Interactive Modelling for AR Applications

John Bastian* Ben Ward† Rhys Hill‡ Anton van den Hengel§ Anthony Dick¶

School of Computer Science, University of Adelaide, Australia

Figure 1: Our method allows the user to recover the 3D shape of a selected object and insert copies of the object into the AR environment.

ABSTRACT

We present a method for estimating the 3D shape of an object from a sequence of images captured by a hand-held device. The method is well suited to augmented reality applications in that minimal user interaction is required, and the models generated are of an appropriate form. The method proceeds by segmenting the object in every image as it is captured and using the calculated silhouette to update the current shape estimate. In contrast to previous silhouette-based modelling approaches, however, the segmentation process is informed by a 3D prior based on the previous shape estimate. A voting scheme is also introduced in order to compensate for the inevitable noise in the camera position estimates. The combination of the voting scheme with the closed-loop segmentation process provides a robust and flexible shape estimation method. We demonstrate the approach on a number of scenes where segmentation without a 3D prior would be challenging.

Index Terms: I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Shape H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 INTRODUCTION

Augmented Reality applications require a degree of scene understanding in order to insert synthetic objects and annotate real objects in the scene. The acquisition of 3D models from video has a long history as a core problem in computer vision; but, understandably, the emphasis in most approaches is on obtaining the most accurate model. Such models are typically obtained by an off-line process using global optimisation over a collection of images. Off-line processes are often ill-suited to the requirements of augmented reality applications which require a 3D reconstruction to be captured in the environment where it is used. Augmented reality applications therefore require a process for rapidly recovering 3D shape in an on-line process. This paper addresses the problem of recovering a 3D model of an object from video in real time. Importantly, this allows the operator to receive feedback about the current estimate of the 3D shape and interact with the model while it is being captured.

The main contribution of this paper is a process for segmenting an object in live video, despite the object's appearance changing as the camera is moved around the object. Previous approaches based only on colour distributions of the foreground and background often cannot reliably segment objects in this instance. In contrast, our approach captures the silhouette of the object from previous frames to make informed predictions about the object's segmentation in future frames. By exploiting this accumulated segmentation information it is possible to achieve robust segmentation results in difficult, cluttered, environments.

The result is a system for image based modelling that only requires the user to 'paint' an object with the camera. Our system provides real-time feedback on the estimated shape of the model, effectively guiding the user to appropriate positions to reconstruct the object. Figure 1 is an example of our approach, illustrating tracking, user-selection, carving and compositing synthetic copies which are correctly occluded by the real object (on the left side of the final frame).

1.1 Related work

Our approach to recovering 3D shape is based on the family of methods known as shape from silhouette. These approaches represent the visual hull by a voxel grid which is recovered by intersecting the back projections of its silhouettes in several images [7]. A common limitation of such methods is their sensitivity to errors in the silhouettes extracted from the images. This is overcome in [17] by using a foreground and background colour model for each voxel as a soft constraint on membership of the 3D model. A graph cut is then used to find the optimal assignment of voxels to the model. As this is a global optimisation, it requires that all data is captured before it begins.

The voxel colouring approach [16] extends shape from silhouette by estimating a colour and alpha for each voxel instead of a binary membership value. However, this relies on strong constraints such as photoconsistency and global visibility calculations, making it difficult to achieve in real-time and for general video.

*e-mail: john.bastian@adelaide.edu.au
†e-mail: ben.ward@adelaide.edu.au
‡e-mail: rhys.hill@adelaide.edu.au
§e-mail: anton.vandenhengel@adelaide.edu.au
¶e-mail: anthony.dick@adelaide.edu.au

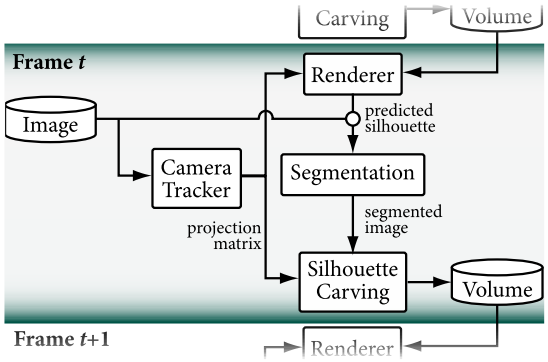


Figure 2: A pipeline overview of our approach. Note that the model update and segmentation processes inform each other over successive frames.

