Robust Structure and Motion from Outlines of Smooth Curved Surfaces

Yasutaka Furukawa (yfurukaw@uiuc.edu)
Amit Sethi (asethi@uiuc.edu)
Jean Ponce (ponce@cs.uiuc.edu)
Beckman Institute, Univ. of Illinois at Urbana-Champaign, Urbana, IL 61801

David Kriegman (kriegman@cs.ucsd.edu)
Computer Science & Engineering, Univ. of California at San Diego, La Jolla, CA 92093

Abstract: This article addresses the problem of estimating the motion of a camera as it observes the outline (or apparent contour) of a solid bounded by a smooth surface in successive image frames. In this context, the surface points that project onto the outline of an object depend on the viewpoint, and the only true correspondences between two outlines of the same object are the projections of frontier points where the viewing rays intersect in the tangent plane of the surface. In turn, the epipolar geometry is easily estimated once these correspondences have been identified. Given the apparent contours detected in an image sequence, a robust procedure based on RANSAC and a voting strategy is proposed to simultaneously estimate the camera configurations and a consistent set of frontier point projections by enforcing the redundancy of multi-view epipolar geometry. The proposed approach is, in principle, applicable to orthographic, weak-perspective and affine projection models. Experiments with nine real image sequences are presented for the orthographic projection case, including a quantitative comparison with the ground-truth data for the six datasets for which the latter information is available. Sample visual hulls have been computed from all image sequences for qualitative evaluation.

Keywords: Image Processing and Computer Vision, Motion, Shape.

1 Introduction

Structure-from-motion algorithms typically assume that correspondences between *viewpoint-independent* scene features such as surface creases, corners, or markings have been established through tracking or some other mechanism [4, 31]. Several proven techniques for computing a projective, affine, or Euclidean scene representation from matching features while estimating the corresponding projection matrices are now available (see [9, 12, 17] for comprehensive surveys). Various robust estimation techniques have been proposed to handle mismatches (or outliers) [32]. M-Estimators reduce the effects of outliers by assigning weights to samples, which problem is formulated as a weighted least-squares. RANSAC [11] has proven to be a very successful technique for the outlier detection, and many variants [33, 34] have also been developed.

When the scene observed by a moving camera consists of solids bounded by smooth surfaces with little texture and few markings, establishing viewpoint-independent correspondences becomes difficult, as the *apparent contour* (or *outline*) of these objects becomes the dominant image feature (See Figure 1). The apparent contour is the projection of the *rim* (or *contour generator*), a surface curve formed by the points whose tangent plane contains the camera's optical center. The rim changes with viewpoints, and the only true stereo correspondences between two different outlines of the same solid are the projections of a finite number of *frontier* points [14]; they are intersections of the contour generators on the surface where the corresponding viewing rays intersect in the tangent plane of the surface (See Figure 2).

This article proposes a robust procedure based on RANSAC for simultaneously estimating the camera configurations and a consistent subset of the *frontier* points formed by all binocular correspondences between the apparent contours found in an image sequence. It uses the *signature* representation of the dual of image outlines originally proposed in the context of object recognition [28] to identify promising correspondences. Briefly, the *signature* of a

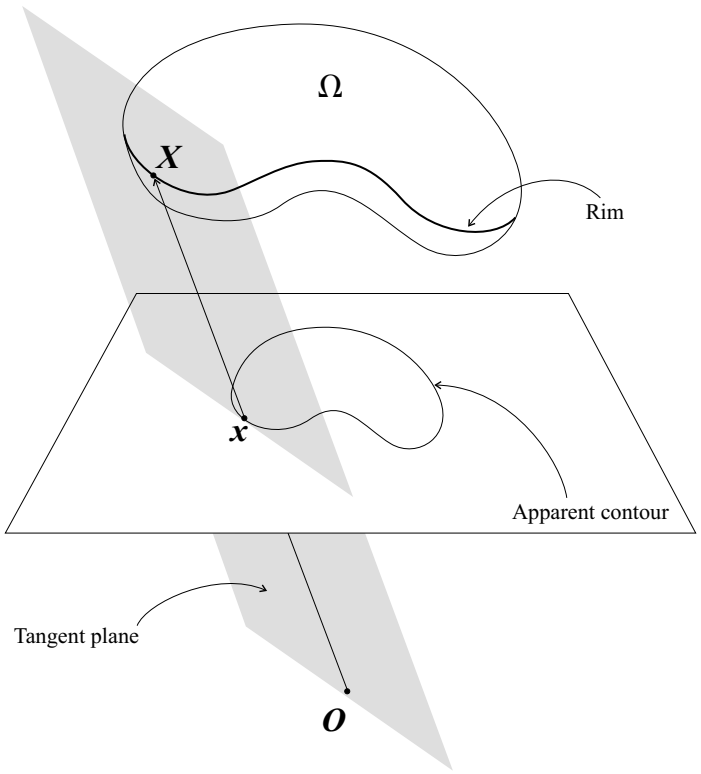


Figure 1: The *apparent contour* (or *outline*) and the *rim* (or *contour generator*) of a solid bounded by a smooth surface Ω. The image point x on the apparent contour is the projection of the surface point X on the rim. The viewing ray passing through the optical center O of the camera and the points x and X lies in the tangent plane to Ω in X. The rays associated with the entire outline form a *viewing cone* that grazes the surface along the contour generator.

planar curve γ is defined by mapping every direction in the plane onto the tuple formed by the distances between successive lines locally tangent to γ that are perpendicular to the direction. Then, the redundancy of multi-view epipolar geometry [24] is exploited to retain the consistent ones. The visual hull [3, 21, 23] of the observed solid is finally reconstructed from the recovered viewpoints. The proposed approach is applicable to orthographic, weak-perspective and affine projection models. We focus here on the orthographic projection case; experiments with nine real image sequences are presented, including quantitative evaluation of recovered motion for six of the datasets for which ground truth is available. As a "proof-of-concept" experiment, an example is also presented for the weak-perspective projection model. Qualitative results in the form of visual hulls are computed from all image sequences.

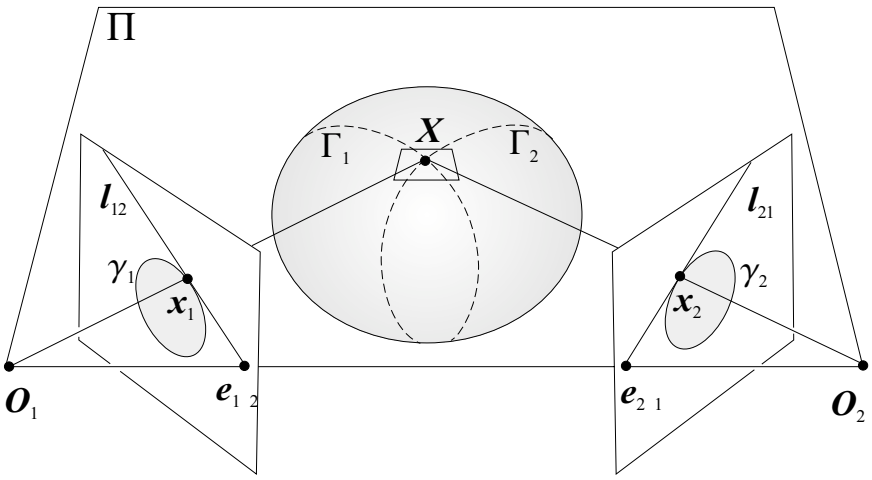


Figure 2: The point X where the rims Γ_1 and Γ_2 cross each other is a *frontier* point. Its two projections x_1 and x_2 are in binocular correspondence, and the corresponding viewing rays lie in the tangent plane Π at X. The tangent plane intersects the two image planes along matching epipolar lines l_{12} and l_{21}, and these are (locally) tangent to the apparent contours γ_1 and γ_2 at x_1 and x_2.

A preliminary version of this article appeared in [13].

2 Background and Approach

2.1 Background

As first demonstrated by Giblin and Weiss [16], it is possible to estimate the *local* second-order shape of a smooth surface along its rim when the trajectory of the camera is known (see [5, 7, 30] for related work). It is also possible to recover the *global* shape of a surface from the outlines found in a set of images when the corresponding camera configurations are known (see, for example, [3, 5, 30]). The earliest—and perhaps the most powerful—approach to the latter problem was proposed by Baumgart in the mid-seventies [3], and it does not rely on the availability of continuous image sequences. Instead, an outer approximation of the observed solid—dubbed the *visual hull* in [21]—is computed as the intersection of the viewing cones associated with all input cameras. Several variants of Baumgart's original algorithm have been proposed over the years, including [1, 8, 25, 29], and the method recently proposed by Lazebnik *et al.* [22, 23] for computing topological mesh models of visual hulls directly in the

image domain is used in our experiments (Section 7).

Methods for recovering both the surface structure *and* the camera motion using a trinocular rig have been proposed by Vaillant and Faugeras [35] and Joshi *et al.* [18]. The single-camera case is more difficult, and the algorithms proposed by Giblin *et al.* [15], Mendonca *et al.* [26], and Wong and Cipolla [38] are limited to circular camera motions.[1] The techniques available today for tackling more general motions suffer from various limitations: Like any non-linear optimization approach, the algorithms proposed by Cipolla *et al* [6], Åström and Kahl [2], and Yezzi and Soatto [39] iteratively update estimations starting from some initial guess, and hence, are susceptible to convergence problems. On the other hand, the non-iterative method proposed by Vijayakumar *et al.* [36] has only been applied to synthetic data. In contrast, the approach presented in the rest of this article demonstrates—for the first time, as far as we know—the fully automated, non-iterative recovery of unconstrained camera motion from monocular surface outlines found in real image sequences.

2.2 Proposed Approach

As mentioned in the introduction, the only true stereo correspondences between two different outlines of a solid bounded by a smooth surface are the projections of a finite number of *frontier points*, where two viewing rays intersect as they graze the surface along the same tangent plane (Figure 2). Equivalently, the frontier points are the intersections of the corresponding contour generators on the surface, and the tangents to the apparent contour at the projections of frontier points are epipolar lines. As will be shown in Section 4, it is a relatively simple matter to estimate the projection matrices associated with $r \geq 2$ views of a smooth surface when the projections of $n \geq 2$ frontier points have been found in each one of the corresponding $r(r-1)/2$ image pairs. Conversely, it is easy to find the frontier points associated with a pair of images once the epipolar geometry is known; they are points whose

[1] This is not strictly true of the method proposed by Wong and Cipolla [38], since pictures taken from arbitrary viewpoints can be added to the original dataset once a reasonable estimate of the scene structure is available. The camera trajectory is, however, limited to a circle during the initial phase aimed at bootstrapping the structure and motion estimation process.

corresponding outline tangents are epipolar lines (Figure 2).

