TELECOM PARIS
école nationale supérieure des télécommunications

Using silhouette coherence for 3D image-based object modeling under circular motion

Utilisation de la cohérence entre silhouettes pour la modélisation d'objets 3D à partir d'images de séquences en rotation

---

Carlos Hernández Esteban
Francis Schmitt

2003D011

2003

Département Traitement du Signal et des Images
CNRS UMR 5141
Groupe Traitement et Interprétation des Images

Dépôt légal : 2003 – 4ème trimestre
Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

Using Silhouette Coherence for 3D Image-based Object Modeling under Circular Motion

Utilisation de la cohérence entre silhouettes pour la modélisation d'objets 3D à partir d'images de sequences en rotation

Carlos Hernández Esteban and Francis Schmitt

Signal and Image Processing Department
Ecole Nationale Supérieure des Télécommunications, Paris, France
{carlos.hernandez,francis.schmitt}@enst.fr

Abstract

We present a new concept of silhouette coherence applied to a set of silhouettes generated by a 3D object. The proposed silhouette coherence concept is used in the problem of camera calibration for image-based 3D object modeling under circular motion. We first discuss the silhouette coherence notion and define a practical criterion to estimate it. This criterion depends both on the silhouettes and on the parameters of the cameras that generated them. These parameters can be optimized for the 3D modeling problem by maximizing the overall silhouette coherence. We present various experimental results illustrating the efficiency of the proposed silhouette coherence criterion.

Résumé

Nous développons le concept de cohérence d'un ensemble de silhouettes générées par un objet 3D pour l'appliquer aux problèmes de recalage 3D et d'estimation de mouvement. Nous discutons d'abord la notion de cohérence de silhouettes et proposons un critère de cohérence ainsi qu'une manière efficace de le calculer. Ce critère dépend simultanément des silhouettes et des paramètres des caméras. Les paramètres des caméras peuvent ainsi être optimisés en maximisant la cohérence des silhouettes. Nous présentons plusieurs résultats expérimentaux qui illustrent l'efficacité du critère de cohérence de silhouettes proposé.

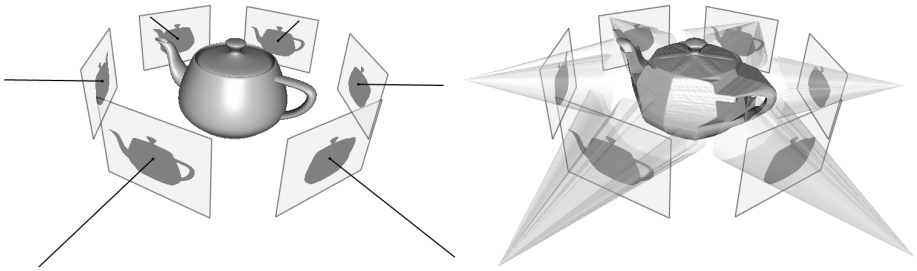


Fig. 1. Silhouette acquisition and its corresponding reconstructed visual hull (case of a rotation sequence). Left: original object and its corresponding silhouettes. Right: reconstructed visual hull using the same set of silhouettes.

1 Introduction

3D image-based modeling is becoming more popular as recent techniques provide high quality models. Shape from silhouette techniques are specially interesting since they provide good initial models for further processing in 3D reconstruction algorithms and are efficient enough for real time applications.

Camera calibration is a very important initial step for 3D reconstruction. The technique presented in this paper makes use of the silhouettes in a novel way in order to recover the camera parameters under circular motion, which is commonly used for 3D image-based modeling. Assuming that the silhouettes are well extracted, we propose the silhouette coherence as a global quality measure for the calibration of a subset of the camera parameters by maximizing the overall silhouette coherence. This allows us to obtain high quality 3D reconstructions without the use of any calibration pattern.

This papers is organized as follows: in section 2 we present the related work. In section 3 we define and discuss the silhouette coherence measure. In section 4 we propose a fast implementation of the coherence criterion. In section 5 we present an application to visual hull registration and in section 6 an application to motion and focal estimation.

2 Related Work

The approach presented in this paper is related to three different kinds of techniques: i) camera calibration, ii) texture registration between images and a 3D reconstructed model, and iii) visual hull computation.

