Projective Surface Refinement for Free-Viewpoint Video

Gregor Miller, Jonathan Starck and Adrian Hilton

Centre for Vision, Speech and Signal Processing,
University of Surrey, UK
{Gregor.Miller, J.Starck, A.Hilton}@surrey.ac.uk

Abstract

This paper introduces a novel method of surface refinement for free-viewpoint video of dynamic scenes. Unlike previous approaches, the method presented here uses both visual hull and silhouette contours to constrain refinement of view-dependent depth maps from wide baseline views. A technique for extracting silhouette contours as *rims* in 3D from the *view-dependent visual hull* (VDVH) is presented. A new method for improving correspondence is introduced, where refinement of the VDVH is posed as a global problem in *projective ray space*. Artefacts of global optimisations are reduced by incorporating rims as constraints. Real time rendering of virtual views in a free-viewpoint video system is achieved using an image+depth representation for each real view. Results illustrate the high quality of rendered views achieved through this refinement technique.

Keywords: Free-viewpoint video, view-dependent rendering, wide baseline stereo, graph cuts

1 Introduction

This paper presents a novel method of dynamic scene reconstruction for free-viewpoint video. High quality view synthesis of real events via multiple view video has been a long term goal in media production and visual communication. Novel view rendering is useful for special effects, unusual perspectives and for scenes where camera placement is limited (e.g. a football stadium or a concert). The aim is to produce virtual view video with a comparable quality to captured video.

Free-viewpoint video systems have been developed to capture real events in studio and outdoor settings. The challenge is to produce good quality views from a limited number of cameras. The *Virtualized Reality*™ system[8] reconstructs dynamic scenes using images captured from a 51 camera hemispherical dome. Narrow baseline stereo is used between views to produce depth maps which are subsequently fused into a single 3D surface. This process relies on stereo matching, which can fail in areas of uniform or regular appearance. View synthesis has been achieved using visual and photo hull to reconstruct dynamic scenes from small numbers of widely spaced cameras[12, 7, 10]. Photo hull can be used to refine the visual hull using photo consistency but may fail if colours across the surface are not distinct.

These techniques have been extended to include temporal consistency as a means to improving surface quality[20, 4, 6]. A volumetric equivalent to optical flow called *scene flow*[20] estimates temporal correspondence using photo consistency across frames, neglecting silhouette contour information. The *bounding edge representation*[4] of the visual hull incorporates colour constraints on the surface to match rigid bodies across frames. This relies upon a unique colour match on a bounding edge, which often fails using photo consistency. Model-based approaches have been developed which construct explicit scene representations[3, 18]. These approaches suffer from artefacts such as ghosting and blur due to the limited accuracy of correspondence between multiple views and the model representation. The visual quality of these approaches is not suitable for applications in high quality view synthesis.

Recent approaches have used surface reconstruction as an intermediary for correspondence and view-dependent rendering to produce high quality views[22, 14]. The novel view system presented in [22] simultaneously estimates image segmentation and stereo correspondence to produce video quality virtual views, but is restricted to a narrow baseline camera setup (8 cameras over 30°). An interactive system for wide baseline views (10 cameras over 360°) was introduced in [14] which produces high quality novel views. A view-dependent visual hull is used as an initial approximation and refined locally where the surface is inconsistent between views. However, the pixel-wise refinement approach produces discontinuous surfaces which occasionally lead to depth artefacts when transitioning between cameras.

A novel method of surface refinement for free-viewpoint video is introduced in this paper. Unlike previous approaches, visual hull and silhouette contours are both used to preserve information from the original images for refinement of view-dependent surfaces. Silhouette contours are represented in 3D as *rims*, and a novel technique is presented for extracting rims from the view-dependent visual hull (VDVH). Given the VDVH as an approximation, a new method for improving correspondence is presented where refinement is posed as a global surface optimisation problem in *projective ray space*. Rims provide local information which constrain the refined surface to lie on known regions of the true surface, and the global optimisation reduces artefacts such as depth discontinuities that can occur with local approaches. Real time rendering of novel views in a free-viewpoint video system is achieved using an image+depth representation for every view.

2 Background Theory

Free-viewpoint video research widely uses the *visual hull* to synthesise novel viewpoints, either directly or as an approximation to the surface for refinement. Given N views, the set of captured images $\mathcal{I} = \{\mathcal{I}_n : n = 1, \ldots, N\}$ is converted into a set of silhouette images $\mathcal{S} = \{\mathcal{S}_n : n = 1, \ldots, N\}$ via foreground segmentation. The *silhouette cone* for the nth view is produced by casting rays from the camera centre $\boldsymbol{c}_n$ through the occupied pixels in the silhouette S_n. The visual hull is the three dimensional shape formed by the intersection of all views' silhouette cones[11].

Many varied techniques exist for constructing the visual hull. A volumetric grid where each element (voxel) is tested against $\mathcal{S}$ is a simple and robust way to generate an approximate surface, but requires an additional quantisation step[17]. The volumetric visual hull can be refined by removing voxels which fail a colour consistency test[10] , although regions of a uniform or regular appearance can lead to unreliable refinement of the surface.

It is not necessary to produce a global representation of the visual hull, some approaches construct its surface with respect to a user-defined viewpoint. Image-based visual hulls[12] use an approximate view-dependent visual hull to efficiently render novel views without explicit reconstruction. The approximations in the previous approaches can introduce artefacts in the rendering of novel views. Exact view-dependent visual hull (VDVH)[13] evaluates the surface in the image domain to produce an accurate depth map representing the visual hull, based on the original contours of $\mathcal{S}$.

Unlike a volumetric representation where space is regularly sampled on a three dimensional grid, the VDVH is represented as a set of intervals in projective ray space. This space is defined by rays from the camera centre passing through pixels in the image. The intersection of these rays with the visual hull surface define the intervals that make up the VDVH.