We define a *match* between two images as a pair of n-tuples of epipolar tangents to the corresponding outlines. In theory, n is always even and greater than or equal to two. We propose the RANSAC-based procedure outlined in Algorithm 1 to robustly estimate the projection matrices associated with an image sequence while identifying geometrically consistent matches between pairs of surface outlines. It consists of three steps: (1) the selection of promising match candidates; (2) a sampling stage to collect a set of seeds, where each seed consists of r randomly selected images together with the corresponding projection matrices estimated from a set of geometrically consistent matches; and (3) a consensus step where the remaining projection matrices are estimated seed by seed through a voting scheme, until one of them allows all the matrices to be successfully estimated.

Three main ingredients play a role in the successful implementation of this algorithm—namely, effective techniques for (A) selecting promising match candidates between a pair of outlines (step (1) of the algorithm); (B) estimating the projection matrices from match candidates associated with individual image pairs (steps (2bi), (2bii) and (3ai)); and (C) assessing the consistency of a set of matches and the corresponding projection matrices with the available geometric information (steps (2bii) and (3aii)). These ingredients are detailed in Sections 3–5.

As noted in the introduction, the approach presented in this article is applicable to orthographic, weak-perspective and affine cameras. For the simplicity of explanations, we will first focus in Sections 3–5 on the orthographic projection case, then come back to general weak-perspective and affine projection models in Section 6.

3 Selecting Promising Match Candidates

This section exploits elementary geometric properties of the orthographic projection process to select a set of potential matches between two images that are candidate projections of frontier points. We will exploit in Section 5 a set of additional multi-view constraints to

Algorithm 1 A RANSAC procedure for estimating the motion of a camera from a sequence of outlines. A *match* between images is a pair of n-tuples ($n \geq 2$) of epipolar tangents to the corresponding outlines. The algorithm takes as input the apparent contours found in m images of the same object, and outputs the set M of m projection matrices associated with the input images. Fixed parameter values of $t = 10$, $k = 50$, $r = 4$ and $s = 2$ are used in all the experiments.

(1) *Precomputation of the match candidates.*
Use the *signature* representation [28] of the m input image outlines to form the set C of potential matches between all $m(m-1)/2$ image pairs. The t most promising matches are retained for each image pair.

(2) *Sampling step: Computation of the set (H^*, M^*) of seeds, where each seed consists of a set H of $r \geq 2$ random images, with the corresponding set M of projection matrices. The geometric consistency is quantitatively assessed for each seed, providing a support measure. The seeds are stored in (H^*, M^*) in decreasing order of support measures.*
For $i \leftarrow 1$ to k do:

(a) Randomly draw a set H of r images.

(b) For each tuple of $r(r-1)/2$ matches in C between images in H do:

(i) estimate the corresponding set M of r projection matrices;

(ii) if the matrices M are consistent with available geometric information, use the set K formed by the remaining $m - r$ images to compute a measure of support for these matrices. Keep the most supported result in step (b).

(c) Add the best result in the previous step (b) to (H^*, M^*), and keep (H^*, M^*) sorted in decreasing order of support measures.

(3) *Consensus step: For each seed (H, M) in (H^*, M^*), add the remaining set K of images one by one in a greedy fashion; the image with the highest support measure is always added first. The measure of support is computed by a voting scheme for projection matrices indexed by their (discretized) viewing directions. This process is repeated, until one of the seeds allows all the camera parameters to be successfully estimated.*
Repeat for each seed (H, M) in (H^*, M^*):
Repeat:

(a) For each image K_i in the remaining set K of images, each tuple H' of $s \geq 2$ images in H, and each tuple of s potential matches in C between K_i and H' do:

(i) estimate the projection matrix $\mathcal{M}$ associated with K_i from the matches;

(ii) if $\mathcal{M}$ is consistent with available geometric information, cast a vote for it.

(b) If no vote has been cast in step (3a), go back to the first *repeat* statement of step (3) and try the next seed (failure). Otherwise, add the image K_i with the highest vote to H, add the corresponding matrix $\mathcal{M}$ to M, and delete K_i from K.

until K is empty.
until H contains all the images.

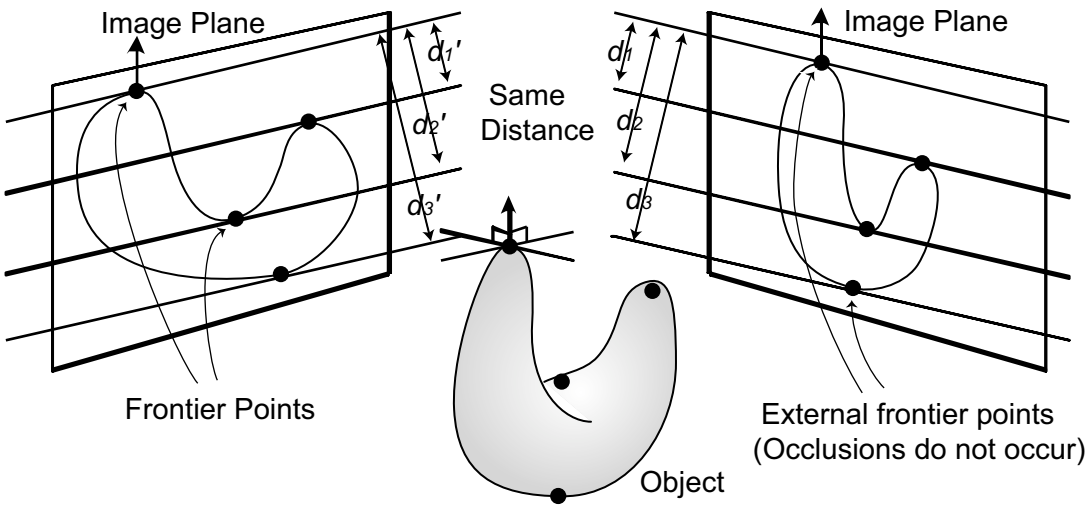


Figure 3: Under orthographic projection, the distances d_i between matching epipolar lines as frontier points are the same in the two images. Under general affine projection, the ratio d_i/d_{n-1}, where n is the number of epipolar lines, is the same in both pictures.

prune the incorrect candidates and retain the correct ones.

Under orthographic projection, the epipolar planes associated with two images are parallel to each other, and the $n \geq 2$ frontier points of a smooth surface can be ordered by picking some arbitrary common orientation for these planes (Figure 3). The epipolar lines are parallel as well, and (barring occlusion) their distances d_i $(i = 0, \ldots, n-1)$ along the perpendicular direction are easily shown to be the same in the two pictures.

This property was used in [28] as the basis for a 3D object recognition algorithm. The *signature* of a planar curve γ is defined by mapping every direction in the plane onto the tuple formed by the distances between successive lines locally tangent to γ that are perpendicular to the direction. Formally, the signature can be thought of as a representation of the set of tangent lines—or *dual*—of γ by a family of curves embedded in subspaces of $\mathbb{R}^{n-1}$ of various dimensions, where n is the maximum number of parallel tangents of γ [28]. In the motion estimation context, the dual interpretation is not necessary: Instead, it is sufficient to note that the signatures of two apparent contours intersect each other at the images of the corresponding frontier points, which affords a simple mechanism for selecting potential frontier points.

Concretely, we construct the signature of each image contour by discretizing the possible tangent directions at 0.25° intervals. Then, for each discretized direction, we compute distances $\boldsymbol{d} = (d_0, \ldots, d_{n-1})$ between epipolar tangents and normalize them by aligning the center of outermost epipolar tangents with the origin: $\widehat{d_i} = d_i - (d_0 + d_{n-1})/2$, where $\widehat{d_i}$ is the normalized distance. To find matching points between the signatures computed from the noisy outlines detected in two images despite the possibility of occlusion, we use the robust matching approach of [28, 34] to determine a meaningful "distance" between two normalized signature points $\widehat{\boldsymbol{d}} = (\widehat{d_0}, \ldots, \widehat{d_{k-1}})$ and $\widehat{\boldsymbol{d}'} = (\widehat{d'_0}, \ldots, \widehat{d'_{l-1}})$, where k may not equal l. The discrepancy between individual entries $\widehat{d_i} \in \widehat{\boldsymbol{d}}$ and $\widehat{d'_j} \in \widehat{\boldsymbol{d}'}$ is computed by the Lorentzian $L_\sigma = \sigma^2/((\widehat{d_i} - \widehat{d'_j})^2 + \sigma^2)$, where 2 pixels is used for the parameter σ in our experiments. Note that the value L_σ is 1 for a perfect match, but is close to zero for large mismatches. Since the tangent lines are naturally ordered, the final score is found by using dynamic programming to maximize the sum of the Lorentzians among all paths with non-decreasing function $j(i)$, and dividing the maximum by $\sqrt{kl}$.

This approach provides a guide for selecting promising matches between two images. We also use a number of filters to reject incorrect ones: First, the object should lie on the same side of matching tangents in both images. Second, the contour curvatures at the two projections of the same frontier point should have the same sign [19]. In practice, we exhaustively search each pair of silhouettes for a set of match candidates,[2] and retain the t most promising ones according to the score described above. The value of t is fixed to 10 in our implementation.

4 Estimating Projection Matrices from Matches

In this section, we present a stratified approach to motion estimation [10, 20] from the match candidates selected in Section 3: Affine projection matrices are computed from the

[2]We could of course use some hashing technique—based, say, on the diameter d_{n-1} of the object in the direction of interest—to improve the efficiency of the search for promising matches, but this is far from being the most costly part of our algorithm.

corresponding object outlines and their pairwise frontier points before Euclidean constraints are used to "upgrade" these matrices to orthographic ones. In a typical structure-from-motion scenario, many correspondences can be found between successive image frames, but distant picture pairs may not have any point in common. Our setting is different, with few (typically a dozen or fewer, but at least two) frontier points shared by *each* pair of images, but only visible there. Therefore, a different approach to motion estimation is called for. We proceed in three steps as described below.

