Largest Silhouette-Equivalent Volume for 3D Shapes Modeling without Ghost Object

Brice Michoud, Saïda Bouakaz, Erwan Guillou, Hector Briceño

Largest Silhouette-Equivalent Volume for 3D Shapes Modeling without Ghost Object

Michoud Brice, Bouakaz Saïda, Guillou Erwan and Briceño Hector

LIRIS - UMR 5205 CNRS / Université Lyon 1 - France

Abstract. In this paper, we investigate a practical framework to compute a 3D shape estimation of multiple objects in real-time from silhouettes in multi-view environments. A popular method called Shape From Silhouette (SFS), computes a 3D shape estimation from binary silhouette masks. This method has several limitations: The acquisition space is limited to the intersection of the camera viewing frusta ; SFS methods reconstruct some ghost objects which do not contain real objects, especially when there are multiple real objects in the scene.
In this paper we propose two contributions to overcome these limitations. First, using a new formulation of SFS approach, our system reconstructs objects with no constraints on camera placement and their visibility. Second, a new theoretical approach identifies and removes ghost objects. The reconstructed shapes are more accurate than current silhouette-based approaches. Reconstructed parts are guaranteed to contain real objects. Finally, we present a real-time system that captures multiple and complex objects moving through many camera frusta to demonstrate the application and robustness of our method.

1 Introduction

Capturing dynamic 3D scenes in real-time allows many applications like gesture recognition, crowd surveillance, behavior analysis, free-viewpoint 3D video, new human-computer interfaces, etc. We wish to perform real-time markerless 3D motion capture of multiple persons. To achieve this, we need a 3D representation of the persons. Shape From Silhouette (SFS) is a popular method that estimates in real-time 3D shapes from silhouette images. Results of this method depend on the number of cameras, their position and can create ghost objects (reconstructed objects where no real objects exist). In this paper, we propose two contributions to make SFS more practical with our context.

SFS methods compute the visual hull of an object relative to a viewing region. The visual hull is defined as the intersection of the silhouette's cones from camera views, which capture all geometric information given by the image silhouettes [1]. A silhouette's cone is given by the back projection in 3D space of the silhouette contours through the associated camera's center. The visual hull is the maximum volume that result in the same silhouettes of real objects from the given viewpoints. The Visual Hull is said Silhouette-Equivalent [2]. However, the volumes produced from a SFS reconstruction suffer from many drawbacks:

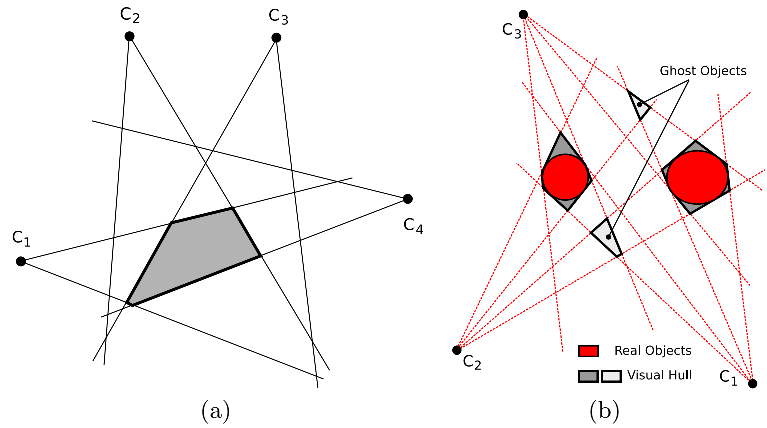


Fig. 1. The main drawbacks of SFS algorithms: (a) the acquisition space (in gray) is limited to the strict intersection of the camera's viewing frustum. (b) shows a typical SFS configuration where ghost objects appear.

Camera Placement: the objects that can be captured must lie in the strict intersection of the field of views of the cameras. Objects that are partially hidden in a certain view will be cut; the capture volume decreases as the number of cameras increases (see Fig. 1(a)).

Ghost Objects: In some cases of visual ambiguities, SFS can reconstruct empty regions as objects which are consistent with silhouettes (see Fig. 1(b)). Ghost objects can greatly interfer with many applications using SFS, especially when these are based on shape analysis, for example markerless motion capture, crowd surveillance, free-viewpoint rendering, etc.

Contribution This paper describes two contributions to overcome these limitations and whose computation is still achievable in real-time.