2.1 Visual Hull Rims

The *bounding edge representation*[4] exploits the unique property of the set of pixels $\mathcal{B}_n$ on the boundary of S_n: the ray cast from $\boldsymbol{c}_n$ through $p \in \mathcal{B}_n$ touches the surface of the scene object tangentially. The visual hull is constructed for $\mathcal{B}_n$ to produce a set of intervals $\mathcal{D}_n$ (bounding edges) in projective ray space, and the surface point is evaluated using colour consistency from neighbouring cameras. A point cloud is produced for every frame in a sequence of multiple view video and used to align subsequent frames to the first (effectively adding cameras to the scene).

The smooth curve through the points on $\mathcal{D}_n$ is called the *rim* of the visual hull. The rim may not be smooth using the bounding edge representation; locating the correct or continuous point on an interval fails when the surface appearance is uniform or regular. Points on adjacent intervals will not necessarily match up correctly.

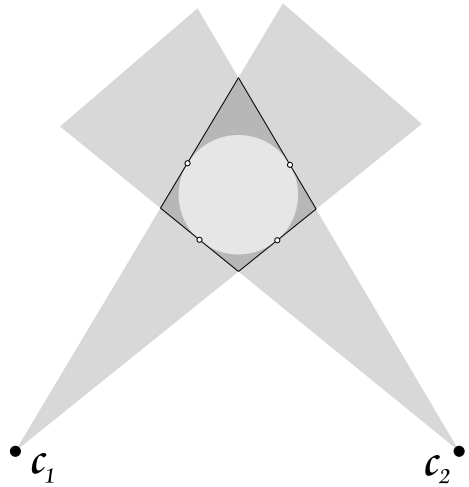


Figure 1: The circle represent the scene. The silhouette cones (light shade) are projected out from the cameras and form the visual hull where they intersect (darker region). The highlighted lines are boundary edges, and the points on them represent the rim points for those edges.

Section 3.2 describes how to retrieve the rims $\mathcal{R}_n$ for the nth view using an optimisation on $\mathcal{D}_n$. The intervals in $\mathcal{D}_n$ are extracted from a multi-layer depth map D_n produced using an extension to the exact VDVH, avoiding the additional quantisation.

2.2 Network Flows and Graph Cuts

Graph cuts on flow networks have become a popular way to solve optimisation problems in computer vision. Recent evaluation of multiple view surface reconstruction[15] show techniques based on graph cuts produce the most accurate results. This paper presents methods to recover the rims and refined surface of the object via graph cuts. The optimisation uses good scores as constraints across regions of similar scores to compensate for unreliable areas.

Previous work has shown how surface reconstruction can be accomplished using graph cuts: stereo reconstruction on a depth map, however it does not use the visual hull to restrict the search space[2]; a multi-view stereo approach, although visual hull or silhouette constraints are not taken into account[9, 21]; rims and surfaces can be constructed using volumetric visual hull, but only for genus zero objects without self-occlusion[16].

A *flow network* $G = (V, E)$ is a graph with vertices V and edges E, where each edge $(u, v) \in E$, $u, v \in V$ has a capacity $c(u, v)$[5]. G has a source $s \in V$ and a sink $t \in V$ defining the direction of flow. A *graph cut* (S, T) of G partitions V into S and $T = V - S$ such that $s \in S$ and $t \in T$. The capacity of a cut is $c(S, T) = \sum_{u \in S, v \in T} c(u, v)$. Finding a flow in G with the maximum value from s to t is known as the maximum flow problem, which, by the *max-flow min-cut theorem*, is equivalent to finding the minimum capacity cut of G.

3 Projective Surface Refinement

This section introduces a novel method for global refinement of the surface visible from a specific view by enforcing depth and silhouette contour constraints in projective ray space.

Global surface refinement techniques produce artefacts where no reliable information is present, for example in a surface region of uniform or regular appearance. This can lead to over- or under-refinement of the surface. Incorporating information from $\mathcal{S}$ (the silhouettes of the scene) additional constraints can be applied to the surface optimisation. The method presented here refines depth maps produced with respect to an existing viewpoint using an extension to the view-dependent visual hull (VDVH)[13]. The rims are evaluated for each view's VDVH using a graph cut on the boundary intervals. These are incorporated as local information into a global optimisation of the visible surface formulated as a graph cut. Vertices are positioned inside the visual hull in projective ray space, and given a score from stereo matching between adjacent views. The graph cut yields the refined surface which is converted into an image+depth representation for real-time rendering.

3.1 Initial Surface Approximation

The refinement technique relies upon an initial approximation to the surface for the following reasons: it directly supplies a narrow search space for refinement; a subset of the true surface can be recovered in the form of rims to constrain the optimisation; and it allows use of wide baseline cameras for stereo matching.

The initial surface approximation is generated by an extended version of the VDVH[13]. The original VDVH uses an image-based method to produce a depth map (single depth-per-pixel) for the required view. The extended method constructs the entire visual hull and represents it as a multi-layer depth map (for the rest of the paper, all depth maps are multi-layered). For every pixel in the depth map, each depth represents an intersection of the ray through that pixel with the visual hull surface. There are an even number of intersections, the odd intersections are the ray entering the surface, and the even ones exiting it. The intersections are grouped into intervals representing the segments of the ray inside the visual hull surface.