4.1 Affine Motion from a Pair of Images

Exploiting the affine ambiguity of affine structure from motion allows us to write the projection matrices associated with two images I and I' in the following *canonical form* (see [12] for example):

$$\hat{\mathcal{M}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \hat{\mathcal{M}}' = \begin{bmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{bmatrix}. \tag{1}$$

Assuming there are n frontier points with three-dimensional coordinates (x_i, y_i, z_i) and coordinates (u_i, v_i) in the first image and (u'_i, v'_i) in the second image $(i = 1, \ldots, n)$, we obtain:

$$\begin{bmatrix} u_i \\ v_i \\ u'_i \\ v'_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & d \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \tag{2}$$

Using the first three equations to solve for x_i, y_i, and z_i, and substituting into the fourth one now yields

$$au_i + bv_i + cu'_i - v'_i + d = 0 \text{ for } i = 1, \ldots, n. \tag{3}$$

This is of course equivalent to the affine epipolar constraint $\alpha u_i + \beta v_i + \alpha' u'_i + \beta' v'_i + \delta = 0$, where the vector of parameters $[\alpha, \beta, \alpha', \beta', \delta]$ is proportional to $[a, b, c, -1, d]$. Given the images of $n \geq 4$ frontier points, the parameters a, b, c, and d can be computed by using linear least squares to solve the over-constrained linear system given by Eq. (3). Exploiting tangent information along the image contour reduces the number of points necessary to

estimate these parameters to $n \geq 2$. In practice, we only use the tangent information when strictly necessary—that is, when there are only two or three frontier points, first because the tangent orientation is noisy, and second because it is not clear how to best combine both types of information.

4.2 Affine Motion from Multiple Images

This section shows how to recover r projection matrices $\mathcal{M}_i$ $(i = 1, \ldots, r)$ in some global affine coordinate system once the $r(r-1)/2$ pairwise epipolar geometries are known, or, equivalently, once the projection matrices are known in the canonical coordinate systems attached to each camera pair.

Let us denote by $(a_{kl}, b_{kl}, c_{kl}, d_{kl})$ the values of the coefficients (a, b, c, d) associated with two images I_k and I_l, and assume that they have been computed from Eq. (3) and known correspondences between I_k and I_l. There must exit some affine transformation $\mathcal{A}$ mapping the canonical form (1) onto $\mathcal{M}_k$ and $\mathcal{M}_l$, i.e.,

$$\begin{bmatrix} \mathcal{M}_k \\ \mathcal{M}_l \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{M}}_k \\ \hat{\mathcal{M}}_l \end{bmatrix} \mathcal{A}. \tag{4}$$

If we write the two projection matrices $\mathcal{M}_k$ and $\mathcal{M}_l$ as

$$\mathcal{M}_k = \begin{bmatrix} \boldsymbol{p}_k^T \\ \boldsymbol{q}_k^T \end{bmatrix} \quad \text{and} \quad \mathcal{M}_l = \begin{bmatrix} \boldsymbol{p}_l^T \\ \boldsymbol{q}_l^T \end{bmatrix},$$

it is a simple matter to eliminate the unknown entries of $\mathcal{A}$ in Eq. (4) and show that

$$a_{kl}\boldsymbol{p}_k + b_{kl}\boldsymbol{q}_k + c_{kl}\boldsymbol{p}_l - \boldsymbol{q}_l = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -d_{kl} \end{bmatrix}. \tag{5}$$

In other words, we have four linear constraints on the entries of the matrices $\mathcal{M}_k$ and $\mathcal{M}_l$. By combining the equations associated with all image pairs, we obtain a linear system of $2r(r-1)$ linear equations in the $8r$ entries of the r projection matrices, whose solutions are only defined up to an arbitrary affine transformation. We remove this ambiguity by fixing two projection matrices to the canonical form given by (1). The solution of the remaining

$2r(r-1)-4$ linear equations in $8(r-2)$ unknowns is again computed by using linear least squares. Two images are sufficient to compute a single solution, and four images yield redundant equations that can be used for consistency checks as explained in the next section.

4.3 Euclidean Upgrade

Let us write the affine projection matrices recovered in the previous section as $\mathcal{M}_i = \begin{bmatrix} \mathcal{A}_i & \boldsymbol{b}_i \end{bmatrix}$ $(i = 1, \ldots, r)$. As shown in [31] for example, once the affine projection matrices are known, there exists an affine transformation, or *Euclidean upgrade*,

$$\mathcal{Q} = \begin{bmatrix} \mathcal{C} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \text{ such that } \mathcal{M}_i \mathcal{Q} = \begin{bmatrix} \mathcal{R}_i & \boldsymbol{b}_i \end{bmatrix},$$

where the 2×3 matrix $\mathcal{R}_i$ is the top part of a 3×3 rotation matrix, and $\mathbf{0} = (0, 0, 0)^T$. It follows that

$$\mathcal{A}_i(\mathcal{C}\mathcal{C}^T)\mathcal{A}_i^T = \mathcal{A}_i \mathcal{S} \mathcal{A}_i^T = \mathrm{Id}_2, \tag{6}$$

where $\mathcal{S} = \mathcal{C}\mathcal{C}^T$, and Id_2 is the 2×2 identity matrix. The r instances of this equation provide $3r$ constraints on the 6 independent entries of the symmetric matrix $\mathcal{S}$, allowing its recovery via linear least squares. Once $\mathcal{S}$ is known, the matrix $\mathcal{C}$ can be recovered using Cholesky factorization for example [27].[3]

5 Assessing the Consistency of Matches and Projection Matrices

The signature of the apparent contour was used in Section 3 to find match candidates between two images. We now use the constraints arising from multiple views in a set of strategies for ensuring that only consistent matches and projection matrices are retained.

[3] This assumes that $\mathcal{S}$ is positive definite, which may not be the case in the presence of noise. See [31] for another approach based on non-linear least squares and avoiding this assumption.

5.1 Filtering out Inconsistent Projection Matrices

As shown in [24] for example, the pairwise epipolar constraints among a set of images are redundant. We propose to exploit this redundancy by enforcing the corresponding geometric constraints. Furthermore, when the input images are part of a video sequence, it is possible to exploit the continuity of the camera motion. These constraints are used to reject inconsistent projection matrices during steps (2bii) and (3aii) of our algorithm.

Concretely, we assume in the rest of this section that the projection matrices associated with q images $\{I_1, \ldots, I_q\}$ have been computed from a set of match candidates. The following four criteria are used to check the consistency of the projection matrices.

1. Let us first assume that a match candidate between two images I_k and I_l $(1 \leq k, l \leq q)$ has *not* been used in the estimation process (this is a common situation because of the epipolar constraints' redundancy). The affine fundamental matrix associated with I_k and I_l is easily computed from the corresponding projection matrices, and it can be used to identify the epipolar tangents and thus *predict* the location of the corresponding frontier points' projections in the two pictures. Due to noise, discretization errors, occlusions, etc., some of the points found in one image may not have matches in the other one. Still, the two outermost—or *external*—frontier points are normally visible in each image (Figure 3), and the distances between their projections should be the same in the two images, i.e., the diameters of the two silhouettes in the direction orthogonal to the epipolar lines should be equal. One can go further and think of Eq. (2) as an over-constrained system of (non-homogeneous) linear equations in the 3D position of a frontier point F_i given its two image projections f_i and f'_i. The residual of this least-squares problem measures the image distance between f_i, f'_i, and the the projections of the recovered point F_i. The two external frontier points give two residual values, whose mean v has proven much more discriminative than the diameter difference in our experiments. We reject projection matrices for which v exceeds 5pixels.

2. This time, let us consider an *existing* match candidate for the two images I_k and I_l $(1 \leq k, l \leq q)$ that may or may not have been used in the computation of the projection matrices. The $n \geq 2$ 3D frontier points $F_1, \ldots, F_n$ associated with this match are easily reconstructed via triangulation, and their projections f_{ij} $(i = 1, \ldots, n;\ j \neq k, l)$ should lie inside the silhouettes of the remaining $q-2$ pictures. We define the *outside distance* between the point f_{ij} and the apparent contour γ_j of I_j as zero if f_{ij} lies inside γ_j, and the Euclidean distance between f_{ij} and γ_j otherwise. We reject projection matrices if the average o of all $n(q-2)$ outside distances exceed 5pixels.

3. When the input images are part of a video sequence, it is possible to exploit the continuity of the camera motion. In particular, we require the angle between the viewing directions associated with I_k and I_l $(1 \leq k, l \leq q)$ to be less than $|k-l|$ times some predefined threshold (15° in our experiments).

4. The least-squares residual associated with the equations determining the affine projection matrices (Eq. [5]) provides another filter for rejecting inconsistent projection matrices: In practice, we reject projection matrices for which the average of the value exceeds some threshold. We have used 0.05 in our experiments, but the accuracy of our algorithm has empirically proven to be quite insensitive to the specific value chosen.

5.2 Filtering out Inconsistent Match Candidates

The smooth camera motion assumption can also be exploited to reject inconsistent match candidates in the consensus step (3) of our algorithm. Suppose the projection matrices associated with images I_k and I_l have already been estimated, while those of the adjacent images I_{k-1}, I_{k+1}, I_{l-1} and I_{l+1} have not. Then, the orientations $(\theta_k^{k,l}, \theta_l^{k,l})$ of epipolar lines associated with I_k and I_l can easily be computed. To enforce the camera motion smoothness, we require that the orientations $(\theta_k^{k,l}, \theta_l^{k,l})$ should be similar to those of the adjacent pairs

(I_{k-1}, I_{l-1}) and (I_{k+1}, I_{l+1}) of images. Concretely, we enforce the following two conditions:

$$|\theta_{k-1}^{k-1,l-1} - \theta_k^{k,l}| + |\theta_{l-1}^{k-1,l-1} - \theta_l^{k,l}| < 2\tau, \quad |\theta_{k+1}^{k+1,l+1} - \theta_k^{k,l}| + |\theta_{l+1}^{k+1,l+1} - \theta_l^{k,l}| < 2\tau. \tag{7}$$

We arbitrarily use $\tau = 15°$ in all our experiments. In practice, for each pair of images (I_k, I_l) whose projection matrices have already been estimated, this condition is tested for all the match candidates associated with the adjacent pairs (I_{k-1}, I_{l-1}) and (I_{k+1}, I_{l+1}) of images, and inconsistent match candidates are discarded. This consistency check not only helps to improve the accuracy of our algorithm, but also effectively speeds up the consensus step (3) by decreasing the number of match candidates to be tested.

5.3 Sampling Step

By using the consistency check methods introduced in Section 5.1, we collect a set (H^*, M^*) of k seeds in the sampling step, where each seed is a set H of $r \geq 2$ randomly selected images, with a corresponding set M of estimated projection matrices.