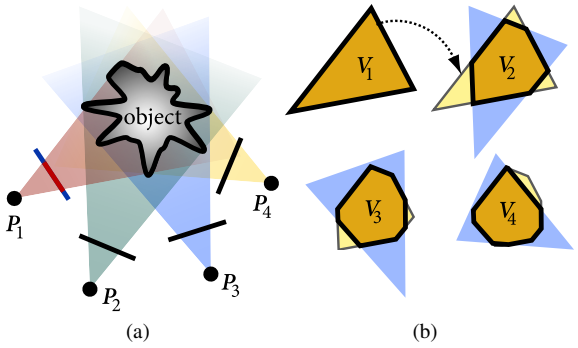


Figure 3: The visual hull is constructed from the intersection of the set of silhouettes back-projected into world-space.

A combination of silhouette and photoconsistency constraints is used in [19] to recover 3D shape and appearance from multiple views. However, this method is based on global optimisation and requires votes from several cameras to calculate a cost function for each voxel, and therefore cannot operate while the video is being captured.

Representing a 3D model as a mesh rather than voxels, Isidoro et al. [5] iteratively estimates the geometry of the mesh and its texture by perturbing the mesh vertices and measuring the resulting change in photoconsistency. Our approach also iterates between the estimation of appearance and shape, but does not require that all images have been captured before our method is run. We also avoid the use of photoconsistency which is a strong constraint that is often violated in real scenes.

A number of approaches to the problem of in-camera modelling have been devised. Chekhlov et al.[4] and Klein et al.[6], for instance, detect planes in the scene in order to calculate how synthetic objects must be rendered within the real environment. The model in this case is very simple, representing a by-product of the system rather than its goal. Many of the in-camera modelling methods come from the Augmented Reality literature, where the approach represents a version of the *immersive modelling* problem (see [8] for a survey). The modelling facilities in these systems are typically not designed to create models that accurately represent objects in the world, however, and using them for this purpose can be somewhat laborious [14, 1, 9].

None of these systems perform any analysis of the image data, meaning that the modeler must fully specify all aspects of each object. Bunnun et al. in [3] propose an image assisted modelling process using a camera attached to a mouse, but this requires that each vertex in the model is individually specified in multiple images. In [18], van den Hengel et al. present a method for interactively generating 3D polygonal models using an image-based modelling process exploiting footage taken from the live video. The method requires significant mouse-based interaction, and is therefore not suitable for in-camera implementation.

Lepetit and Berger in [10] approached the problem of segmenting objects in live video by modelling the required scene geometry in two static views. The authors make use of stereo geometry to estimate the motion of the camera and its uncertainty. The user-specified boundary is then projected into the current view and refined in order to inform the occlusion rendering process. This method produces good results and does not require a pre-existing model, but it does not run in real-time. Mooser et al. [11] have built upon Lepetit and Berger's approach but use optical flow to transfer an object boundary into a new image.

Pan et al describe a system for on-line modelling by moving an object in front of a stationary camera [13]. The structure-from-motion point cloud is converted into a convex hull of tetrahedra via Delaunay tetrahedralisation. The object's surface is recovered by removing tetrahedra whose surfaces do not support the inclusion of observed feature points. Their system is able to recover geometry in real-time, but accuracy is only possible for sufficiently textured, faceted objects. Furthermore, as their system must segment objects via background subtraction, it is not directly amenable for recovering geometry from a moving camera.

Newcombe et al recover dense reconstructions in real-time from a single moving camera [12]. Their approach estimates a 'base-surface' from 3D feature-points that is used to predict the position of the surface in new views. The displacement between the predicted locations and observed scene-flow is used to update the surface. Although the base-surface is initialised via a level-set through feature-points, scene-flow is used to recover per-pixel depth estimates and is therefore not as reliant on features as Pan's approach.