A large collection of methods for camera calibration exists [10]. They rely on correspondences between the same primitives detected in different images. In the particular case of circular motion, the methods in [11, 12] work well when the images contain enough texture to make a robust detection of the primitives. Otherwise silhouettes can be used instead. Calibration based on silhouettes is used for two main applications: visual hull 3D registration [27, 23, 6] and motion estimation [21, 13]. Although [27] does not explicitly propose the registration of two different sequences of silhouettes, they

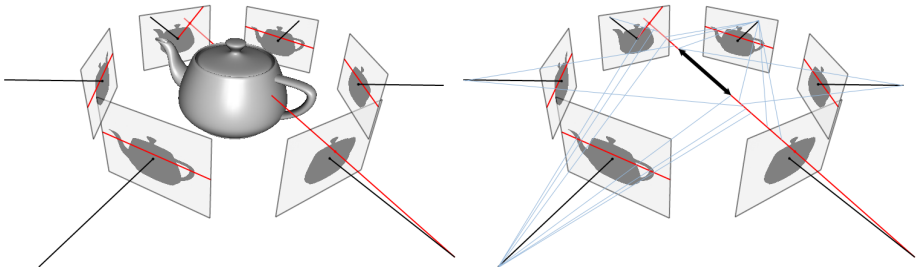


Fig. 2. Computation of the 3D optic ray intersection by back projection of the silhouette intervals.

propose a way of estimating the pose of a 3D model relative to a set of silhouettes, so silhouette registration can in fact be achieved by first reconstructing a 3D model using one of the sequences followed by a pose estimation relative to the second sequence of silhouettes. In [23] two different reconstructions are obtained from every silhouette sequence. Then, the two 3D reconstructed models are matched using tangent planes and stability constraints. [6] also start with different reconstructions. They use them to speed up stereo matching to locate 3D points, and with the 3D points they are able to register/refine both visual hulls. In [21, 13] motion recovery is achieved using the notion of epipolar tangencies [26, 24], i.e., points on the silhouette contours that belong to an epipolar line tangent to the silhouette. Although this method gives good results, its main drawback is that there is a limited number of epipolar tangencies per pair of images, generally only two, and their detection is very sensitive to noise.

Concerning texture registration, there exists a number of algorithms that try to register a set of textured views with a 3D representation of the object using silhouettes [16, 19, 22]. Calibration is accomplished by minimizing the error between the contours of the silhouettes and the contours of the projected object. In [19, 22] the error is defined as the sum of the distances between a number of sample points on one contour and their nearest points on the other. In [16] a hardware-accelerated method is used to compute the similarity between two silhouettes as the area of their intersection.

Finally, visual hull computation is a very active area since it is one of the fastest and most robust ways of obtaining an initial estimation of a 3D object. It can be precise enough for real time rendering applications such as [17] or used as an initial estimation for further processing in 3D reconstruction algorithms [9].

In this paper we define a silhouette coherence criterion inspired from [16] but the main difference is that we do not need a known 3D model corresponding to the real object. The 3D model is *implicitly* reconstructed from the silhouettes at the same time as the camera calibration by a visual hull method [1, 4, 7, 8, 14, 18, 20]. In particular, the use of the technique described in [20] allows all the computations to be done in the image domain, which overcomes the need for a 3D representation as in [27, 23, 6].

3 A Measure of Silhouette Coherence

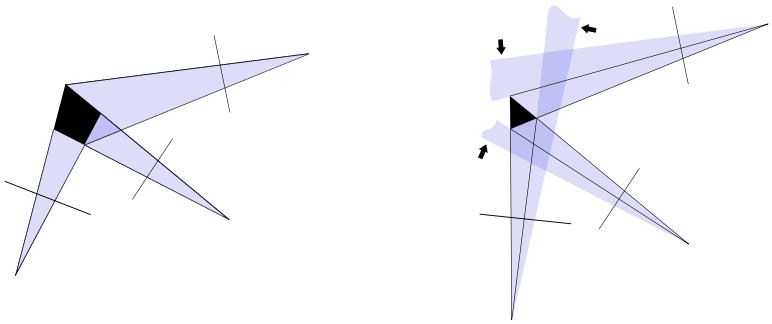
It seems that the notion of silhouette coherence has not already been developed in the literature. However, [3] study the same problem but with a different point of view. They provide the notion of *silhouette compatibility* to state if a given set of silhouettes of a same object are possible or not. They give some rules to detect if a set of silhouettes is compatible or not, but only for the special case of orthographic cameras and without providing a means to compute their amount of *incompatibility*. In the following we describe a criterion of silhouette *coherence* that allows us to measure a degree of coherence and thus offers us the possibility of using optimization methods to recover some of the camera parameters.

