Synchronization and Calibration of Camera Networks from Silhouettes

submitted to ICPR'04.

Sudipta N. Sinha Marc Pollefeys.
Dept. of Computer Science, University of North Carolina at Chapel Hill.

Abstract

We propose an automatic approach to synchronize a network of uncalibrated and unsynchronized video cameras, and recover the complete calibration of all these cameras. In this paper, we extend recent work on computing the epipolar geometry from dynamic silhouettes, to deal with unsynchronized sequences and find the temporal offset between them. This is used to compute the fundamental matrices and the temporal offsets between many view-pairs in the network. Knowing the time-shifts between enough view-pairs allows us to robustly synchronize the whole network. The calibration of all the cameras is recovered from these fundamental matrices. The dynamic shape of the object can then be recovered using a visual-hull algorithm. Our method is especially useful for multi-camera shape-from-silhouette systems, as visual hulls can now be reconstructed without the need for a specific calibration session.

1. Introduction

Shape-from-Silhouette methods [2, 8], attempt to compute the visual hull [6] of an object, which is the maximal shape that produces the same set of silhouettes seen from multiple views. The rays through the center of a calibrated camera and points on the silhouette define a viewing cone [6]. Intersecting viewing cones backprojected from silhouettes in multiple views produces the visual hull of the object. Methods dealing with dynamic objects [2], require synchronized image frames to accurately reconstruct its shape. For calibration most multi-camera systems use specific offline procedures which require moving a planar pattern [9] or a LED in the camera's field of view to acquire the calibration data. This requires physical access to the observed space and precludes reconfiguration of cameras during operation. Some of the past approaches for structure-from-motion for silhouettes are impractical for arbitrary unknown camera configurations since they may require specific camera setups (ie. at least partially circular) [8].

In [1], a camera network was calibrated from synchronized video sequences of a dynamic object using only its silhouettes. This was done by robustly computing the epipolar geometry from synchronized sequences, repeatedly for many view-pairs. In surveillance camera networks, recording is often triggered by moving objects, and different cameras could be activated at different instants in time. Hence any two video sequences from the network could have a time-shift between them (assuming identical frame-rates). In this paper we extend the method proposed in [1], to deal with such unsychronized sequences and compute both the epipolar geometry and the synchronization offset simultaneously. We robustly synchronize the camera network after computing the time-shift between enough view-pairs. The camera calibration from the fundamental matrices and the reconstruction using synchronized video frames is described in [1]. Our method uses moving objects as cues for spatio-temporal alignment [3], and verifies a hypothesized epipolar geometry similar to [3]. It can deal with large temporal offsets and does not require rough alignment like [10].

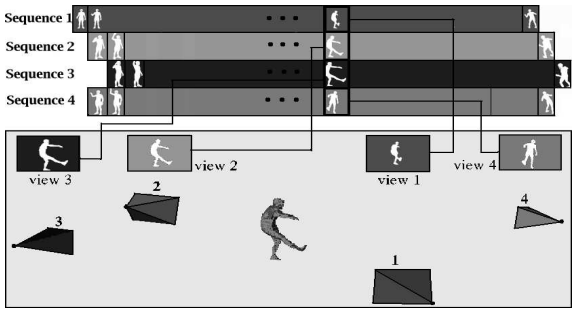


Figure 1. (a)Visual-hull reconstruction of person and recovered camera calibration from four unsynchronized video sequences.

2 Background and Previous Work

The algorithm in [1] is based on the constraints arising from the correspondence of frontier points and epipolar tangents [8, 7]. These are points on an objects' surface which

project to points on the silhouette in two views. In Fig 2(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 are guaranteed to correspond. The image of the frontier points corresponding to the outer-most epipolar tangents [8] must lie on the convex hull of the silhouette. The silhouettes are stored in a compact data structure called the tangent envelope, (refer [1], see Fig. 2(b)). Video sequences of dynamic objects contain many different silhouettes, yielding many constraints that must be satisfied. In [1], a RANSAC [4] based approach is used to search for the true epipoles in each view. At every step, a random hypothesis for the epipolar geometry is generated and subsequently verified. A pair of frames, one in each view are randomly chosen. Two directions are randomly sampled in each frame and the intersection of tangents in these directions generates the hypothesis for the epipoles. Another frame pair is randomly chosen, and tangents to its silhouettes from the hypothesized epipoles are computed. The three pair of matching lines produces an epipolar line homography [5]. Next a pencil of tangents is computed from each epipole to the silhouette sequence in each view. Tangents from the first pencil are transferred to the second view and compared with the tangents in that view. This is the verification step. Probable hypotheses are refined through a non-linear minimization stage in which the symmetric transfer error is being minimized. For unsychronized video, these constraints still exist upto an unknown parameter, the temporal offset. In this paper, we use random sampling for exploring a possible range of temporal offsets, in addition to searching the $4D$ space of epipoles and for handling outliers in the silhouette data.