The key idea to address the first limitation is to use a subset of cameras when deciding if a 3D point represents a foreground object as opposed to SFS which uses the complete set. To circumvent the second limitation, we propose a formalism to describe and remove ghost objects. The key idea is that if a pixel inside a silhouette is derived from exactly one 3D connex part then that connex part contains a real object.

2 Previous Work

There are mainly two ways that SFS algorithms estimate the shape of objects: Surface-based approaches and Volumetric-based approaches.

Surface-based approaches compute the intersection of silhouettes' cones. First, silhouettes are converted into polygons. Each edge in the silhouette is extruded away from the camera to form a 3D polygon. The intersection of these extrusions are assembled to form an estimation of the polyhedral shape (see [3, 4]).

Volumetric approaches usually estimate shape by processing a set of voxels [5, 6]. The object's acquisition area is split up into a 3D grid of voxels (volume elements). Each voxel remains part of the estimated shape if its projection in all images lies in all silhouettes.Volumetric approaches are well adapted for real-time shape estimation and robustness to noisy silhouettes.

From the methods that compute a 3D model, we note that the classical SFS algorithms require that all viewing fustra intersect.This intersection describes the capture volume of the system (see Fig. 1(a)). If parts of the subject leave this capture volume they will not be reconstructed. One solution to increase the capture volume is to increase the number of cameras. Additional cameras would have to be placed farther away to increase the field of view, and would have to have a higher resolution. To overcome the limitations on camera placement, Franco and Boyer [7] use a probabilistic 3D representation of scene contents and an occupancy grid. This method can reconstruct part of the objects seen by a subset of cameras and is resistant to badly segmented silhouettes. The drawback is that the Bayesian formulation is time consuming (more than 10 seconds to process one frame) thus unsuitable for the proposed applications. Michoud *et al.* [6] propose a deterministic extension of the classical SFS to allow parts of the object to exit the intersection of all cones. Their implementation works in real-time. The first limitation of their approach is that if one cameras sees nothing, no object is reconstructed. The second limitation is that if there are multiple objects in the scene, the proposed method removes non-ghost objects that are outside of the strict intersection of cameras frusta.

The problem of removing ghost objects (see Fig. 1(b)) from a SFS reconstruction has not been adequately addressed in previous research. One solution to decrease the reconstruction of ghost objects is to increase the number of cameras, nevertheless artifacts still occurs. In [8], Yang *et al.* propose heuristics on size and use temporal filtering to remove ghost objects. This approach can be unreliable with dynamic scenes with small and large objects. Miller and Hilton define in [9] the concept of Safe Hulls. This is a per-point approach to remove ghost parts. Their algorithm is not guaranteed to produce a completely correct result, the right picture of the Fig. 3 shows an example where this approach fails. However this approach is fully automatic and does not require any additional information such as object correspondence. In [10], Bogomjakov and Gotsman present an algorithm that does remove ghost objects, but it requires additional information like Depth-maps or correspondence between the silhouettes. Depth-maps come from particular sensors. Silhouette correspondence impose that each view see each object and labeling processing can be unstable with similar objects. Thus, their approach is unsuitable for our applications. Other approaches [11, 12] can obtain reconstructions without ghost objects using additional information like color cues. Unfortunately, these methods rely on computationally intensive statistics, and sometimes need pixel matching and correspondences, which are expensive operations and are far from real-time, thus unsuitable for our needs.

In this paper, we propose a framework reconstruct multiple 3D objects in real-time from multiple views. Using a new formulation of SFS approach, our system is able to reconstruct object parts with no constraints on camera placement and visibility. Our method identifies and removes ghost objects. Reconstructed parts are guaranteed to contain real objects. The reconstructed shapes are more accurate than current silhouette-based approaches.

This paper is organized as follows. In the next section, we present the Largest Silhouette-Equivalent Volume concept, which removes the constraints on camera placement. Section 4 presents our approach to detect and remove ghost objects. Section 5 demonstrates our algorithm under real scenarios. We summarize our contribution and and give the future perspectives for our work in Section 6.

3 Largest Silhouette-Equivalent Volume

In this section we describe a novel approach which extends the acquisition space by relaxing the camera placement constraints.

SFS algorithms deduce the shape of an object from its silhouettes seen by multiple cameras. It is a concept based on the visual hull of objects. The visual hull (VH) is defined as **the maximum volume consistent with the observed silhouettes**. A 3D point will be classified as "belonging to real object" or occupied if all of its projections into each camera lie in silhouettes.

