

Efficiently Determining Silhouette Consistency

Li Yi, David W. Jacobs

Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742

{liy, djacobs}@umiacs.umd.edu

Abstract

Volume intersection is a frequently used technique to solve the Shape-From-Silhouette problem, which constructs a 3D object estimate from a set of silhouettes taken with calibrated cameras. It is natural to develop an efficient algorithm to determine the consistency of a set of silhouettes before performing time-consuming reconstruction, so that inaccurate silhouettes can be omitted. In this paper we first present a fast algorithm to determine the consistency of three silhouettes from known (but arbitrary) viewing directions, assuming the projection is scaled orthographic. The temporal complexity of the algorithm is linear in the number of points of the silhouette boundaries. We further prove that a set of more than three convex silhouettes are consistent if and only if any three of them are consistent. Another possible application of our approach is to determine the miscalibrated cameras in a large camera system. A consistent subset of cameras can be determined on the fly and miscalibrated cameras can also be recalibrated at a coarse scale. Real and synthesized data are used to demonstrate our results.

1. Introduction

Silhouettes are used widely in computer vision and graphics. Background subtraction generally outputs silhouettes [19]. Gait recognition, an area in biometrics, often uses silhouettes as input [15]. Silhouettes are also used in 3D reconstruction [5, 6]. Volume intersection [16] is performed by back-projecting the silhouettes taken from calibrated cameras. Accurate reconstruction is desirable but many factors make this difficult. In particular, information from some cameras used in reconstruction may be unreliable, either because background subtraction can be inaccurate, or a camera may become uncalibrated. Therefore, it would be useful to have a fast algorithm that can determine whether the silhouettes found using a set of cameras are really consistent with a single, 3D object. We develop such an algorithm, and show how to use it to determine which, if

any of a set of silhouettes is unreliable.

Given a set of silhouettes, are they consistent with a single object? This is the main question addressed in our paper. We provide an effective answer when the projection is scaled orthographic with unknown depth. We begin our journey by considering the consistency of three silhouettes with simple cross-sections (which we will define) from arbitrary but known viewpoints. We develop a verification algorithm with complexity $O(n)$ where n is the number of points on the silhouette boundaries. For the case of convex, 3D objects, we generalize this result to a method for determining the consistency of more than three silhouettes. In this case, we prove that m silhouettes are consistent if and only if all triples pass our test for consistency. Therefore, the answer to the consistency problem immediately follows after we verify every triple of silhouettes, and the overall complexity is $O(m^3 \times n)$. All these results can be applied to silhouettes of general objects, providing necessary conditions on their consistency.

Silhouette consistency verification can be used in various practical scenarios. For example, we show that we can use our test to efficiently determine which silhouettes are causing problems, because they are part of many inconsistent triples. Then we can discard inconsistent silhouettes prior to reconstruction. Camera re-calibration [22] is another possible application. It is good to calibrate only few uncalibrated cameras in a large camera system if the system becomes out of calibration. Robotics is also a possible domain. If an autonomous robot has viewed a landmark from a few directions, our results can be used to allow it to determine whether it might be looking at the same landmark when it moves to another position. Therefore, our problem has both theoretical and practical implications.

There are alternative approaches to solving the consistency problem. For example, we can recalibrate the whole camera system frequently, to avoid inconsistent silhouettes, but this can be time consuming and does not address the problem of determining when the background subtraction results are bad. Alternatively, we can perform volume intersection, project the 3D estimate to the image planes, and

compare new silhouettes with the inputs. However, while this can signal inconsistency, it can be slow and does not necessarily tell us which silhouettes have caused the inconsistency.

We organize this paper as follows. We provide a brief review of topics related to the silhouette consistency problem in Section 2. Section 3 presents our fast verification algorithm for three silhouettes. Section 4 proves a theorem about the consistency of a set of convex silhouettes. Experiments are shown in Section 5 to demonstrate how to use our algorithm in practical situations. This paper ends with a short discussion.