A significant advantage of the method we propose is that it is not necessary to interactively model the foreground object. The user is required only to point the camera at the object as the camera is moved, reducing the required interaction to the point whereby the camera can be used as the interface. Whereas Newcombe et al recover continuous, dense surfaces, our approach is aimed at recovering specific objects that are segmented from the background. This has the advantage that the user can manipulate a specific object, including applying transformations and copying to it, or by adding the object to a library of pre-existing objects that can be readily identified in future images. Our system requires only that the object can be segmented from the background on the basis of colour distributions. Consequently, unlike Pan's approach, we neither require a stationary camera nor do we assume the multi-planar scenes with a sufficiently detailed, trackable texture. We demonstrate this advantage by recovering smooth, partially reflective surfaces that cannot be reliably tracked.

2 Method

Our aim is to estimate the 3D shape of a user-selected, rigid object from a single moving camera. In order to make this process tractable in real-time from a hand-held camera, we make a number of assumptions about the 3D model: namely, we assume that the entire structure may be encapsulated by a bounding volume and that its boundary can be identified with each image. The implications of these constraints is that our method cannot represent objects with infinite extent where every pixel of the video back-projects to the object of interest. This approach, however, greatly simplifies the problem of representing the object's 3D structure while being amenable for a variety of augmented reality applications. Finally, we aim to quickly capture the object while allowing it to be ma-

(a) (b) (c) (d) (e)

Figure 4: The user selects the object of interest by 'painting' the object, either with the mouse or by moving the camera while ensuring the object's projection lies within a designated area of the image. The inclusion brush (indicated by green) adds pixels to the initial segmentation. The user may be required to correct the segmentation in some cases, marking out erroneous foreground pixels with the red exclusion brush.

nipulated by the user, therefore capitalising on the immediacy of augmented reality applications.

Our approach is based on the observation that the projection of every point on an object's surface must lie within its silhouette in all views [7]. Beginning with a super-set of the model's shape, silhouette carving progressively refines this surface by comparing the projection of each point against the set of silhouettes. Points that fall outside a silhouette in any view are removed until the volume converges to the object's *visual hull*, the intersection of all silhouettes that have been back-projected into world-space.

Our approach extends this idea into the domain of augmented reality applications, where a user captures an object with a hand-held camera. Part of the novelty of our approach is that segmentation is informed by current shape estimate, and this silhouette is used to update the shape estimate. This feedback loop, illustrated in Figure 2, makes our approach significantly more robust to errors in camera pose estimation and segmentation, which would otherwise critically undermine the quality of the recovered 3D shape.

2.1 Overview

Our goal is to recover a volumetric model $\mathcal{V}$ of an object from an image set $\{I_t\}$ taken from different perspectives with associated projection matrices $\{\mathbf{P}_t\}$. The volume model $\mathcal{V}$ is defined by a set of binary-labelled voxels $\mathcal{V} = \{v^i\}$, $i \in [1,N]$ which are mapped into world-space by the transformation $W(v^i) \rightarrow \Re^3$. Each voxel has an associated label denoting whether or not it is part of the object's 3D shape, where $v^i = 1$ denotes that the space spanned by the voxel v^i in world-space is occupied by the 3D surface and $v^i = 0$ denotes that the space is unoccupied.

Silhouette carving is the process of labelling free-space by comparing the volume's projection against the set of observed silhouettes of the model. A voxel $v^i \in \mathcal{V}$ is not part of the 3D object if

$$\{\mathbf{P}_t W(v^i)\} \cap \mathcal{B} \neq \emptyset \qquad \forall t \in [0,T] \tag{1}$$

where $\mathcal{B}$ is the set of pixels in frames that do not back-project to points on the object. This equation describes the evolution of the object's *visual hull* and is illustrated in Figure 3.