3. Computing the Synchronization Offset and Epipolar Geometry

The 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 once the position of the epipoles are known, the frontier points can be determined, and the remaining degrees of freedom of the epipolar geometry can be computed from them. The pencil of epipolar lines in each view centered on the epipoles, is considered as a $1D$ projective space [5] [Ch.8, p.227]. The epipolar line homography between two

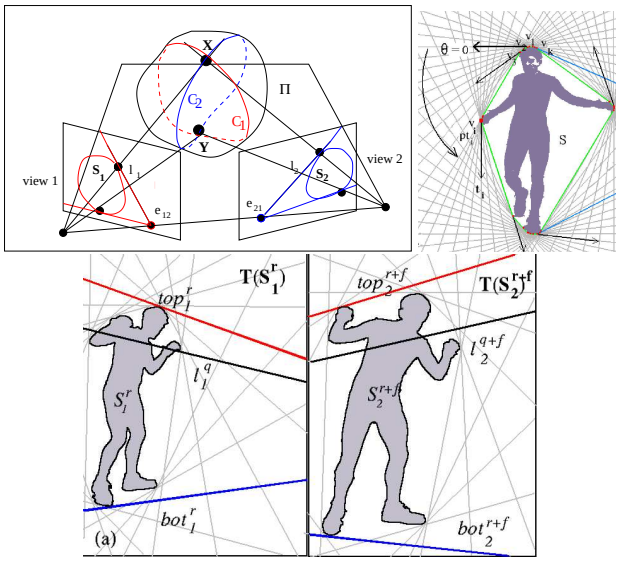


Figure 2. (a) Frontier points and epipolar tangents. (b) The Tangent Envelope. (c) The hypothesis step (epipoles not in picture).

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 $F_{ij} = [e_{ij}]_\times H_{ij}$.

A hypothesis-verification step is at the core of Algorithm 1. At every step, a hypothesis for the temporal offset, f is randomly chosen and a frame is randomly picked from the first sequence. Let this be frame r. Frame $r+f$ is chosen from the second sequence. As shown in Fig. 2(c), we randomly sample independent directions top_i^r from $T(S_i^r)$ and top_j^{r+f} from $T(S_j^{r+f})$ for one of the tangents in each view. A second pair of directions are chosen, bot_i^r=top_i^r+x_i from $T(S_i^r)$ and bot_j^{r+f}=top_j^{r+f}+x_j from $T(S_j^{r+f})$ where x_i and x_j are drawn from the normal distribution $N(180°, 30°)$. The intersections of these two pair of tangents in each image produces the epipole hypothesis (e_{ij} , e_{ji}). A second pair of frames, q and q+f are randomly picked. Either the top or the bottom pair of tangents from e_{ij}, e_{ji} are computed for these two frames. These tangents are denoted by l_i^q and l_j^{q+f}. The epipolar line homography H_{ij}, is determined from these three pair of corresponding lines. Thus (f,e_{ij},e_{ji},H_{ij}) represents the model hypothesis.

This model is now verified for consistency. The two outer tangents from the epipole to silhouettes in the whole

sequence are computed for each view separately. These form two tangent pencils passing through each epipole. Every tangent in the first pencil is transferred through H_{ij} to the second view and the reprojection error of the transferred line from the point of tangency of the original tangent is computed. Tangents that exceed a reprojection error threshold (we choose 8 pixels) are outliers. We throw away our hypothesis if the outlier count exceeds a certain percentage of the expected inlier count. Hence we abort early for most incorrect hypotheses. The actual choice of outlier percentage threshold results in a trade-off between accuracy and speed. All the probable solutions obtained are ranked using a more strict inlier count, maintained using a lower threshold (we choose this to be 1.5 pixels).

Algorithm 1 ComputeSyncAndFMatrix (S_i, S_j, s_o, s_r)
Input: S_i,S_j : video sequences, s_o+/-s_r : sync search range.
Output:F_{ij} & f,v : estimated frame offset and its variance.