This approach has many limitations and generates artifacts. One of the most important limitation is related to the acquisition space or capture volume. The objects to be captured must lie in the strict intersection of the field of views of all cameras. Objects that leave this space will be cut. The placement and the video resolution of the cameras will determine the granularity and precision of what can be reconstructed. The capture volume decreases as the number of cameras increases.

Our approach decides that a 3D point is inside the VH from the subset of cameras which can see this 3D point, not from the complete set (as supposed in SFS).

Notations

For a better comprehension, we introduce the following notations:

- C_i is one of the n cameras with $i \in [1, \cdots, n]$,
- π_i is the image plane of C_i,
- I_i is the image seen by C_i,
- S_i is a point subset of I_i, which are inside the silhouette of the interest objects,
- $Proj_{\pi_i}(x)$ is the projection of the point x on the image plane π_i.
- To be concise, we adopt the same notation for the projection of set E on the image plane π_i $\quad Proj_{\pi_i}(E)$.

Let VS_x the subset of camera indices of the silhouettes and VI_x the subset of camera indices of the images, where the projection of point x lies in, then:

$$VS_x = \{i \in [1, \cdots, n], Proj_{\pi_i}(x) \in S_i\} \tag{1}$$

$$VI_x = \{i \in [1, \cdots, n], Proj_{\pi_i}(x) \in I_i\} \tag{2}$$

SFS computes the set of 3D points whose projections lie inside all the silhouettes. According to the above notation, the reconstruction based on SFS using n cameras can be written as:

$$SFS = \{x \in R^3, VS_x = VI_x, Card(VS_x) = n\} \tag{3}$$

The SFS reconstruction is consistent with the silhouettes if all scene's objects are inside the strict intersection of the field of views of all the cameras. If a 3D point is out of sight of even one camera, SFS cannot accept it.

However, if we want to extend the acquisition space, a 3D point x could be visible by less than n cameras.

The visual hull is the maximum volume that result in the same silhouettes of real objects from the given viewpoints. To reconstruct 3D points in volumes space seen by less than n cameras we introduce *Largest Silhouette-Equivalent Volume* $LSEV$, an extension of SFS, defined by:

$$LSEV = \{x \in R^3, VS_x = VI_x\} \tag{4}$$

The volume estimated by $LSEV$ is guaranteed to contain all the real objects. $LSEV$ can create ghost object, especially in space seen by a small number of cameras. The number of ghost object decreases as the camera number increases. A simple solution to reduce ghost object reconstruction, is to suppose that all real objects are seen by more than n_{min} cameras.

Let $LSEV_{n_{min}}$ be the *largest silhouette-equivalent volume* of the real objects seen by at least n_{min} cameras:

$$LSEV_{n_{min}} = \{x \in LSEV, Card(VI_x) \geq n_{min}\} \tag{5}$$

where n_{min} is a threshold that represents the minimum number of cameras. Points in space which are not potentially visible by at least n_{min} cameras, will never be reconstructed. The acquisition volume can then be controlled by this parameter at the expense of accuracy and resolution. The value n_{min} is important to prevent some infinite parts in the VH. Usually $n_{min} \geq 2$. The Figure 2 underlines reconstruction of the VH using our $LSEV_{n_{min}}$ approach.

This contribution extends the volume where 3D points can be tested, then reconstructed. Our formulation is directly based one the definition of the volumetric visual hull, proposed by Aldo Laurentini in [2].

In this section we described a novel approach to compute the Visual Hull of multiple objects in the scene, with only one constraint to the camera placement: objects will be reconstructed if they are inside at least n_{min} fields of view of

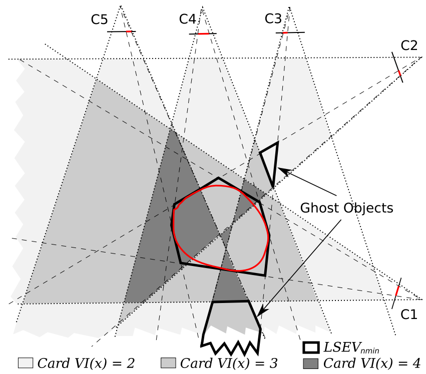


Fig. 2. 2D representation of a $LSEV_{n_{min}}$ reconstruction of one object (circle in red) using $n_{min} = 2$ and $n = 5$. Please note that the intersection of all camera frustum views is empty, and that usual SFS is unable to reconstruct the VH. The acquisition space is defined as the union of space where $Card(VI_x) \geq n_{min}$ (in gray). With this configuration $LSEV_{n_{min}}$ creates some ghost objects.

