Silhouettes Fusion for 3D Shapes Modeling with Ghost Object Removal

Brice Michoud, Erwan Guillou, Héctor M. Briceño and Saïda Bouakaz

LIRIS - CNRS UMR 5205 - Université Claude Bernard Lyon 1

Abstract. In this paper, we investigate a practical framework to compute a 3D shape estimation of multiple objects in real-time from silhouette probability maps 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; Lastly, the results depend heavily on quality of silhouette extraction.

In this paper we propose two major contributions to overcome these limitations. First, using a simple statistical approach, our system reconstructs objects with no constraints on camera placement and their visibility. This approach computes a fusion between all captured images. It compensates for bad silhouette extraction and achieves robust volume reconstruction. 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. The users should be unconstrained: the capture system should be robust, markerless and there must be a large acquisition space. Furthermore, to be of most use, the system should work with multiple persons and in real-time. Shape from Silhouette (SFS) methods provide a good approach as they are robust, markerless, operate in real-time and are easy to implement.

SFS methods compute the visual hull of an object relative to a viewing region. The visual hull is defined as the intersection of 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 cameras's center. Intuitively, 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-Consistent [2]. However, the volumes produced from a SFS reconstruction suffer from many drawbacks:

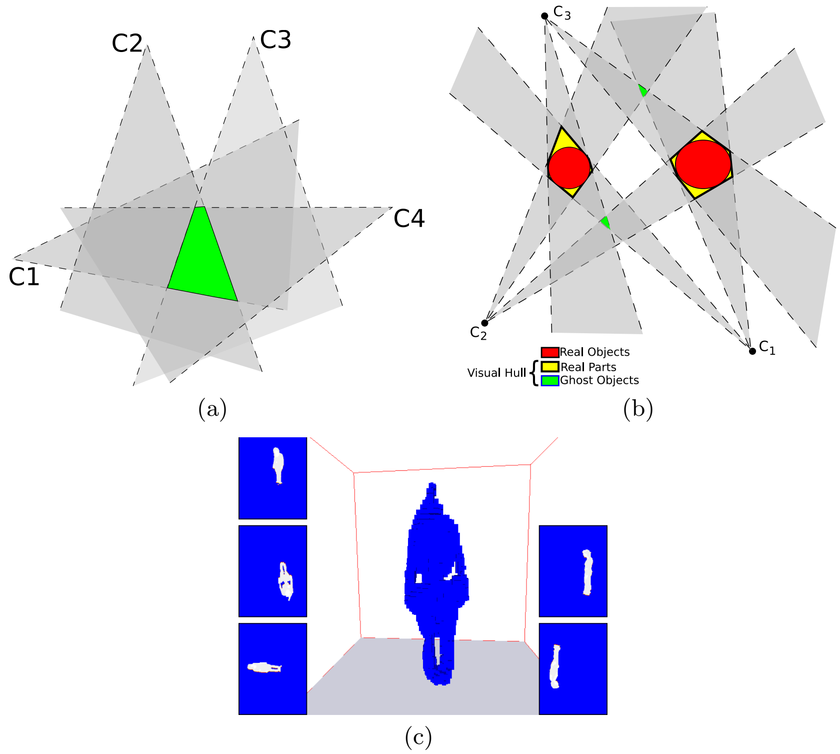


Fig. 1. The main drawbacks of SFS algorithms: (a) the acquisition space (in green) is limited to the strict intersection of the camera's viewing frustum. (b) shows a typical SFS configuration where ghost objects appear. (c) underlines that SFS reconstruction accuracy depends on silhouette's extraction quality. When holes and noise appear in the silhouette images, SFS reconstructs a corrupted shape.

Camera Placement: the objects that we can capture 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)).

Influence of Silhouette Extraction on Reconstruction Accuracy: the extracted silhouette can become incomplete or of bad quality. Missing information in one or more silhouettes has a negative impact over the whole SFS reconstruction. (see Fig. 1(c))

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 interfere with many applications of SFS, especially when they 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 having computation 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 use the complete set. To deal with the second problem, we adopt a probabilistic approach to confront information issued from all the images. To circumvent the third 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 must belong to 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 [5–8]. 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 the intersection of all viewing frusta.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[9] 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.* [8] 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. [10] 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. [11] defines 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, right picture of the Fig. 4 shows an example where this approach fails. However it is fully automatic and do not require any additional information such as object correspondence. [12] obtain ghost object removal but require additional information like Depth-maps or correspondence between the silhouettes. Depth-maps come from particular sensors, and correspondence impose that each view see each object, and labeling processing can be unstable with similar objects. Their approach is unsuitable for our applications. [13, 14] 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 to robustly reconstruct multiple 3D objects in real-time from multiple views. Using a simple statistical 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 Shape From Silhouette's Probability Maps 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 give the perspectives in Section 6.

