A Homographic Framework for the Fusion of Multi-view Silhouettes

Saad M. Khan Pingkun Yan Mubarak Shah
School of Electrical Engineering and Computer Science
University of Central Florida, USA

Abstract

This paper presents a purely image-based approach to fusing foreground silhouette information from multiple arbitrary views. Our approach does not require 3D constructs like camera calibration to carve out 3D voxels or project visual cones in 3D space. Using planar homographies and foreground likelihood information from a set of arbitrary views, we show that visual hull intersection can be performed in the image plane without requiring to go in 3D space. This process delivers a 2D grid of object occupancy likelihoods representing a cross-sectional slice of the object. Subsequent slices of the object are obtained by extending the process to planes parallel to a reference plane in a direction along the body of the object. We show that homographies of these new planes between views can be computed in the framework of plane to plane homologies using the homography induced by a reference plane and the vanishing point of the reference direction. Occupancy grids are stacked on top of each other, creating a three dimensional data structure that encapsulates the object shape and location. Object structure is finally segmented out by minimizing an energy functional over the surface of the object in a level sets formulation. We show the application of our method on complicated object shapes as well as cluttered environments containing multiple objects.

1. Introduction

Visual hull based methods have had an enormous impact on a variety of applications including 3D modeling, object localization, object recognition and motion capture applications amongst others. Most of these methods attempt to fuse silhouette information from multiple views in 3D space thereby requiring calibrated views. Camera calibration is itself a challenging problem with a large literature devoted to it. There are many situations in real life where calibration is a cumbersome task that may be best avoided. A common case is when multiple non-stationary cameras (with possibly different internal parameters) are used to capture different views of an object in the absence of a calibration pattern.

In this paper we present a novel approach to silhouette fusion that does not require calibrated views. The method delivers the affine structure of objects which can be augmented with a metric measurement from the scene for full Euclidean structure. In many cases, though, the affine structure would suffice for applications like multi-object localization, object recognition, generation of novel views, motion capturing and activity recognition among others, some of which we demonstrate in this paper. Our approach gets its inspiration from body part reconstruction using CAT (Computed Axial Tomography) scans in medical imaging. The basic method is quite simple, the CAT scanner uses X-rays that penetrate into the body of the object to capture a 2D cross-sectional slice of the object. By moving the scanner along the body of the object a series of successive slices are obtained that are stacked on top of each other to obtain the structure of the object.

In this spirit we consider objects to be composed of an infinite number of cross-sectional slices, with the frequency at which we sample the slices being a variable determining the granularity of the reconstruction. We state the problem of determining a slice of an object as finding the region on a hypothetical plane that is occupied by the object. To this end, we show that by homographic warping of silhouette information from multiple views to a reference view we can achieve visual hull intersection on a $plane$. If foreground information is available in each view then this process delivers a 2D grid of space occupancies: indeed a representation of a slice of the scene objects cut out by the plane. Starting with homographies between views due to a reference plane in the scene (usually the ground plane) we show that homographies of successively parallel planes can be obtained in the framework of plane to plane homologies using the vanishing point of the reference direction (the direction not parallel to the reference plane). This enables us to obtain an arbitrary number of occupancy grids/slices along the body of the object each being a discrete sampling of 3D space of object occupancies. Finally, the slices obtained are stacked up in the reference direction and the object structure segmented out by minimizing an energy functional over the surface of the object.

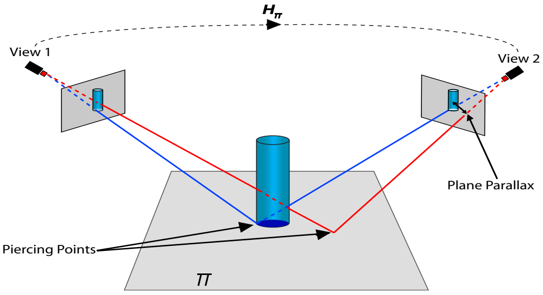


Figure 1. H_π is the homography induced by the planar surface between view 1 and view 2. Warping a pixel from view 1 with H_π amounts to projecting a ray on to the plane at the piercing point and extending it to the second camera. Pixels that are image locations of scene points off the plane have plane parallax when warped. This can be observed for the red ray in the figure.

2. Related Work

Visual hull methods [1] can yield surprisingly accurate shape models. Over the years a number of variants have evolved including surface representation [12] [11], voxel representation [10], or image-based representation [6] [7]. Typically, the approach is to start with an estimate of the silhouettes or boundaries of the object that are projected in 3D space for visual hull intersection (alternatively voxels are projected back to test if the silhouettes carve them out). The process is quite sensitive to segmentation and calibration errors since a small error in even a single view can have a dramatic effect on the resulting 3D model. This is why recent focus has been to move away from deterministic approaches and to make the process more statistical, thereby delay the act of decision making/thresholding to as late in the process as possible [17][5] [2].

Several methods have also been proposed to bypass silhouette estimation altogether, as many algorithms reconstruct the scene structure based only on photometric information [4] [8] [3]. Crucially, this class of methods must deal with the visibility of points on the object's surface (occlusion reasoning) making them more complicated and computationally expensive. This is why there are still many situations where silhouette-based methods are preferred e.g. VR platforms or real-time interactive systems. For further details the reader is directed to an excellent recent survey of the area [13].