.

cameras. Our reconstruction is always silhouette-equivalent if all objects are seen by at least n_{min} cameras. Because we aim to track the motion of the filmed persons, the $LSEV$ approach guarantees that the 3D reconstruction includes real object shapes, even when objects are visible only in a subset of all the cameras. Of course when object part is visible in a subset of all the cameras, this part is reconstructed with a reduced quality compared with SFS having all the cameras seeing this object part.

Usual SFS approaches suffer from ghost objects reconstruction. As we relax the camera placement constraint, $LSEV$ can create more ghost object than SFS (see Fig.2). In the next section we propose an approach which automatically removes all ghost objects of the VH. Our solution does not require any additional information such as the number of real objects or correspondence between silhouettes. In the rest of the paper we suppose that $LSEV$ has been used for visual hull estimation.

4 Ghost Object Removal

We are interested in removing ghost objects, this section attempts to give a formal definition to characterize a ghost object and underlines several qualitative and quantitative properties.

Fairly simple and straightforward, if there is a single object in the scene, the silhouette's cones, intersect themselves exactly over the object (e.g. there is no possible ambiguity). However, if there is more than one object in the scene, the regions of intersection of vision cones generated by the silhouettes can admit component outside the boxes encompassing objects; this intersection includes empty boxes. We call these regions ghost objects. In the following we describe an approach which guarantees that kept parts, are not ghost objects (*i.e.* contain real objects).

We recall that the visual hull (VH) is the largest volume to be consistent with silhouettes, then

$$\bigcup_i Proj_{\pi_i}(VH) = \bigcup_i S_i \tag{6}$$

Our goal is to compute the subset of the connex components (connected components) of VH that contain real objects. In the following we note CC_j one of the connex component of the VH :

$$\bigcup_{j=1}^{m} CC_j = VH \tag{7}$$

with m the number of connex components in VH.

Definition 1. *A connex component CC_j of VH is a ghost object if CC_j does not contain a real object.*

Proposition 1. *Let $p \in S_i$ be a pixel belonging to the silhouette S_i*
If

$$\textit{there exists only one } CC_l \subset VH \textit{ with } p \in Proj_{\pi_i}(CC_l)$$

Then

$$CC_l \textit{ is not a ghost object.}$$

Proof. According to definition of the silhouettes, $\forall p \in S_i$ there exists a least a real object Obj such that $p \in Proj_{\pi_i}(Obj)$.
If there exists a unique connex component $CC_l \subset VH$, such that $p \in Proj_{\pi_i}(CC_l)$ then there exists real object Obj:

$$Obj \subset CC_l \text{ and } \exists P \in Obj \ , \ Proj_{\pi_i}(P) = p$$

Because the uniqueness of connex component $CC_l \subset VH$ whose projection contains pixel p it becomes clear that CC_l contains at least the object Obj.

To remove ghost objects, our algorithm checks the connex component of the VH which satisfies the Proposition 1.

We introduce the notion of Real Shape Hulls (RSH) as the union of these connex components:

$$RSH = \bigcup CC_l \subset VH, \exists p \in Proj_{\pi_i}(CC_l), p \notin Proj_{\pi_i}(CC_k) \tag{8}$$

with $p \in S_i$, $CC_k \subset VH$ and $\forall CC_k \neq CC_l$.

One important property of RSH, is that it contains no ghost object. Furthermore RSH is easy to implement with new GPU capabilities, thus obtaining a real-time implementation. In our implementation of RSH, we project each connex component of the VH into all the image planes (using the GPU) which create