3 Shape From Silhouette Probability Maps

In this section we describe a novel approach which computes a fusion between all captured images which can compensate bad silhouette segmentation. We use the silhouette information in a probabilistic setting. Our method extends the acquisition space because it relaxes the camera placement constraints. Robust volume reconstruction is achieved.

SFS algorithms deduce the shape of an object from its silhouettes given 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).

Another limitation is that the SFS method is not robust with badly segmented silhouettes. If a pixel is wrongly classified as background, the projection of 3D points that lie in this pixel will be classified as non-object points. This results in a incomplete reconstruction. Several reasons account for badly segmented silhouettes, in particular perturbations due to camera sensor noise, ambiguities between foreground objects and background color, changes in the scene's lighting (including shadows of foreground objects). Controlling all these factors at the same time is no easy task.

Our key observation, is that a 3D point can be rejected by one badly segmented silhouette, thus our approach delays the occupancy decision to an upper stage of the system to confront information issued from all the images. It is possible to compensate a noisy silhouette by information from other images.

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 foreground 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 $\quad Proj_{\pi_i}(E)$.

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:

$$SFS(n) = \{x \in R^3, \forall i \in [1, \cdots, n], Proj_{\pi_i}(x) \in S_i\} \tag{1}$$

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.

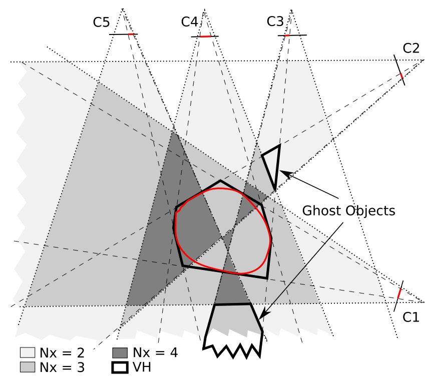


Fig. 2. 2D representation of a $ESFS$ reconstruction of one object (in red) using $n_{min} = 2$ and $n = 5$. Please note that the intersection of all camera frustum views is empty, and that usual SFS is unable to reconstruct the VH. The acquisition space is defined as the union of space where $n_x \geq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

.

Let VS_x the subset of cameras which can see the point x, then:

$$VS_x = \{i \in [1, \cdots, n], Proj_i(x) \in I_i\} \tag{2}$$

n_x is defined by:

$$n_x = Card(VS_x). \tag{3}$$

To reconstruct 3D points in volumes space seen by less than n cameras we introduce $ESFS$, an extension of SFS, defined by:

$$ESFS(n, n_{min}) = \{x \in R^3, n_x \geq n_{min}, \forall i \in VS_x, Proj_{\pi_i}(x) \in S_i\} \tag{4}$$

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 $ESFS$ approach.

This contribution extends the volume where 3D points can be tested, then reconstructed. But the reconstruction is not more robust to bad (noisy) silhouette than the usual SFS approach. To overcome this constraint we need to compute VH from the probabilities that each pixel represents a foreground object.

The silhouette information is generally deduced from statistical background modeling (see [15, 16]) using:

$$\forall y \in S_i, P_{fg}(y) > T. \tag{5}$$

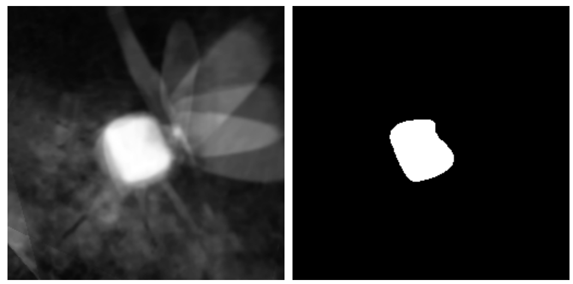


Fig. 3. The left picture shows 2D representation of $P_{VH}(x)$ values with back color corresponds to 0 and white stands for 1. Right picture underlines $SFSPM$ reconstruction with $T_{iso} = 0.8$.

