Visual-Hull Reconstruction from Uncalibrated and Unsynchronized Video Streams

Sudipta N. Sinha and Marc Pollefeys
{ssinha,marc}@cs.unc.edu
Department of Computer Science
University of North Carolina at Chapel Hill, USA.

Abstract

We present an approach for automatic reconstruction of a dynamic event using multiple video cameras recording from different viewpoints. Those cameras do not need to be calibrated or even synchronized. Our approach recovers all the necessary information by analyzing the motion of the silhouettes in the multiple video streams. The first step consists of computing the calibration and synchronization for pairs of cameras. We compute the temporal offset and epipolar geometry using an efficient RANSAC-based algorithm to search for the epipoles as well as for robustness. In the next stage the calibration and synchronization for the complete camera network is recovered and then refined through maximum likelihood estimation. Finally, a visual-hull algorithm is used to the recover the dynamic shape of the observed object. For unsynchronized video streams silhouettes are interpolated to deal with subframe temporal offsets. We demonstrate the validity of our approach by obtaining the calibration, synchronization and 3D reconstruction of a moving person from a set of 4 minute videos recorded from 4 widely separated video cameras.

1. Introduction

In surveillance camera networks, live video of a dynamic scene is often captured from multiple views. We aim to automatically recover the 3D reconstruction of the dynamic event, as well as the calibration and synchronization, using only the input videos. Different pairs of archived video sequences may have a time-shift between them 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 [3, 4, 14] as visual-hulls can now be reconstructed from uncalibrated and unsynchronized video of moving objects.

Different existing structure-from-motion approaches using silhouettes [21, 20, 22] either require good initialization or only work 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 [23] but this requires physical access to the observed space. This is often impractical and costly for surveillance applications and could be impossible for remote camera networks or networks deployed in hazardous environments. Our method can calibrate or recalibrate such cameras remotely and also handle wide-baseline camera pairs, arbitrary camera configurations and a lack of photometric calibration.

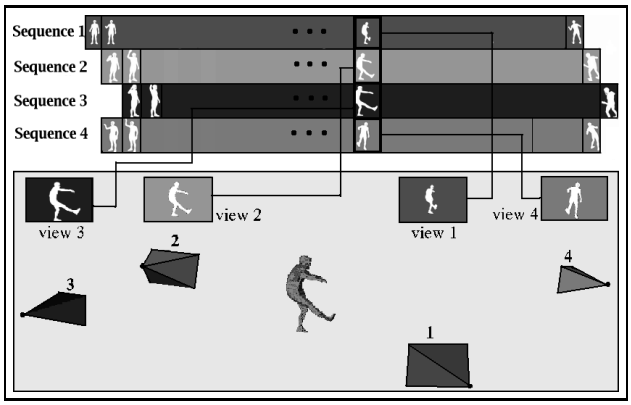


Figure 1: Synchronization, calibration and 3D visual-hull reconstruction from 4 video streams.

At the core of our approach is a robust RANSAC [5] 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 [21, 13, 2] 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 [13, 2]. Here we take advantage of the fact that video sequences of dynamic objects will contain many different silhouettes, yielding many constraints that must be satis-

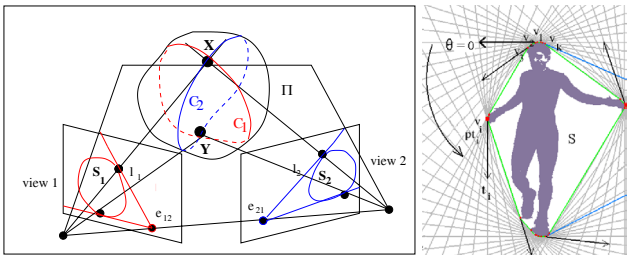


Figure 2: (a)Frontier Points and Epipolar Tangents.(b) The Tangent Envelope.

fied. We use RANSAC [5] not only to remove outliers in silhouette data but also to 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 pairs of unsynchronized video sequences acquired at the same frame rate. A method to synchronize the whole camera network is then presented. Finally, the deforming shape is reconstructed using a shape-from-silhouette approach taking subframe temporal offsets into account through interpolation.

