

Exact View-Dependent Visual Hulls

Gregor Miller and Adrian Hilton
University of Surrey, UK
{gregor.miller, a.hilton}@surrey.ac.uk

Abstract

The visual hull is widely used to produce three dimensional models from multiple views, due to the reliability of the resulting surface. This paper presents a novel method for efficiently evaluating the exact view-dependent visual hull without using approximations. Methods for selecting intersections and ordering them via the cross ratio are presented. Results show the high quality of the surfaces produced using this method.

1. Introduction

Synthesis of novel viewpoints via shape-based reconstruction techniques is well established in media production. One of the most widely used is the *visual hull*, a three dimensional shape produced using silhouettes from multiple views. A *silhouette cone* is produced by casting rays from the camera centre through the observed view’s silhouette (obtained from foreground segmentation). The visual hull is the result of the intersection of all views’ silhouette cones. There are many varied techniques to compute the visual hull, mostly using volumetric or polyhedral approximations. In this paper we present a novel method for efficiently computing the exact view-dependent visual hull (VDVH), a surface which corresponds exactly to the surface of the visual hull visible from a specific viewpoint. The vertices of the exact VDVH lie on the surface of the visual hull. This avoids the arbitrary quantisation of a volumetric grid or polyhedral contour approximation associated with many previous visual hull algorithms. An efficient method of evaluating the VDVH using the cross ratio to compute the silhouette intersections in 2D without 3D reconstruction is presented in Section 2. Results of the approach are presented in Section 3.

Laurentini introduced the visual hull [7] as a volume-based construct which completely encloses an object in a scene given a set of silhouette images. Other researchers employed volumetric approaches to reconstruct scenes from smaller numbers of widely spaced cameras [6, 9, 3]. The ac-

curacy of the result depends on voxel size, so higher accuracy results often require large memory overheads. Franco et al. [4] presents a method to recover the exact representation of the visual hull corresponding to a polyhedral approximation of the silhouette contour. Brand et al. [2] describe a technique where differential geometry is applied to obtain a close estimate to the exact visual hull surface from silhouette contours. Image-based visual hulls [8] used an approximation view-dependent visual hull to render novel views without explicit reconstruction. Silhouette outlines for individual camera views are approximated by a piecewise linear polygonal representation. This approximation results in inexact computation of the view-dependent visual hull introducing additional artefacts in the rendering of novel views.

Given a set of observed silhouette images of a scene the exact view-dependent visual hull (VDVH) for a virtual viewpoint is the intersection of the virtual camera rays with the visual hull surface. In the following section we show how the exact visual hull geometry can be calculated efficiently for an arbitrary virtual camera view. Constraints on the ordering of silhouette intersections along the virtual image rays, together with projective invariants between intersections for different views, are used to efficiently evaluate the exact intersection with the visual hull surface.

2. Exact view-dependent visual hull

In this section we introduce a novel method to efficiently compute the exact visual hull surface visible to a virtual viewpoint from a set of silhouette images of a scene.

For VDVH computation in this work the exact intersection of the silhouette cones is found in the image domain, taking advantage of multiple view geometry and projective invariants. We first present the VDVH for a single view and then extend it to multiple views. Throughout this paper we use the notation, $\mathbf{x} \in \mathbb{R}^3$, to denote points or vectors in three dimensions and $x^j \in \mathbb{R}^2$ to denote the projection of a point or vector in the image plane of the j^{th} camera.

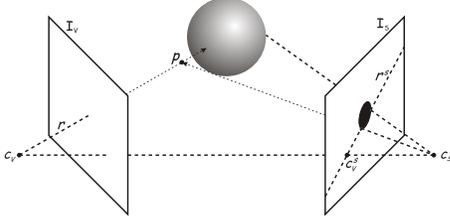


Figure 1. Rays are cast from c_s through intersections between r^s (the epipolar line) and the silhouette boundary, and triangulated with r to find the points on the visual hull.