The common feature amongst all these methods is the requirement of fully calibrated views and the use of 3D constructs like voxels or visual cones being intersected in the 3D world. Herein lies the novelty of our approach. We present a completely image-based approach that uses only 2D constructs like planar homographies for silhouette fusion *in the image plane* without requiring to go in 3D space.

3. Approach

We begin with a description of planar homographies. Let **x** $= (x, y, 1)$ denote the image location (in homogeneous coordinates) of a 3D scene point in one view and let $\mathbf{x}' = (x', y', 1)$ be its coordinates in another view. Let H_π denote the homography between the two views with respect to scene plane π as depicted in figure 1. Warping the first view onto the second using H_π, the point **x** is transformed to $\mathbf{x}_w$, where $\mathbf{x}_w = [H_\pi]\mathbf{x}$. For scene points on plane π, $\mathbf{x}_w = \mathbf{x}'$, while for scene points off π, $\mathbf{x}_w \neq \mathbf{x}'$. The misalignment $\mathbf{x}_w - \mathbf{x}'$ is called the plane parallax. Geometrically speaking, warping **x** from the first image to the second using homography H_π amounts to projecting a ray from the camera center through pixel at location **x** and extending it until it intersects the plane π at the point often referred to as the 'piercing point' of **x** with respect to plane π. The ray is then projected from the piercing point onto the second view. The point in the image plane of the second view that the ray intersects is $\mathbf{x}_w$. In effect $\mathbf{x}_w$ is where the image of the piercing point is formed in the second camera. As can be seen in figure 1, scene points on the plane π have no plane-parallax, while those off the plane have considerable plane-parallax.

Using this concept, in the next section we show how cross-sectional slices of the visual hull cut out by arbitrary planes in the scene can be obtained by the homographic fusion of multiple silhouettes onto a reference view.

3.1. Obtaining Object Slices

Consider figure 2(a). The scene is viewed from several angles with the cylinder object detected as foreground (white regions) in each view. One of the views, say I_1, is chosen as the reference view. Warping view I_i to the reference view using homography $H_{i_\pi 1}$ induced by scene plane π, first every foreground pixel in I_i is projected to its piercing point on π. This process can be viewed as the foreground object casting a *shadow* on π (an analogy if the cameras are replaced by point light sources), as depicted by the light blue regions in figure 2(a). The shadow is then projected onto the reference view to complete the operation of the homographic warping.

Clearly computing the shadow is equivalent to determining the region on π that falls inside the visual hull of the object image in I_i. The fusion of these shadows projected from various views therefore amounts to performing visual hull intersection on plane π, depicted by the dark blue region in figure 2(a). This process is performed implicitly, when we warp all the views onto the reference view and fuse them to obtain the red region in the reference view I_1. Without loss of generality, reference image plane I_1 after homographic fusion of foreground data can be viewed as a projectively transformed planar slice of the object (strictly

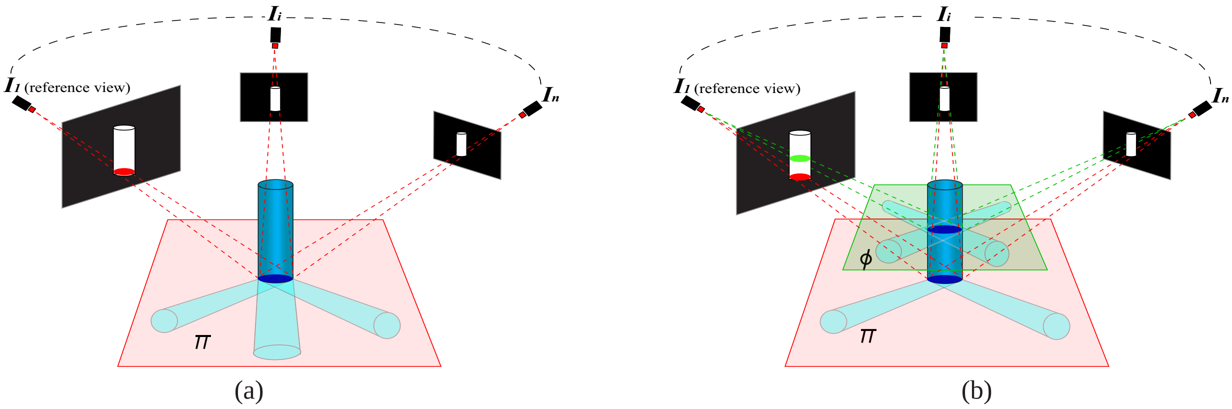


Figure 2. Warping the silhouettes of an object from the image plane to a plane in the scene using a planar homography is equivalent to projecting the visual hull of the object onto the plane. If the camera center is considered as a point light source this can be interpreted as the object casting its shadow on a plane. Figure (a) demonstrates this for a cylinder viewed from different angles. The intersection of these *shadows* amounts to performing visual hull intersection on the plane. The result is the dark blue region that can be considered a slice of the cylinder cut out by π. This process is *implicitly* performed when we warp and fuse silhouette information from other views on to reference view I_1 and is depicted by the red region. (For the sake of clarity projection of the shadows are not shown in the reference view, and only the intersection of these projections i.e. the slice in red is shown). Figure (b) demonstrates that the same process can be performed on a second plane ϕ delivering another slice of the cylinder.

speaking a perspectivity with only 6dof).