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. Section 6 described our reconstruction approach. Experimental results are presented in different sections of the paper and we conclude with scope for future work in Sec. 7.

2. Background and Previous Work

Our algorithm exploits the constraints arising from the correspondence of frontier points and epipolar tangents [21, 13]. Frontier points on an objects' surface are 3D points which project to points on the silhouette in the 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 correspond. Morever, the image of the frontier points corresponding to the outer-most epipolar tangents [21] must lie on the convex hull of the silhouette. The silhouettes are stored in a compact data structure called the tangent envelope, [16] (see Fig. 2(b)). We only need of the order of 500 bytes per frame.

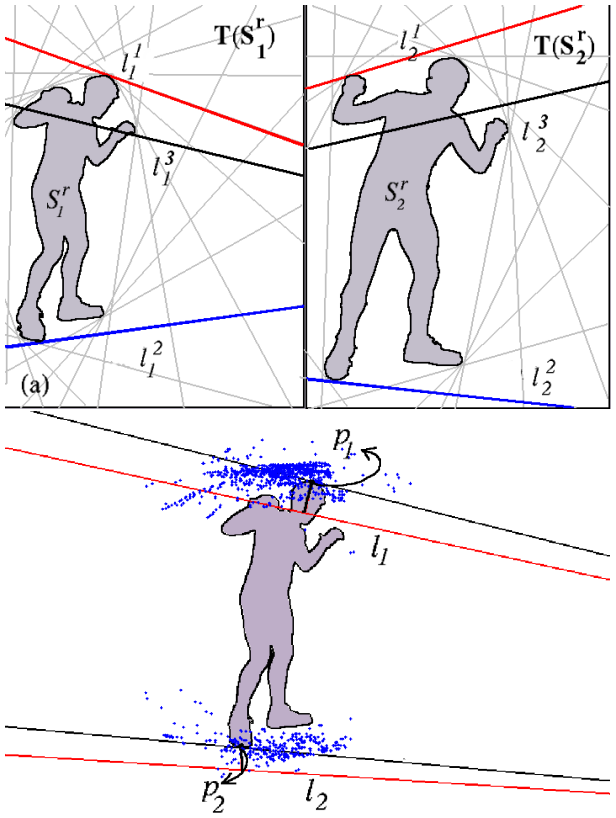


Figure 3: (a) The 4D hypothesis of the epipoles (not in picture). (b) All frontier points for a specific hypothesis and a pair of transferred epipolar lines l_1, l_2.

Video of dynamic objects contain many different silhouettes, yielding many constraints that are satisfied by the true epipolar geometry. Unlike other algorithms, e.g. [6], 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 for many silhouettes. Only using the outermost epipolar tangents allows us to be far more efficient because the data structures are simpler and there are no self-occlusions. 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

Computing the epipolar geometry from silhouettes is not as simple as computing it from points. The reason is that having two corresponding silhouettes doesn't immediately yield usable equations. We first need to know where the frontier points are and this is dependent on the epipolar geometry. This is thus a typical "chicken and egg" problem.

However, this is not as bad as it seems. We do not need the full epipolar geometry. The location of the epipoles (4 out of 7 parameters) is sufficient to determine the epipolar tangents and the frontier points. Our approach will thus consist of randomly generating epipole hypotheses and then verify that the bundle of epipolar tangents to all the silhouettes are consistent. One of the key elements to the success of our algorithm is to have a very efficient data representation and to generate a proper distribution of epipoles in our sampling process. This approach is explained more in detail in the remainder of this section.

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 from them. The pencil of epipolar lines in each view centered on the epipoles, is considered as a $1D$ projective space [7] [Ch.8, p.227]. The epipolar line homography between two such $1D$ projective spaces can be represented by a $2D$ homography to be applied to the $2D$ representation of the lines. 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 uniquely 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. 3(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}). 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 3(a)). H_{ij} is computed from $(l_i^k \leftrightarrow l_j^k; k = 1...3)$[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. 3(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 distance between the tangency point and the transferred epipolar line (see Fig. 3(b)) 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. An outline of the algorithm is given in Algorithm 1.