2. Related Work

Much prior work uses silhouettes in different settings with different assumptions. One approach is to reconstruct 3D using only a single silhouette. The type of 3D object, such as algebraic surfaces [11], is often given so that 3D properties can be inferred. Alternatively, a large number of images from known viewpoints may be collected allowing fewer assumptions about the object [8, 21], perhaps using background subtraction [19, 10] to obtain the silhouettes. Volume intersection [16] is then used for reconstructing the 3D visual hull [3, 17]. Viewpoints of calibrated cameras can either be fixed [20], or vary continuously when objects are placed on a turntable [5, 6]. Frontier points [7] are introduced as epipolar constraints [2]. They are formed by two viewing rays intersecting in the tangent plane of the surface, which can be used as true stereo correspondences [13]. Other cross-sections contain additional information. Therefore, we use cross-sections to analyze silhouette consistency in this paper.

The problem of determining the consistency of a set of silhouettes taken from unknown viewpoints has been previously addressed. [14] considers the case of three silhouettes produced by scaled orthographic projection, with viewing directions that are coplanar. They show that the consistency of silhouettes with simple cross-sections can be determined by solving a linear program. [4] presents the case in which more than three silhouettes are considered. They derive a set of inequalities for determining the consistency of a set of silhouettes, with coplanar viewing directions and orthographic projection. Most recently, [12] presents an algorithm to determine the silhouette consistency from uncalibrated motion sequences. They consider the problem as the generalization of an epipolar tangency constraint and demonstrate the system using a turntable. We address the problem when viewing directions are arbitrary but given, and assume scaled orthographic projection with unknown depth.

Our derivation also makes use of Helly's theorem on convex sets. Helly's theorem [9] states that the intersection of a collection of convex sets in d dimensional space R^d is

non-empty if and only if every $d + 1$ of them have a non-empty intersection. Our use of Helly's theorem is somewhat related to, and inspired by, its use in solving line transversals problems in computational geometry [1]. It motivates us to explore an efficient algorithm for determining the consistency of more than three convex silhouettes because the computational cost is greatly reduced. We work directly in the image domain instead of the 3D voxel domain. This means we can be very precise and still very efficient, which is the key difference between our ideas and the visual-hull based approaches.

3. The Consistency of Three Arbitrary Silhouettes

The consistency of two silhouettes is discussed in [4], and provides limited information. Therefore, we study the consistency problem of three silhouettes taken from arbitrary but known directions in this section. It is natural to begin by examining the case in which viewing directions are orthogonal. Then we show that we can transform space so that arbitrary viewing directions are handled using the case in which views are orthogonal.

Without loss of generality, the rotation matrices from World Coordinate System (*WCS*) to each Camera Coordinate System (*CCS*) are shown in Fig.1, and we denote as S_X , S_Y , and S_Z silhouettes viewed from the orthogonal basis X , Y , and Z^1 , respectively (Fig.1).

$$\begin{aligned}
 & \text{CCS}_z \quad R_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 & \text{CCS}_y \quad R_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\
 & \text{CCS}_x \quad R_x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

Figure 1. The rotation matrices from world coordinate system *WCS* to camera coordinate systems CCS_x , CCS_y , and CCS_z .

We assume a horizontal cross section of S_X and S_Y has a single line segment². Generally, it might have multiple ones. This simple contour assumption of the first two silhouettes implies that if we apply our algorithm to general silhouettes, we check necessary, but not sufficient conditions for consistency. In this case, we use enclosing lines of multiple line segments as the input.

¹It means the viewing directions are opposite to the directions of the X , Y , and Z axes.

²This does not necessarily mean the silhouette must be convex because it only requires the horizontal cross-sections to be single lines.

In this section, we first briefly show two observations. Our algorithm is then presented and the transformation procedure follows. We assume scaled orthographic projection with unknown depth in the discussion. The algorithm is linear in the number of points of the silhouette boundaries. Variables discussed in this section are continuous.

3.1. Observations

We observe two important properties of points in a silhouette.

- Observation 1: Any point p in the silhouette S_p corresponds to a line L_p in 3D, which intersects the image plane in p (Fig.2). Suppose O is the reconstructed 3D object using other silhouettes, then L_p must intersect O if the silhouette S_p is consistent with others.

This observation holds for different assumptions about the projection. For example, L_p is parallel to the viewing direction if the projection is orthographic, or goes through the camera center if the projection is perspective. Therefore, part of the consistency problem is to determine the existence of points in L_p for every possible p .