In our implementation, instead of using binary foreground maps, we pursue a more statistical approach and model the background [9] in each view to obtain foreground likelihood maps, thereby using cameras as statistical occupancy sensors (foreground interpreted as occupancy in space). In the case of non-stationary cameras, object detection is achieved in a plane+parallax framework [16] assigning high foreground likelihood where there is high motion parallax. The reason to adopt a $soft$ approach is to delay the act of thresholding preventing any premature decisions on pixel labelling; an approach that has proven to be very useful in visual hull methods [17], due to their susceptibility to segmentation and calibration errors. Let us restate I_i as the foreground likelihood map (each pixel value is likelihood of being foreground) in view i of n. Consider a reference plane π in the scene inducing homographies $H_{i_\pi j}$, from view i to view j. Warping I_i's to a reference view I_{ref}, we have the warped foreground likelihood maps: $\hat{I}_i = [H_{i_\pi ref}]I_i$.

Visual hull intersection on π (AND-fusion of the shadow regions) is achieved by multiplying these warped foreground likelihood maps:

$$\theta_{ref} = \prod_{i=1}^{n} \hat{I}_i, \qquad (1)$$

where θ_{ref} is the projectively transformed grid of object occupancy likelihoods. Arguably a more elaborate fusion model can be used at the expense of simplicity, but that is not the primary focus of this research. Indeed, a sensor fusion strategy that explicitly models pixel visibility, sensor reliability, scene radiance as in [2], can be transparently incorporated, without affecting our underlying approach of fusing at slices in the image plane rather than in 3D space.

Each value in θ_{ref} is saying what the likelihood is of this grid location being inside the body of the object; indeed representing a slice of the object cut out by plane π. It should be noted that the choice of reference view is irrelevant, as the slices obtained on all image planes and the scene plane π are projectively equivalent. This computation can be performed at an arbitrary number of planes in the scene, each giving a new slice of the object. Naturally, this does not apply to planes that do not pass through the object's body, since visual hull intersection on these planes will be empty, therefore a separate check is not necessary. Figure 2(b) demonstrates a second slice of the cylinder obtained using our approach.

Starting with a reference plane in the scene (typically the ground plane), we perform visual hull intersection on successively parallel planes in the up direction along the body of the object. The probabilistic occupancy grids θ_is obtained in this fashion can be thresholded to obtain object slices, but this creates the problem of finding the optimum threshold at each slice level. Moreover, the slices have a strong dependency on each other as they are parts of the same object/s, and should as such be treated as a whole. Our approach is to model this dependency by stacking up the slices, creating a three dimensional data structure $\Theta = [\theta_1; \theta_2; \ldots \theta n]$. Θ is not an entity in the 3D world or a collection of voxels. It is, simply put, a logical arrangement of planar slices, representing discrete samplings of the continuous occupancy space. Object structure is then segmented out from Θ i.e., simultaneously from all the slices as a smooth surface that divides the space into the object

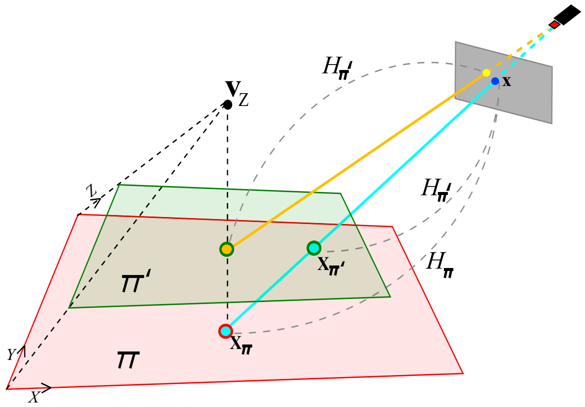


Figure 3. The diagram illustrates the geometrical relationship of the homography of an image plane to two parallel scene planes π and π'. $\mathbf{v}_Z$ is the vanishing point of the direction normal to π and π'. Given the homography H_π from the image plane to π, $H_{\pi'}$ can be computed by adding a scalar multiple of the vanishing point $\mathbf{v}$ to the last of column of H_π.

and background. The details of this process are delayed until section 4. In the next section, we present an image-based approach using the homography of a reference plane in the scene to compute homographies induced between views by planes parallel to the reference plane.

3.2. Extending to Successive Planes

Consider a coordinate system XYZ in space. Let the origin of the coordinate frame lie on the reference plane, with the X and Y -axes spanning the plane. The Z-axis is the reference direction, which is thus any direction not parallel to the plane. The image coordinate system is the usual xy affine image frame, and a point $\mathbf{X}$ in space is projected to the image point $\mathbf{x}$ via a 3×4 projection matrix $\mathbf{M}$ as:

$$\mathbf{x} = \mathbf{MX} = [m_1 \quad m_2 \quad m_3 \quad m_4]\mathbf{X},$$

where $\mathbf{x}$ and $\mathbf{X}$ are homogenous vectors in the form: $\mathbf{x}$ = $(x, y, w)^T$, X = $(X, Y, Z, W)^T$, and '=' means equality up to scale. The projection matrix $\mathbf{M}$ can be parameterized as:

$$\mathbf{M} = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \hat{\mathbf{l}}],$$

where $\mathbf{v}_X$, $\mathbf{v}_Y$ and $\mathbf{v}_Z$ are the vanishing points for X, Y and Z directions respectively and $\hat{\mathbf{l}}$ is the vanishing line of the reference plane normalized [14].