Results. This approach works well in practice and has been demonstrated on multiple datasets recorded by ourself and by others [16]. Here we show some results obtained from the 4-view dataset that is used throughout this paper. In Figure 4 the computed epipolar geometry F_{14}, F_{24} and F_{34} are shown. The black epipolar lines correspond to the initial epipolar geometry computed as discussed in this section, the colored epipolar lines correspond to the epipolar geometry once it is made consistent over a triplet of cameras[3]. One can notice the significant improvement for pair 2-4 once three-view consistency is enforced. The result for pair 3-4 is less accurate. This is due to clipping of the silhouette at the feet in most of the frames, therefore only yielding a small number of extremal frontier points at the

[1] We use $\sigma = 60$ in our experiments. In case silhouettes are clipped in this frame, the second pair of directions is 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] Note that the final epipolar geometry is refined even further through bundle adjustment, see next section

Algorithm 1 Outline of our RANSAC algorithm

```
do // start RANSAC loop

  // generate hypothesis
  pick 2 corresponding frames
  pick 2 random tangents in each
  compute hypothesized epipoles
  pick 1 more tangent pair in new frames
  compute homography

  // verify hypothesis
  for all tangents (as long as promising)
    compute symmetric transfer error
    update inlier count
  end

  update best hypothesis

until hypothesis good enough

refine epipolar geometry
```

bottom of the image. The result is still sufficient to successfully initialize the bundle adjustment of the next section.

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 [9] 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. 5(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) [7], choosing P_1 and P_2 as follows :

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

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 [9]. 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) initializes 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 [12] to estimate the rectifying transform for each of the projective cameras. We rectify these projective cameras into metric cameras, and use them to initialize a Euclidean bundle adjustment [19]. The Euclidean bundle adjustment step produces the final calibration of the full camera network.

Figure 4: Three computed epipolar geometries. Points in the left column are transferred to epipolar lines in the right column.

Results. Here we present results from full calibration of the 4-view video dataset which was 4 minutes long

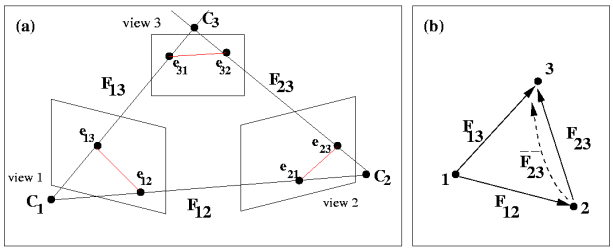


Figure 5: (a) Three non-degenerate views for which we estimate all F matrices. (b) The three-view case. $\overline{F}_{23}$ is the closest approximation of F_{23} we compute. (c)&(d) The induction steps used to resolve larger graphs using our method.

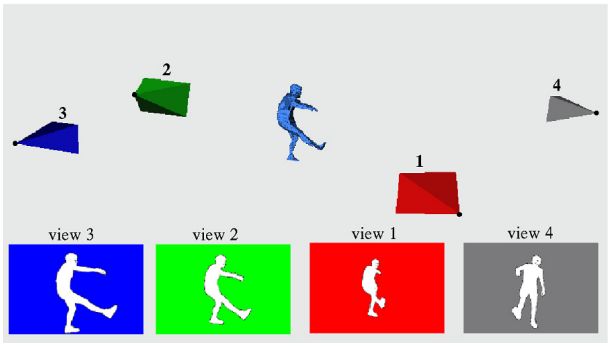


Figure 6: Recovered camera configuration and visual-hull reconstruction of person.

and captured at 30 fps [14] (see Fig. 6). We computed the projective cameras from the fundamental matrices $F_{12}, F_{13}, F_{23}, F_{14}, F_{24}$. On average, we obtained one correct solution, one which converged to a global minimum after non-linear refinement for every 5000 hypothesis[4]. This took approximately 15 seconds of computation time on a 3GHz 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.