As shown in Section 4, $r \geq 2$ images, together with $n \geq 2$ frontier points per image pair, are sufficient to estimate the corresponding r projection matrices and the $r(r-1)/2$ pairwise epipolar geometries. This assumes, of course, that all the matches used in the estimation process are correct, and thus requires an effective method to assess the consistency of estimated projection matrices with the available geometric information. Concretely, we use the process illustrated by Figure 4: We randomly select r images $H = \{H_1, \ldots, H_r\}$ from the input images, and for each possible tuple of promising matches among them, we estimate the corresponding r projection matrices $M = \{\mathcal{M}_1, \ldots, \mathcal{M}_r\}$ by using these matches. Next, we evaluate how well the estimated matrices M are supported by the remaining images $K = \{K_1, \ldots, K_{m-r}\}$. Then we select the pair (H, M) with best support among all the tuples, and add it to (H^*, M^*). This process is repeated k times, and k sets of images and matrices stored in (H^*, M^*) are sorted in decreasing order of support measures.

Our measure of support is defined as follows (Figure 4): For each image K_i $(i = 1, \ldots, m-r)$ in K, we randomly draw a single subset H' of $s \geq 2$ images in H. Since the projection

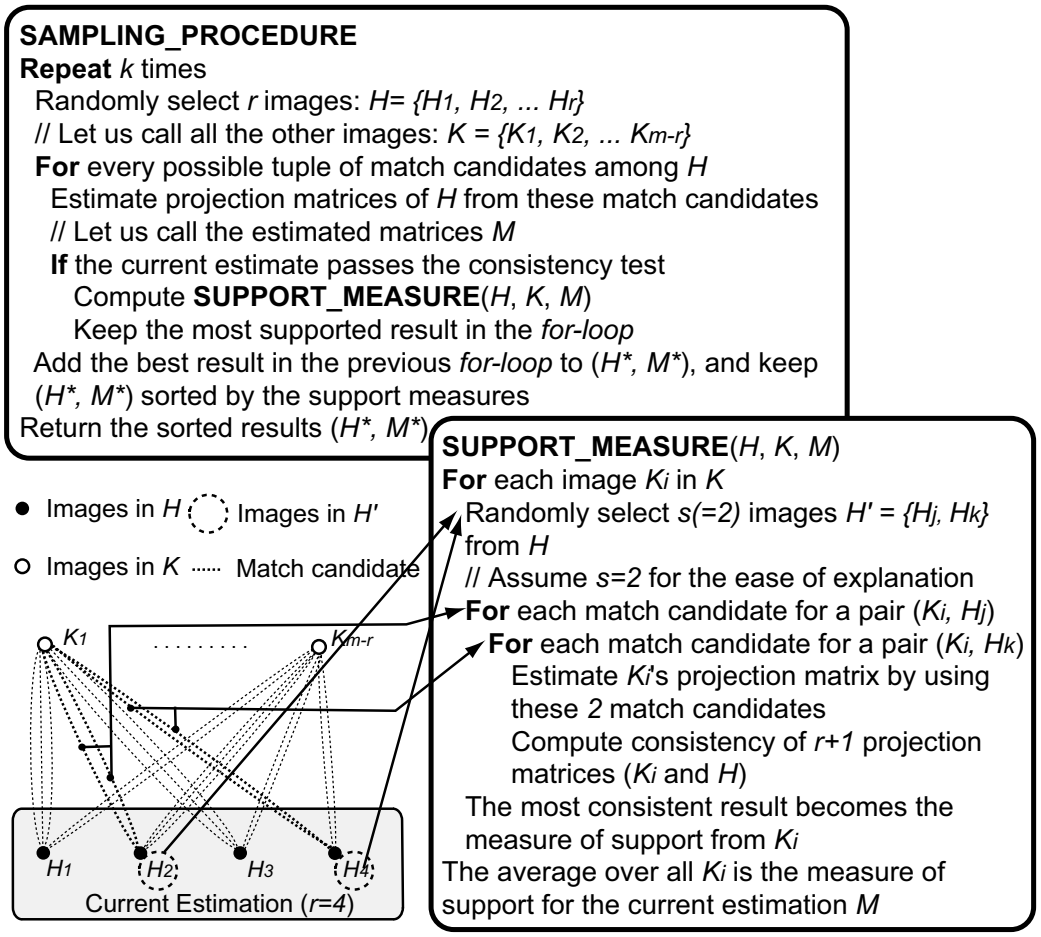


Figure 4: A sampling procedure to estimate sets of geometrically-consistent projection matrices for a minimal set of images, and the method used by this procedure to compute how well r projection matrices are supported by all the other images. We assume $s = 2$ for simplicity in this figure.

matrices associated with the elements of H have already been estimated, it is sufficient to match K_i with two elements of H' to estimate its projection matrix $\mathcal{M}_i$.[4] Thus, for each pair (H_j, H_k) of images in H' and each pair of matches associated with the images (K_i, H_j) and (K_i, H_k), we do the following: (a) estimate $\mathcal{M}_i$ from the pair of matches; (b) compute the consistency score v (defined as in the previous section) associated with K_i and every image H_l, with $l \neq j, k$ in H, and report the mean $\hat{v}$ of these $r-2$ scores; (c) compute the consistency scores o (defined again as in the previous section) associated with the two matches, and report their average $\hat{o}$; (d) report $\hat{v}+\hat{o}$ as the final consistency score associated with the pair (H_j, H_k) and the two matches. The maximum score over all matches and all pairs in H' is taken as the measure of support $S(K_i)$ of the image K_i for H. The overall measure of support for H is computed as the average of the individual measures, or $\sum_{i=1}^{m-r} S(K_i)/(m-r)$.

5.4 Consensus Step

In the sampling step, multiple sets (or seeds) of r projection matrices have been obtained. In the consensus step, the rest of the parameters ($m-r$ projection matrices) are estimated for each seed, until one seed is found that allows all the parameters to be successfully estimated without breaking consistency conditions. Let us suppose that r projection matrices associated with images $H = \{H_1, \ldots H_r\}$ are obtained as one of the seeds, and consider the problem of adding one more image K_i from $K = \{K_1, \ldots, K_{m-r}\}$ to H (Figure 5).

We use a greedy approach and always select the image K_i best supported by the pictures in H. The process is similar to the one used in the previous section, except that, this time, (1) the elements of H support the image K_i instead of the opposite, (2) all subsets H' of size s are used instead of a single one in the computation of the support measure, and (3) a non-linear (voting) scheme is used instead of a linear (averaging) one to compute this

[4]The only unknowns are the eight parameters of the projection matrix associated with K_i, and one match candidate gives us four constraints as shown by Eq. (5). Therefore, $s \geq 2$ match candidates are enough for this estimation.

measure—a choice justified empirically by our experiments.

Concretely, we tessellate the unit sphere and represent each projection matrix by its viewing direction on the sphere. We loop over each subset H' of size $s \geq 2$ of H and, for each pair (H_j, H_k) of pictures in H' and each pair of matches associated with the images (K_i, H_j) and (K_i, H_k), estimate the projection matrix $\mathcal{M}_i$ associated with K_i. If the projection matrix is consistent according to the criteria of Section 5.1, we cast a vote for the cell associated with the corresponding viewing direction. The cell receiving the largest number of votes is declared the winner, and the projection matrix $\mathcal{M}_i$ finally assigned to K_i is taken to be the mean of the estimates associated with these votes. Note that motion smoothness can be enforced in this scheme by limiting the voting space to the intersection of disks on the sphere centered at the viewing directions associated with the images H_k and H_l (Figure 5). All images are added one by one to the set H (and deleted from K) by using this simple voting strategy repeatedly. Note that every time a new projection matrix is estimated, the consistency check introduced in the Section 5.2 is applied to discard inconsistent match candidates.

As already mentioned, this procedure may fail in estimating one of the matrices, because all the votes could be rejected by the consistency checks. In this case, we discard all the intermediate results, and start this process from the beginning with the next seed.

6 The Weak-Perspective and Affine Projection Cases

The robust approach to motion estimation from surface outlines proposed in Sections 2 to 5 has been presented in the context of orthographic projection. It is also applicable, with minor modifications, to general weak-perspective and (uncalibrated) affine projection models. In this setting, epipolar tangents to the apparent contour at the projections of frontier points are still parallel to each other, but individual distances are not preserved anymore. However, (barring occlusion) the ratio d_i/d_{n-1} is the same in the two pictures, where n is the number of epipolar tangents and d_{n-1} becomes the distance between the two outermost epipolar

CONSENSUS_PROCEDURE (*H*, *K*)
// (*H*, *K*) is one of the seeds obtained in the sampling process.
// *H* is a set of images whose projection matrices are already
// estimated, and *K* is a set of all the other images.
Repeat while *K* is not empty
  For each image K_i in *K*
    For each tuple *H′* of size *s* of images in *H*
      For each tuple of match candidates between K_i and *H′*
        Estimate the projection matrix of K_i
        If the current estimate passes the consistency test
          Vote onto the tessellated sphere with the projection matrix
    Remember the higest count and its associated projection matrix
  Among all the images in *K*, the image K_i with the highest count
  is the winner
  If no vote has been casted, discard the current seed and try
    the next one (failure).
  Otherwise, add K_i to *H* and delete it from *K*
All the projection matrices are estimated

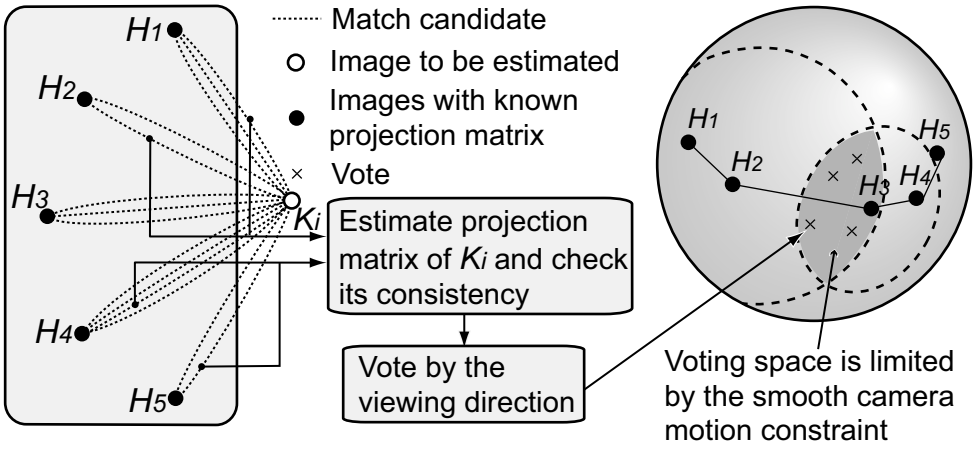


Figure 5: The procedure used to add images to the current estimates one by one while estimating the corresponding projection matrices: Two match candidates are selected to cast a vote. When a camera motion is known to be smooth, the third consistency check is applied, and the voting space is limited to the intersection of disks on the sphere.

lines. Therefore, frontier point candidates can be identified by simply comparing ratios of distances instead of the distances themselves.