Suppose the world coordinate system is translated from the plane π onto the plane π' along the reference direction(Z) by z units as shown in figure 3, then it is easy to show that we can parameterize the new projection matrix $\mathbf{M}'$ as:

$$\mathbf{M}' = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \alpha z \mathbf{v}_Z + \hat{\mathbf{l}}],$$

where α is a scale factor. Columns 1, 2 and 4 of the projection matrices are the three columns of the respective plane to image homographies. Therefore, the plane to image homographies can be extracted from the projection matrices, ignoring the third column, to give:

$$H_\pi = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \hat{\mathbf{l}}], \quad H'_\pi = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \alpha z \mathbf{v}_Z + \hat{\mathbf{l}}].$$

In general:

$$H_\gamma = H_{ref} + [0|\gamma \mathbf{v}_{ref}], \qquad (2)$$

where H_{ref} is the homography of the reference plane γ is a scalar multiple encapsulating α and z, [0] is a 3x2 matrix of zeros and $\mathbf{v}_{ref}$ is the vanishing point of the reference direction. Using this result it can be shown (see appendix A) that if we have the homography $H_{i_\pi j}$ induced by a reference scene plane π between views i and j, then the homography $H_{i_\phi j}$ induced by a plane ϕ parallel to π in the reference direction is given by:

$$H_{i_\phi j} = (H_{i_\pi j} + [0|\gamma \mathbf{v}_{ref}])(I_{3x3} - \frac{1}{1+\gamma}[0|\gamma \mathbf{v}_{ref}]). \qquad (3)$$

In our implementation, we used the ground plane as the reference scene plane and the up direction as the reference direction. The ground plane homographies between views were automatically computed with SIFT [18] feature matches and using the RANSAC algorithm [19]. Vanishing points for the reference direction were computed by detecting vertical line segments in the scene and finding their intersection in a RANSAC framework as in [20]. It should be noted that the particular values of γ are not significant, we are only interested in the range of γ for planes that span the body of the object (e.g., if the object is a person, then starting from the ground plane to a plane parallel to the ground plane but touching the tip of the head). The computation of this range for γ is quite straightforward since outside this range visual hull intersection on the corresponding planes will be empty. In the next section, we describe how we segment out the object from the occupancy grid data.

4. Object Segmentation

As described earlier slices computed along the body of the object are stacked, creating a three dimensional data structure Θ that encapsulates the object structure. To segment out the object we evolve a parameterized surface $\mathcal{S}(q) : [0,1] \rightarrow \mathbb{R}^3$, that divides Θ between the object and the background similar to the approach in [3]. This is achieved by formulating the problem in a variational framework, where the solution is a minimizer of a global cost functional that combines a smoothness prior on slice contours and a data fitness score. Our energy functional is defined as:

$$E(\mathcal{S}) = \int_{\mathcal{S}} g\left(|\nabla\Theta(\mathcal{S}(q))|\right)^2 dq + \int_{\mathcal{S}} \left|\frac{\partial \mathcal{S}(q)}{\partial q}\right|^2 dq, \qquad (4)$$

where $\nabla\Theta$ denotes gradient of Θ, and g denotes a strictly decreasing function: $g(x) = 1/(1+x^2)$. The first term at the right side of (4) represents external energy. Its role is to attract the surface towards the object boundary in Θ. The second term, called the internal energy computes, the area of the surface. Given the same volume, smoother surface will have smaller area. Therefore, this term controls the smoothness of the surface to be determined. When the overall energy is minimized, the object boundary will be approached by a smooth surface.

Minimizing energy functional (4) is equivalent to computing geodesic in a Riemannian space:

$$E(\mathcal{S}) = \int g\left(|\nabla\Theta(\mathcal{S})|\right) \left|\frac{\partial\mathcal{S}}{\partial q}\right| dq. \tag{5}$$

With the Euler-Lagrange equation deduced, this objective function can be minimized by using the gradient descent method by an iteration time t as

$$\vec{\mathcal{S}_t} = g\left(|\nabla\Theta(\mathcal{S})|\right)\kappa\vec{\mathcal{N}} - (\nabla g\left(|\nabla\Theta(\mathcal{S})|\right)\cdot\vec{\mathcal{N}})\vec{\mathcal{N}}, \tag{6}$$