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.

[4]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.

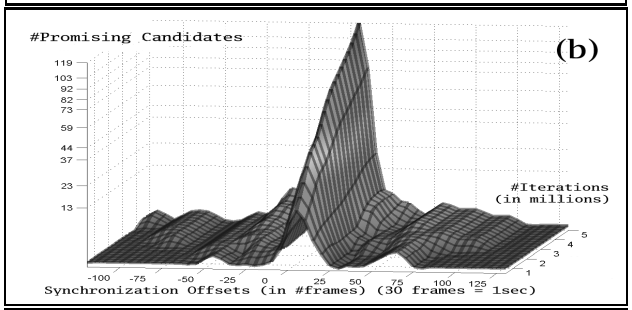
5. Camera Network Synchronization

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 we use a multi-scale approach. Frames containing slow moving and static silhouettes allow to efficiently obtain an approximate synchronization. Therefore, the tangents accumulated in the angular bins during keyframe selection are sorted by angular speed. While selecting the initial keyframes we select the ones with the slowest silhouettes. Once a rough temporal alignment is obtained, a more exhaustive set of keyframes is 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 clock offsets, 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 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.

(a)

e_{ij}	$range$	t_{ij}	σ_{ij}	$\overline{t}_{ij}$	$\hat{t}_{ij}$
e_{12}	*[-13,-3]*	*-8.7*	*0.80*	*-8.50*	*-8.32*
e_{13}	*[-11,-1]*	*-8.1*	*1.96*	*-8.98*	*-8.60*
e_{14}	*[-12,-2]*	*-7.7*	*1.57*	*-7.89*	*-7.85*
e_{23}	*[-5,5]*	*-0.93*	*1.65*	*-0.48*	*-0.28*
e_{24}	*[-5,5]*	*0.54*	*0.72*	*0.61*	*0.47*
e_{34}	*[-6,4]*	*1.20*	*1.27*	*1.09*	*0.75*



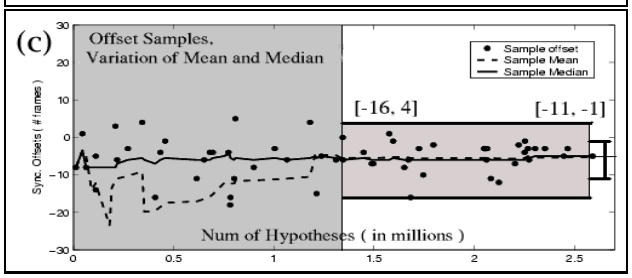


Figure 7: (a) Results of camera network synchronization. (b) Typical sync. offset distribution. (c) Sample offset distribution for rough alignment phase.

Results. We tried our approach on the same 4-view video dataset that was manually synchronized earlier (see Fig. 6). 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 offsets we compute are approximatively within $\frac{1}{100}s$ of the true temporal offsets. Fig. 7(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 $\overline{t}_{ij}$, 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, 2 & 3, Fig. 7(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. 7(c) shows a typical distribution of offsets obtained during a particular run and shows the converging search intervals.

Figure 8: Visual-hull reprojection error (white) induced by subframe temporal offset.

6. Visual Hull Reconstruction from Unsynchronized Video Streams

Once the calibration and synchronization are available, it becomes possible to reconstruct the shape of the observed person using visual hull techniques [3, 10, 8]. However, one remaining difficulty is that the temporal offset between the multiple video streams is in general not an integer number of frames. Given a specific frame from one video stream, the closest frame in other 30Hz video streams could be as far of as $\frac{1}{60}s$. While this might seems small at first, this can be significant for a moving person. This problem is illustrated in Figure 8 where the visual hull was reconstructed from the closest original frames in the sequence. The gray area represents what is inside the visual hull reconstruction and the white area corresponds to the reprojection error (points inside the silhouette carved away from another view). The motion of the arm and the leg that takes place during the small temporal offset between the different frames is sufficient to cause a significant error.