If the projection is orthographic, we have the following observation:

- Observation 2: With orthographic projection, when the viewing directions are orthogonal to the z direction, silhouettes are consistent if and only if they have the same maximum z value, and the same minimum z value. Any cross-section that is orthogonal to the z direction then produces a line segment in each silhouette (Fig.3 and 4), if they are simple silhouettes. These two line segments have the same z value, and the back-projection of these line segments form an enclosing rectangle of the cross-section of the object that generates them.

Based on these observations, we introduce an efficient algorithm in Section 3.2, and prove the theorem in Section 4.

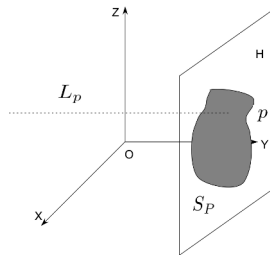


Figure 2. A point p of silhouette S in image plane H corresponds to a 3D line L_p . The illustration uses orthographic projection.

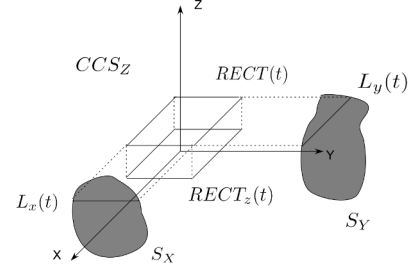


Figure 3. A horizontal cross-section and its orthographic projections. Viewing directions are parallel to the XY plane.

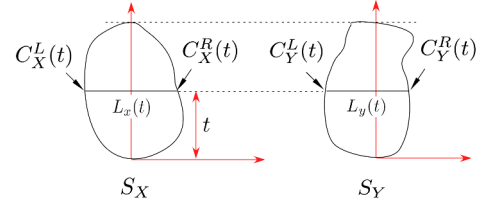


Figure 4. Two consistent silhouettes and two consistent line segments.

3.2. The Consistency of Three Silhouettes from Orthogonal Viewing Direction

3.2.1 Necessary and Sufficient Conditions

Scale is determined by the distance from the camera center to the mass center of the object³. The viewing directions of S_X and S_Y are parallel to the XY plane in WCS . Suppose $L_x(t_x)$ and $L_y(t_y)$ are line segments from S_X and S_Y , respectively, and parallel to the x axis in the image coordinate system (Fig.4). They are parameterized by the distance t to the lowest horizontal points (or line, if the bottom is flat). If they are from the same horizontal cross-section (Observation 2), we have $t_x = t_y = t$. Back-projection of $L_x(t)$ and $L_y(t)$ must form an enclosing rectangle $RECT(t)$ of the cross section in the 3D space (Fig.3). Based on Observation 1, every point in $L_x(t)$ and $L_y(t)$ must correspond to at least one 3D point inside $RECT(t)$. Projecting $RECT(t)$ to the image plane of CCS_Z we have $RECT_z(t)$ and its $EDGE_z(t)$. The rectangles must satisfy Theorem 1.

- Theorem 1: Three silhouettes from orthogonal viewing directions (Fig.1) are consistent if and only if 1) $S_Z \subset \bigcup_t RECT_z(t)$ and 2) $S_Z \cap EDGE_z(t) \neq \emptyset$, $0 \leq t \leq t(H(S_X))$ (Fig.5).

The proof of Theorem 1 is as follows. We only need to verify silhouettes' boundaries because silhouettes are solid. By Observation 1, every point in the boundary of S_Z corresponds to a 3D line l parallel to the Z axis. Any such line l must intersect at least one $RECT(t)$ (the recon-

³We consider that the scale is unknown and it forms part of the consistency test. The images can be further normalized to the same height.

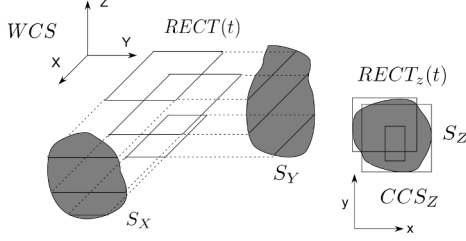


Figure 5. The necessary and sufficient condition of the consistency for three silhouettes taken from orthogonal basis. Only three cross sections are shown for simplicity.

structed 3D object from S_X and S_Y), which is $S_Z \subset \bigcup_t RECT_z(t)$. Similarly, edges of the enclosing rectangles $RECT(t)$ correspond to points of the boundaries of S_X and S_Y . Therefore, edges must be inside the volume formed by the back-projection of S_Z in 3D, which is equivalent to $S_Z \cap EDGE_z(t) \neq \emptyset$ for all possible t . A naive algorithm to verify these constraints would compare every point on the boundary of S_Z to every rectangle, taking $O(n^2)$ time. We now provide a fast, $O(n)$, verification algorithm.