Given a set of silhouettes of a same 3D object taken from different points of view, and the corresponding set of camera projection matrices, we would like to measure the coherence of both the silhouette segmentation and the camera projection matrices. The only information contained in a silhouette is a classification into two categories of all the optic rays that go through the optic center of the associated view: those that intersect the object and those that do not, depending on if the pixel that defines the optic ray belongs to the silhouette or not. Let us consider an optic ray intersecting the object. Its projection into any other view *must* intersect the corresponding silhouette. The back projection of this intersection onto the 3D optic ray defines one or several 3D intervals where we know that the optic ray intersects the real object surface (see Fig.2). In the case of only two views, the corresponding silhouettes will not be coherent if there exists at least *one* optic ray classified as intersecting the object by one of the silhouettes and whose projection does not intersect the other silhouette. In the case of N views, the lack of coherence is defined by the existence of at least one optic ray where the depth intervals defined by the $N-1$ other silhouettes have an empty intersection. This lack of coherence can be measured simply by counting how many optic rays in each silhouette are not coherent with the other silhouettes. Two examples of coherent and non-coherent silhouettes in the 2D case are shown in Fig.3. The optic ray pencils that are not coherent with the other silhouettes are shown by an arrow in Fig.3.b.

A first way of computing a coherence measure is the following: i) compute the reconstructed visual hull defined by the silhouettes, ii) project the reconstructed visual hull back into the cameras, and iii) compare the reconstructed visual hull silhouettes to the original ones. In the situation of ideal data, i.e., perfect segmentation and perfect projection matrices, the reconstructed visual hull silhouettes and the original silhouettes will be exactly the same (see Fig. 3.a). With real data, both the silhouettes and the projection matrices will not be perfect. As a consequence, the original silhouettes and the reconstructed visual hull silhouettes will not be the same, the reconstructed visual hull silhouettes being always contained in the original ones. This can be explained mathematically in the following way:

Let S_i be the i^{th} image silhouette, P_i the corresponding camera projection matrix and p a 2D point. We can define the cone C_i generated by the silhouette S_i as the set of lines l_{iv} which verifies

$$C_i \equiv \{l_{iv} = P_i^{-1} p, p \in S_i\}.$$



a) b)

Fig. 3. 2D examples of different silhouette coherences. The reconstructed visual hulls are the black polygons. a) Perfectly coherent silhouette set. b) Set of 3 silhouettes with low silhouette coherence. Incoherent optic ray pencils are indicated by the arrows.

The reconstructed visual hull V defined by the silhouette set S_i, $i = 1, \ldots, n$ can be written as the following cone intersection:

$$V = \bigcap_{i=1,\ldots,n} C_i.$$

The projected silhouette S_i^V is defined as the set of the 2D points p in the i^{th} image, whose optic ray intersects the reconstructed visual hull:

$$S_i^V = \{p, P_i^{-1} p \cap V \neq \emptyset\}.$$

By construction

$$S_i^V \subset S_i, \forall i.$$

If the silhouettes and the projection matrices are perfect, then $S_i^V = S_i \; \forall i$.

A first possible measure of similarity between the two silhouettes S_i and S_i^V of the i^{th} image can be the ratio of areas between the two silhouettes. But because the silhouettes area can be very large for big images (we use images of up to 4Kx4K pixels), for small differences between silhouettes, the dynamic range of this measure will be very small and not accurate enough for the applications we are considering. A better measure uses the ratio of the silhouette contour lengths rather than the ratio of the silhouette areas. As we will see in the implementation of the algorithm, there is also a performance reason for this choice.

Let ∂_i denote the contour of the original silhouette S_i, and ∂_i^V the contour of the reconstructed visual hull silhouette S_i^V. A measure $\mathcal{C}$ of coherence between these two silhouettes can be defined as the ratio between the length of their common contours $\partial_i^V \cap \partial_i$ and the total length of ∂_i:

$$\mathcal{C}(S_i, S_i^V) = \frac{\int (\partial_i^V \cap \partial_i)}{\int \partial_i} \in [0, 1].$$

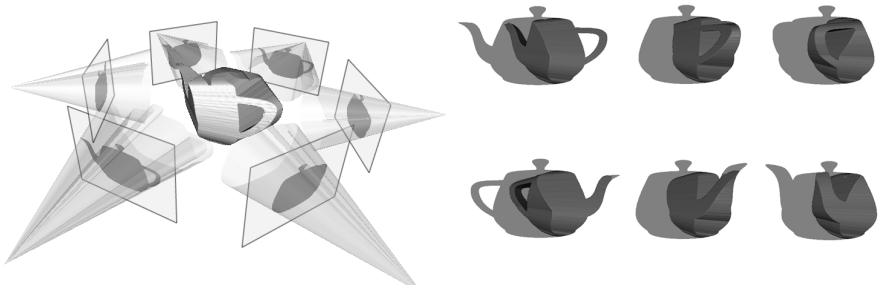


Fig. 4. Reconstructed visual hull of the teapot sequence with bad extrinsic parameters: total coherence of 39.11%. Left: reconstructed visual hull by cone intersection. Right: the 6 original silhouettes superposed with the projections of the reconstructed visual hull.