To deal with this problem, we propose to use temporal silhouette interpolation. Given two frames i and $i+1$, we compute the distance $d_i(\mathtt{x})$ and $d_{i+1}(\mathtt{x})$ to the closest point on each silhouette for every pixel $\mathtt{x}$ [15, 11]. For the purpose of interpolation we can limit ourselves to the convex hull of both silhouettes. Then we compute an interpolated silhouette for subframe temporal offset $\Delta \in [0,1]$ as the 0-level set of $S(\mathtt{x}) = (1-\Delta)d_i(\mathtt{x}) - \Delta d_{i+1}(\mathtt{x})$.

Results. In Figure 9 an example is shown. Given three consecutive frames, we generate the frame in the middle of frames 1 and 3 and compare it to frame 2. The result it satisfying.

We use this approach in combination with the subframe temporal offsets computed in the previous section to improve our visual hull reconstruction results. We choose some frames recorded from view 3 as a reference and generate interpolated silhouettes from the other viewpoints that correspond to the appropriate temporal offset. From the table in Fig. 7 we obtain $\Delta_0 = 0.98$ (after taking into account

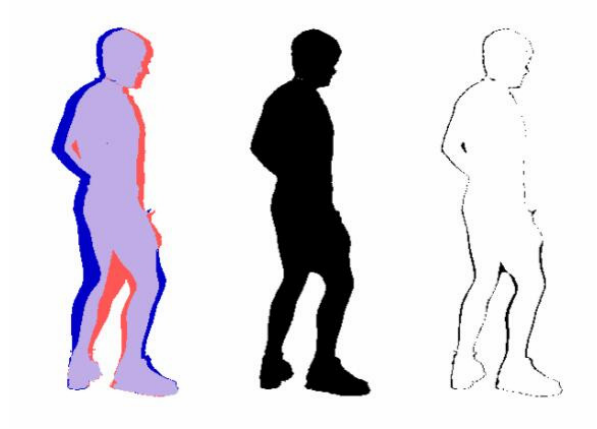


Figure 9: Silhouettes for frame 1 and 3 overlapping (left), interpolated silhouette for frame 2 (middle) and difference between interpolated and original frame 2 (right).

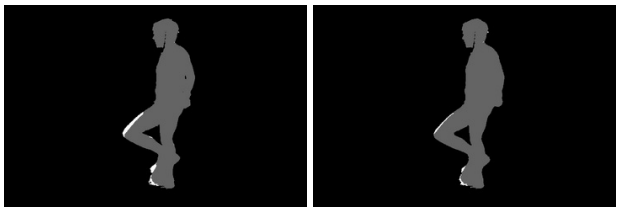


Figure 10: Reprojection error reduced from 2.9% to 1.3% of the pixels contained in the silhouette (834 to 367 pixel). The overall improvement for the 4 corresponding silhouettes was from 1.2% to 0.6% reprojection error.

an integer offset of 8 frames), $\Delta_1 = 0.48$, $\Delta_3 = 0.09$ (integer offset of 1 frame). In Figure 10 and 11 the visual hull reprojection error is shown with an without subframe silhouette interpolation. We show an overal improvement by a factor of two or better of the reprojection error.

7. Conclusions and Future Work

In this paper we have presented a complete approach to determine the 3D visual-hull of a dynamic object from silhouettes extracted from multiple videos recorded using an uncalibrated and unsynchronized network of cameras. The key element of our approach is a robust algorithm that efficiently computes the temporal offset between two video sequences and the corresponding epipolar geometry. The proposed method is robust and accurate and allows calibration of camera networks without the need for acquiring specific calibration data. This can be very useful for applications where sending in technical personnel with calibration targets for calibration or re-calibration is either infeasible or impractical. We have shown that for visual-hull reconstructions from unsycnhronized video streams subframe silhouette interpolation allows to significantly improve the quality of the results.

Figure 11: Reprojection error reduced from 10.5% to 3.4% of the pixels contained in the silhouette (2785 to 932 pixel). The overall improvement for the 4 corresponding silhouettes was from 5.2% to 2.2% reprojection error.