3.2.2 The Verification Algorithm

Four edges of a rectangle are in one-to-one correspondence to four vertices. Each vertex of $RECT_z(t)$ can be represented by two points of the boundary of S_X and S_Y respectively (Fig.3). Since these points have the same y coordinate in the image plane, as described in Observation 1, we then parameterize their x coordinate as $C_X^L(t)$ and $C_X^R(t)$, or $C_Y^L(t)$ and $C_Y^R(t)$ (Fig.4), where L and R denote the left side or the right side of the silhouette. Therefore, we represent the vertices of $RECT_z(t)$ in CCS_z using these functions respectively (Fig.5a). For example, if we use the transformations in Fig. 1, the top right vertex is:

$$V_{TR}(t) = (-C_Y^L(t), C_X^R(t)), 0 \leq t \leq H(S_X). \quad (1)$$

where T and R (and B, L in later paragraphs) in Eq.1 denote top, right, bottom and left. We also define:

$$R_{TR}(t) = (x, y) : x \leq -C_Y^L(t), y \leq C_X^R(t) \quad (2)$$

$$R_{TL}(t) = (x, y) : x \geq -C_Y^R(t), y \leq C_X^R(t) \quad (3)$$

$$R_{BR}(t) = (x, y) : x \leq -C_Y^L(t), y \geq C_X^L(t) \quad (4)$$

$$R_{BL}(t) = (x, y) : x \geq -C_Y^R(t), y \geq C_X^L(t) \quad (5)$$

To verify $S_Z \subset \bigcup_t RECT_z(t)$, we observe that ⁴:

$$RECT_z(t) = R_{TR}(t) \cap R_{TL}(t) \cap R_{BR}(t) \cap R_{BL}(t) \quad (6)$$

⁴There is redundancy in this equation, but the notation is for discussion in later context.

Therefore, if we denote $R_{TR} = \bigcup_t R_{TR}(t)$, $0 \leq t \leq H(S_X)$, etc, we have ⁵:

$$\begin{aligned} & \bigcup_t RECT_z(t) \\ &= \bigcup_t (R_{TR}(t) \cap R_{TL}(t) \cap R_{BR}(t) \cap R_{BL}(t)) \\ &= (\bigcup_t R_{TR}(t)) \cap (\bigcup_t R_{TL}(t)) \cap (\bigcup_t R_{BR}(t)) \cap (\bigcup_t R_{BL}(t)) \\ &= R_{TR} \cap R_{TL} \cap R_{BR} \cap R_{BL} \end{aligned} \quad (7)$$

We show how to compute the boundaries of these regions efficiently. Without loss of generality, we discuss R_{TR} . As shown in Fig.6b, $R_{TR}(t_1) \in R_{TR}(t_2)$ if $V_{TR}(t_1)$ (denoted by intersection of dashed lines) is lower left to $V_{TR}(t_2)$ (denoted by solid lines). Therefore, if we denote $C_{TR}(t)$ as the boundary of R_{TR} , $C_{TR}(t)$ at $x = -C_Y^L(t_1)$ is the maximal value of $C_X^R(t_2)$ for all t_2 such that $-C_Y^L(t_2) \geq -C_Y^L(t_1)$, which is:

$$C_{TR}(t) = \max(C_X^R(t_0)), \forall C_Y^L(t_0) \leq C_Y^L(t) \quad (8)$$

for $t_0, t \leq H(S_X)$. This causal equation can be efficiently solved. Similarly, C_{TR} , C_{BR} , and C_{BL} can be computed so the verification of $S_Z \subset \bigcup_t RECT_z(t)$ can be quickly determined.

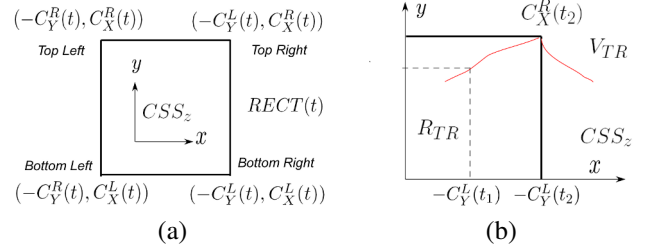


Figure 6. (a) The parameterization of the vertices of $RECT_z(t)$; (b) The boundary of R_{TR} (The red curve is $V_{TR}(t)$). $R_{TR}(t_1) \in R_{TR}(t_2)$ therefore $C_{TR}(t_1) = C_X^R(t_2)$.