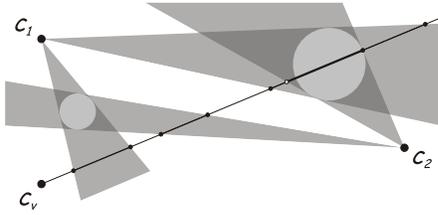


Figure 2. Cross-section of the silhouette intersections along a ray from c_v with silhouettes for two cameras with centres of projection c_1 and c_2 . The first visible intersection point on the visual hull surface marked an o .

2.1. Single view visual hull intersection

Consider the case of a single observed silhouette image I_s with camera c_s . The visual hull for a virtual view I_v results from the intersection of its rays with the silhouette cone from c_s . Equivalently, this is defined by two view projective geometry illustrated in Figure 1, where the intersection can be performed in the image plane.

For a ray r from the virtual camera centre c_v through a pixel in I_v the projection r^s of r is a two dimensional epipolar line in the image plane of I_s . All epipolar lines pass through the epipole, $c_v^s = P_s c_v$, the projection of c_v onto I_s , where P_s is the camera projection matrix. The intersections on r representing the visual hull surface are given by the reprojection of the points of intersection of r^s with the silhouette boundary in I_s . This operation returns a set of intersection points, $\{\mu_k\}_{k=1}^K \in \mathbb{R}^2$, ordered along r^s starting from c_v^s . Whenever the epipole is inside the silhouette the next intersection on the epipolar line is removed. This follows from the assumption that all intersections must be in front of the camera (in other words objects are not behind

the camera and the camera is not inside an object).

It is important to note that the point where the epipolar line enters the silhouette corresponds to a possible visible surface point. More formally, the following condition must be satisfied:

Visual Hull Visible Intersection Condition: For a silhouette intersection, μ_k , to correspond to an intersection of ray r with a visible part of the visual hull surface the intersection number k along the epipolar line r^s must be odd.

This condition guarantees that the intersection point μ_k is visible (the surface normal points towards the camera viewpoint).

The exact VDVH for a single view is given by the first silhouette intersection on the epipolar line of every ray through the virtual image I_v .

2.2. Multiple view intersection selection

Given N views we have an ordered set of silhouette intersections $U^j = \{\mu_k^j\}_{k=1}^{K_j}$ for the epipolar line r^j of ray r projected onto the silhouette image I_s^j for the j^{th} view. Then the silhouette intersection which corresponds to the first intersection of r with the visual hull is given by the following theorem:

Visual Hull Visible Intersection Theorem: The silhouette intersection $\mu \in \{U^j\}_{j=1}^N$ corresponds to the first intersection of ray r with the visual hull surface if it is the first silhouette intersection which satisfies the condition that for each of the views there is an odd number of silhouette intersections on the projection of ray r from the virtual camera centre c_v up to and including the point $p(\mu)$.

Proof: If there is an even number of intersections for any view j on the line segment between c_v and $p(\mu)$ then for the j^{th} view the projection of $p(\mu)$ is observed as outside the silhouette corresponding to empty space. Consequently if the projection of $p(\mu)$ is not inside or on the silhouette for all views then it does not correspond to a point on the visual hull. Therefore the *visual hull visible intersection condition* must be satisfied in all views for μ to be on the visual hull. This requires an odd number of silhouette intersections along the corresponding epipolar line r^j for all views.

This can be seen intuitively from the previous observation that whenever a projected ray enters a silhouette, the number of this intersection must be odd. For an intersection to correspond to the visual hull surface, the ray r must have entered every silhouette, and therefore every view must have an odd number of intersections.

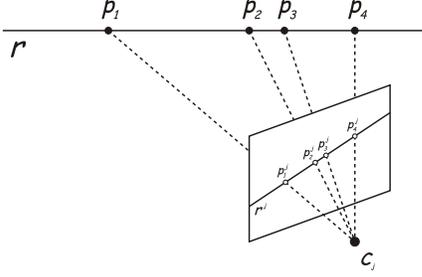


Figure 3. The cross ratio of p_{1-4} on r is equal to the cross ratio of $p_{1-4}^j = P_j p_{1-4}$ on r^j in the j^{th} view.