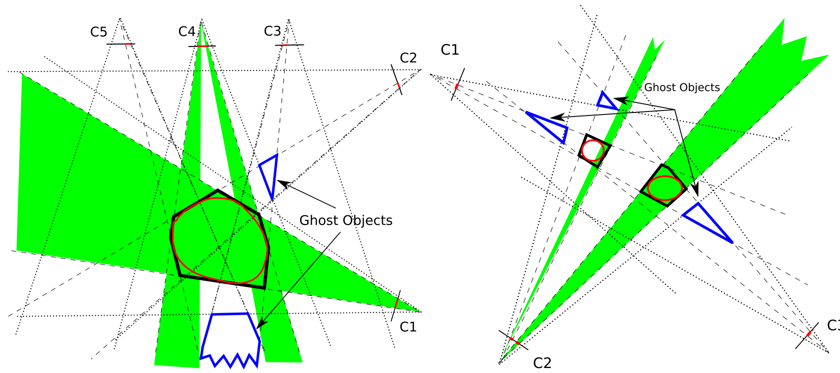


Fig. 3. 2D representations of RSH results. The real objects are represented in red. (a) Configuration presented in Fig.2. The black connected component is kept because it satisfies the Proposition 1. Green parts indicate where connected components will be accepted (exactly one object is projected in these regions). Ghost objects (Blue) components are rejected. (b) RSH keeps the black connected components; it conserves connected component of the VH which contains real objects. With this configuration, the Safe Hulls reconstruction proposed by [9] results as the intersection of VH connected components, and green parts. Their method fails to keep all real parts.

.

images Ip_i. For each pixel in all the images Ip_i, if there exists only one Connex component of the VH which has marked this pixel, the Connex component is kept as containing real object.

With our approach, we guarantee that real objects which are inside connex components of VH satisfying the Proposition 1, are contained in RSH. In other words, real objects which are inside connex components of VH are contained in RSH, if there is no other connex component of VH which completely occlude these connex components in all views. With multiple objects in the scene with similar sizes and different shapes, it is unlikely that RSH will miss any real object.

As a limitation, RSH does not guarantee that all the real objects are represented. This limitation comes from the fact that it exists a non-finite number of configurations of real objects that produce the same silhouettes, thus the same Visual Hull (see the paper [2] for a precise study). Then the goal of removing all ghost objects without removing any real objects, is not attainable without strict hypothesis on real object configuration, placement, number, etc. Our algorithm will not reconstruct a real object if it lies *completely inside* the silhouettes of other objects; in practice, this is rarely or never the case. In [9] Miller and Hilton method sounds similar to our work. Our approach can be rewritten in the following way: A 3D connex component of VH is kept if it exists a ray coming from a silhouette's pixel in a least one camera, which intersect this 3D connex component only. In [9] a 1D connex component of VH is kept if it exists a ray coming from a silhouette's pixel in a least one camera, which intersect this 1D (interval) connex component only. RSH removes real parts from the VH only when real objects are **totally occluded** by others object in **all the views at the same time**. This is a slight assumption that it exists at least one pixel in all the views, that is inside the projection of only one real object. The method of Miller and Hilton removes real parts from the VH when real object is **partially occluded** by others object in all the views. This is a strong assumption that no object lies partially inside the silhouette of another object. Our formulation is less restrictive. The right illustration of the Fig.3 outlines an example where our approach keep all real objects, and the [9] approach removes some parts of real objects.

5 Results

This section presents results which demonstrate the robustness and effectiveness of $LSEV_{n_{min}}$ and RSH methods.

The capture setup is composed of five firewire cameras capturing at 30fps with a resolution of 640x480 pixels. Each camera is connected to a computer that does the silhouette map extraction using the approach proposed by [13], and sends the information to a server. Intrinsic and extrinsic parameters were estimated using a popular calibration method [14]. Reconstruction ($LSEV_{n_{min}}$) and ghost removal (RSH) steps are computed on a ATHLON X2 5600+ with a Nvidia 7900GTX graphics card.

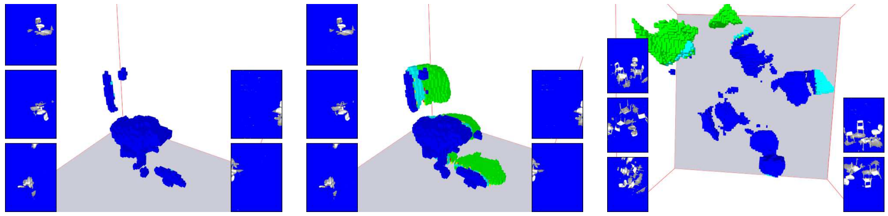
5.1 $LSEV_{n_{min}}$

In the first experiment (see Fig. 4), we compare the standard SFS and our approach. There are five cameras, two of which only have a partial view. The traditional SFS breaks down because it cannot reconstruct anything outside the strict intersection of the camera's viewing frusta. In contrast, in spite of partial views, our algorithm computes the correct visual hull.