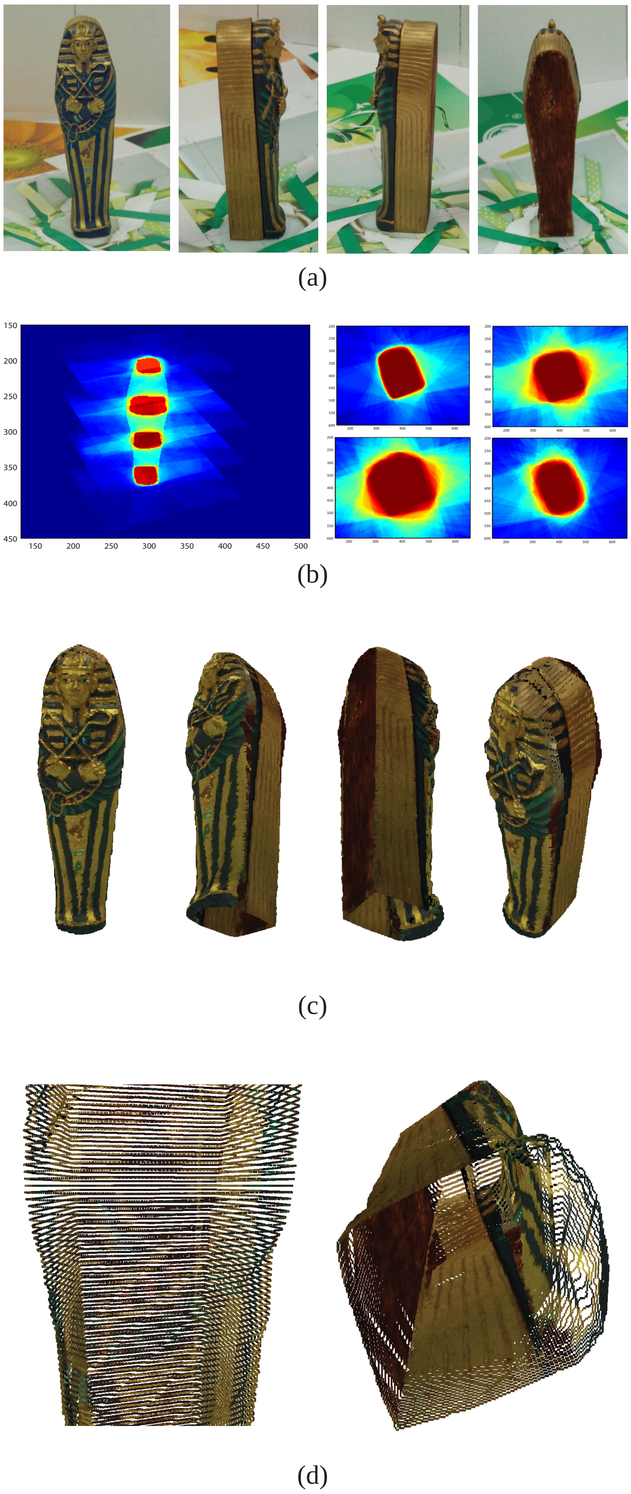
where κ is the surface curvature, and $\vec{\mathcal{N}}$ is the unit normal vector of the surface. Since the objects to be reconstructed may have arbitrary shape and/or topology as shown in our experiments, the segmentation is implemented using the level set framework [15]. Level sets based methods allow for topological changes to occur without any additional computational complexity, because an implicit representation of the evolving surface is used. The solution (6) can be readily cast into level set framework by embedding the surface $\mathcal{S}$ into a 3D level set function Ψ with the same size as Θ, i.e. $\mathcal{S} = \{(x,y,z)|\Psi(x,y,z) = 0\}$. The signed distance transform is used to generate the level set function in our work. This yields an equivalent level set update equation to the surface evolution process in (6):

$$\frac{\partial\Psi}{\partial t} = g\left(|\nabla\Theta|\right)\kappa|\nabla\Psi| + \nabla g\left(|\nabla\Theta|\right)\cdot\nabla\Psi. \tag{7}$$

Starting with an initial estimate for $\mathcal{S}$ and iteratively updating the level set function using (7) leads to a segmentation of the object.

5. Results and Applications

In the absence of metric calibration patterns (as is very commonly the case in natural scenes) lifting the requirement for full calibration in every view and relying on homographies induced by a dominant plane (typically the ground) in the scene can greatly simplify the acquisition process. A popular scenario is when we have a monocular sequence of a single camera flying around the object in an arbitrary and irregular motion path as is the case in the result shown in figure 4. Also due to the inherent computational simplicity of processing 2D data (conducive to graphics hardware



(a)

(b)

(c)

(d)

Figure 4. Object Reconstruction: (a) 4 of the 30 views of a mummy statue used in our experiment. (b) The left image is the foreground likelihood map in the reference view with the fusion of 4 of the 200 slices overlaid. Image on the right are the 4 slices shown in log scale (hotter is higher likelihood). (c) Object structure after segmentation from the stacked slices is rendered with point rendering algorithm together with color mapping from the original images. (d) A closeup of segmented slices. The one on the right is showing a view from the bottom of the object looking up.