Further work is needed to develop a more general silhouette interpolation scheme that can deal with faster visual motion and/or lower frame rates and with some specific topological degeneracies of our approach. We intend to explore the use of approaches such as [1]. Eventually, we would also like to be able to deal with asynchronous image streams which do not have a fixed frame rate.

To record events in large environments we are exploring the possibility to extend this work to networks of active pan-tilt-zoom cameras [18]. In this case one has to solve significant additional challenges, e.g. background segmentation becomes harder, the observed events need to be actively tracked and calibration needs to be maintained. However, such a system would offer a far greater flexibility than exisisting systems with fixed cameras.

Acknowledgements

We would like to thank Peter Sand [14] for providing us the 4-view dataset from MIT. The partial support of the NSF Career award IIS 0237533 is gratefully acknowledged.

References

[1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH*, 2000.

[2] K. Astrom, R. Cipolla, and P. Giblin. Generalised epipolar constraints. In *ECCV*, pages II:97–108, 1996.

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

[4] 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*, pages II: 375–382, 2003.

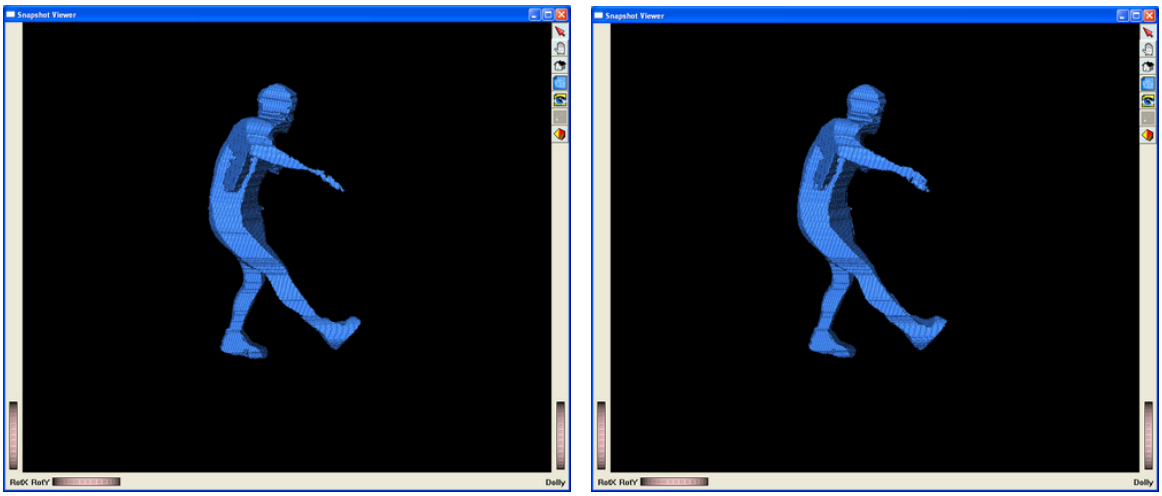


Figure 12: 3D visual hull reconstruction from silhouettes extracted from original video frames (left) and from interpolated video frames (right).

[5] 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*, pages 637–643, 1981.

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

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

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

[9] N. Levi and M. Werman. The viewing graph. In *CVPR03*, pages I: 518–522, 2003.

[10] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *Siggraph*, pages 369–374, 2000.

[11] S. Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, 2003.

[12] M. Pollefeys, R. Koch, and L.J. Van Gool. Self calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *IJCV*, 32(1):7–25, August 1999.

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

[14] P. Sand, L. McMillan, and J. Popovic. Continuous capture of skin deformation. In *Siggraph*, pages 578–586, 2003.

[15] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, volume 94, pages 1591–1595, 1996.

[16] S.N. Sinha and M. Pollefeys. Camera network calibration from dynamic silhouettes. In *CVPR*, 2004.

[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.

[19] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

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

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

[22] A.J. Yezzi and S. Soatto. Structure from motion for scenes without features. In *CVPR*, pages I: 525–532, 2003.

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