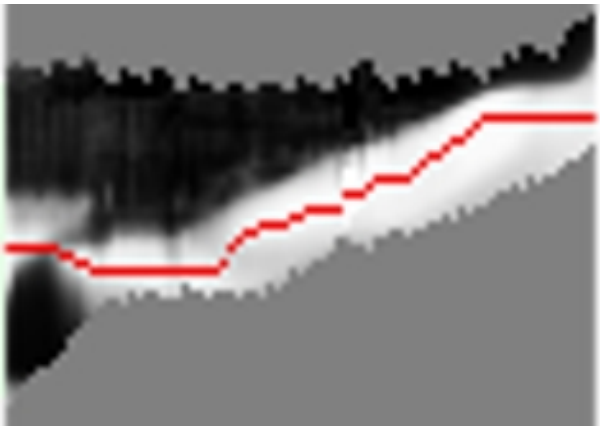


Figure 3: Diagram showing a graph cut on a chain: intervals are along the x-axis, depths down the y-axis. Good stereo scores are represented as white, and bad scores as black. The dark line through the white region is the graph cut, representing the rim.

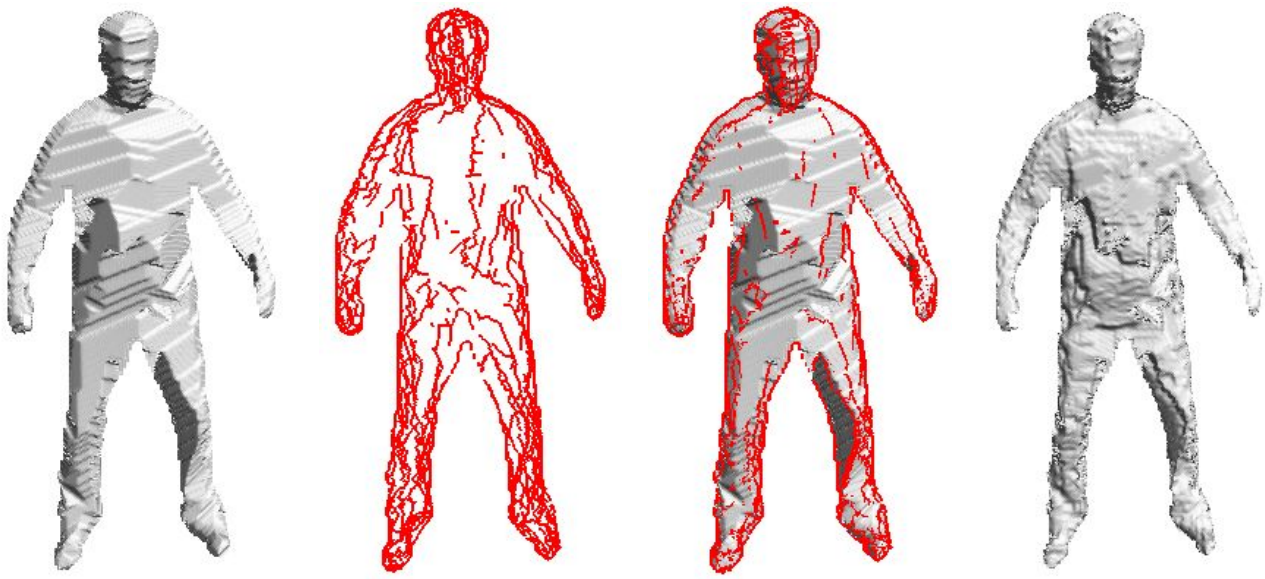
3.2 Rim Recovery

The set of rims $\mathcal{R}_n$ for the nth view can be recovered by finding the points on the rays through pixels on the silhouette contour $\mathcal{B}_n$ which correspond to the true surface. On a depth map M_n produced using VDVH, the surface point lies on the interval corresponding to $M_n(u)$, $u \in \mathcal{B}_n$. For this work, only contour points with one interval in M_n are considered since those with multiple intervals may represent phantom volumes, an artefact of visual hull resulting from occlusion or multiple objects in a scene.

The rim for a single genus-zero object with no self-occlusion is a smooth continuous curve. This scene constraint has been invoked in other work[16], however the goal of this paper is to find the rims on visual hulls representing people. The technique must therefore deal with occlusion, either from one object occluding itself or from the presence of multiple objects. Occlusions appear in the depth map as depth discontinuities.

As with any visual hull based technique, it is important to have good camera calibration and image matting. For a synthetic scene where calibration and matting are perfect then the contour of the silhouette will directly correspond to the contour of the depth map silhouette (an image constructed from a depth map by setting pixels with depths as foreground and those without as background). In practice, calibration and matting both have some degree of error, so the silhouette used to construct the rims is taken from the depth map.

Before constructing the rims the contour of the silhouette must



(a) VDVH (b) Rims (c) VDVH + Rims (d) Refined Depth Map

Figure 2: Stages of surface reconstruction for a specific viewpoint from the initial VDVH approximation to the globally optimised surface.

by analysed to detect occlusions. This process will produce a set of *pixel chains* $\mathcal{C} = \{\mathcal{C}_i : \mathcal{C}_i \subseteq \mathcal{B}_n, i = 1, \ldots, N_\mathcal{C}\}$ where $\mathcal{C}_i$ is an ordered set of pixels on $\mathcal{B}_n$ and $N_\mathcal{C}$ is the number of chains.

To produce the chains $\mathcal{B}_n$ is represented as an ordered set of pixels $\mathcal{P}_n$. $\mathcal{P}_n$ is analysed to produce pixel chains: if the interval $M_n(p), p \in \mathcal{P}_n$ overlaps the interval $M_n(p-1)$, p is added to the current pixel chain $\mathcal{C}_i$. Otherwise p marks a depth discontinuity (occlusion), so $\mathcal{C}_i$ is saved and $\mathcal{C}_{i+1}$ begun. For a scene with no occlusions, a single pixel chain is produced.