Only a single background pixel is required to prove that a voxel v^i does not belong to the model and may therefore be irrevocably removed. As a consequence, the 3D shape is critically dependent on accurate segmentation and estimation of the camera-pose. In practice, however, it is often difficult to accurately estimate the camera-pose and reliably segment the object for every frame, particularly in real-time. Unfortunately, these errors would irrevocably destroy the greedy volume update step described in (1).

In order to reconstruct a stable 3D model despite these errors, we use a number of techniques to guide the segmentation process and to account for the uncertainty in the camera-pose estimation. Firstly, rather than irrevocably remove a voxel given only one observation where it falls outside the silhouette, we exploit the fact that we have a video stream by requiring consensus over a number of frames. Our second technique guides the segmentation by using the current estimate of the volume model to predict the model's silhouette in a given image. These techniques makes the volume significantly more robust to errors in segmentation and camera pose estimation while requiring only a small number of additional frames until the volume converges to the visual hull.

2.2 Camera tracking

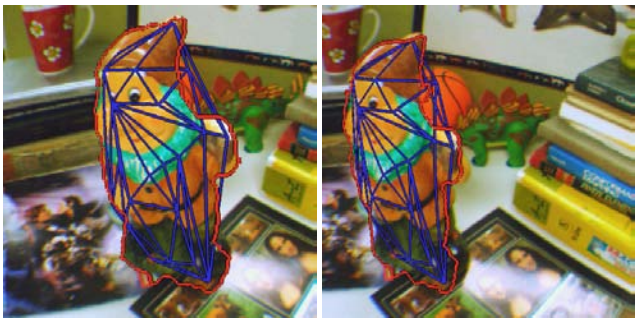
Estimating the pose of the camera is critical for predicting the object's silhouette in a given image, updating the 3D model on the basis of the silhouette, and finally to render synthetic imagery as part of the augmented reality environment. We use the Parallel Tracking and Mapping (PTAM) method of Klein and Murray [6] to estimate the projection matrix $\mathbf{P}_t$ corresponding to the image I_t. PTAM combines feature tracking with incremental bundle-adjustment, and is capable of tracking a single camera at interactive frame-rates while building maps with in excess of $10,000$ 3D feature points.

The key insight behind PTAM is that tracking the camera can be decoupled from estimating the map. The camera's pose is estimated by minimising the displacement between the predicted image of the map and the tracked points in the current frame. In parallel, but not at frame-rate, the map is estimated by applying bundle adjustment to a set of key-frames from the original video. This allows a significantly larger map of the environment to be maintained while still providing a camera trajectory estimate at a satisfactory frame rate. By applying PTAM to the live video, we obtain an estimate of the current camera relative to a fixed world coordinate system and a map of 3D scene point locations.

2.3 Segmentation

The object of interest must be segmented from every image I_t in order to update the volumetric model on the basis of the current camera pose estimate. To segment the object from its background, we must first identify which of the numerous potential objects in the scene is the one that we wish to model. We solve this problem by requiring the user to paint the object in the video. This process is used to initialise a colour descriptor of the object and its background which may then be used to segment the object in subsequent frames.

There has been considerable success in using foreground and background colour descriptors for segmenting images. Rother's Grab-Cut [15], for example, is often able to extract a region of interest with only a user-specified bounding box. Segmentation methods based on colour alone, however, are not directly suitable for segmenting an object from a video stream because the object and the background colour distributions are likely to change as the camera moves around the object. Furthermore, Grab-cut relies on a



(a) Initial model. (b) Predicted position.

Figure 5: An example of the initial triangle mesh and its predicted projection in a subsequent frame.

(a) (b)

Figure 6: Only pixels within a band around the predicted silhouette are classified. By default, pixels within the band are labelled foreground (marked in green) and pixels outside the band are labelled background (marked in red).

user-specified bounding box as a strong foreground prior, which is not available when the object's projection changes as the camera is moved.