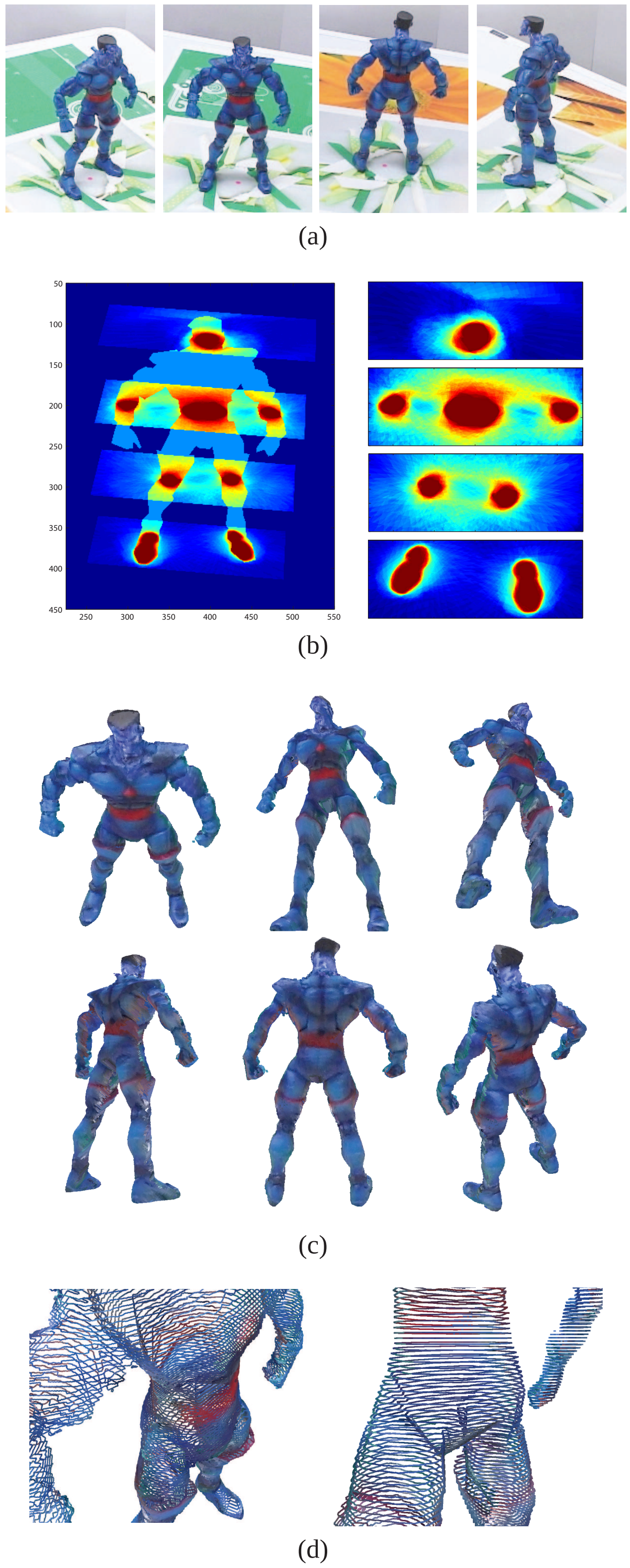


Figure 5. Object Reconstruction: (a) 4 of the 60 views of an action figure model used in our experiment. (b) The left image is the foreground likelihood map in the reference view with the fusion of 4 of the 200 slices overlaid. Image on the right are the slices shown in log scale (hotter is higher likelihood). (c) Rendering of the object structure after segmentation from the stacked slices. (d) A closeup of segmented slices.

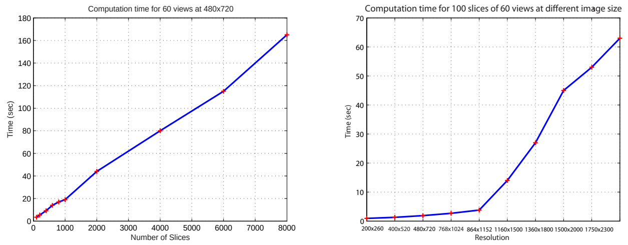


Figure 6. Computation time for homographic fusion on a Nvidia Geforce 7300 GPU. (a) Number of slices vs. Time for 60 views each at 480x720. (b) Image Resolution vs. Time for fusing 100 slices from 60 views.

acceleration) our approach has an advantage over other approaches that perform expensive 3D computations, which can be intractable for complex shapes. Our current implementation of homographic fusion runs on a Nvidia Geforce 7300 GPU. It is capable of fusing 60 views (480x720 pixels) at the rate of 50 slices/second (see figure 6).

Though, it may be debated that we lose robustness by processing data on cross sections of 3D grids and not on volumetric sections in world space. We believe in cases where full calibration is impractical, and computation efficiency is important the advantages of our approach convincingly outweigh the reduction in robustness, if any. This is corroborated by our experimental results and applications, some of which are discussed next.

5.1. Object Reconstruction

Figures 4 and 5 show two of the objects that we used in our reconstruction experiments. The data was captured using a digital camera set at a resolution of 480x720. The mummy sequence in figure 4 is a monocular video captured with the camera flying around the object in a very arbitrary/unconstrained motion path (see video sequence here: www.cs.ucf.edu/ smkhan). The blue model sequence in figure 5 was captured with a camera stationary and the object on a turntable. Figures 4(a) and 5(a) show 4 of the 30 and 60 views used for each object respectively. In figures 4(b) and 5(b) we show the reference views in each sequence overlaid with 4 of the 200 occupancy grids/slices computed for each object. The slices are also shown separately in log scale (hotter is higher likelihood). Figures 4(c) and 5(c) show our reconstruction results. Only contour points of the slice data (after segmentation from Θ) were rendered using a point rendering algorithm. Texture mapping was achieved by reverse warping the slice contour points to the original images for color lookup. Artifacts are visible near the top part of the reconstructions (see head portion of the object in figure 5(c)). These are due to small errors in the homographies of the reference plane that get propagated to homographies of the upper planes. In figures 4(d) and 5(d) we show closeups to emphasize that the reconstruction is slice data rather than a 3D mesh. Notice the detail in which fine curvatures of the objects are captured. It should be pointed out that the result of our method is the affine structure. This is because at no step in the process did we use metric (calibration) information from the scene. Though metric information from the scene can be used to rectify the slice data for full Euclidean structure, it may not be necessary for visualization purposes. For instance, in figure 5(c) we used the typical aspect ratio (height to width) of an adult human male.