As far as our algorithm is concerned, the weak-perspective projection only differs from orthographic projection in that the Euclidean upgrade constraint used in Section 4.3 takes the form $\mathcal{A}_i \mathcal{S} \mathcal{A}_i^T = \lambda_i \mathrm{Id}_2$, where λ_i is a scalar depending on the input image. In particular, r instances of this equation provide $2r$ constraints on the six independent entries of the 3×3 symmetric matrix $\mathcal{S}$, allowing its recovery and the upgrade computation from three images instead of two in the orthographic case. Another potential difference is the dimensionality of the voting space in the consensus step (3): The scale changes associated with a weak-perspective camera should be taken into account to index the projection matrix, which increases the dimension of the voting space from two to three. However, the same voting space, i.e., the discretized surface of a sphere indexed by viewing direction, has been empirically shown to also work well with the weak-perspective camera in our experiments.

In principle, going from weak-perspective to general affine projection simply entails dropping the Euclidean upgrade constraint. In practice, however, matching is expected to become more difficult in this case, with additional degrees of freedom to account for (eight independent entries per projection matrix instead of five under orthographic projection), yet fewer constraints available to solve them (no Euclidean upgrade). In addition, the dimensionality of the voting space could be an issue due to additional degrees of freedom.

7 Implementation and Results

Our experiments focus on the orthographic projection case, and this section presents the quantitative and qualitative results for nine real image sequences taken under the orthographic projection. A qualitative, "proof-of-concept" experiment with a weak-perspective projection model is presented in Section 8.

7.1 Implementation Details

The image-processing part of our implementation is rather straightforward: After a rough manual initialization, image contours are extracted using B-spline snakes and gradient vector flow [37] (Figure 6). This method allows the localization of contour points with sub-pixel precision, which is extremely important for the reliability of the proposed motion estimation process.

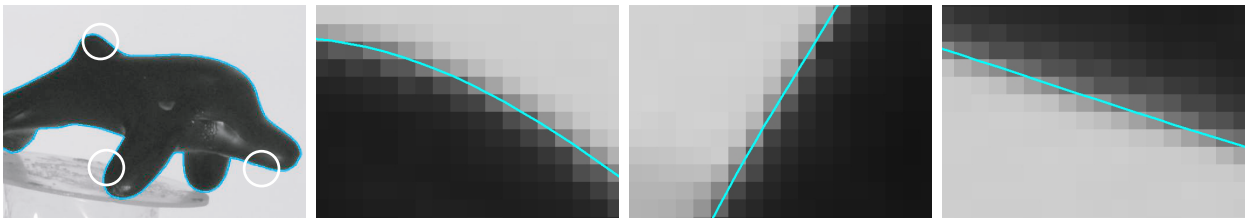


Figure 6: An image contour extracted by a B-spline snake, and close-ups demonstrating a sub-pixel localization.

We exploit the smooth camera motion and Euclidean upgrade constraints in all our experiments. The main parameters governing the behavior of our algorithm are σ of the Lorentzian function used to evaluate matching scores, the number t of match candidates stored for each image pair, the number k of random samples drawn in its sampling stage, the number r of images in each sample, and the number s of images used to compute the measure of support. Empirically, taking $\sigma = 2$, $t = 10$, $k = 50$, $r = 4$, and $s = 2$ has consistently given good results in all our experiments. r is a size of the sample set in a RANSAC procedure, and should be as small as possible in general. In practice, using a slightly redundant set of images (four instead of two) improves the reliability of our algorithm. σ is probably the only parameter that may need to be adjusted. σ should depend on the image size and the dimensions of signatures. We chose $\sigma = 2$pixels for an object around 500pixels in diameter and signatures of at most 20 in dimensions. A larger value should be set for bigger objects and signatures of lower dimensions. The running time of our algorithm is largely dominated by steps (2) and (3), that respectively take (on average) about 20 and 30 minutes each for

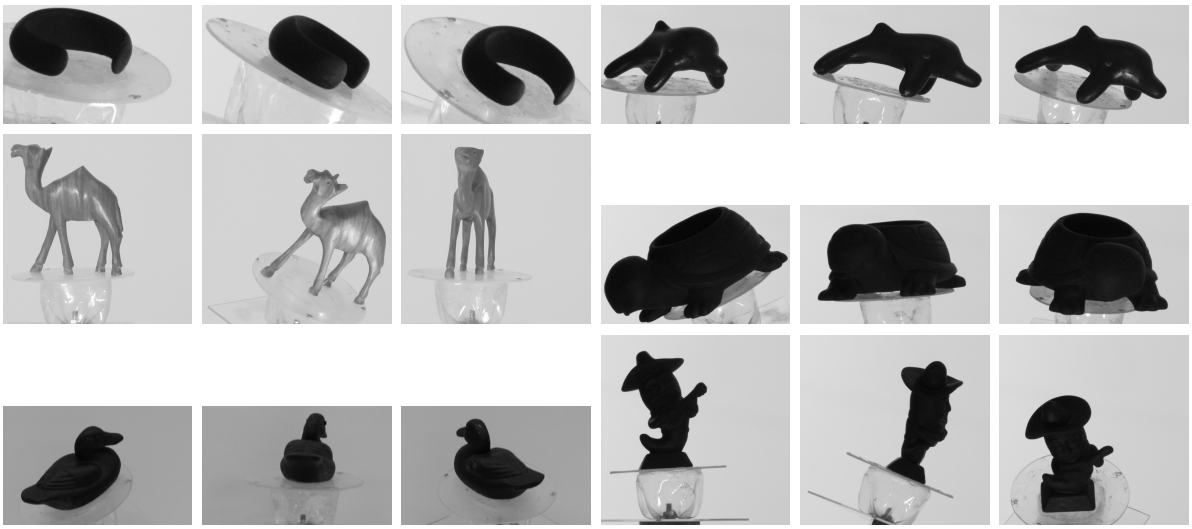


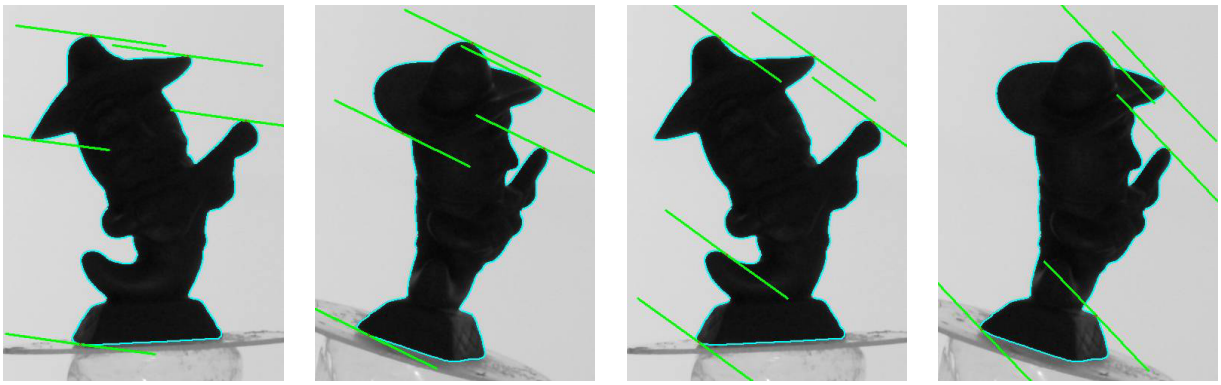
Figure 7: Sample images of six objects used in our experiments.

our C/C++ implementation running on a Pentium4 1.3Ghz equipped with the Windows operating system. In contrast, step (1) typically takes about 5 minutes.

7.2 Quantitative Evaluation

Six objects (a bracelet, a toy dolphin, a toy turtle, a toy duck, a toy camel, and a Mexican doll) have been used in our quantitative experiments under orthographic projection. Each sequence consists of 21 images acquired using a fixed camera and a pan-tilt head providing the ground truth for the viewing angles. Figure 7 shows sample images for the six objects. Image sizes range from 400×400 to 600×600 pixels. In all the sequences, pan angles between successive frames are around 8°, and tilt angles are around 3°. Figure 8 shows examples of (correct and incorrect) matches found using outline signatures: The incorrect match in the second image pair receives a high score because the distances between the corresponding epipolar tangents are almost the same, even though they are not in true correspondence. Algorithm 1 is capable of rejecting such incorrect matches by exploiting multi-view geometric constraints.

Figure 9 compares the camera trajectories recovered by our algorithm to the ground-truth



Correct match Incorrect match

Figure 8: Sample matches found using the signature of two outlines. The first match is correct, the second is not.

data from the pan-tilt head. In each case, the corresponding camera coordinate frames are first registered by a similarity transformation before being plotted on the unit sphere. For the purpose of performance evaluation, trajectories are plotted for the first four seeds found in the sampling phase. The number at the bottom right hand corner of each figure represents the number of estimated projection matrices; since each sequence consists of 21 images, the consensus step *succeeded* where the number is 21. For each sequence, the first successful trajectory, shown by a black bounding box, is the final output of our algorithm for the sequence. As can be seen from the figures, the estimated trajectories are quite accurate, especially for the first four objects, and trajectories are recovered well for most of the seeds.