To address the limitations of Grab-Cut for model segmentation over video, we use the current estimate of the model's shape as a prior on its silhouette in subsequent frames. We use two different models to represent the current estimate of the 3D object. The first model is a triangle mesh which is used to bootstrap the process while more information about the underlying structure can be collected, but is replaced by a volumetric model when the object has been seen from a sufficient number of views. Irrespective of which model is used, its projection defines a foreground prior which is used to guide Grab-Cut's segmentation.

2.3.1 Initial segmentation

Our approach allows the user to mark out the object of interest by 'painting' onto the live video stream. In practice, we use the mouse to scribble over the object within the image. The user may, however, select the object by moving the camera while ensuring the object's projection lies within a designated area of the image. An example of this process for initially segmenting an object is illustrated in Figure 4. Here, a user selects an object with the 'inclusion' brush by painting green strokes with the mouse 4(b), giving the initial segmentation 4(c). The exclusion brush may be used to correct any errors 4(d) until the segmentation converges to the correct outline 4(e). For comparison, the toy in Figure 1 was painted by appropriately moving the camera rather than requiring mouse interaction.

Painting the object serves as an initial estimate of the set of foreground and background pixels for the initial frame. As in Grab-Cut, we build a set of foreground and background colour distributions in the form of a Gaussian Mixture Model for each class by sampling pixels within and outside the painted regions respectively. The image I_t is segmented into foreground pixels $\mathcal{F}$ and background $\mathcal{B}$ by finding the set of labels I^α where $I^\alpha(\mathbf{p}) \in \{\mathtt{F},\mathtt{B}\}$ by minimising

$$J(I^\alpha) = \sum_{I_\alpha \mathbf{p}} \psi(\mathbf{p}) + \sum_{\mathbf{p}} \sum_{\mathbf{q}\in N(\mathbf{p})} \vartheta(\mathbf{p},\mathbf{q}). \tag{2}$$

The unary potential $\psi(\mathbf{p})$ measures the cost of assigning a particular label to $I^\alpha(\mathbf{p})$, where

$$\psi(\mathbf{p}) = -\log \Pr(I(\mathbf{p}) \in \mathcal{L} \,|\, \mathcal{C}_\mathcal{L}). \tag{3}$$

Here, $\mathcal{L}$ is either the foreground $\mathcal{F}$ or background $\mathcal{B}$ sets with associated colour-model $\mathcal{C}_\mathcal{L}$ depending on whether $I^\alpha(\mathbf{p})$ is labelled either F or B respectively. The probability that a pixel $\mathbf{p}$ belongs to a set with colour model $\mathcal{C} = \{\{\mu_1,\Sigma_1,\omega_1\},\cdots,\{\mu_K,\Sigma_K,\omega_K\}\}$ is given by

$$\Pr(I(\mathbf{p}) \,|\, \mathcal{C}) \propto \prod_n \omega_n \frac{e^{-\frac{1}{2}(I(\mathbf{p})-\mu_n)^\top \Sigma_n^{-1}(I(\mathbf{p})-\mu_n)}}{|\Sigma_n|^{\frac{1}{2}}} \tag{4}$$

where ω_n is the weight of the n^{th} colour Gaussian with mean μ_n and covariance Σ_n. The binary potential

$$\vartheta(\mathbf{p},\mathbf{q}) = (I^\alpha(\mathbf{p}) \neq I^\alpha(\mathbf{q}))e^{-\beta\|I(p)-I(q)\|^2} \tag{5}$$

operates over a neighbourhood $N(\mathbf{p})$ of pixels surrounding $\mathbf{p}$, acting as a regularisation term to encourage a contiguous, edge-preserving segmentation. The objective function (2) may be solved using graph-cuts [2] to partition the pixels into two sets, giving an optimal labelling of foreground and background pixels I^α corresponding to the object's segmentation in image I.

2.4 Initial model

As it would be tedious to mark the object in a number of key-frames around the object, we would like to be able to automatically segment the model in future frames. Grab-Cut is unlikely to segment the object when the camera moves because the object will be in different positions of the image, and the background and foreground colour distributions are likely to change. To this end, we introduce a number of model-based priors to guide the segmentation for subsequent frames without requiring further user intervention.