We next provide a way to check $S_Z \cap EDGE_z(t) \neq \emptyset$ in the following discussion. Observation 1 tells us that if there is consistency, all edges must intersect at least one point in S_Z . Without loss of generality, we discuss the rightmost edge of $RECT_z(t)$, whose endpoints are $V_{TR}(t)$ and $V_{BR}(t)$ ($V_{BR}(t) < V_{TR}(t)$ for all t). Assume when $t_1 < t < t_2$, $C_Y^L(t)$ is monotonically increasing (Fig.7). Then all the rightmost edges of the enclosing rectangles $RECT_z(t)$ form a region R , which is bounded by four curves $-C_Y^L(t_2)$, $-C_Y^L(t_1)$, $V_{TR}(t)$ and $V_{BR}(t)$, for $t_1 < t < t_2$, in CCS_z . Since every vertical line in R is a rightmost edge of some enclosing rectangle, they must all intersect S_Z . Therefore, the projection of $S_Z \cap R$ to x axis can only be a single line, which is equivalent to:

$$lsNum(Proj_x(S_Z \cap R)) = 1 \quad (9)$$

⁵This equality does not hold in general, but it can be shown that it is an identity in our case.

Discretized, this algorithm is linear in the number of points of the silhouettes. The algorithm uses boundary points of the silhouettes only as input and visits every point only twice. Therefore, it greatly reduces the computational complexity. In the next section we will present a procedure to transform space so that arbitrary viewpoints can be treated using the method described above.

In this section we show that we can transform space so that three arbitrary viewing directions are handled using the case in which views are orthogonal. Intuitively, suppose we have three silhouettes taken from arbitrary viewpoints. We can determine a 3D linear transformation that will transform these viewpoints to the x , y and z axes. The three original silhouettes will be consistent with a 3D object if and only if the linearly transformed silhouettes are consistent with the linearly transformed object. So, to determine consistency of the original silhouettes we need to calculate the silhouettes that will appear in the transformed space, and then apply the algorithm from Section 3.2.

1. The transformation from its camera coordinate system

$$T_1 = [V_1, V_2, V_3]^{-1} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (10)$$

- $$T_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

- #### 4. The Consistency of a Set of Convex Silhouettes

To prove our theorem, we use a specific case of Helly’s theorem in low dimension as follows. Given a set of closed 1D intervals, Helly’s theorem says they must share at least one point iff any two intervals have an intersection. That is:

- The proof is described as follows. Assume we have m silhouettes. Without loss of generality, Observation 1 says any point p of the silhouette S_P in the image plane denotes a line L_p , which is parallel to the viewing direction V_P and intersects the image plane in p . The projection of L_p to other cameras is a line, which intersect other silhouettes if

S_P is consistent with them. The intersection with the other $m - 1$ silhouettes are line segments, which correspond to intervals I_1, \dots, I_{m-1} ($m > 3$) in L_P . Because any three silhouettes are consistent, the intersection of every two intervals I_i, I_j ($i, j = 1 \dots m - 1$) must be non-empty. By Lemma, $\cap_{k=1}^{m-1} I_k \neq \phi$.

This shows that for every point in S_P we can choose a 3D point that will project to that point in S_P , and project to the inside of every other silhouette. The union of these points for all S_P will account for all silhouettes.

5. Experiments

In this section, we present our experiments and demonstrate how our theory can be used for applications. Real and synthesized data are used in the experiments. Image sequences taken from 15 calibrated cameras [20] are used. The resolution is 640 by 480.

We first define the consistency rate, and then two kinds of miscalibration, the change of the viewpoint and the in-plane rotation, are shown. Finally, we present the detection of uncalibrated cameras in a camera system using our algorithm and theory.

5.1. Results for the Verification Algorithm

Real images are taken from synchronized cameras and background subtraction is performed (Fig.8a, 8b, and 8c). Connected component analysis based on simple rules and mathematical morphological operations are used to select the largest components in the image (Fig.8d, 8e, and 8f). To generate synthetic data, we use the camera calibration data to arbitrarily place a 3D object (tetrahedron in our examples) in the space, and then generate the silhouette images using scaled orthographic projection (Fig.8j, 8k, and 8l).