{**Phase I: Compute Approx. Sync Offset**}
$K_i \leftarrow$ buildStationaryKeyframeSet(S_i)
execute Step A with K_i, S_j & 33% outlier-threshold.
refine search interval after every 40 offset samples.
if size(interval)<=20, set outlier-threshold to 10%.
$f \leftarrow$ median of the last 40 offset samples.
{**Phase II: Compute Sync Accurately**}
set search interval to $(f-5, f+5)$.
$K_i^E \leftarrow$ buildExhaustiveKeyframeSet(S_i)
execute Step A using K_i^E, S_j sufficient # of times.
f,$v \leftarrow$mean & variance of offsets of probable solns.
{**Phase III: Compute F at sync. offset 'f'**}
execute Step A using K_i^E, S_j sufficient # of times.
pick best soln. $M = e_{ij}$,e_{ji},H_{ij}.
$F \leftarrow$ computeF(e_{ij},e_{ji},H_{ij})
repeat nonlinear-min(F) until #inliers is stable.
return (f**,** v**,** F**)**

Step A: Make Hypothesis and Verify
randomly pick offset $f \epsilon$ (s_o-s_r,s_o+s_r)
randomly pick keyframe r from the first sequence.
top_i^r, bot_i^r, top_j^{r+f}, $bot_j^{r+f} \leftarrow$ pickTangents(S_i,S_j)
e_{ij}, $e_{ji} \leftarrow$ findEpipoles(top_i^r, bot_i^r, top_j^{r+f}, bot_j^{r+f})
randomly pick keyframe $q \epsilon K_i$.
l_i^q, $l_j^{q+f} \leftarrow$ tangentsFromEpipoles(e_{ij}, e_{ji}, q, q+f)
$H_{ij} \leftarrow$ epipLineH(top_i^r,top_j^{r+f},bot_i^r,bot_j^{r+f},l_i^q,l_j^{q+f})
if (($res \leftarrow$ evalHypothesis(e_{ij},e_{ji},H_{ij})! =0)
then record solution (f,e_{ij},e_{ji},H_{ij},res).

3.1 Keyframe Selection and Coarse to Fine Search for Synchronization Offset

In typical sequences, the frontier points and epipolar tangents remain stationary over long periods. Such static frames are redundant and representative keyframes must be chosen to make the algorithm faster. The frames in the middle of static subsequences are special, since they would allow a search over a wider interval of temporal offsets and provide a rough alignment. As directly searching a wide interval would require too many hypotheses, we adopt a coarse-to-fine strategy for this search. We start by coarsely sampling a large interval and verifying the hypotheses using a high outlier percentage threshold of 33%. For every 40 promising hypotheses, a 99% confidence interval for the sample mean is computed. This becomes the new search interval. Once an interval size is less than 20 frames, further search is done using a lower, (hence more strict) outlier percentage of 10%. The median of the next 40 samples roughly estimates the peak of the offset distribution.

A list of key-frames from the middle of static sequences are computed during pre-processing as follows. A pencil of tangents are computed to the silhouettes in every video frame from hypothetical epipoles (at the 4 corners of the image). The angular speed of each of these tangents are also computed within a window of +/-5 frames. Each of these frames are inserted into a high-resolution angular bin of size 0.2 degrees, sorted by the combined angular speed of the two tangents. Each of the angular bins store the frame numbers in a priority queue. The keyframe list is then constructed by choosing frames with low angular speed (The choice for cutoff angular speed and window size depends on the data). For an angular speed of 0 and a window of 10, we ended up with about 90-150 out of 7500 frames(4 mins. at 30Hz). Phase I of Algorithm 1 uses only these keyframes from the first sequence. Once the offset distribution's median is roughly determined, the exact offset and its variance are computed in Phase II, within an interval of +/-5 frames of the median, using a more exhaustive set of keyframes. Using the synchronized sequences, a robust value of F_{ij} can be computed in Phase III. The exhaustive list of keyframes for Phase II and III respectively, are built by ignoring angular speeds of tangents and ensuring at least one frame from every occupied bin is chosen.