This measure evaluates the *coherence* between the silhouette S_i and all the other silhouettes $S_{j,j\neq i}$ that contributed to the reconstructed visual hull. To compute the total coherence between all the silhouettes, we simply compute the mean coherence of each silhouette with the $n-1$ other silhouettes:

$$\mathcal{C}(S_1,\ldots,S_n) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{C}(S_i, S_i^V)$$

To be able to efficiently exploit this coherence measure we need first to know its application limits. As we stated above, if we have perfect silhouettes and perfect camera matrices, then $\mathcal{C}(S_1,\ldots,S_n) = 1$. This never happens in practice. Let us assume that the silhouettes are perfectly segmented. We can maximize the silhouette coherence by adjusting the camera parameters in order to reduce mismatches between silhouettes. This can be seen as a kind of bundle adjustment of the silhouettes. But maximizing coherence between silhouettes does not mean finding the right camera parameters [5]. This depends on:

- **The object shape**. The worst case corresponds to the sphere for which there is no unique solution to the problem of silhouette coherence maximization because of the sphere symmetry. Thus, in general, we can not guarantee the uniqueness of the solution.
- **The number of silhouettes**. If we use only a small number of silhouettes, then the coherence criterion is less accurate and may be maximized for a large set of solutions. However, if we take a sufficient number of pictures, real objects are asymmetric enough to guarantee a unique solution as will be shown in the practical examples.
- **The interdependence between parameters of the camera**. Some parameters can affect the coherence criterion in a similar way and thus we are not able to distinguish between them during the coherence maximization.

So the coherence criterion is not the ultimate criterion for recovering all the parameters in a global optimization, but it can work quite well for some particular scenarios

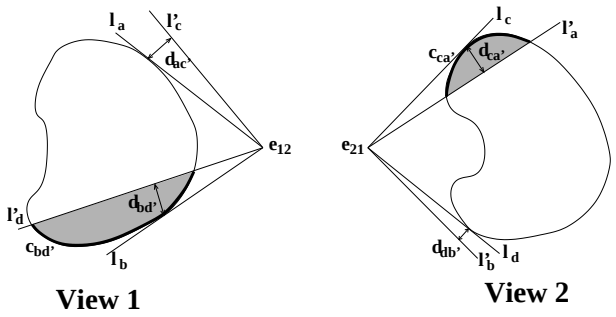


Fig. 5. Epipolar tangency and silhouette coherence criteria for a pair of silhouettes.

where the number of parameters to recover is not very high and silhouettes provide enough information to optimize them. In the case of a sequence of cameras with circular motion and constant intrinsic parameters, [21, 13] demonstrate the feasibility of recovering the motion using epipolar tangencies. We show below that the proposed silhouette coherence criterion is a more general approach that includes the criterion of epipolar tangencies. For a given pair of views, as shown in Fig.5, the epipolar tangency approach minimizes the distance between epipolar tangents of one view (l_a and l_b in view 1, l_c and l_d in view 2) and the transformed epipolar tangents of the other view (l'_c and l'_d in view 1, l'_a and l'_b in view 2). That is, it minimizes $\mathcal{C}_{et} = d_{ac'} + d_{bd'} + d_{ca'} + d_{db'}$. For the same pair of silhouettes, the coherence criterion will minimize the length of the contours $c_{ca'}$ and $d_{bd'}$. So we can see that, except for pathological configurations, both criteria try to minimize the sectors defined by the epipolar tangents in one view and their corresponding epipolar tangents on the other view. This implies that if we maximize our coherence criterion on a set of silhouettes by pairs, we get the same behavior as [13], and thus we are able to recover the camera motion using silhouette coherence. When using the proposed silhouette coherence, silhouettes are not just taken by pairs but all at the same time. This makes that the information we exploit is not only on the epipolar tangencies but all over the silhouette. Although we have the theoretical possibility of recovering some parameters, it is well known that the circular motion is a critical motion for self-calibration [10]. The question is which parameters can be theoretically recovered and if in practice they affect the coherence criterion enough to be retrieved by maximization. Recovering the camera principal point (u_0, v_0) and translation vector (t_x, t_y, t_z) under circular motion is a very difficult task. In the projection of a 3D point the translation introduces in the projection equation a pixel offset proportional to the focal length f and inversely proportional to the distance of the 3D point: $\Delta x(z) = f t_x/(z + t_z)$, $\Delta y(z) = f t_y/(z + t_z)$. If, as it is in practice, the depth variations of the 3D object along the apparent contours are small compared to their distance to the camera, $(\Delta x(z), \Delta y(z))$ are almost constant and behave in the same way as (u_0, v_0). The focal length is a different case. Its main problem is that very large variations of the focal length may produce only small variations of the silhouette coherence. Even with an infinite focal length (orthographic projection) high values of coherence may still be obtained. However, focal length can still be recovered together with the motion by always initializing it with a lower bound of its expected value. We present