One rim segment is produced for every chain $\mathcal{C}_i \in \mathcal{C}$. For every $p \in \mathcal{C}_i$, the interval $M_n(p)$ is sampled regularly, and each sample is given a score based on a stereo comparison between two camera views with good visibility.

Previous methods found the point on the interval with the highest photo consistency score[4], but this approach leads to a discontinuous rim, because surfaces may have uniform appearance or repetitive patterns which give false positives. We propose that an optimisation problem be formulated for each pixel chain, to produce a smooth continuous curve for its rim segment. Each chain is set up as a flow network and the optimum path (the rim) through the intervals is found via a graph cut.

Each interval on the chain is sampled regularly, using the effective sampling resolution of the nearest camera at the current depth. Every sample is given a score using normalised cross-correlation stereo matching between two adjacent cameras with the best visibility of the point. The score for each sample is mapped to the range $[0, 1]$. Visibility maps are constructed in a similar way to [12], except exact computation is used (from the exact VDVH). At a sample which is not visible to two adjacent views but is visible to at least two views, a photo consistency test is performed to attach a score to the sample. In regions where there is zero visibility (for example, under the arms) the samples are given scores of 0.5, which should not bias the optimisation and allow interpolation over these regions.

Stereo windows in the original images are constructed using a base plane in 3D, set up perpendicular to the surface to improve correlation scores. The derivative of the silhouette contour is found and rotated $90°$ to give a 2D perpendicular vector pointing out of the silhouette. The equivalent 3D normal is evaluated and used to construct a 3D window at the required point on the interval with the same normal as the surface point. The 3D window is projected onto each image to produce two images for comparison.

A flow network for each chain is constructed as a set of vertices $\mathcal{V}_{\mathcal{C}_i}$ based on the sample points, and a set of edges $\mathcal{E}_{\mathcal{C}_i}$ based on the scores. The first vertex of every interval is connected to the source $s \in \mathcal{V}_{\mathcal{C}_i}$ and the last to the sink $t \in \mathcal{V}_{\mathcal{C}_i}$. A 4-connected neighbourhood is set up on the rest of the graph. Adjacent vertices on an interval are connected by an edge, and vertices at equivalent depths between intervals are connected. The capacity of each edges is $c(u, v) = 1 - \frac{s(u)+s(v)}{2}$, $u, v \in \mathcal{V}_{\mathcal{C}_i}$, where $s(u)$ is the score at vertex u. Stereo scores are maximal, whereas for a flow network a good score should have a low capacity, so the average score is subtracted from 1.

The graph cut is applied to $\mathcal{C}_i$ to retrieve the rim segment's path through the interval, as in the example in Figure 3. This is mapped into 3D using the depths on the interval to recover the actual rim segment. This process is performed for every $\mathcal{C}_i \in \mathcal{C}$ to retrieve $\mathcal{R}_n$. $\mathcal{R}$, the complete set of rims, is found by applying this process for every viewpoint, which is important for constraining the global optimisation.

3.3 Constrained Global Optimisation

The refined surface for rendering is produced by performing a global optimisation on the view-dependent surface (the depth map). Refining depth maps has been proposed before, but has either neglected silhouette constraints[2] or performed a local refinement which produces a discontinuous surface[14]. The novelty of this work is to first constrain the problem using VDVH to define the search range (allowing use of wide baseline views), and secondly to use rims to provide local information to achieve a higher quality surface reconstruction.

The technique for performing a global optimisation on a depth map produced using VDVH without enforcing contour constraints is defined first.

3.3.1 Global Optimisation of Depth Maps

Let $\mathcal{P}_n = \{p \in M_n : p \textit{ is non-empty}\}$, then $\forall p \in \mathcal{P}_n$ the possible location of the surface is defined strictly by the interval $M_n(p)$. The set of intervals $\{M_n(p) : p \in \mathcal{P}_n\}$ exist in projective ray space: the intervals are defined on rays cast through $\mathcal{P}_n$ from the camera centre $\boldsymbol{c}_n$. The intervals are sampled at regular depths to produce vertices on a 3D projective grid. Each vertex is given a score from the stereo comparison between view j and an adjacent viewpoint (chosen based on visibility). A normalised cross-correlation on a window around the pixel in I_n and the window around the projection of the vertex to the adjacent view is used to produce a correspondence score (mapped to the range $[0, 1]$).

The optimisation for the nth view is formulated as a flow network $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with vertices $\mathcal{V}_n$ and edges $\mathcal{E}_n$, illustrated in Figure 4(a). The first vertex of every interval is connected to the source $s \in \mathcal{V}_n$ and the last to the sink $t \in \mathcal{V}_n$. A 6-connected neighbourhood of edges is set up for the internal vertices. Vertices at equal depth on horizontally and vertically adjacent intervals are connected by an edge, using the capacity function $c(u, v)$, $u, v \in \mathcal{V}_n$ from Section 3.2. Adjacent vertices on an interval are connected by an edge using $c(u, v)$ with a smoothing multiplier k. As the value of k increases, the resulting surface moves toward the best scores per interval with less constraint. Correspondingly, as k decreases the surface is more constrained so that at $k = 0$ the surface is flat.

The refined surface is produced by separating the graph into two regions using the max-flow min-cut algorithm. Only edges along the intervals are checked to see if they were part of the cut, and the vertices on the edges which were cut are extracted for the surface (the vertex further away from the camera is chosen).

This method for global optimisation works very well in detailed regions of the surface, and performs a 'best guess' in regions with similar scores. Unfortunately this can lead to deformed surfaces (see Figure 5 in the results section).

3.3.2 Rim-Constrained Optimisation

The novel approach presented here incorporates the rims into the optimisation problem to provide local constraints, preserving the original information from the silhouette contours.

