Visual Hull

David Schneider

This article gives an overview of the concept of the Visual Hull and what its advantages and disadvantages are.

The Visual hull is a concept of a 3D reconstruction by a Shape-From-Silhouette (SFS) technique. This kind of 3D scene reconstruction first has been introduced by Baumgart in his PhD thesis in 1974 [1]. Since then there have been several different variations of the Shape-From-Silhouette method. The basic principle is to create a 3D representation of an object by its silhouettes within several images from different viewpoints. Each of these silhouettes by different camera views form in their projection a cone, called visual cone and an intersection of all these cones form a description of the real object's shape (see figure 1.a for a 2D example).

By using this basic idea there are many advantages in using Shape-From-Silhouette techniques. First of all the calculation of the silhouettes is easily to implement, when we assume an indoor environment with special conditions, like static light and static cameras. Without these assumptions it can become difficult to calculate an accurate silhouette out of the images, because of shadows or moving backgrounds. Further techniques for this problem will not been discussed any further in this article. On the other hand are the implementations of the SFS-algorithm straight forward and especially compared to other techniques for shape estimations, like multi-baseline stereo far less complex. The result of the SFS construction is an upper bound of the real object's shape in contrast to a lower bound, which is a big advantage for obstacle avoidance in the field of robotic or visibility analysis in navigation. Another application for SFS estimations are for instance the field of motion capturing [5].

On the other hand there are also disadvantages for these techniques. So there are time consuming testing steps, which are a bottleneck for real-time applications or the silhouette calculations, which are relative sensitive for errors, like noise or wrong camera calibrations. These ends up in problems for the intersection of the visual cones and therefore bad results for the resulting 3D shapes.

Furthermore is the result of each SFS algorithm just an approximation of the actual object's shape (like we will prove later), especially if there are only a limited number of cameras and therefore is this approach not practical for applications like detailed shape recognition or realistic rerendering of objects [5].

The main problem of the SFS-based algorithms are that they are not able to perform an accurate reconstruction of concave objects, like figure 2 (as long as we assume that the camera views are not too near to the object). An obvious question, which occurs in this context is which parts of an object can be reconstructed by standard SFS techniques, or what are the limits of these approaches?

To denote this difference Laurentini introduced the term of the Visual Hull in 1991 [2]. His formal definition of the Visual Hull is the following:

> "The visual hull $VH(S, R)$ of an object S relative to a viewing region R is a region of E^3 such that, for each point $P \in VH(S, R)$ and each viewpoint $V \in R$, the half-line starting at V and passing through P contains at least a point of S." [3]

Two more informal definitions given by Laurentini are the following ones:

> "...the visual hull of an object S is the envelope of all the possible circumscribed cones of S. An equivalent intuition is that the visual hull is the maximal object that gives the same silhouette of S from any possible viewpoint." [3]

If we consider these definitions it is easy to see, that $S \leq VH(S)$. The proof for this statement is rather forward: First of all we show that $VH(S, R)$ includes the object's silhouettes with respect to R. According to the first definition, each projection of any point $P \in VH(S, R)$ belongs to the silhouette of S, otherwise it would not be a part of the Visual Hull, therefore is $S \leq VH(S)$. The reason, why $VH(S, R)$ is the maximum silhouette equivalent is that, if there would be a point $P' \notin VH(S, R)$ and we would go along a line starting from V'/inR and passing P', then this line would not intersect S. Therefore P' would not belong to the silhouette of S, according to V' and P' does not belong to the shape of the object.

Other interesting propositions by Laureneti are that $VH(S, R)$ is the closest approximation of S, that can be archived by using volume intersection techniques. Furthermore we can see that if we choose our R that way that $R > R'$, then $VH(S, R) > VH(S, R')$, so if we want to have a higher precision we need more different viewpoints and so more visual cones. A general conclusion of Laureneti is the following: $VH(S, R) <= CH(S)$ [3], so that the Visual Hull is always smaller or equal to the convex hull.

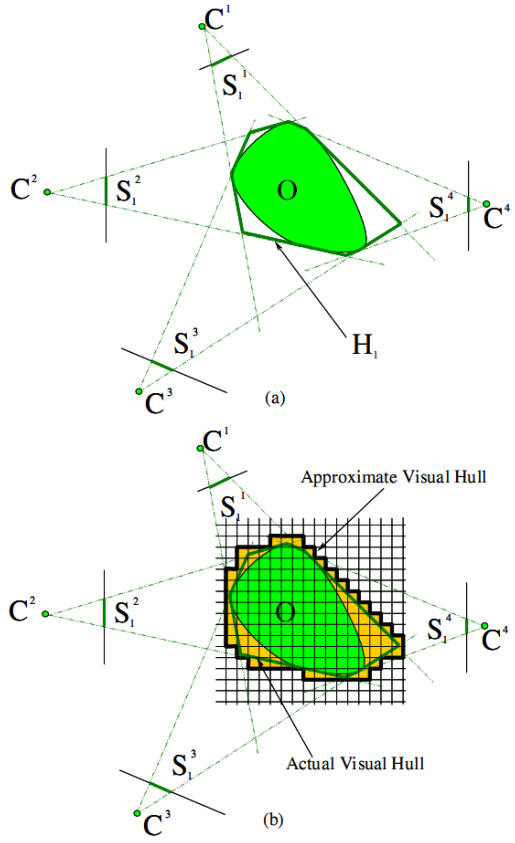

Figure 1: This figure shows a 2D example of the visual cone. Figure (a) shows different viewpoints $C^1, C^2, \ldots C^4$, which all have a different view at the object O and therefore differentsilhouettes $S_1^1, S_1^2, \ldots, S_1^4$. The intersection of the projected silhouettes form the Visual Hull H_1. Figure (b) clarify the difference between the Visual Hull of an object and its approximation by a certain algorithm. (Figure taken from Cheung, 2003[5]).