5.2. Object Localization and Detection

Our method can be used in much harder conditions for object detection and localization or to initialize a more elaborate photometric method. The presence of high levels of noise, occlusions and low resolution limit the use of the method for precise 3D modeling; however, the method can still be used reliably to locate objects in the scene. Our test data is quite challenging as can be seen in figure 7(a). It is a surveillance scenario containing multiple people viewed by four wide-baseline cameras covering a relatively large area (parking lot). The cameras have different resolutions and aspect ratios (240x360, 240x320), gamma corrections and the scene has considerably poor contrast, causing noisy background subtraction. The most challenging feature, though, is the severity of inter-occlusions between people limiting the visibility. Due to low resolution on the objects (approx 50 pixels in the longer direction) only a small number of slices could be meaningfully generated. We limited our results to 25 slices. Despite all these factors our method was able to generate surprisingly good results as shown in figure 7(b). Though there are a few artifacts (regions not pruned by visual hull intersection), these can be resolved by increasing the number of views.

6. Conclusion

In this paper we have presented an image-based visual hull approach for fusing foreground silhouette information from multiple views. Unlike other visual hull based methods that require calibrated views and use 3D constructs like voxels, 3D visual cones or polygonal meshes, our method uses only minimal geometric information i.e. homographies between views and the vanishing points of a reference direction. We perform visual hull intersection *in* the image plane without without requiring to go in 3D space. Each planar computation delivers object occupancies on a plane representing a cross-sectional slice of the scene objects cut out by the particular plane. Our method also avoids making hard decisions about silhouette labelling in images, which would have required tedious per-image parameter settings. Instead, foreground likelihood values are directly fused and object segmentation is performed on all the slices simultaneously using a gradient based approach.

We have tested our approach on applications including fine 3D structure modelling as well as multi-object localization and detection in cluttered and noisy conditions. Many new ideas and applications can be explored using our method. Different modalities, like infra-red imagery, can be seamlessly integrated into our method. Full body tracking of multiple occluding objects is a direct application that we are currently exploring. Currently, our method assigns equal importance to every view. Considering views with different resolutions, quality, noise levels and vantage points are fused, it makes sense to have a mechanism of assigning different weights based on quality and reliability of data. Another interesting topic that we are currently investigating is the integration of photo-consistency constraints in our framework to eliminate the need for prior detection of silhouette information.

APPENDIX A

Let $H_{i_\pi j}$ be the homography between views i and j induced by scene plane π. Now $H_{i_\pi j}$ can be decomposed as the product of two homographies first from i to π and then from π to j:

$$H_{i_\pi j} = (H_{\pi_{to} j})(H_{i_{to}\pi}). \tag{8}$$

Similarly the homography $H_{i_\phi j}$ induced by a plane ϕ that is parallel to π can be written as:

$$H_{i_\phi j} = (H_{\phi_{to} j})(H_{i_{to}\phi}). \tag{9}$$

Now from equation (2) we have:

$$H_{\phi_{to} j} = H_{\pi_{to} j} + [0|\gamma \mathbf{v}_{ref}]. \tag{10}$$

$$\begin{aligned} & H_{i_{to}\phi} = inv(H_{\phi_{to} i}) = inv(H_{\pi_{to} i} + [0|\gamma \mathbf{v}_{ref}]) \\ = \quad & H_{i_{to}\pi} - \frac{1}{1+g} H_{i_{to}\pi}[0|\gamma \mathbf{v}_{ref}] H_{i_{to}\pi}, \end{aligned} \tag{11}$$

where $g = trace([0|\gamma \mathbf{v}_{ref}] H_{i_{to}\pi})$. Replacing (10) and (11) into (9) we have:

$$\begin{aligned} & H_{i_\phi j} = (H_{\pi_{to} j} + [0|\gamma \mathbf{v}_{ref}])(H_{i_{to}\pi} \\ - & \frac{1}{1+g} H_{i_{to}\pi}[0|\gamma \mathbf{v}_{ref}\mathbf{v}_{ref}] H_{i_{to}\pi}.) \end{aligned} \tag{12}$$

Since $H_{i_{to}\pi}$ is a central projection from one plane to another (2D perspectivity with 6 DOF) the last row is [0 0 1]; therefore, $g = trace([0|\gamma \mathbf{v}_{ref}] H_{i_{to}\pi}) = \gamma$. Plugging this and (8) into (12) and with some matrix algebra we reach:

$$H_{i_\phi j} = (H_{i_\pi j} + [0|\gamma \mathbf{v}_{ref}])(I_{3\text{x}3} - \frac{1}{1+\gamma}[0|\gamma \mathbf{v}_{ref}]). \tag{13}$$

Acknowledgement

We would like to thank Kevin Boulanger for providing the 3D point rendering code and Pavel Babenko for the GPU optimizations. This work is supported in part by the the U.S. Government VACE program. The views and conclusions are those of the authors, not of the US Government or its agencies.