Figure 2 illustrates the silhouette intersections for a virtual camera ray with two silhouette images with multiple objects. The first visible intersection of the ray with the visual hull surface is the first point which is inside the silhouette for both camera views. This is given by an odd number of silhouette intersections for each camera view as stated in the *Visual Hull Visible Intersection Theorem*.

2.3. Ordering by projective invariants

The theorem introduced in the previous section states that for a set of silhouette images the exact view-dependent visual hull intersection of a virtual camera ray r can be determined from the ordering of silhouette intersections for each view. In this section we show how projective invariants can be used to evaluate the relative ordering of silhouette intersections for different views without explicit computation of the three dimensional points $p(\mu)$ along the ray. This allows computationally efficient evaluation of the exact intersection of each ray with the silhouette boundaries.

The *cross ratio* of four collinear points, p_{1-4} , is the only invariant in projective geometry [5], defined by:

$$\chi(p_{1-4}) = \frac{|\overrightarrow{p_1 p_2}| |\overrightarrow{p_3 p_4}|}{|\overrightarrow{p_1 p_3}| |\overrightarrow{p_2 p_4}|} \quad (1)$$

where $\overrightarrow{p_k p_l} = p_l - p_k$. The cross ratio is constant across all domains for the same points, i.e. $\chi(p_{1-4}) = \chi(p_{1-4}^j)$, as illustrated in Figure 3. This property can be used to order silhouette intersections along the virtual camera ray r by comparison of their cross ratio along the epipolar lines for different views. To evaluate the cross ratio, three points are generated on the ray and projected onto all images. For example, $p_1 = c_v - 2r$, $p_2 = c_v - r$ and $p_3 = c_v$. These common points are projected onto view j to obtain three points on the epipolar line, r^j : p_1^j , p_2^j and $p_3^j = c_v^j$. The cross ratio χ_k^j of the projected points with a silhouette in-

tersection point $p_4^j = \mu_k^j$ is calculated from Equation 1, and used to sort the points by increasing distance from the camera centre.

Ordering of silhouette intersections along the virtual camera ray r , using the cross ratio χ_k^j , is used to identify the silhouette intersection μ_k^j which corresponds to the first visible intersection with the visual hull surface. The corresponding 3D point on the visual hull surface $p(\mu_k^j)$ is reconstructed as the distance $d(\mu_k^j)$ along the ray from c_v . Repeating this process for virtual rays corresponding to each pixel in the virtual image, I_v , the exact view-dependent visual hull is obtained as a depth map.

2.4. Complexity analysis

Given a $n \times n$ virtual view, n^2 rays must be cast from the virtual camera centre and intersected with the silhouettes (also containing n^2 pixels). Assuming a silhouette boundary to have approximately n pixels, if every boundary was traversed once to find all intersections the running time would be $O(n^3)$, equivalent to a brute-force volumetric approach. Efficiency is improved by reducing the cost of performing line-boundary intersections, by splitting up the silhouette boundary into a large number of small pixel chains. Each chain is indexed by the angle of the epipolar line. This look-up table reduces the running time to $O(n^2)$ for a single camera, and $O(vn^2)$ where v is the number of cameras (the number of intersection tests per ray is assumed to be low enough to be considered constant). This exact method has a cost equivalent to that of the approximate solution presented in image-based visual hulls[8].

3. Results

In this section we present results for VDVH construction of people. The studio setup comprised ten cameras equally spaced in a circle of radius 5m at a height of 2.5m looking towards the centre of the space. Images were captured at 720×576 with Sony DXC-9100P 3-CCD colour cameras. Camera calibration[1] gives a maximum reprojection error of 1.6 pixels (0.6rms) averaged across the cameras which is equivalent to a reconstruction error in the order of 10mm at the centre of the volume.