These accurate definition of the resulting shape of a SFS-technique helps us to describe the resulting approximation of the real object.

The direct way of the actual construction of the visual hull would be by intersecting the visual cones. An 2D example of this is given in figure 1.a. The problem of this approach is that the visual hull in general consists of a curved and irregular surface and therefore requires a complex geometrical representation for its cones. This leads to a higher complexity and numerical instability, which encourage scientists to choose approximate representation by using a polyhedral shape instead, while intersecting the visual cones.

Another more efficient way to calculate an approximation of the visual hull is a volume based approach ([10], [9], [11], [8]). Figure 3 shows a pseudo code version of the algorithm, whereas $\prod$ descries the projection of the silhouettes into the space.

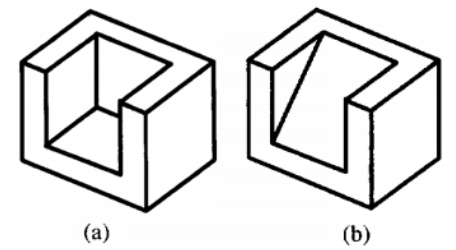

Figure 2: This figure shows different object shapes. Figure (a) is a real world object, whereas figure (b) represents its reconstruction by a SFS-approach. (Figure taken from Laurentini, 1994 [3]).

1. Divide the space of interest into $N \times N \times N$ discrete voxels v_n, $n = 1, \ldots, N^3$.
2. Initialize all the N^3 voxels as inside voxels.
3. For $n = 1$ to N^3 {
   For $k = 1$ to K {
   (a) Project ν_n into the k^{th} image plane by the projection function $\prod^k()$;
   (b) If the projected area $\prod^k(\nu_n)$ lies completely outside S_j^k, then classify ν as $outside$ voxel;
   }
   }
4. The Visual Hull H_j is approximated by the union of all the inside voxels.

Figure 3: This figure shows a pseudo code for a volumne-based SFS-algorithm.

As we know from the previous definition, the voxel-based SFS computation uses the same principles like the visual cone intersection, just that in this version the final shape representation is done by 3D volume elements (voxels). So we have deviant the room into sections, which we have declared as $inside$ and sections which we have declared as $outside$, according if they are in the visual cones.

Even though this technique is very easy and fast, it has a big disadvantage. the resulting shape is significantly larger then the true object shape, which makes it only of those applications feasible, in which only an approximation is used [6].

The modern approaches use surface-based representations instead of the volumetric representation of the scene, which allows to use regularization in a energy minimization

framework. These techniques results in a higher robustness to outliers and erroneous camera calibration. Furthermore these approaches try to overcome the inability to reconstruct concavities, due to the fact that they do not affect the silhouettes by using in addition stereo-based methods. They are used to repeatedly inrode inconsistent voxels and so result in smoother reconstruction. So that in addition the aim is to archive a photoconsistency [7].

At all SFS-approaches gives us a good approximation of the object's shape, which can been used for further calculations. The Visual Hull on the other side gives us a tool to describe the limitations of SFS-techniques and therefore their use in certain applications.

References

[1] B.G Baumgard. *Geometric Modeling for Computer Vision*. PhD thesis, University of Stanford, 1974.

[2] Aldo Laurentini. *The visual hull: A new tool for contour-based image understanding*. Proc. 7th Scandinavian Conf. Image Analysis, pp. 993-1002, 1991.

[3] Aldo Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Trans. Pattern Anal. Mach. Intell., 150–162, 1994.

[4] Aldo Laurentini. How Far 3D Shapes Can Be Understood from 2D Silhouettes IEEE Trans. Pattern Anal. Mach. Intell., 188–195, 1995.

[5] Kong Man German Cheung *Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering*. PhD thesis, Carnegie Mellon University, 2003.

[6] Kong Man German Cheung *Visual Hull Alignment and Refinement Across Time: A 3D Reconstruction Algorithm Combining Shape-From-Silhouette with Stereo*. CVPR (2), 375–382, 2003.

[7] Kalin Kolev and Maria Klodt and Thomas Brox and Selim Esedoglu and Daniel Cremers *Continuous global optimization in multiview 3d reconstruction*. In International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2007

[8] R. Szeliski *Rapid octree construction from image sequences*. Vision, Graphics and Image Processing: Image Understanding, 58(1):23?32, July 1993.

[9] H. Noborio, S. Fukuda, and S. Arimoto *Construction of the Octree Approximating a Three-Dimensional Object by Using Multiple Views*. IEEE Trans. Pattern Anal. Mach. Intell., 769–782, 1988

[10] M. Potmesil *Generating octree models of 3D objects from their silhouettes in a sequence of images*. Comput. Vision Graph. Image Process., 1–29, 1987

[11] Ahuja,, Narendra and Veenstra,, Jack E. *Generating Octrees from Object Silhouettes in Orthographic Views*. IEEE Trans. Pattern Anal. Mach. Intell., 137–149, 1989