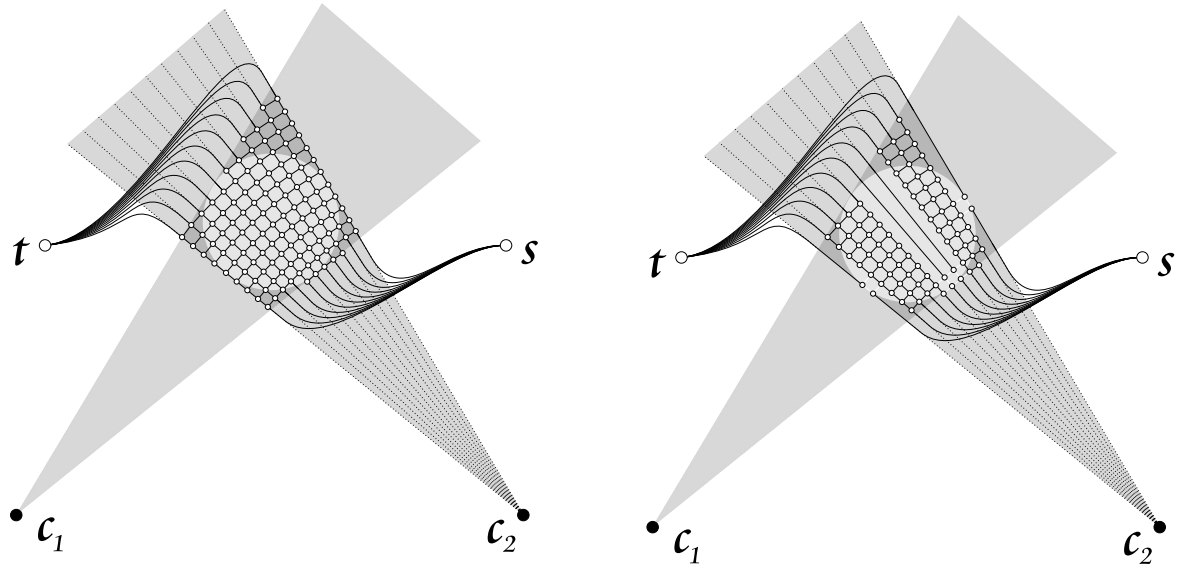
The rims are added to the flow network as it is set up, with one pre-computed step. A set of points $\mathcal{R}_n^v = \{\boldsymbol{p} \in R : \boldsymbol{p} \textit{ visible to view } n, R \in \mathcal{R}_j, j = 1, \ldots, N\}$ is extracted from the set of rims if they are visible to the current view. Every $\boldsymbol{p} \in \mathcal{R}_n^v$ is projected onto the image plane of the nth view. Edges are not added to the graph between the four pixel centres surrounding it, or to the vertices on the intervals corresponding to the four pixels. Instead, for each of the four pixels an edge is added between depths at the depth of the rim with a capacity of zero; horizontal and vertical edges are added for the vertices at those depths to adjacent intervals and among the four, as shown in Figure 4(b). Allocating a capacity of zero to the edges corresponding to the rim's location guarantees that edge becomes part of the cut, and the rest of the cut is bound to this depth.

The surface in Figure 5(d) show the benefit of adding rims to the global surface optimisation, compared to the surface without rims in Figure 5(c).

4 Rendering

The refinement operation produces N image+depth sufaces per frame; identical topology is used to produce a mesh of each surface for free-viewpoint rendering. Novel views are synthesised in real-time by rendering the N meshes in back-to-front order. The depth buffer is cleared after rendering each mesh to remove small artefacts from overlapping regions (due to differences in the view-dependent geometry).

View-dependent rendering of each mesh is performed by blending the texture from images I_m and I_n when transitioning between views m and n. The colour from each image is weighted according to the angle between the camera and the rendered viewpoint. This ensures a smooth transition



(a) Global optimisation (b) Rim constrained optimisation

Figure 4: (a) Example of a graph set up on the visual hull from Figure 1 in projective ray space with respect to $\boldsymbol{c}_2$. Vertices are marked as white circles, connected by edges marked in black. The first vertex of every interval is connected to the source s, and the last is connected to the sink t. (b) The graph with rim constraints included. Vertices are removed where the surface is known not to exist, and vertices connected by zero capacity edges (white).

between views using the estimated correspondence.

The use of multiple local representations over a single global representation gives the best correspondence between adjacent views in the presence of camera calibration error and reconstruction ambiguity[19]. High quality rendering with accurate reproduction of surface detail is achieved using locally refined surfaces.