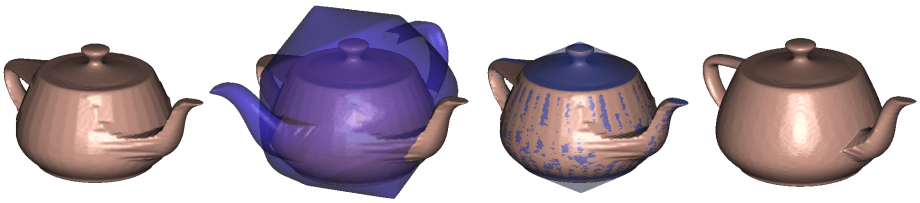
in sections 5 and 6 two useful applications of the silhouette coherence: the registration of two different sets of silhouettes of a same object under circular motion (7 parameters: rotation+translation+scale) and a partial calibration of a circular motion sequence (4 parameters: motion + focal length) that allows in practice the auto-calibration of the sequence in order to use 3D object reconstruction techniques without the use of a calibration pattern.

4 Coherence Criterion Implementation

The silhouette coherence criterion $\mathcal{C}$ being defined, a first implementation, and probably the simplest, would be: 1) for a given set of cameras and silhouettes, compute the reconstructed 3D visual hull by using any of the multiple existing techniques, 2) project the reconstructed visual hull into the cameras, 3) compute the coherence criterion. This approach has two drawbacks: computation time and volume sampling. The former may be a problem since it may take several minutes to compute a 3D visual hull, and the latter because constructing a 3D visual hull usually needs a discrete 3D representation, such as a volume grid or an octree. If we want a highly accurate visual hull, we need a high resolution 3D model of the visual hull, which is computationally very expensive. In addition, we are not interested in a 3D representation in itself but in comparing 2D views of it with the original silhouettes. Therefore, it seems a waste of time to completely reconstruct the visual hull when only some views of it are required. The imaged-based visual hull (IBVH) technique [20] does not compute a 3D representation of the reconstructed visual hull but only 2D views of it. The key idea is as follows: for any given camera, a depth map of the visual hull is obtained by a ray-casting approach. For each pixel we compute the intersection between its optic ray and the visual hull. According to the definition of the visual hull, this is equivalent to (see Fig.2): 1) projecting the optic ray into each silhouette, 2) computing the 2D intersection intervals between the projected ray and each silhouette, 3) back projecting all the 2D intervals onto the original 3D optic ray, 4) computing the intersection on the 3D optic ray of all the intervals of all the silhouettes. For each pixel of the depth map, we have a set of remaining depth intervals, possibly empty, which represent the intersection between its optic ray and the implicit visual hull.

To compute the 2D intersection between the projected ray and a silhouette, two approaches can be used: a Bresenham-like intersection test or an interval intersection algorithm. Even if the Bresenham intersection is a full integer algorithm, the images are too big and the precision of the intersecting point is too low for it to be a valid choice. With the approach of interval intersection, the silhouette contours are coded as a closed sequence of segments, defined by two consecutive pixels along the silhouette contour. For a given projected ray, we compute the intersection of the half line with the sequence of segments, which gives us the 2D intervals directly. This procedure can be done in a very efficient way as described in [20].

Concerning the coherence algorithm, we can approximate the contours of the silhouette by the discrete list of the pixels that define them, the contour length simply becoming the cardinal of this list. So we are interested in computing the image-based visual hull only on the contour pixels of the silhouette. Furthermore, we do not need all



a) b) c) d)

Fig. 6. Synthetic teapot of 10 cm bounding size. (a) First reconstructed visual hull; self coherence of 100%. (b) Unregistered second reconstructed visual hull; self coherence of 100%; mutual coherence of 51.71%. (c) Reconstructed visual hulls after registration; mutual coherence of 100%. (d) Resulting reconstructed visual hull after registration.

the intervals for every pixel, we only want to know if the intersection of all the intervals is empty or not, i.e., if the ray intersects the reconstructed visual hull or not. The pseudo code of the coherence algorithm $\mathcal{C}(S_i, S_i^V)$ between a silhouette and its corresponding reconstructed visual hull silhouette is as follows:

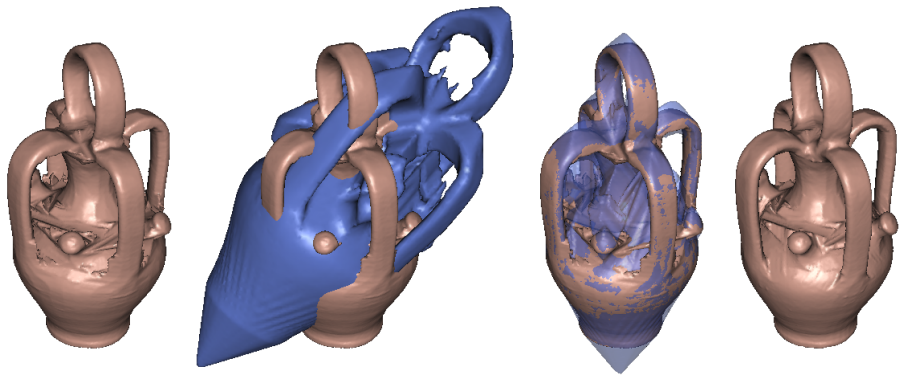
```
FLOAT Coherence (SILHOUETTE silRef,SILLIST silList)
INTEGER emptyPixels = 0
INTEGER totalPixels = 0
For each PIXEL p in Contours(silRef)
   INTERVALS intervals = [0,inf]
   VEC3D ray = InverseProjection(silRef,p)
   For each SILHOUETTE sil in silList
      VEC2D epipole = Projection(sil,CameraOrigin(silRef))
      VEC2D direction = Projection(sil,ray)
      INTERVALS int2D = Intersect(sil,epipole,direction)
      INTERVALS int3D = InverseProjection(sil,int2D)
      intervals = Intersect(intervals,int3D)
      If intervals = void, then emptyPixels++; break
   end
   totalPixels++
end
return (totalPixels-emptyPixels)/totalPixels
```

5 Registration of two Calibrated Sequences

We consider the silhouettes of two different rotation sequences $\mathcal{S}^1$ and $\mathcal{S}^2$ of the same object, each sequence being calibrated independently by a pattern-based calibration method [15]. We would like to register both sequences in order to reconstruct the object using all the views available. The two sequences are related by: a rotation (Euler angles $[\alpha, \beta, \gamma]$), a translation ($[t_x, t_y, t_z]$), and a scaling factor s. This makes 7 parameters to optimize $v = [\alpha, \beta, \gamma, t_x, t_y, t_z, s]$. The mutual coherence function can be defined as:

$$\mathcal{C}(\mathcal{S}^1, \mathcal{S}^2, v) = \frac{1}{2}(\frac{1}{n}\sum_{i=1}^{n}\mathcal{C}(S_i^1, \mathcal{S}^2, v) + \frac{1}{m}\sum_{j=1}^{m}\mathcal{C}(S_j^2, \mathcal{S}^1, v)).$$

A non-linear optimization algorithm has been used to maximize the coherence function. Gradient, simplex and Powell's methods have been tested [25]. Powell's method



a) b) c) d)

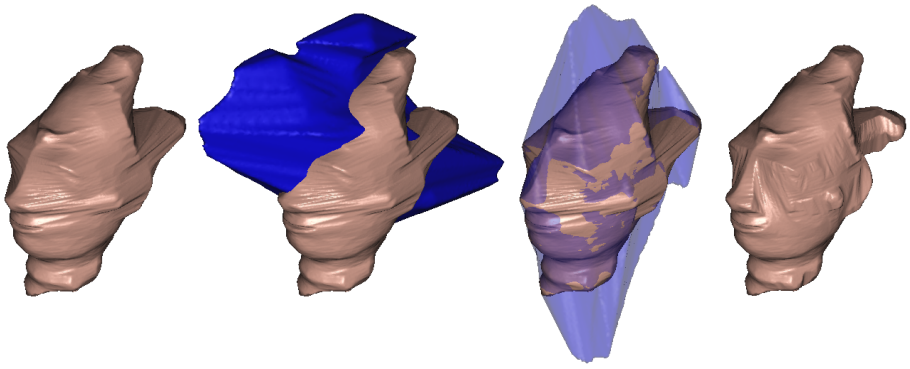
Fig. 7. Pitcher of 30 cm bounding size. (a) First reconstructed visual hull; self coherence of 98.19%. (b) Unregistered second reconstructed visual hull; self coherence of 90.99%; mutual coherence of 54.46%. (c) Reconstructed visual hulls after registration; mutual coherence of 95.66%. (d) Resulting reconstructed visual hull after registration.

leads to better convergence results. Concerning the initialization, the translation is simply initialized by matching the bounding box centers defined by the two silhouette sequences. In practice the initial rotation does not need to be very accurate and an exhaustive sampling of the Euler angles with a grid step of 20 degrees suffices.

We present here the results for three different objects: the digital teapot, which we use to evaluate the algorithm (Fig.6), and two real objects (Fig.7 and Fig.8). For all the objects we dispose of two sequences of 36 views.