Figure 10 shows some statistics for the four consistency check methods proposed in Section 5.1: The number of votes that have been rejected by each of four consistency check methods, the number of votes that have passed all the tests, and the number of total votes that have been tested. These six numbers are counted for each newly estimated projection matrix in the consensus step. Note that the sum of the first five numbers is not equal to the number of total votes, because a single vote can be rejected by multiple consistency checks at the same time. As can be seen from the graph, the number of total votes first increases, then decreases. However, in principle, as more projection matrices are estimated, more images are used to

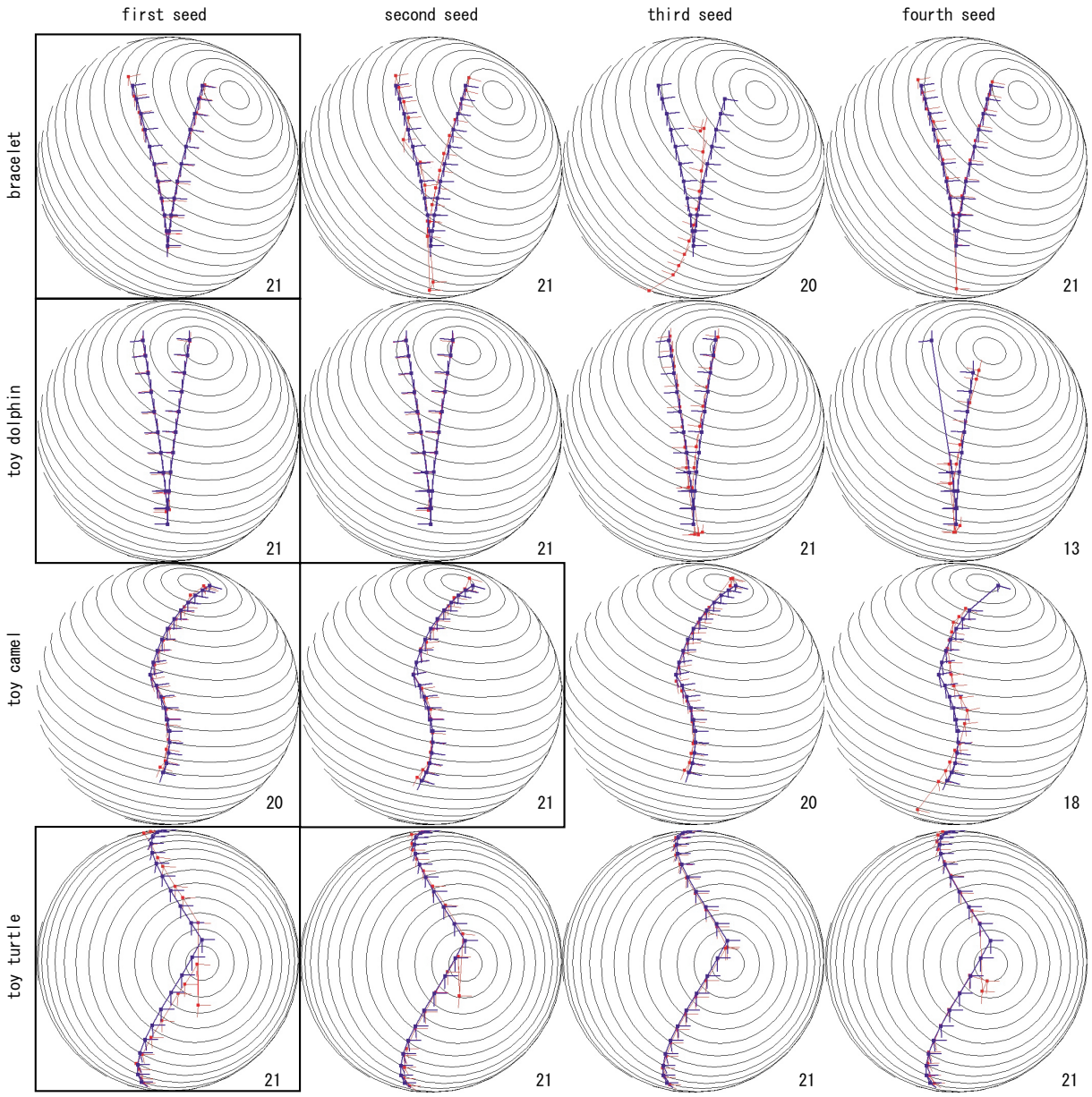


Figure 9: In the figures, thick lines represent ground truth data and thin lines represent our estimates of the camera trajectories. The viewing direction is depicted as a point on the sphere and the orientation of the image plane is shown as a pair of coordinate axes. For each object, results are shown for the first four seeds. The number at the bottom right hand corner of each figure represents the number of estimated projection matrices. Each image sequence consists of 21 pictures, therefore the number being 21 means the success; all the camera parameters are estimated. A trajectory with a black bounding box is the first success of each object, and the output of our algorithm for the object.

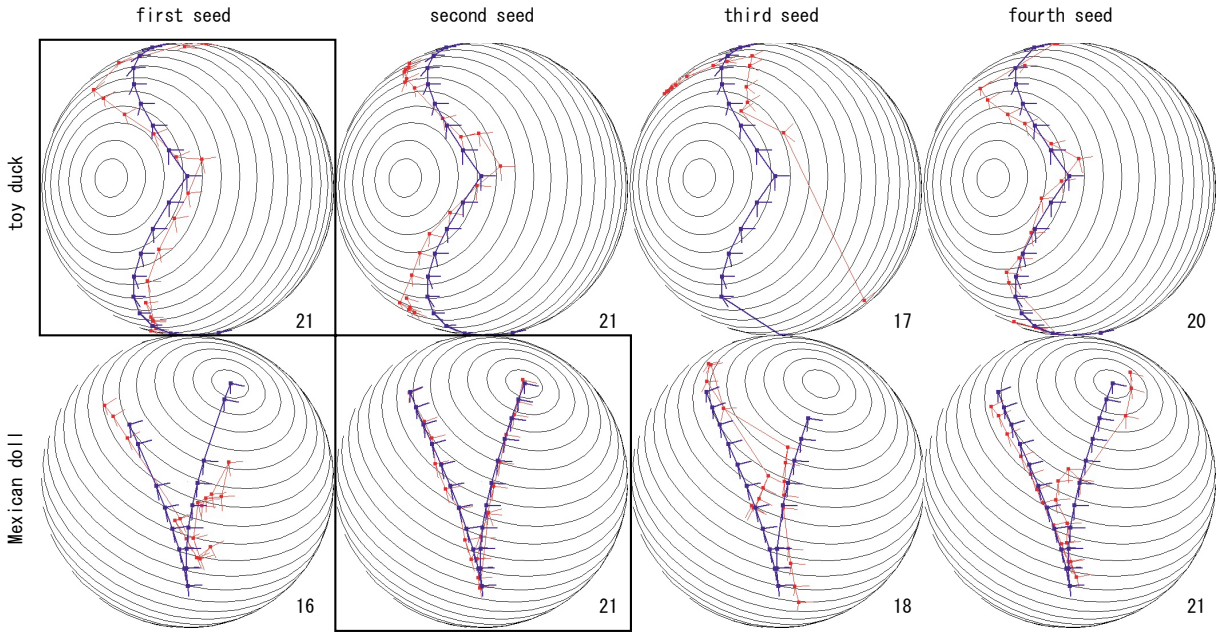


Figure 9: Continued

cast votes, and hence, the total number of votes should keep increasing by following x^s, where x is the number of estimated projection matrices, and $s = 2$ is the number of images used to compute a support measure as explained in Section 5.4. This illustrates the fact that inconsistent match candidates are effectively filtered out by the consistency check proposed in Section 5.2: As more and more projection matrices are estimated, more constraints in the form of (7) are enforced, and more inconsistent match candidates are filtered out. The number of total votes also shows the difficulty (mainly due to the occlusions) of each image. For an image with severe occlusions, a large proportion of identified match candidates are incorrect and discarded by the consistency checks, and hence, fewer votes are cast for difficult images. Since projection matrices are estimated in a greedy fashion in the consensus step, the image with the most votes is always estimated first, images getting more and more difficult as the consensus step proceeds. This is clearly visible, for example, in the 18th through 21st images of the toy duck sequence, where the number of total votes dramatically decreases.

Figure 11 presents some quantitative results. The top two graphs show that the angular