By using the space transformation procedure, we have Fig.8g, 8h, and 8i (real data), and Fig.8m, 8n, and 8p (synthesized data). Because of the discretization error, it is possible that some consistent silhouettes are incorrectly detected as inconsistent ones. Also, it is possible that the silhouettes are still consistent when viewpoints are miscalibrated for some special cases. Therefore, we define the consistency rate, to measure the degree of consistency for each camera. For every camera, we can have $\binom{m-1}{2}$ choices for two other cameras to use as the input of the verification algorithm, assuming there are m cameras. Therefore, we define the consistency rate if k consistent cameras are detected:

$$CR(i) = \frac{k}{\binom{m-1}{2}} \quad (12)$$

for the i^{th} camera. The consistency rate is shown in Fig.9 for synthesized and real data. The average rate for synthesized data is 95%, and for real data it is 75% in our exper-

iments. We apply this to detecting uncalibrated cameras in the next section.

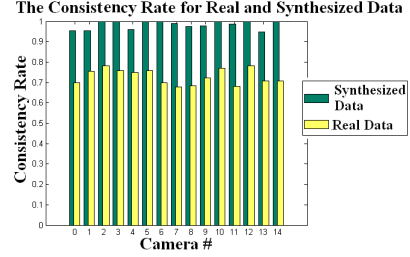


Figure 9. The consistency rate for real and synthesized data.

5.2. Uncalibrated Camera Detection

In this section, we show some experiments to illustrate how our theory and algorithm can be used for detecting uncalibrated cameras in a camera system. Camera calibration for multiple camera systems is a tedious but necessary pre-processing step for most applications. Also, cameras can become uncalibrated due to unexpected change of 1) viewing direction and 2) in-plane rotation. Therefore, it is necessary to efficiently determine the inconsistent cameras.

Viewpoints are obtained in the experiment, and then one or more of them are reasonably perturbed. By computing the convex hull of the silhouettes, or more directly, placing an arbitrary convex object in the scene, our theory says correctly calibrated cameras must be consistent with each other. Therefore, we first show that the consistency rate decreases if we perturb one camera. If more than one camera becomes inconsistent, we show that the uncalibrated cameras can still be detected because their consistency rates decrease much faster than others.

5.2.1 Change of the Viewpoint

Define a camera's coordinate system so that the viewing direction is $[0, 0, 1]$ (z axis). We miscalibrate the camera so that the viewpoint is $[p, q, 1]$, for various $p, q \in [-0.1, 0.1]$ and then calculate the consistency rate. Fig.10 depicts this change in the World Coordinate System. Synthesized data are used in this demonstration. Results are shown for changes of 0.005 as the step size in each direction (Fig.11). The average angle of miscalibration is 4.48° and the maximal one is 8.04° .

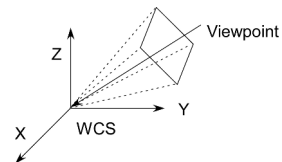


Figure 10. The change of the viewpoint.