where T is a threshold and $P_{fg}(y)$ denotes the probability that the 2D point y of π_i comes from a foreground object; $P_{fg}(y)$ is given by:

$$P_{fg}(y) \begin{cases} \in [0-1] & \text{if } y \in I_i \\ = 0 & \text{otherwise} \end{cases} \tag{6}$$

hence we introduce the probability maps of the camera C_i as the set of $P_{fg}(p)$ with $p \in I_i$. We use the information issued from all probability maps to decide if a 3D point is inside the VH. Silhouette segmentation can be noisy on one or multiple views. This notion can be defined as the probability that a 3D point x is consistent with all the possible silhouettes. Let $P_{VH}(x)$ be this probability:

$$P_{VH}(x) = \begin{cases} \dfrac{1}{n_x} \displaystyle\sum_{i=1}^{n} P_{fg}(Proj_{\pi_i}(x)) & \text{if } n_x \geq n_{min} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Finally we define the Shape From Silhouette Probability Maps ($SFSPM$) by:

$$SFSPM(n, n_{min}, T_{iso}) = \{x \in R^3, P_{VH}(x) \geq T_{iso}\} \tag{8}$$

The parameter T_{iso} is a threshold on the probability value, it defines the accuracy of the reconstructed shape. Figure 3 shows 2D representations of $P_{VH}(x)$ values (on the left) and the corresponding $SFSPM$ reconstruction (on the right).

3D reconstructions of the VH with $SFSPM$ are underlined on Fig .6. To use the confrontation scheme to decide if x is in the VH or not, the approach need that $n_{min} \geq 2$. Values of T_{iso} and n_{min} will be discussed in Section 5. For a 3D point x, higher values for n_x allows better correction of bad silhouette extraction using $SFSPM$.

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 cameras. Our reconstruction is always silhouette-consistent if all objects are seen by at least n_{min} cameras. Furthermore, $SFSPM$ can be used to improve silhouette extraction in our input images, by back projecting, using point information from other input views.

Usual SFS approaches suffer from ghost objects reconstruction. As we relax the camera placement constraint, $SFSPM$ can create more ghost object than SFS (see Fig.2 and Fig.6). 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 $SFSPM$ has been used for silhouette correction and 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. For reading convenience, some of these properties are explained in informal but pragmatical fashion.

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 ambiguity)

However, if there is more than one object in the scene, the regions of intersection of cones vision generated by the silhouettes can admit component outside the box encompassing objects. So 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

$$\bigcup_i Proj_{\pi_i}(VH) = \bigcup_i S_i \tag{9}$$

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 with

$$\bigcup_{j=1}^{m} CC_j = VH \tag{10}$$

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 satisfy the Proposition 1.

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

$$RSH = \bigcup CC_l \subset VH, \exists p \in Proj_{\pi_i}(CC_l), p \notin Proj_{\pi_i}(CC_k) \tag{11}$$

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.

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.

Limitations: 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 term of keeping real objects, our approach is more accurate than the one proposed in [11]. Our approach can be rewritten in the following way: A 3D connex component of VH is kept if it exist a ray coming from a silhouette's pixel in a least one camera, which intersect this 3D connex component only. [11] defines the Safe Hulls concept: A 1D connex component of VH is kept if it exist

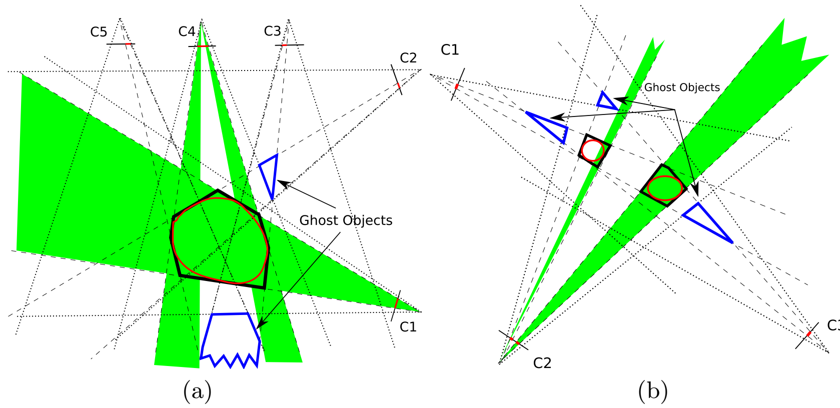


Fig. 4. 2D reprensentations of RSH results. The real objects are represented in red. (a) Configuration presented in Fig.2. The black connected component is kept because it satifies the Proposition 1. Green parts indicate where connected components will be accepted (exactly one object is projected in 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 [11] results as the intersection of VH connected components, and green parts. Their method fails to keep all real parts.

.

a ray coming from a silhouette's pixel in a least one camera, which intersect this 1D (interval) connex component only. Our formulation is less restrictive and the right illustration of the Fig.4 outlines an example where our approach keep all real objects, and the [11] approach removes some parts of real objects.