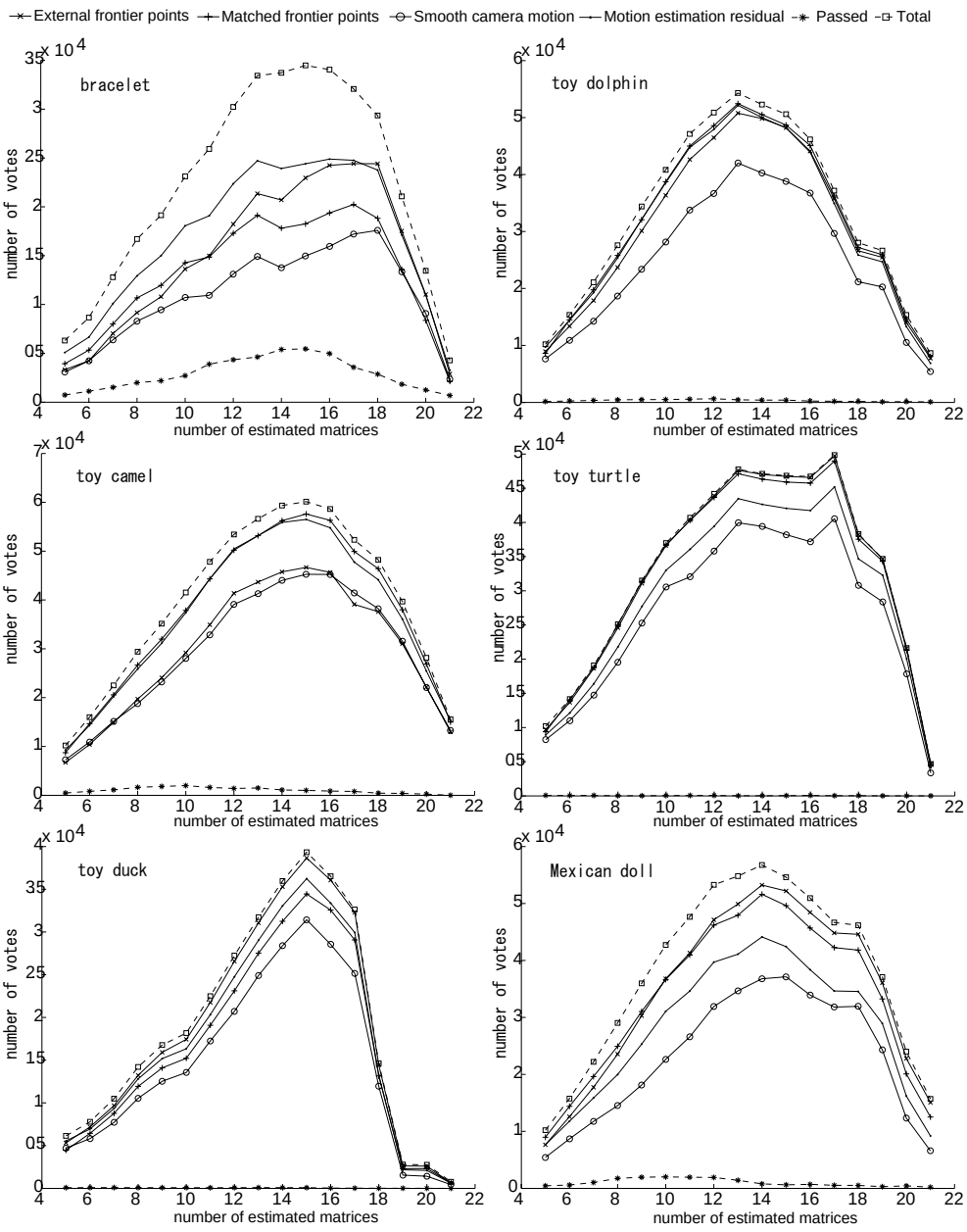


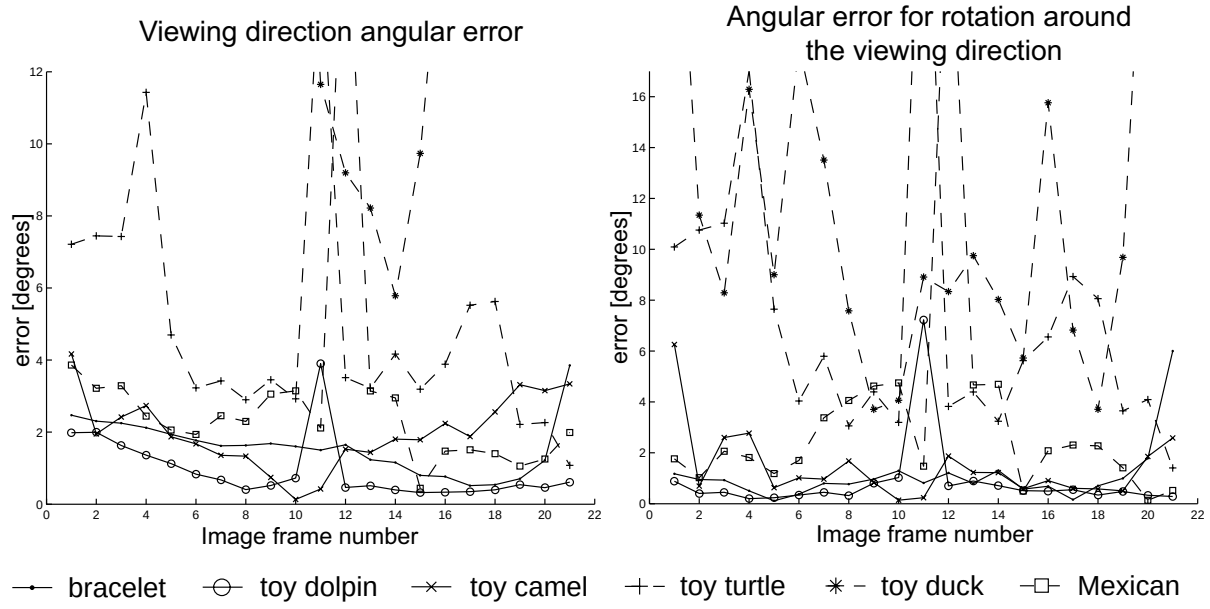
Figure 10: This figure illustrates how the four consistency check methods work in the consensus step by plotting six numbers: The number of votes that have been rejected by each of four consistency checks, the number of votes that have passed all the tests, and the number of total votes. The numbers are counted for each newly estimated projection matrix.

errors tend to decrease in the middle of image sequences, which intuitively corresponds to the fact that pictures in that range are supported by a larger number of neighbors than images at either end of the sequence. As shown by the bottom table, the projection matrices recovered from the bracelet and dolphin sequences are quite accurate, with angular errors around one degree. The camel and turtle sequences are interesting, because their contours are not smooth everywhere; yet, the results are reasonably good. This is mainly because frontier points tend to occur at high curvature points, and outlines do not have to be smooth everywhere. Rather large errors are obtained for the duck sequence, however; this is due to a few erroneous projection matrices at the beginning and the end of the sequence, with accurate estimates in its middle part. The Mexican doll has a complicated shape, which results in relatively larger errors compared to the bracelet or the toy dolphin.

7.3 Qualitative Evaluation

To give a qualitative idea of the accuracy of our motion estimation procedure, we show the visual hulls of our six objects, computed from the recovered camera motion and image silhouettes by the algorithm presented in [23]. As can be seen in Figure 12, the surfaces are recovered quite well. In fact, most inaccuracies are not so much due to errors in the recovered projection matrices as to the fact that a limited set of camera positions was used to construct each visual hull model.

This is illustrated further with three additional image sequences of the toy camel, a yellow duck, and a toy featuring a man sitting in a flying saucer. Figure 13 shows the recovered visual hulls of the three objects with textures mapped on them. The visual hull of the toy camel is of a much higher quality than before, simply due to the fact that we use more images (33 images) to compute it. The texture for the man in the flying saucer is not mapped accurately on the visual hull. This is due to the limited set of viewpoints used to construct the model, and to the fact that concave parts of a surface never show up on its visual hull. Note that in all three sequences, the final outputs are obtained from the first seed.



Mean and standard deviation of angular errors

Sequence		bracelet	dolphin	camel	turtle	duck	mexican
Viewing direction angular error [degrees]	Mean	1.09	0.84	1.41	7.42	12.7	3.38
	Standard Deviation	1.16	1.45	1.32	6.02	9.62	4.96
Angular error for rotation around the viewing direction [degrees]	Mean	1.59	0.93	1.99	4.96	25.0	3.00
	Standard Deviation	0.75	0.84	0.97	3.27	11.6	3.43

Figure 11: Quantitative experimental results. Viewing direction angular errors and angular errors for rotation around the viewing direction are plotted for 6 sequences. The mean and the standard deviation of these errors are also shown in the bottom table.

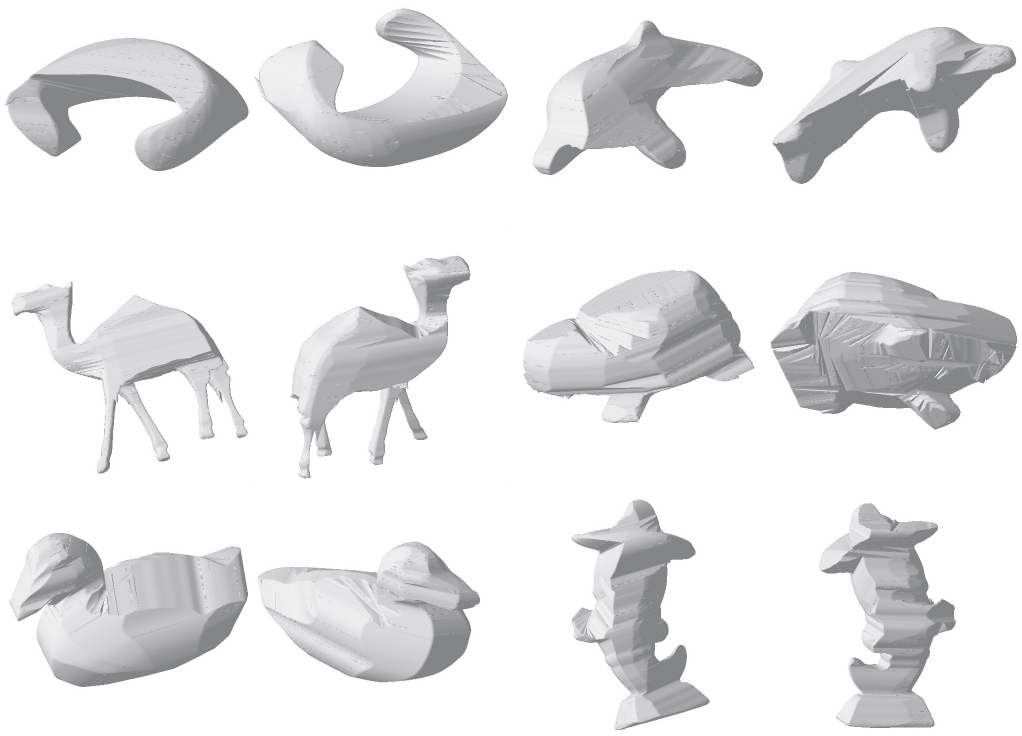


Figure 12: Visual hull models constructed using the recovered camera projections.

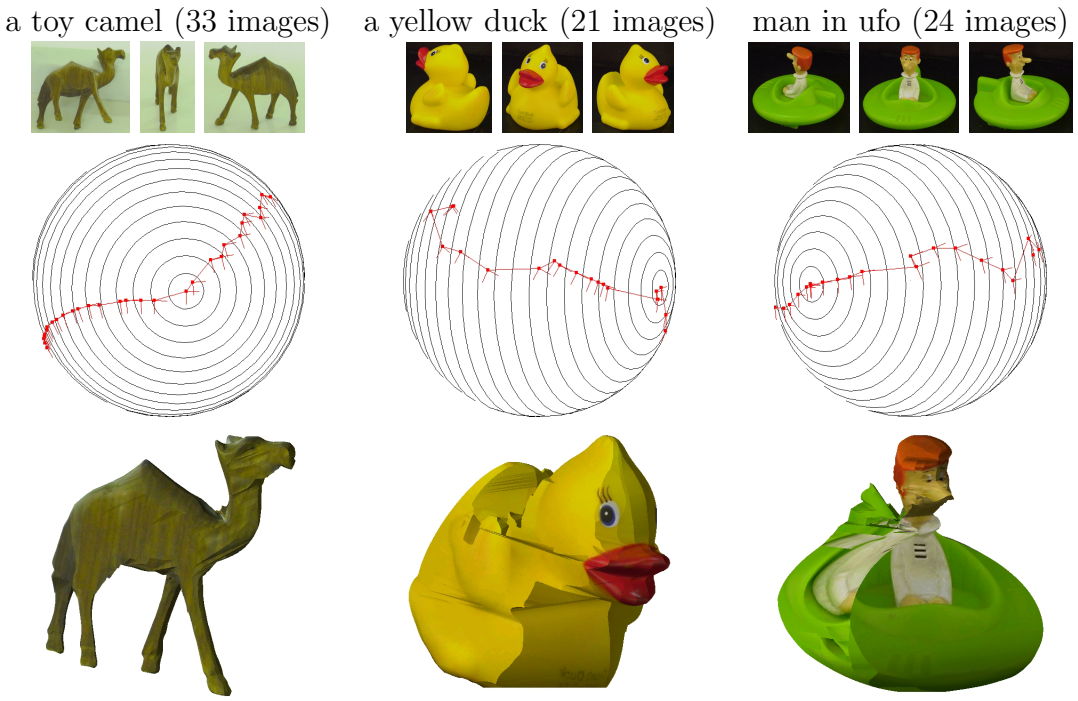


Figure 13: Input images, estimated motion trajectories, and recovered surfaces with textures whose ground truth motion data are unavailable.