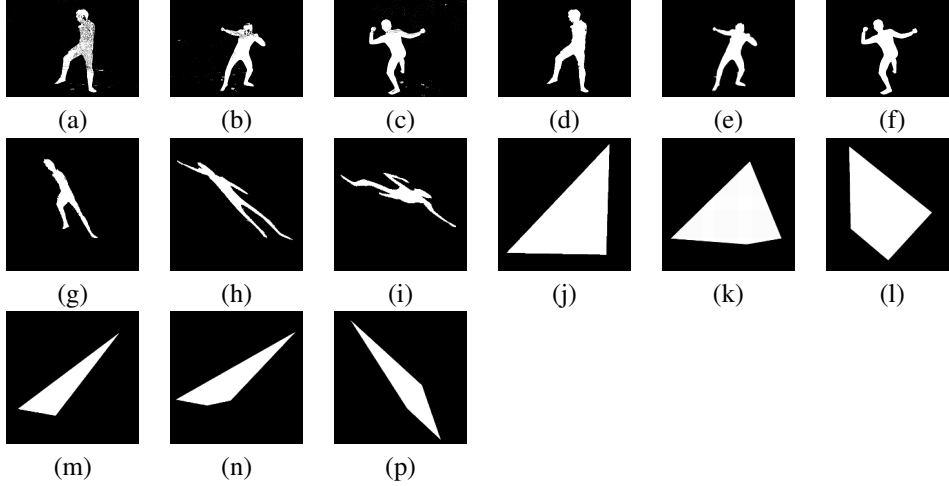


Figure 8. Examples of silhouettes for experiment. (a), (b) and (c): Silhouettes taken from three different viewpoints (Human); (d), (e) and (f): Silhouettes processed using connected component analysis and mathematical morphology; (g), (h), and (i): Real data transformed for orthogonal viewing directions; (j), (k) and (l): Silhouettes of a synthesized 3D object (Tetrahedron); (m), (n) and (p): Synthesized data after transformation.

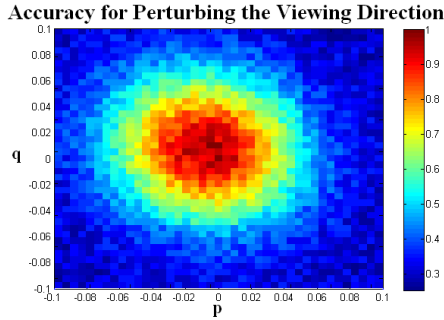


Figure 11. The result for change of viewpoint.

We observe a local maxima in Fig.11. Theoretically, it might not be the global maxima. But practically speaking, if the viewpoint is changed a little bit, the local maxima is probably the correct viewpoint. Therefore, it is possible to reach the local maxima. Therefore, our method can detect the uncalibrated camera and calibrate the camera at a coarse scale⁶ if the scaled orthographic assumption holds.

5.2.2 In-plane Rotation

Another type of miscalibration is in-plane rotation. Instead of changing the viewpoint, the image plane is rotated a small angle about the z axis (in CCS). This section demonstrates the consistency rate when one camera is rotated $[-10^\circ, 10^\circ]$ in the image plane. Real data are used and convex hulls of silhouettes are computed. Because they are perspective projected, the consistency rate is lower than the simulation. None-the-less, we observe the same trend

⁶We do not expect precise calibration with just silhouettes and the precision is determined by the step size used in searching.

(Fig.12).

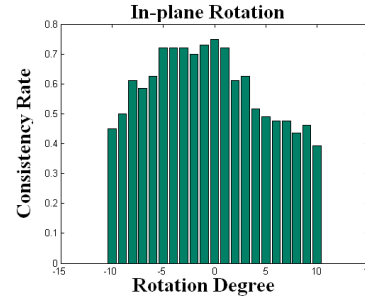


Figure 12. The result for in-plane rotation.

5.2.3 Detecting Uncalibrated Cameras in a Camera System

We have shown how to detect inconsistent cameras in the previous section. Miscalibrated cameras can also be calibrated at a coarse scale. In practice, more than one camera in a camera system can become uncalibrated at the same time. We simulate this scenario by arbitrarily perturbing some calibration data in this section.

We use real and synthesized data in this experiment. At most five cameras (33%) are randomly selected, and we arbitrarily perturb them and compute the consistency rate, using the method in Section 5.2.1 and 5.2.2. For example, Fig.13 shows the consistency rate when camera # 0,1,2,6,10 are randomly changed using synthesized data. The consistency rate of miscalibrated cameras decreases faster than other cameras, and a simple thresholding technique (e.g., [18] in MATLAB) can separate the two groups. We repeat this experiment a few times, and the average result is shown

in Fig.14. In this example, uncalibrated cameras can still be automatically detected even if 1/3 of the cameras are uncalibrated.

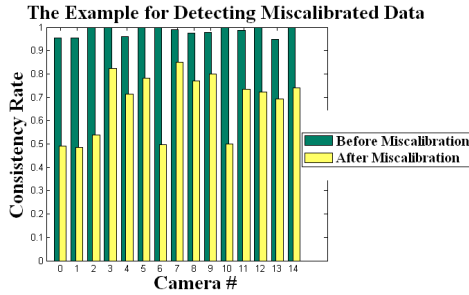


Figure 13. Example for detecting miscalibrated cameras.