5 Results

This section presents results which demonstrate the robustness and effectiveness of $SFSPM$ 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 [15], and sends the information to a server. Intrinsic and extrinsic parameters were estimated using a popular calibration method [17]. To enforce coherence between the cameras, color calibration is done using the method proposed by N.Joshi [18]. Reconstruction ($SFSPM$) and ghost removal (RSH) steps are computed on a ATHLON X2 5600+ with a Nvidia 7900GTX graphics card.

5.1 SFSPM

In the first experiment (see Fig. 5), we compare the standard SFS and our approach. There are five cameras, two of which only have a partial view. The

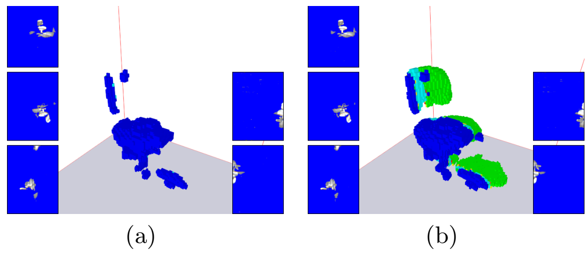


Fig. 5. Comparison of SFS and SFSPM reconstructions of a chair. Two cameras (right two frames of each figure) have a partial view of the object. Classical SFS clips parts that lie outside the strict intersection of viewing frusta. Note that with exact and accurate background substraction, $ESFS$ and $SFSPM$ give the same results.

.

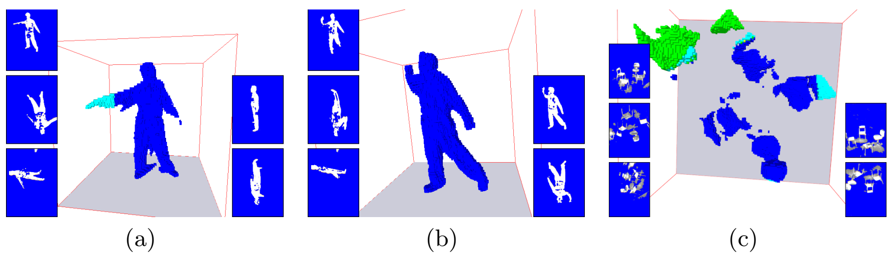


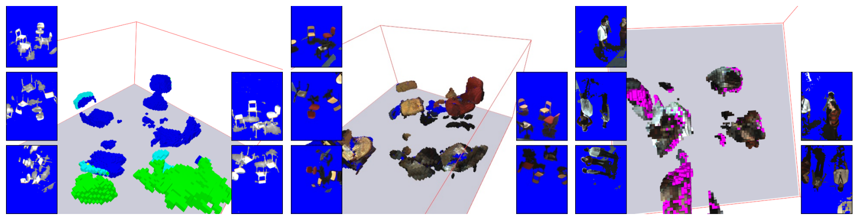
Fig. 6. Different results using the $SFSPM$ approach with $n_{min} = 2$ and $T_{iso} = 0.8$. (a) A human shape is reconstructed from some partial silhouettes. The color indicates the number of cameras that see a 3D point. (b) $SFSPM$ achieves a robust reconstruction from very noisy silhouettes (notice the holes in the silhouettes). (c) $SFSPM$ (without RSH) is also able to reconstruct a complex scene composed of four chairs, but ghost objects appear (green parts).

.

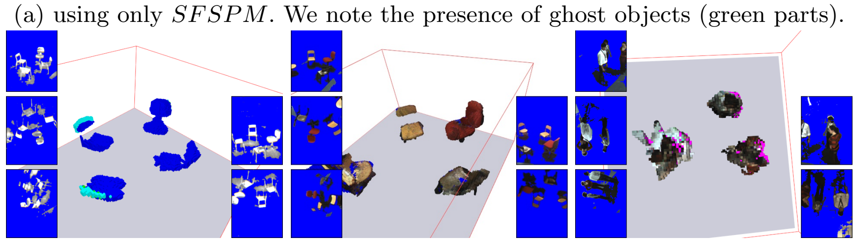
traditional SFS breaks down because it cannot reconstruct anything outside the strict intersection of the camera's viewing frusta. In contrast, in spite of partial views, our algorithm computes the correct visual hull.

Figure. 6 presents the results of $SFSPM$ with $n_{min} = 2$ and $T_{iso} = 0.8$. Figures 6(a) and 6(b) show that $SFSPM$ filters noisy silhouettes to provide a corrected 3D VH. Figure 6(c) outlines VH estimation from a complex scene with noisy silhouettes, nevertheless ghost objects are constructed.