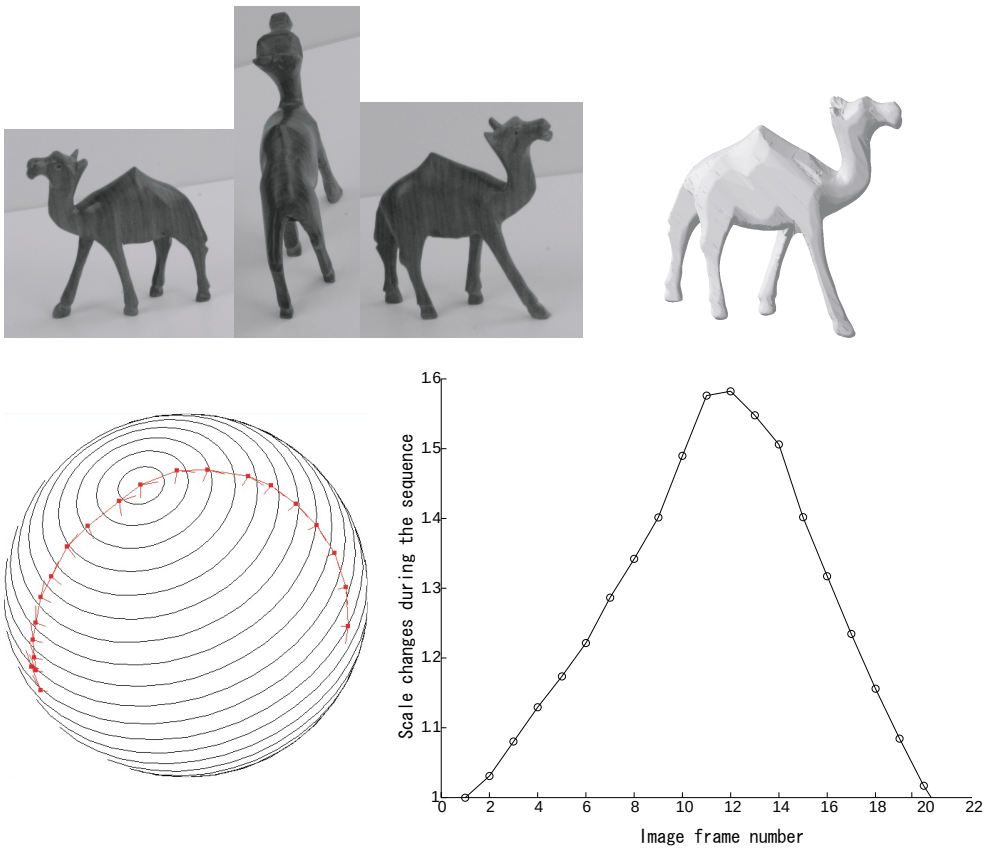


Figure 14: An image sequence taken under the weak-perspective projection. Input images, the estimated motion trajectory, the recovered surface and the scale changes during the sequence are presented.

8 Conclusion

We have presented an effective RANSAC procedure for estimating the motion of orthographic, weak-perspective, or affine cameras from a sequence of image silhouettes. The proposed algorithm uses the signature representation of the dual of image outlines to identify promising epipolar geometries for each pair of images. A minimal set of projection matrices are estimated by making most of the redundancy of multi-view epipolar geometry, and a voting scheme is taken in the consensus step to estimate rest of the projection matrices, where each vote is computed from a set of geometrically consistent epipolar geometry candidates.

Unlike other implemented methods that either restrict the range of camera motion or rely on an iterative optimization process, our method can handle arbitrary continuous camera motions, and it does not require an initial estimate of the camera trajectory. Quantitative and qualitative experiments with nine real image sequences taken under the orthographic projection have demonstrated the accurate recovery of both the camera motion and the object structure in the form of visual hulls. We claimed earlier that our algorithm could be extended to the weak-perspective case. As a proof of concept, we show in Figure 14 the results of an experiment using weak-perspective projection. In the sequence, the camera first approaches the object, then moves away from it. The corresponding scale changes are illustrated by the graph shown in the figure.

By design, our algorithm is also amenable to a recursive implementation where additional images are added one by one to refine the motion and structure estimation after a minimal set has been used to bootstrap the process. A robust non-iterative algorithm for estimating arbitrary motions from outlines observed under full perspective projection is still missing. Of course, one could use the output of our algorithm as an initial guess for one of the existing iterative approaches to the same problem [6, 2, 38, 39]. An alternative is to go beyond pure epipolar geometry, and actually exploit the 3D structure of the observed surface in identifying frontier points by making the most of the current estimates of projection matrices. We are currently exploring this line of research in the projective visual hull framework proposed in [22, 23].

Acknowledgments. This research was partially supported by the National Science Foundation under grant IIS-0312438, and the Beckman Institute. We thank Svetlana Lazebnik for providing the original visual hull software used in our implementation.

References

[1] N. Ahuja and J. Veenstra. Generating octrees from object silhouettes in orthographic views. *IEEE Trans. Patt. Anal. Mach. Intell*, 11(2):137–149, 1989.

[2] Kalle Åström and Fredrik Kahl. Motion estimation in image sequences using the deformation of apparent contours. *IEEE Trans. Patt. Anal. Mach. Intell*, 21(2):114–127, 1999.

[3] B.G. Baumgart. *Geometric modeling for computer vision*. PhD thesis, Stanford University, 1974.

[4] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, 1998.

[5] Edmond Boyer and Marie Odile Berger. 3d surface reconstruction using occluding contours. *Int. J. of Comp. Vision*, 22(3):219–233, 1997.

[6] Roberto Cipolla, Kalle E. Åström, and Peter J. Giblin. Motion from the frontier of curved surfaces. In *Proc. Int. Conf. Comp. Vision*, pages 269–275, 1995.

[7] Roberto Cipolla and Andrew Blake. Surface shape from the deformation of apparent contours. *Int. J. of Comp. Vision*, 9(2):83–112, 1992.

[8] C.I. Connolly and J.R. Stenstrom. 3D scene reconstruction from multiple intensity images. In *Proc. IEEE Workshop on Interpretation of 3D Scenes*, pages 124–130, Austin, TX, November 1989.

[9] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[10] O.D. Faugeras. Stratification of 3D vision: projective, affine and metric representations. *Journal of Optical Society America A*, 12(3):465–484, March 1995.

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[12] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2002.

[13] Yasutaka Furukawa, Amit Sethi, Jean Ponce, and David J. Kriegman. Structure and motion from images of smooth textureless objects. In *ECCV (2)*, pages 287–298, 2004.

[14] P. Giblin and R Weiss. Epipolar curves on surfaces. *Image and Vision Computing*, 13(1):33–44, 1995.

[15] Peter Giblin, Frank E. Pollick, and J. E. Rycroft. Recovery of an unknown axis of rotation from the profiles of a rotating surface. *Journal of Optical Society America*, pages 1976–1984, 1994.

[16] Peter Giblin and Richard Weiss. Reconstruction of surface from profiles. In *Proc. Int. Conf. Comp. Vision*, pages 136–144, 1987.

[17] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[18] Tanuja Joshi, Narendra Ahuja, and Jean Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. In *Proc. Int. Conf. Comp. Vision*, pages 290–295, 1995.

[19] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[20] J.J. Koenderink and A.J. Van Doorn. Affine structure from motion. *Journal of Optical Society America A*, 8:377–385, 1990.

[21] A. Laurentini. How far 3D shapes can be understood from 2D silhouettes. *IEEE Trans. Patt. Anal. Mach. Intell*, 17(2):188–194, Februrary 1995.

[22] S. Lazebnik. Projective visual hulls. Technical Report MS Thesis, University of Illinois at Urbana-Champaign, 2002.

[23] Svetlana Lazebnik, Edmond Boyer, and Jean Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 156–161, 2001.

[24] Noam Levi and Michael Werman. The viewing graph. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 518–522, 2003.

[25] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH*, 2001.

[26] Paulo Mendonca, Kwan-Yee K. Wong, and Robert Cipolla. Camera pose estimation and reconstruction from image profiles under circular motion. In *Proc. Euro. Conf. Comp. Vision*, pages 864–877, 2000.

[27] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Patt. Anal. Mach. Intell*, 19(3):206–218, March 1997.

[28] Amit Sethi, David Renaudie, David Kriegman, and Jean Ponce. Curve and surface duals and the recognition of curved 3d objects from their silhouette. *Int. J. of Comp. Vision*, 58(1):73–86, 2004.

[29] S. Sullivan and J. Ponce. Automatic model construction, pose estimation, and object recognition from photographs using triangular splines. *IEEE Trans. Patt. Anal. Mach. Intell*, 20(10):1091–1096, Oct. 1998.

[30] Richard Szeliski and Richard Weiss. Robust shape recovery from occluding contours using a linear smoother. *Int. J. of Comp. Vision*, 28(1):27–44, 1998.

[31] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. of Comp. Vision*, 9(2):137–154, 1992.

[32] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. of Comp. Vision*, 24(3):271–300, 1997.

[33] P.H. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *Proc. Int. Conf. Comp. Vision*, pages 1037–1042, 1995.

[34] P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[35] Régis Vaillant and Olivier D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. Patt. Anal. Mach. Intell*, 14(2):157–173, 1992.

[36] B. Vijayakumar, David J. Kriegman, and Jean Ponce. Structure and motion of curved 3d objects from monocular silhouettes. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 327–334, 1996.

[37] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Structure-adaptive b-snake for segmenting complex objects. In *IEEE International Conference on Image Processing*, 2001.

[38] Kwan-Yee K. Wong and Robert Cipolla. Structure and motion from silhouettes. In *Proc. Int. Conf. Comp. Vision*, pages 217–222, 2001.

[39] Anthony J. Yezzi and Stefano Soatto. Structure from motion for scenes without features. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages I: 525–532, 2003.