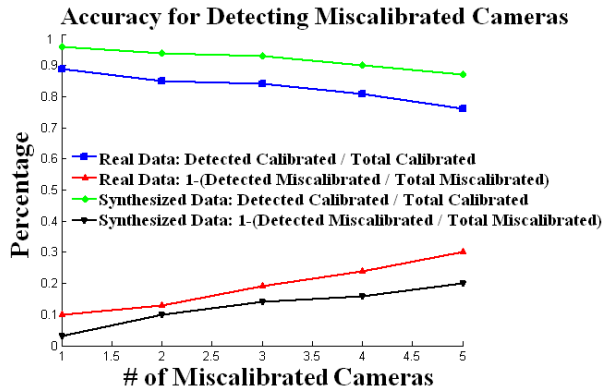


Figure 14. Detecting uncalibrated cameras in a camera system.

6. Conclusion

We explore the silhouette consistency problem in this paper. Assuming the projection is scaled orthographic, we show three silhouettes from known viewing direction can be transformed to three orthogonal ones, and then their consistency can be tested with a fast algorithm. We further show that there is a nice property for convex 3D objects that all silhouettes in the set are consistent if and only if any three of them are consistent. We demonstrate how to detect inconsistent cameras in a camera system by changing viewpoints and rotating image planes. Experiments show that our algorithm can detect miscalibration even when 33% of cameras are miscalibrated.

There are many possible future directions to explore in this topic. For example, detecting consistency with unknown viewing directions is interesting, but multiple solutions might exist based just on the shape of silhouettes. Other future work includes more comprehensive simulations and developing applications based on the silhouette consistency.

References

- [1] N. Amenta. Helly theorems and generalized linear programming. In *SCG '93*, pages 63–72, New York, NY, USA, 1993. ACM Press.
- [2] K. Astrom, R. Cipolla, and P. Giblin. Generalised epipolar constraints. *Int. J. Comput. Vision*, 33(1):51–72, 1999.
- [3] B. Baumgart. Geometric modeling for computer vision. Technical Report AIM-249, Stanford University, 1974. Ph.D. Thesis. Department of Computer Science.
- [4] A. Bottino and A. Laurentini. Introducing a new problem: Shape-from-silhouette when the relative positions of the viewpoints is unknown. *IEEE Trans. PAMI.*, 25(11):1484–1493, 2003.
- [5] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part i: Theory and algorithms. *IJCV*, 62(3):221–247, May 2005.
- [6] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *IJCV*, 63(3):225–245, July 2005.
- [7] R. Cipolla, K. E. Astrom, and P. J. Giblin. Motion from the frontier of curved surfaces. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 269, Washington, DC, USA, 1995. IEEE Computer Society.
- [8] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *IJCV*, 9(2):83–112, 1992.
- [9] J. Eckhoff. Helly, radon, and caratheodory type theorem. In *Handbook of Convex Geometry*, pages 389–448, 1993.
- [10] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, volume 90, pages 1151–1163, July 2002.
- [11] D. A. Forsyth. Recognizing algebraic surfaces from their outlines. *IJCV*, 18(1):21–40, 1996.
- [12] M.-C. H. Francis Schmitt and M.-R. Cipolla. Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):343–349, 2007.
- [13] Y. Furukawa, A. Sethi, J. Ponce, and D. J. Kriegman. Structure and motion from images of smooth textureless objects. In *ECCV (2)*, pages 287–298, 2004.
- [14] D. W. Jacobs, P. N. Belhumeur, and I. Jermyn. Judging whether multiple silhouettes can come from the same object. In *IWVF*, pages 532–541, 2001.
- [15] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. Chowdhury, V. Krüger, and R. Chellappa. Identification of humans using gait. *IEEE Trans. Im. Proc.*, 13(9):1163–1173, 2004.
- [16] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. PAMI.*, 16(2):150–162, 1994.
- [17] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH*, 2001.
- [18] N. Otsu. A threshold selection method from grey-level histograms. *IEEE Trans. System, Man and Cybernetics*, 9(1):62–66, January 1979.
- [19] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 2246–2252, 1999.
- [20] A. Sundaresan and R. Chellappa. Segmentation and probabilistic registration of articulated body models. In *Proc. Int'l Conf. Pattern Recognition*, pages 92–96, 2006.
- [21] R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *IJCV*, 28(1):27–44, 1998.
- [22] T. Ueshiba and F. Tomita. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *ICCV*, pages 966–973, 2003.