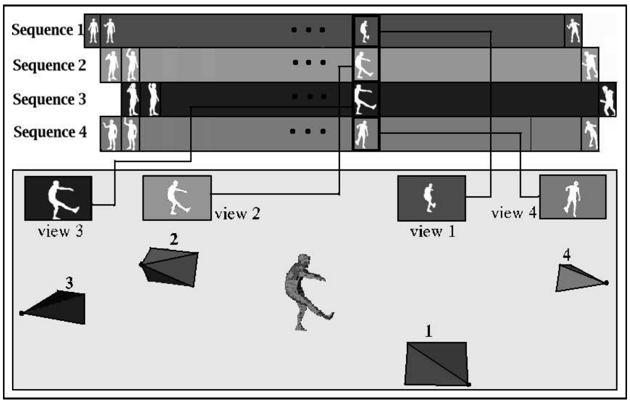


Figure 7: (d) Final Results: synchronization, calibration and reconstruction of the 4-view dataset only from video sequences.

6. CONCLUSIONS AND FUTURE WORK

We presented an approach to determine the calibration and synchronization of a network of cameras from possibly unsynchronized videos of moving objects, observed by them. Our method is based on a robust algorithm that efficiently computes the temporal offset between two sequences and the epipolar geometry of the respective views. The proposed method is robust and accurate and allows calibration of camera networks without the need for acquiring specific calibration data. In future, we intend to explore the possibility of calibrating active pan tilt zoom (PTZ) camera networks using this approach. Preliminary calibration results in this direction are described in [18] (see Fig. 8(b)). Fig. 8 shows an example of a high-resolution calibrated panoramic mosaic computed automatically by a rotating camera. Most PTZ cameras can be modelled as a static omnidirectional camera with a fixed center of projection that coincides with the center of rotation and zoom of the camera. By registering video frames from an active PTZ camera to its pre-computed calibrated panorama such as shown in Fig. 8 using the background in the image, we could adopt our approach described in this paper to extract the epipolar geometry of camera pairs using the warped silhouettes. Morever multiple silhouettes observed at different spots in a wide-area environment could be used to obtain more accurate estimates of the epipolar geometries.

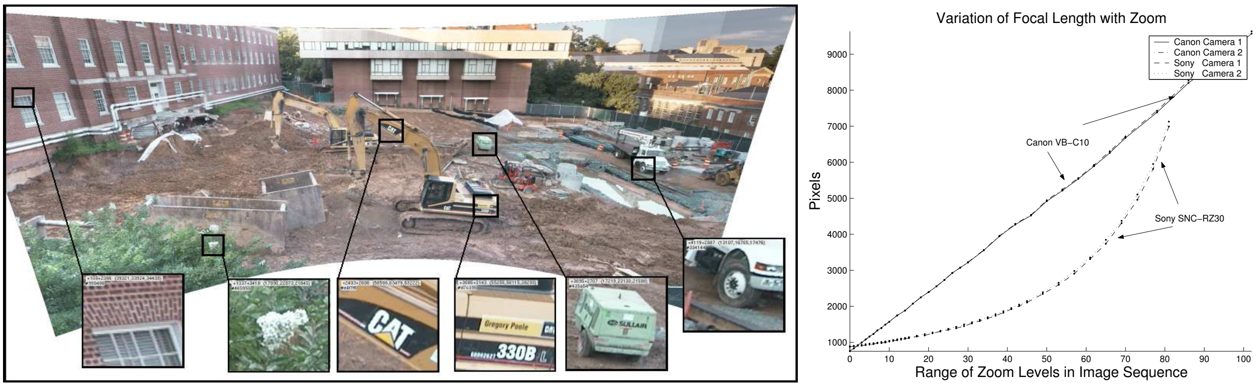


Figure 8: (a)High-resolution panorama (6000 × 6000 pixels) built from 119 images at 5X zoom. Note the zoomed-in regions of the panorama, displayed in the original scale.(b) The variation of focal length of pan-tilt-zoom (PTZ) cameras with zoom for Canon and Sony camera models recovered from full-range zoom calibration.

Acknowledgements

We would like to thank Peter Sand [3] for providing us the 4-view dataset from MIT and Aravind Sunderesan for giving us archived video sequences from Keck Lab, UMD.

7. REFERENCES

[1] C. Buehler, W. Matusik, and L. Mcmillan, "Polyhedral visual hulls for real-time rendering," in *Eurographics Workshop on Rendering*, 2001.

[2] G.K.M. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: a 3d reconstruction algorithm combining shape-from-silhouette with stereo," in *CVPR03*, 2003, pp. II: 375–382.

[3] P. Sand, L. McMillan, and J. Popovic, "Continuous capture of skin deformation," in *Siggraph*, 2003, pp. 578–586.

[4] K.Y.K. Wong and R. Cipolla, "Structure and motion from silhouettes," in *ICCV01*, 2001, pp. II: 217–222.

[5] B. Vijayakumar, D. Kriegman, and J. Ponce, "Structure and motion of curved 3d objects from monocular silhouettes," in *CVPR*, 1996, pp. 327–334.

[6] A.J. Yezzi and S. Soatto, "Structure from motion for scenes without features," in *CVPR*, 2003, pp. I: 525–532.

[7] Z.Y. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *ICCV*, 1999, pp. 666–673.

[8] M.A. Fischler and R.C. Bolles, "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *IJCAI81*, 1981, pp. 637–643.

[9] J. Porrill and S. Pollard, "Curve matching and stereo calibration," *IVC*, vol. 9, pp. 45–50, 1991.

[10] K. Astrom, R. Cipolla, and P. Giblin, "Generalised epipolar constraints," in *ECCV*, 1996, pp. II:97–108.

[11] S.N. Sinha and M. Pollefeys, "Camera network calibration from dynamic silhouettes," in *CVPR*, 2004.

[12] Y. Furukawa, A. Sethi, J. Ponce, and D. David Kriegman, "Structure and motion from images of smooth textureless objects," in *ECCV*, 2004.

[13] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[14] N. Levi and M. Werman, "The viewing graph," in *CVPR03*, 2003, pp. I: 518–522.

[15] M. Pollefeys, R. Koch, and L.J. Van Gool, "Self calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *IJCV*, vol. 32, no. 1, pp. 7–25, August 1999.

[16] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – A modern synthesis," in *Vision Algorithms: Theory and Practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds., LNCS, pp. 298–375. Springer Verlag, 2000.

[17] S.N. Sinha and M. Pollefeys, "Synchronization and calibration of camera networks from silhouettes," in *ICPR*, 2004.

[18] S.N. Sinha and M. Pollefeys, "Towards calibrating a pan-tilt-zoom camera network," in *OMNIVIS*, 2004.