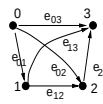
4. Camera Network Synchronization

A camera network is represented as a directed graph $G(V,E)$. V is the set of N cameras each with an offset x_i and E, the set of edges between them. An edge, $e_{ij} \epsilon E$ consists of a sync. offset estimate t_{ij}, alongwith a standard deviation σ_{ij} between cameras i and j. A consistently synchronized camera network should satisfy ($\sum_{e \epsilon C} e = 0$) $\forall$ cycles $C \epsilon G$. Our method in general will not produce a consistent graph. Each edge in the graph contributes a constraint: $t_{ij} = x_i - x_j$. Stacking all the eqns. produces a $|E| \times N$ system of equations T=AX, (T: vector consisting of all t_{ij}'s, X: vector consisting of x_i for

$i = 1 \ldots N$). A Maximum Likelihood Estimate of the N offsets is obtained by solving for X using Weighted Linear Least Squares (each eqn. is weighted by σ_{ij}). These offsets (fixing the first offset at zero) are optimal provided no outliers are present. For every edge $e \in E$, we check ($\sum_{s \in t} s$), $\forall$ triangles t in G containing e. An outlier edge would have only significantly non-zero triangles and could be easily detected and removed. This method will produce very robust estimates for complete graphs. However a fully connected graph with at least N-1 edges is still sufficient to synchronize the whole network upto a certain level of accuracy.

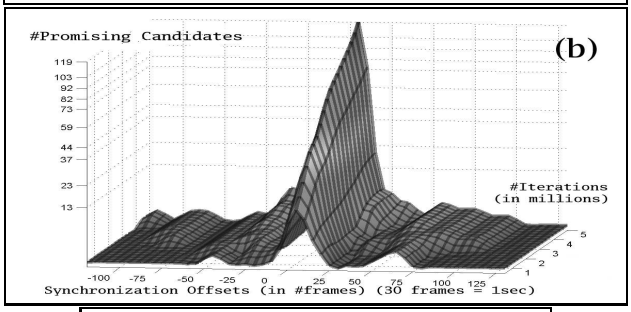
5. Experimental Results

(a)



4-view network graph

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
e_{13}	[-5,5]	0.54	0.72	0.61	0.47
e_{23}	[-6,4]	1.20	1.27	1.09	0.75



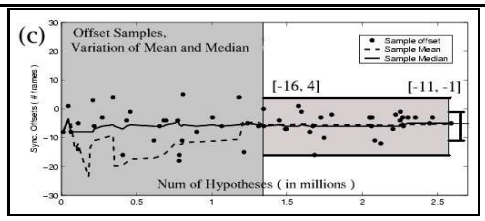


Figure 3. (a) Results of camera n/w synchronization. (b) Typical sync. offset distribution. (c) Sample offset distribution for Phase I.

We applied our techniques to a 4-view video dataset that was about 4 mins. long, captured at 30 fps. We synchronized all six view-pairs, starting each search in a 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. Fig. 3(a) tabulates for each view-pair, the +/-5 $interval$ computed from Ph. I, the estimates (t_{ij},σ_{ij}) from Ph. II, the optimal consistent offset $\bar{t}_{ij}$, and the ground truth $\hat{t}_{ij}$. Ph. I typically required 1.3-2.9 million hypotheses, and 60-120 seconds on a 3 GHz Pentium IV PC with 1 GB RAM. The approx. epipoles computed in Ph. I were used to bias epipole sampling to make Ph. II and III faster.

For view-pair 1 & 2, Fig. 3(b) shows the offset distribution within +/-125 frames of the true offset for 5 million hypotheses. 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. 3(c) shows a typical distribution of offsets in Ph. I and shows the converging search intervals. The results of camera calibration from the synchronized sequences, and visual-hull reconstruction using the computed calibration are shown in Fig. 1.

6. Summary and Conclusions

We have presented a complete method to determine the calibration and synchronization of a network of cameras from a set of unsynchronized silhouettes sequences. At the core of the approach is a RANSAC-based 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 extensions of our approach to asynchronous video streams and wide-area active camera networks.

References

[1] Anonymous. Camera network calibration from dynamic silhouettes. *submitted to CVPR*, 2004.

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

[3] Y. Caspi, D. Simakov, and M. Irani. Feature based sequence to sequence matching. In *Vision and Modelling of Dynamis Scenes workshop, with ECCV*, 2002.

[4] M. Fischler and R. Bolles. A ransac-based approach to model fi tting and its application to fi nding cylinders in range data. In *IJCAI81*, pages 637–643, 1981.

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

[6] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, 1994.

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

[8] K. Wong and R. Cipolla. Structure and motion from silhouettes. In *ICCV01*, pages II: 217–222, 2001.

[9] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, pages 666–673, 1999.

[10] C. Zhou and H. Tao. Dynamic depth recovery from unsynchronized video streams. In *CVPR*, pages 351–358, 2003.