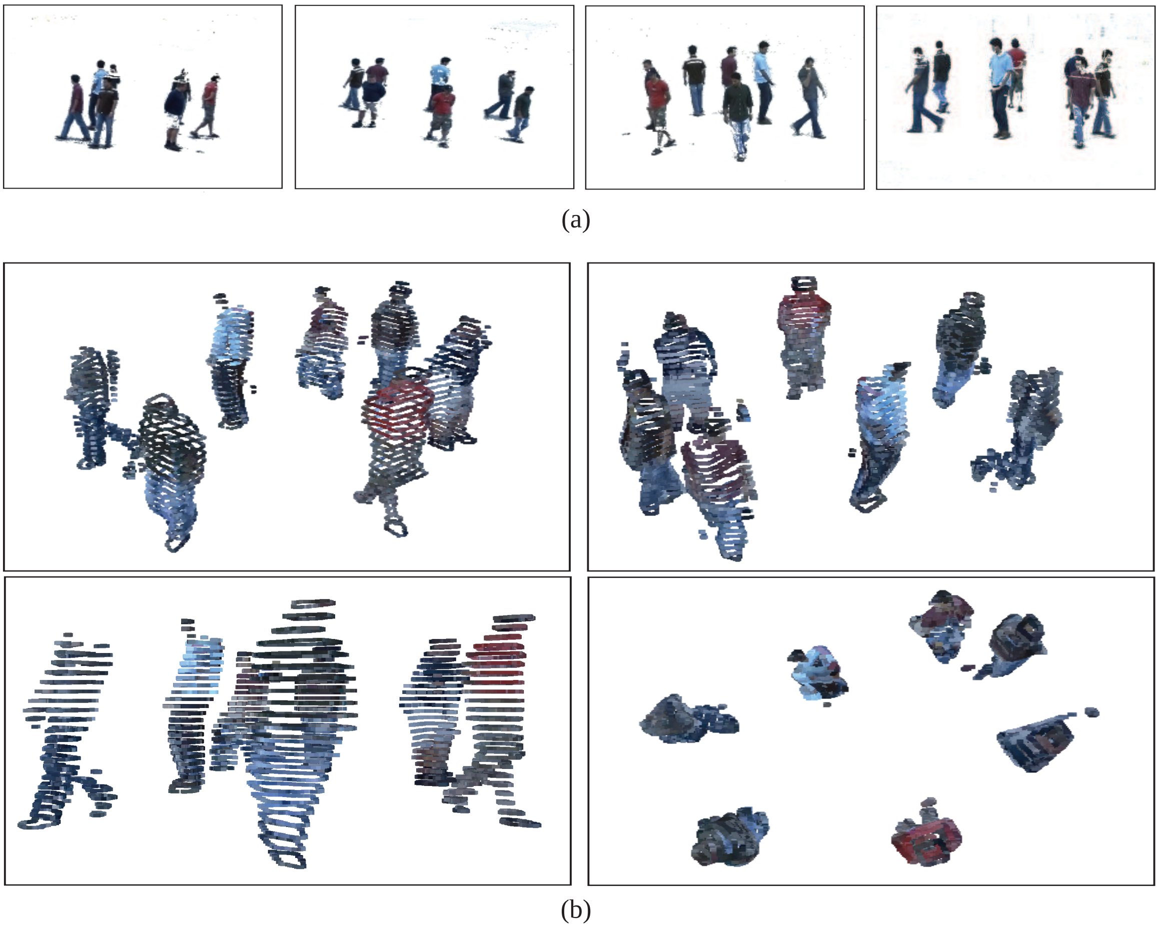
References

[1] A. Laurentini. The Visual Hull Concept for Silhouette- Based Image Understanding. IEEE TPAMI, 1994.

[2] J. S. Franco, E. Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupance Grid. IEEE ICCV, 2005.

[3] A. Yezzi and S. Soatto, Stereoscopic segmentation, IJCV, vol.53(1), pp. 31–43, 2003.

[4] K. Kutulakos and S. Seitz. A Theory of Shape by Space Carving. IJCV, 38(3):199.218, 2000.



(a)

(b)

Figure 7. (a) The scene contains seven people and is viewed by 4 cameras. Notice the low contrast in the scene that makes background subtraction quite noisy and cluttered. In (b) we show the results of our method. Only 25 slices were computed yet the localization and reconstruction is quite good. Also notice the artifacts in the form of ghost objects. These are due to the lack of visibility created by the inter occlusions and limited number of views. The bottom right image is a top view.

[5] K. Grauman, G. Shakhnarovich, and T. Darrell. A bayesian approach to image-based visual hull reconstruction. IEEE CVPR 2003.

[6] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image Based Visual Hulls. In ACM Siggraph 2000.

[7] K. Wong and R. Cipolla. Structure and motion from silhouettes. IEEE ICCV 2001.

[8] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In CVPR, 2000.

[9] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In CVPR, 1999.

[10] R. Szeliski. Rapid Octree Construction from Image Sequences. Computer Vision, Graphics and Image Processing, 58(1):23.32, 1993.

[11] S. Sullivan and J. Ponce. Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs using Triangular Splines. IEEE TPAMI, 1998.

[12] S. Lazebnik, Y. Furukawa, J. Ponce. Projective Visual Hulls. IJCV 2006.

[13] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. IEEE CVPR 2006.

[14] A. Criminisi, I. Reid and A. Zisserman. Single View Metrology, IJCV, 1999.

[15] S. Osher, and J. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations Journal of Computational Physics 1988.

[16] M. Irani and P. Anandan. A Unified Approach to Moving Object Detection in 2D and 3D Scenes. IEEE TPAMI 1998.

[17] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for the Space Carving algorithm. In ICCV, 2001.

[18] D. G. Lowe. Distinctive Image Features from Scale Invariant Keypoints, IJCV, 60, 2 (2004), pp. 91-110.

[19] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision, Cambridge University Press.

[20] C. Rother. A new approach for vanishing point detection in architectural environments. In BMVC, 2002.