The teapot object allows us to measure the theoretical precision of the coherence criterion. The coherence of each independent sequence is 100% (Fig.6.a and Fig.6.b): segmentation and camera parameters being perfect, the original silhouettes and the reconstructed visual hull ones are identical. However, mutual coherence at the initial position is much lower: 51.71% (Fig. 6.b). After convergence, the mutual coherence is completely maximized. The algorithm precision for the 3 translation parameters is about 1/1000 of the bounding box size (images of 1024x768), for the rotation about 6/100 degree and for the scale factor better than 10^{-4} (see table 1).

For the real objects (Fig. 7 and Fig. 8), we have used images of 2008x3040 pixels and the silhouettes have been binarized by an automatic color-segmentation technique. In the pitcher sequence we dispose of a complete calibration of the system, obtained using the calibration technique described in [15]. Since the object did not move between the acquisition of both sequences, we can recover the transformation between the sequences and compare it with the one obtained by maximizing the coherence criterion. We observe the difference between the self coherences obtained for the two sequences: 98.19% and 90.99% respectively (see Fig. 7.a and Fig. 7.b). The lower score of the second sequence is due to the worse quality of its extracted silhouettes. Despite these segmentation errors, the precision obtained after optimization is rather good (see table 1). As we can appreciate in figure 7.b, the disparity of the reconstructed visual hulls makes the use of other registration methods such as the ICP [2] very difficult.



a) b) c) d)

Fig. 8. Statuette (a) First sequence visual hull; self coherence of 82.63%. (b) Unregistered second sequence visual hull; self coherence of 84.47%; mutual coherence of 41.27%. (c) Visual hulls after registration; mutual coherence of 93.67%. (d) Resulting reconstructed visual hull after registration.

In the second experimental sequence, the statuette has been manually moved between both acquisitions, so there is no way of finding out the transformation with a classical calibration technique. Due to automatic segmentation, silhouettes are not perfect. This is reflected by the low self coherence values of 82.63% and 84.47% respectively(Fig.8.a, Fig.8.b).

It may seem strange that, after convergence, the mutual coherence is higher than the self coherences in the two real examples. This is simply due to the fact that the two measures do not compare the same silhouettes.

Once we have registered the two sequences, we can compute the visual hull with the silhouettes of both sequences, which improves the quality of the reconstructed models (see Fig. 6.d, Fig. 7.d and Fig. 8.d).

	rotation (degrees)			translation (mm)			scale
Teapot	α	β	γ	t_x	t_y	t_z	s
initial	180.000	90.000	180.000	8.0000	0.0000	12.0000	1.20000
real	150.000	0.000	180.000	10.0000	-10.0000	10.0000	1.00000
recovered	150.063	0.010	179.998	9.9986	-10.0010	10.0012	0.99993
error	0.063	0.010	0.002	0.0014	0.0010	0.0012	0.00007
Pitcher							
initial	103.000	6.000	8.000	-10.000	460.000	650.000	1.20000
calibrated	63.776	-4.605	-2.497	-36.680	478.329	656.895	1.00000
recovered	63.813	-4.560	-2.487	-36.492	478.125	656.915	1.00014
error	0.037	0.045	0.009	0.188	0.204	0.020	0.00014

Table 1. Registration results for the teapot and pitcher sequences.

6 Motion and Focal Recovery from a Single Rotation Sequence

In this scenario, we seek to recover the circular motion parameters and the focal length of a single rotation sequence. For a single axis rotation sequence (see [11] for a detailed explanation), the motion parameters are reduced only to the rotation axis direction, the angles of rotation between two consecutive views and the camera translation direction from the axis origin. If we suppose the angle steps known, we only have 2 parameters to code the rotation axis. The camera translation direction can be coded only with one parameter (see [11]).

The algorithm has been tested with the synthetic teapot sequence and the real pitcher sequence. In the teapot sequence the rotation axis direction has been modified by 30 degrees, the camera translation direction by 1.4 degrees and the focal length by 33%. The camera translation angle is a very sensitive parameter in the image formation. A modification of 1.4 degrees gives, already, a very low coherence of 22.82%, worse than the example shown in Fig. 4. After optimization, we get a residual error of $4.8\ 10^{-4}$ degrees on the translation, 0.03 degrees on the rotation axis, and 71 pixels ($< 1\%$) on the focal length (see table 2), but the coherence is already perfect.

The real sequence corresponds to the first sequence of the pitcher with bad parameters. The coherence before optimization is of 62.95%. After optimization, the coherence is of 98.43%. We can see in table 2 the results of the motion and focal estimation.