Figure 4(c) outlines VH estimation from a complex scene, nevertheless ghost objects are constructed.

In our experiments we set $n_{min} = 3$. This parameter controls the minimum number of cameras that must see a point in order to be reconstructed. A. Laurentini [2] has shown that the higher the number of cameras seeing a point, the more accurate the estimation of VH is. To maximize the capture volume, n_{min} should be close to 1. Setting it to 1 is discouraged as it allows an infinite volume. Setting n_{min} close to n the camera number, will yield a more accurate reconstructions, albeit with a smaller capture volume.

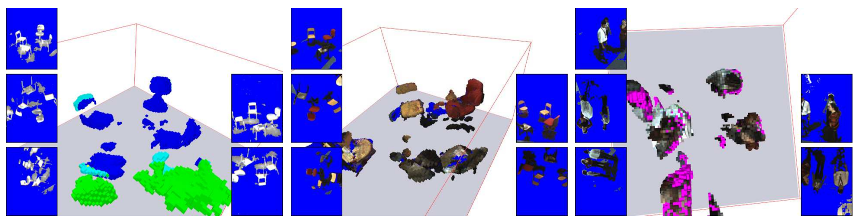
Our implementation of $LSEV_{n_{min}}$ works in real-time. We chosen to sample VH with a 3D regular grid, to use GPU processing power. With a grid of 128^3 in a box of $4 \times 4 \times 2$ meters and $n = 5$ cameras, our implementation computes more than 100 reconstruction per second.



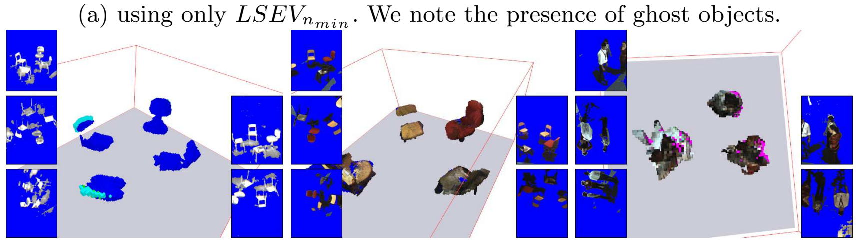
(a) (b) (c)

Fig. 4. Comparison of SFS and $LSEV_{n_{min}}$ reconstructions of a chair. On (a) and (b) pictures, two cameras (right two frames of each figure) have a partial view of the object. (a) Classical SFS clips parts that lie outside the strict intersection of viewing frusta. (b) $LSEV_{n_{min}}$ keeps all object parts. (c) $LSEV_{n_{min}}$ is also able to reconstruct a complex scene composed of four chairs, but ghost objects appear. The color indicates the number of cameras that see a 3D point : blue = 5, cyan = 4, green = 3.

.



(a) using only $LSEV_{n_{min}}$. We note the presence of ghost objects.



(b) using $LSEV_{n_{min}}$ and RSH

Fig. 5. RSH results: First row (a) represents the VH estimated with $LSEV_{n_{min}}$. Second row (b) shows the VH cleaned of ghost objects using the RSH concept.In the second and the third columns, color is shown for a better comprehension (voxels whose color cannot be deduced are shown in purple).

.

5.2 *RSH*

Having accurate silhouettes is not enough to filter out ghost objects. Figure 5(a) shows the reconstruction of different frames use only $LSEV_{n_{min}}$. Although the silhouettes are less noisy, there are many ghost objects. In contrast, Figures 5(b) and 6 using RSH removes all the ghost parts of VH. And as we can see in the camera views (small frames on the sides), the silhouettes of the objects *can* overlap.