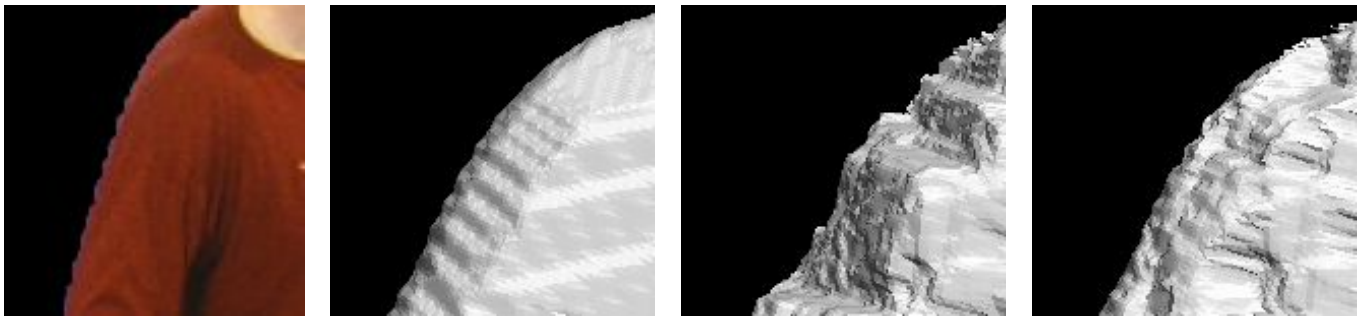
5 Results

This section presents results and evaluation of projective surface refinement for free-viewpoint rendering. Multiple view video capture was performed in a studio with eight cameras equally spaced in a ring of radius $6m$ at a height of $2.5m$ looking towards the centre of the studio. Each camera pair had a baseline of $4.6m$ with a $45°$ angle between them, and the capture volume was approximately $8m^3$. A comparative evaluation of the proposed method was performed against results from previous work[14]. The studio setup for these results comprised eight cameras, seven in an arc spanning $110°$ of radius $4m$ with a baseline of $1.2m/18°$ and approximate capture volume of $2.5m^3$ (the eighth camera gave a view from above). Synchronised video sequences were captured at 25Hz PAL resolution (720×576) progressive scan with Sony DXC-9100P 3-CCD colour cameras. Intrinsic and extrinsic camera parameters were estimated using the public domain calibration toolbox [1].

The rendering software was implemented using OpenGL, and tests were performed on an AMD 3100+ Sempron with 1GB RAM and an nVidia 6600 graphics card. The eight camera scene was rendered interactively at 28 frames per second for novel viewpoints, though this could be much improved by using hardware based view-dependent rendering. Projective surface refinement takes approximately twenty minutes to refine eight depth maps for one frame.

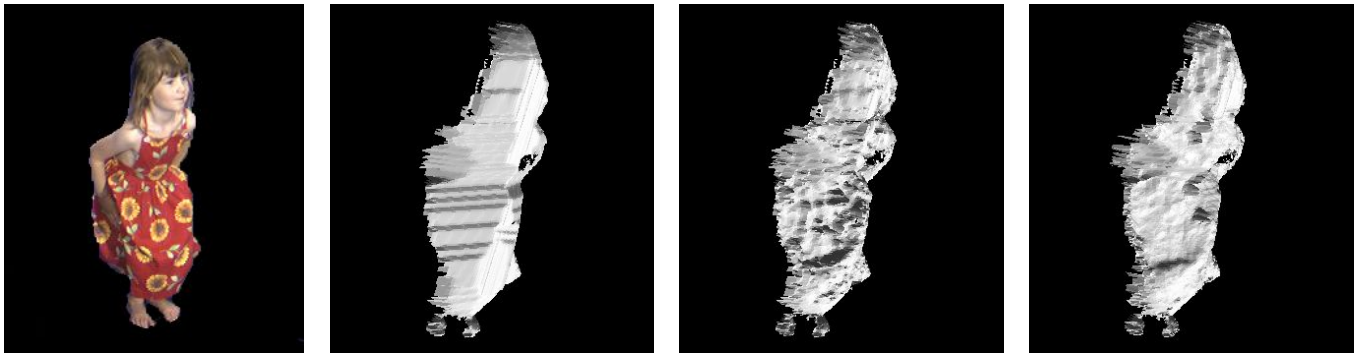
Figure 5 displays a comparison of view-dependent visual hull and optimisations with and without silhouette contour constraints. As can be seen from Figure 5(a) there is not much variation in surface appearance, and the optimisation without silhouette constraints over-refines the surface (Figure 5(c)). Figure 5(d) shows the result after adding rims to constrain the problem: the surface regains its original shape plus refinement.

The images in Figure 6 show the difference between the proposed method and work previously demonstrated[14], using the eight camera studio setup. Figure 6(c) displays the result of a local refinement performed on inconsistent areas of the surface, to produce consistent colour when transitioning between views. Figure 6(d) shows the reconstruction proposed using the presented approach which eliminates the depth map spikes and resulting render artefacts. The high variation in surface normal in Figure 6(c) makes this surface unsuitable for relighting, unlike the method proposed in this paper which



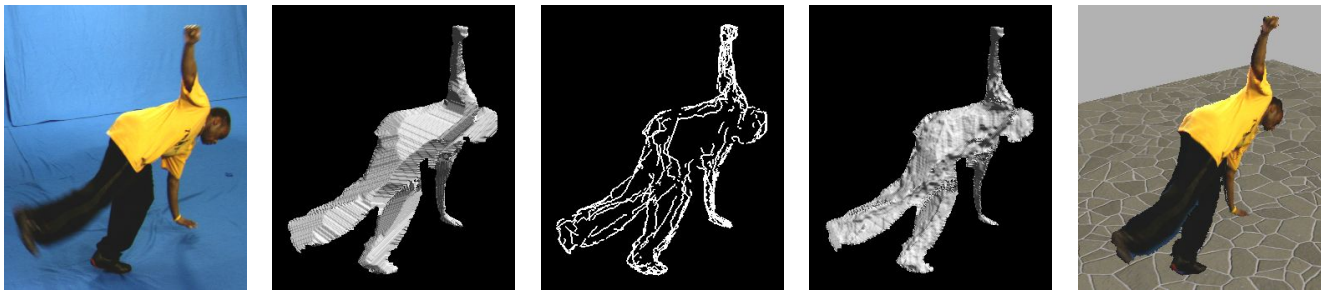
(a) Original colour | (b) VDVH | (c) Global optimisation | (d) Global optimisation constrained by rims

Figure 5: Comparison of visual hull, global refinement and refinement with rim constraints ((a) taken from a different angle to the surfaces, to provide a better view of the colour)



(a) Original view | (b) VDVH | (c) Local refinement | (d) Proposed method

Figure 6: The results of this method compared to a previous local refinement method. Image (c) shows the depth artefacts associated with local refinement, whereas the global refinement in (d) produces a smooth surface.