Teapot	rotation (degrees)		translation (degrees)	focal (pixels)
	θ	ϕ	α	f
initial	106.000	110.000	1.4	6000
real	86.626	90.576	0.0	9000
recovered	86.611	90.554	$4.8\ 10^{-4}$	8929
error	0.015	0.022	$4.8\ 10^{-4}$	71

Pitcher	rotation (degrees)		translation (degrees)	focal (pixels)
	θ	ϕ	α	f
initial	90.000	90.000	0.000	2000
calibrated	99.671	90.343	0.427	6606
recovered	99.774	90.338	0.429	6573
error	0.103	0.005	0.002	33

Table 2. Motion and focal results for the teapot and pitcher sequence.

Concerning the non-recovered principal point (u_0, v_0), its value is not critical and choosing the image center as the principal point seems to be sufficient for 3D modeling. This is due to the fact that the error in the principal point position is compensated by the translation vector optimization. The resulting distortion on the 3D object reconstruction due to this weak calibration is indeed very low when the width of the object remains small compared to its distance from the camera.

7 Conclusions

We have presented a new approach to camera parameters estimation based on the similarity between a set of silhouettes and the silhouettes of their visual hull. This approach has been successfully tested for different estimation problems such as independent sequence registration or motion and focal recovery from a single rotation sequence. The high precision of the calibration results are due to the use of thousands of pixels in the

computation, whereas classic methods use a few hundred at most. Its main drawback is the need for an accurate segmentation of the object, which sometimes is not an easy task. We will further investigate the robustness of this technique for more complicated scenarios.

References

1. B.G. Baumgart. *Geometric Modelling for Computer Vision*. PhD thesis, Standford University, Palo Alto, 1974.
2. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
3. A. Bottino and A. Laurentini. Introducing a new problem: Shape-from-silhouette when the relative positions of the viewpoints is unknown. *Accepted for publication on IEEE Transactions on Pattern Analysis and Machine Intelligence*.
4. E. Boyer. Object models from contour sequences. In *Proc. ECCV*, pages 109–118, 1996.
5. K.M. Cheung. *Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering*. PhD thesis, Carnegie Mellon University, 2003.
6. K.M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 375–382, 2003.
7. C.H. Chien and J.K. Aggarwal. Computation of volume/surface octrees from contours and silhouettes of multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 250–255, 1986.
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. C. Hernández Esteban and F. Schmitt. A snake approach for high quality image-based 3d object modeling. In *to appear in Variational, Geometric and Level Set Methods in Computer Vision*, October 2003.
10. O. Faugeras and Q.T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.
11. A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open images sequences. In *ECCV*, pages 311–326, 1998.
12. G. Jiang, H.T. Tsui, L. Quan, and A. Zisserman. Single axis geometry by fitting conics. In *ECCV*, volume 1, pages 537–550, 2002.
13. K.-Y. K. and R. Cipolla. Structure and motion from silhouettes. In *8th IEEE International Conference on Computer Vision*, volume II, pages 217–222, Vancouver, Canada, July 2001.
14. A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Trans. on PAMI*, 16(2), 1994.
15. J. M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *Proc. ECCV*, volume 1, pages 158–174, 1998. Germany.
16. H. Lensch, W. Heidrich, and H. P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Journal of Graphical Models*, pages 245–262, July 2001.
17. M. Li, M. Magnor, and H.P. Seidel. Improved hardware-accelerated visual hull rendering. In *to appear in Vision, Modeling, and Visualization*, November 2003.
18. Y. Matsumoto, K. Fujimura, and T. Kitamura. Shape-from-silhouette/stereo and its application to 3-d digitizer. In *Proceedings of Discrete Geometry for Computing Imagery*, pages 177–190, 1999.

19. K. Matsushita and T. Kanedo. Efficient and handy texture mapping on 3d surfaces. *Computer Graphics Forum*, 18:349–358, 1999.
20. W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. *SIGGRAPH 2000*, pages 369–374, July 2000.
21. P. R. S. Mendonça, K.-Y. K. Wong, and R. Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):604–616, June 2001.
22. P. J. Neugebauer and K. Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18:245–256, 1999.
23. S.Y. Park and M. Subbarao. Pose estimation of two-pose 3d models using the base tangent plane and stability constraints. In *VMV 2002*, pages 379–386, 2002.
24. J. Porrill and S. B. Pollard. Curve matching and stereo calibration. *Image and Vision Computing*, 9(1):45–50, February 1991.
25. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C (The Art of Scientific Computing)*. Cambridge University Press, 1992.
26. J. H. Rieger. Three dimensional motion from fixed points of a deforming profile curve. *Optics Letters*, 11(3):123–125, March 1986.
27. S. Sullivan and J. Ponce. Automatic model construction, pose estimation, and object recognition from photographs using triangular splines. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(10):1091–1096, 1998.