In our experiments we set $n_{min} = 2$. 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.



(a) using only $SFSPM$. We note the presence of ghost objects (green parts).



(b) using $SFSPM$ and RSH

Fig. 7. RSH results: First row (a) represents the VH estimated with $SFSPM$. Second row (b) shows the VH cleaned of ghost objects using the RSH concept. Each column represent on particular frame. In the second and the third columns, color is shown for a better comprehension (voxels whose color cannot be deduced are shown in purple).

.

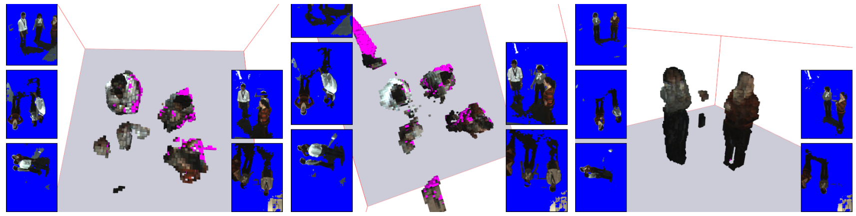
The second parameter T_{iso} of $SFSPM$ defines the accuracy of the silhouette segmentation. This is threshold on the probability that a 3D point lies inside the VH. We use $T_{iso} = 0.8$. This accepts a small error on the silhouette extraction without adding too much noise. Our implementation of $SFSPM$ 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 and $n = 5$ cameras, our implementation computes more than 100 reconstruction per second.

5.2 RSH

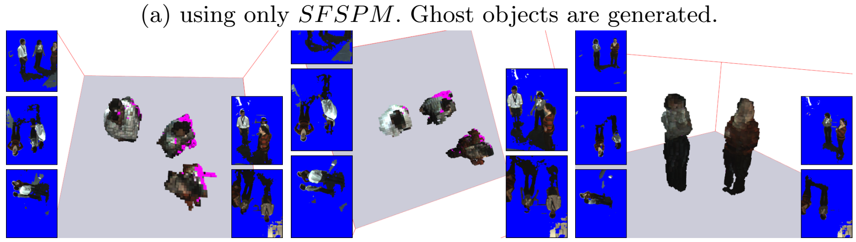
Having accurate silhouettes is not enough to filter out ghost objects. Figures 7(a) and 8(a) shows the reconstruction of different frames use only $SFSPM$. Although the silhouettes are less noisy, there are many ghost objects. In contrast, Figures 7(b) and 8(b) using RSH removes all the ghost parts of VH. And as we can see in the camera views (small frames on the sides), the silhouettes of the objects *can* overlap.

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

Our implementation of RSH processes in real-time with more than 25 corrections per second. RSH is slower than $SFSPM$ because of computing associations between connex components, and silhouette pixels. The complete process ($SFSPM$ and RSH) works at more than 20 frames per second. Computation time linearly depends on the n (number of cameras) parameter.



(a) using only $SFSPM$. Ghost objects are generated.



(b) using $SFSPM$ and RSH. Ghost objects are removed.

Fig. 8. RSH results: represents the VH of a complex scene with multiple persons, estimated with $SFSPM$. Second row (b) shows the VH cleaned of ghost objects using the RSH concept (voxels whose color cannot be deduced are shown in purple).

.

6 Conclusions

In this paper we have presented two major contributions to overcome three 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. We proposed a statistical approach which make the reconstruction of the VH more robust to bad silhouette extraction. Our method compensates for noisy silhouette with information from other images. 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 $SFSPM$ 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)
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. Liu, X., Yao, H., Yao, G., Gao, W.: A novel volumetric shape from silhouette algorithm based on a centripetal pentahedron model. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition, Washington, DC, USA, IEEE Computer Society (2006) 9
7. Guan, L., Franco, J.S., Pollefeys, M.: 3D occlusion inference from silhouette cues. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, (USA). (2007)
8. 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
9. 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)
10. 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
11. Miller, G., Hilton, A.: Safe hulls. In: Proc. 4th European Conference on Visual Media Production, IET (2007)
12. Bogomjakov, A., Gotsman, C.: Reduced depth and visual hulls of complex 3D scenes. In: Computer Graphics Forum 27(2) (Proc. of Eurographics '08). (2008)
13. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 199–218
14. 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
15. 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
16. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: ICPR 2004. Proceedings of the 17th International Conference on Pattern Recognition. Volume 2. (2004) 28–31 Vol.2
17. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: ICCV. (1999) 666–673
18. Joshi, N.: Color calibration for arrays of inexpensive image sensors. Technical report, Stanford University (2004)