(a) Original view | (b) VDVH | (c) Visual hull rims | (d) Refined | (e) Virtual viewpoint

Figure 7: Different stages of the refinement: VDVH is constructed from all views (b), rims are recovered (c) and the VDVH depth map refined in projective ray space (d). A rendered virtual view is shown in (e).

produces a consistent surface with fewer depth artefacts.

Results of the different stages of the method are shown in Figure 7. The refined mesh is a more accurate representation of the surface, as can be seen in the rendered shape. The VDVH in Figure 7(b) gives a coarse shape approximation, while the refined shape constrained by the rims in Figure 7(d) is a more accurate approximation of surface shape. The surface was slightly over-refined around the torso area (Figure 7) due to the lack of rims in that region to constrain the optimisation. Results of the graph cut can be improved by varying the smoothness multiplier or altering the size of the stereo window.

Figures 8 and 9 show novel rendered views of a person using an eight camera studio setup from $45°$ views. The virtual viewpoints in Figure 8 are at the mid-point between two cameras, and show a static actor. The novel view in Figure 9 is fixed and the images show a dynamic sequence of the actor dancing. The rims for the visual hulls were recovered using an $8cm^2$ 3D stereo window, and stereo scores for the depth map optimisation used 9×9 windows on the original images. This window size was chosen instead of something larger due to the wide baseline of the cameras in the studio. The results images demonstrate the high quality of the rendered views, correctly reproducing details of the face and wrinkles in the clothing, from a limited set of cameras in a complete circle surrounding the scene.

6 Conclusions

Refinement of view-dependent surfaces in projective ray space for application in free-viewpoint video has been presented. The method narrows the search space for refinement using the VDVH allowing the use of wide baseline views. Rims are recovered using silhouette contours from the original views by constructing a graph optimisation problem from the boundary of the VDVH. Surface refinement is formulated as a graph optimisation problem in projective ray space with rim constraints from all views. Results demonstrate that using rims reduce artefacts due to excessive refinement in global optimisation. Multiple view image+depth is used to represent the reconstructed scene by adding a depth channel to the captured images.

Free-viewpoint video is rendered at above 25Hz on consumer graphics hardware allowing interactive viewpoint control. Results for a wide baseline studio setup have demonstrated the high quality images possible with this approach. Detailed surface areas in the clothing and face are accurately reproduced in the rendered results.

The work could be improved by adding the concept of uncertainty to the rims to account for calibration and matting errors. For pixel chains where no detailed features exists or visibility of the intervals from the cameras is low the extracted rim will not be very reliable. An additional score could be added to the rims in the global refinement representing the reliability of their location. As with all work using visual hull, segmentation of images is an important area to reduce errors, and further work is needed to optimise the boundary of the silhouette.

7 Acknowledgements

This research was supported by: EPSRC grants GR/M88075 ‘Video-Based Representation of Dynamic Scenes’ and EPSRC GR/S27962 in collaboration with BBC Research and Development; the DTI Technology programme project ’iview: Free viewpoint video for interactive entertainment production’ TP/3/DSM/6/I/15515 and EPSRC Grant EP/D033926, lead by BBC Research and Development (http://www.bbc.co.uk/rd/iview).

References

[1] J.-Y. Bouguet. Camera calibration toolbox for matlab: www.vision.caltech.edu/bouguetj/calib-doc. Technical report, MRL-INTEL, 2003.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, pages 377–384, 1999.

[3] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *Proceedings ACM SIGGRAPH*, 22(3):569–577, 2003.

[4] K. M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[6] B. Goldluecke and M. Magnor. Space-Time Isosurface Evolution for Temporally Coherent 3D Reconstruction. In *CVPR*, pages S–E, Washington, D.C., USA, July 2004. IEEE Computer Society, IEEE Computer Society.

[7] O. Grau. A studio production system for dynamic 3d content. In *Proceedings of Visual Communications and Image Processing, Proceedings of SPIE*, 2003.

[8] T. Kanade and P. Rander. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(2):34—47, 1997.