Figure 4 shows novel rendered views of a human body at full resolution (720×576), demonstrating the quality of the reconstructed surface. Figure 5 shows frames of a sequence from a single novel viewpoint, where the surfaces produced contain detail of the face and hair. The right foot in image 2 of Figure 5 has slightly disappeared due to segmentation error. Each surface is produced in approximately 14 seconds on a 1.8GHz PC with 1GB RAM (for comparison purposes, a surface produced from 4 cameras at 320×240 takes approximately 0.75 seconds). The implementation could be

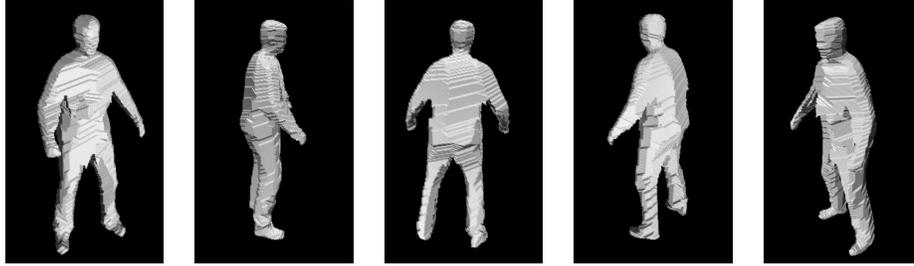


Figure 4. VDVH surface from novel views around the person for a single time instant

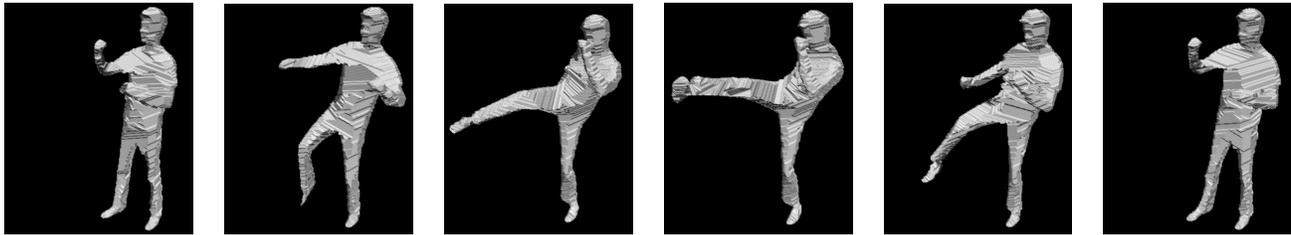


Figure 5. VDVH surface from a single novel viewpoint for a sequence of a person kicking

improved in a number of ways, for example using more than one PC or employing graphics hardware. For static viewpoints parts of the process could be done once and stored for subsequent frames.

4. Conclusions

A novel algorithm for efficient computation of the exact View-Dependent Visual Hull has been presented which produces a sampled representation of the true visual hull surface. The cross ratio is used to order silhouette intersections in 2D and reduce the number of calculations required. A *Visual Hull Visible Intersection Theorem* is introduced to efficiently select the intersection corresponding to the Visual Hull surface. Advantages of the VDVH algorithm are: (1) exact computation of intersection points on the visual hull surface without requiring an intermediate approximation or quantisation step; and (2) efficient computation of intersections in 2D using the cross ratio. As with all visual hull methods, the algorithm is limited by the quality of the silhouettes. Further work is required to optimise the segmentation of the input images.

5. Acknowledgements

This research was supported by EPSRC grants GR/M88075 ‘Video-Based Representation of Dynamic Scenes’ and EPSRC GR/S27962 in collaboration with BBC R&D.

References

- [1] J.-Y. Bouguet. Camera calibration toolbox for matlab: www.vision.caltech.edu/bouguetj/calib-doc. Technical report, MRL-INTEL, 2003.
- [2] M. Brand, K. Kang, and D. B. Cooper. An algebraic solution to visual hull. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [3] W. B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. In *International Workshop on Vision Algorithms*, Corfu, Greece, 1999.
- [4] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. In *Fourteenth British Machine Vision Conference (BMVC)*, pages 329–338, September 2003. Norwich, UK.
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [6] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. Technical Report TR692, University of Rochester, 1998.
- [7] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, 1994.
- [8] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.
- [9] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR*, pages 1067–1073, 1997.