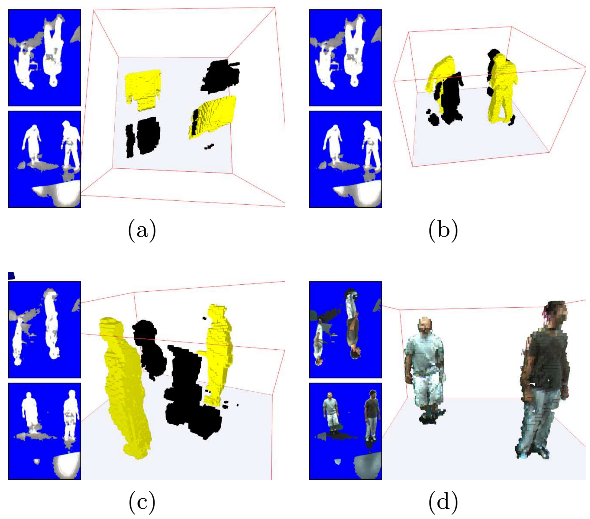


Fig. 6. RSH results: represents the VH of a complex scene with two persons estimated from two cameras from a voxel grid of 64^3 in a box of $2.5 \times 2.5 \times 2.5$ meters. (a) and (b) represent the same reconstruction of RSH in yellow from two different virtual views. Ghost objects are successfully recognized (in black). (c) and (d) underline the efficiency of the RSH approach for the same scene. (d) shows the colored reconstruction provided by RSH

.

We emphasize that RSH removes ghost objects for a given VH and is independent of $LSEV_{n_{min}}$. Thus RSH can be used with all other SFS methods.

With a grid of 128^3 in a box of $4 \times 4 \times 2$ meters and $n = 5$ cameras, our implementation of RSH processes in real-time with more than 25 corrections per second. RSH is slower than $LSEV_{n_{min}}$ because of computing associations between connex components, and silhouette pixels. The complete process ($LSEV_{n_{min}}$ and RSH) works at more than 20 frames per second. Computation time linearly depends on the n (number of cameras) parameter.

6 Conclusions

In this paper we have presented two contributions to overcome two of the usual drawbacks of Shape From Silhouette algorithms. Our approach is able to reconstruct the visual hull (VH) of a scene even if cameras see only part or even no part of the object. While most previous approaches assume that the complete silhouette has to be visible, this system is much more flexible in the camera placement, and therefore allows extending the acquisition space. As our new approach computes the VH, the reconstruction is always silhouette equivalent. The other major contribution we have presented is a theoretical approach to remove ghost objects which result in scenes with multiple objects. Our solution does not require any additional information such the number of real objects or correspondence between silhouettes. This greatly enhances the uses for SFS algorithms, and with $LSEV_{n_{min}}$ it achieves great results.

In the future, we plan to add temporal coherence to increase the efficiency and accuracy of the system. We would also like to address a minor limitation of the system: RSH guarantees that there are no ghost objects in the reconstructions, but it is theoretically possible to miss the reconstruction of a real object, even if we have never seen this in practice. This can be addressed by color matching, temporal coherence among other methods.

References

1. Baumgart, B.G.: Geometric modeling for computer vision. PhD thesis, Stanford (1974)
2. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Trans. on Pattern Analysis and Machine Intelligence **16** (1994) 150–162
3. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: Proceedings of the Fourteenth British Machine Vision Conference. (2003) 329–338 Norwich, UK.
4. Li, M., Magnor, M., Seidel, H.: Improved hardware-accelerated visual hull rendering. Vision, Modeling and Visualization 2003 (VMV-03) : proceedings (2003)
5. Cheung, K.M., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (2003)
6. Michoud, B., Guillou, E., Bouakaz, S.: Shape From Silhouette: Towards a Solution for Partial Visibility Problem. In C.Hansen, D., ed.: Eurographics 2006. Eurographics 2006 Short Papers Preceedings (2006) 13–16
7. Franco, J.S., Boyer, E.: Fusion of multi-view silhouette cues using a space occupancy grid. In: Proc. of the 10th International Conference on Computer Vision. (2005)
8. Yang, D.B., Héctor H. González-Ba n., Guibas, L.J.: Counting people in crowds with a real-time network of simple image sensors. In: ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision, Washington, DC, USA, IEEE Computer Society (2003) 122
9. Miller, G., Hilton, A.: Safe hulls. In: Proc. 4th European Conference on Visual Media Production, IET (2007)
10. Bogomjakov, A., Gotsman, C.: Reduced depth and visual hulls of complex 3D scenes. In: Computer Graphics Forum 27(2) (Proc. of Eurographics '08). (2008)
11. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 199–218
12. Mordohai, P., Medioni, G.: Dense multiple view stereo with general camera placement using tensor voting. In: 3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium on (3DPVT'04), Washington, DC, USA, IEEE Computer Society (2004) 725–732
13. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II, London, UK, Springer-Verlag (2000) 751–767
14. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: ICCV. (1999) 666–673