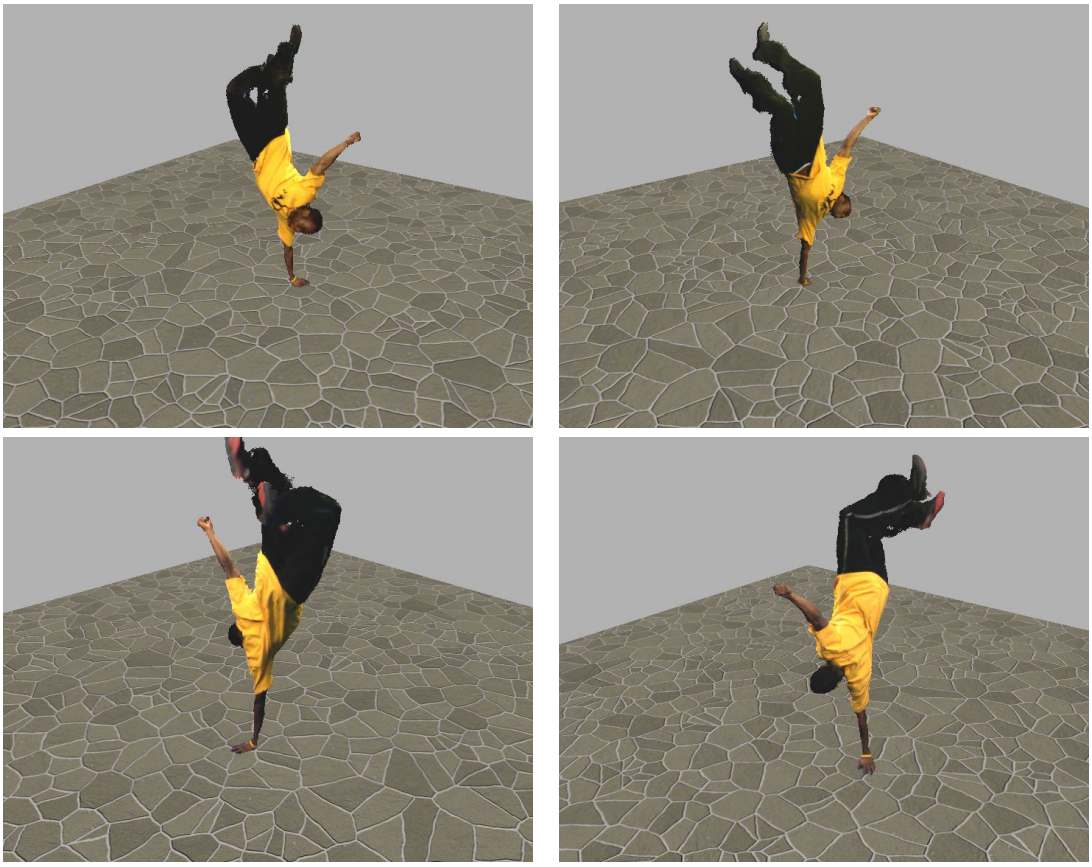


Figure 8: Virtual views rendered around a static subject, each view at the mid-point between two existing views (with a 45° baseline)

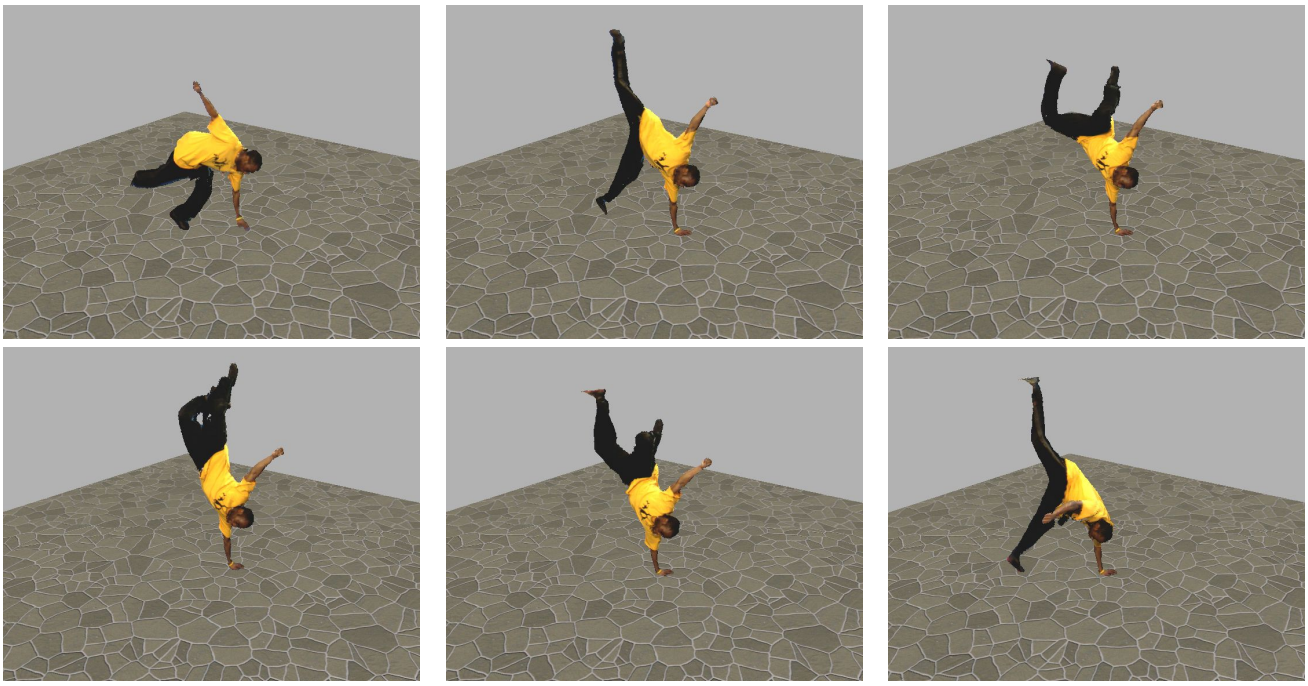


Figure 9: Novel rendered views from a static viewpoint for a dynamic scene, illustrating the high quality this of method (with a 45° baseline).

[9] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV (3)*, pages 82–96, 2002.

[10] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. Technical Report TR692, University of Rochester, 1998.

[11] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, 1994.

[12] 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.

[13] G. Miller and A. Hilton. Exact view-dependent visual hulls. In *Proc. 18th International Conference on Pattern Recognition*. IEEE Computer Society, August 2006.

[14] G. Miller, A. Hilton, and J. Starck. Interactive free-viewpoint video. In *Proc. 2nd European Conference on Visual Media Production*. IEE, November 2005.

[15] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. Computer Vision and Pattern Recognition*, 2006.

[16] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *ICCV*, pages 349–356, 2005.

[17] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *Proc. of the Joint IEEE TCVG and Eurographics Workshop*. Springer Computer Science, 2001.

[18] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *IEEE International Conference on Computer Vision*, pages 915–922, 2003.

[19] J. Starck and A. Hilton. Virtual view synthesis from multiple view video. *Submitted Journal of Graphical Models*, 2003.

[20] S. Vedula, S. Baker, and T. Kanade. Spatio-temporal view interpolation. *Eurographics Workshop on Rendering*, pages 1–11, 2002.

[21] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 391–398, Washington, DC, USA, 2005. IEEE Computer Society.

[22] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *SIGGRAPH*, pages 600—608, 2004.