An initial 3D model is constructed to bootstrap the segmentation, allowing the system to collect more observations of the objects which may then be used to evolve the model. Because a single silhouette back-projects to infinity, we cannot estimate a suitable transformation $W(\mathcal{V})$ to map voxels into world-space until the infinite volume is clipped by the frustum of a second view. To overcome this problem, we construct a temporary model on the basis of the 3D feature points estimated by PTAM for predicting the object's silhouette until a volume may be created.

A triangular mesh is used as a proxy for the volumetric model until the 3D shape can be contained. We assign a depth label for all points within the initial segmentation by triangulating PTAM's 3D scene points whose projection lies within the silhouette. A Gaussian kernel is used along the periphery of this triangulation to hallucinate depths for points within the initial silhouette but outside the convex hull. These depths define points in world-space which are used to extend the surface until it covers the initial silhouette. An example of an initial triangle mesh and the predicted silhouette in a subsequent frame is illustrated in Figure 5.

2.5 Model-guided segmentation

The triangulated depth map is projected into subsequent views to give an approximation of the object's new position. This prediction is used in conjunction with the foreground and background colour models to generate a new segmentation of the object, which is then used to improve the estimate of the model and update the foreground and background colour distributions. The unary potential $\psi(\cdot)$ from equation (2) is modified to include a structure term

$$\psi(\mathbf{p}) = \Pr(I(\mathbf{p}) \in \mathcal{L} \,|\, \mathcal{C}_{\mathcal{L}}) \Pr(\mathbf{p} \,|\, I^{\hat{\alpha}}) \tag{6}$$

where $I^{\hat{\alpha}}$ is the image of the predicted silhouette. The probability distribution

$$\Pr(\mathbf{p} \,|\, I^{\hat{\alpha}}) = \int \frac{\mathcal{E}^{\hat{\alpha}}(\mathbf{q})}{2\pi\sigma^2} e^{-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma^2}} d\mathbf{q}, \tag{7}$$

is designed to weight pixels near the boundary of the predicted silhouette. Here, $\mathcal{E}^{\hat{\alpha}}(\cdot)$ indicates edges in the predicted silhouette, ie. $\mathcal{E}^{\hat{\alpha}}(\mathbf{p})$ is the magnitude of the derivative of $\mathcal{S}^{\hat{\alpha}}|_{\mathbf{p}}$, where $\mathcal{S}^{\hat{\alpha}}(\mathbf{p}) = f(I^{\hat{\alpha}}(\mathbf{p}))$, $f(\mathtt{F}) = 1$ and $f(\mathtt{B}) = 0$.

We only optimise the segmentation within a band of the predicted silhouette to preserve the stability of the segmentation. Two bands of pixels are constructed from the predicted silhouette. The outer band is used to update the background colour model; all pixels contained within are labelled background. The inner region is marked as foreground and is used to update the object's colour model. The segmentation is only performed on the inner band ≈ 10 pixels either side of the projected silhouette. An example of the band around the predicted silhouette and the final segmented result is illustrated in Figure 6.

2.6 Shape from silhouette

The estimate of the current 3D volume is updated by interatively testing the projection of each voxel against the object's silhouette. As described in Section 2.4, we begin with a triangle mesh to represent the surface until a volume can be constructed Until then, we store the segmented images I_t^{α} and the corresponding projection matrices $\mathbf{P}_t$ to be used in conjunction with new views of the object. The set of 3D points that are used to build the initial triangle mesh and subsequent 3D feature-points within the silhouette I_t^{α} are also kept. These points are used to estimate the world-transform $W(\cdot)$ when the intersection of the initial silhouette back-projected into world-space is clipped to a finite volume by a second camera's frustum.

The object's silhouette is used to refine the estimate of the current 3D volume by iteratively testing each voxel and removing those whose projection in I^{α} is marked as background. Given a segmented image I_t^{α}, projection matrix $\mathbf{P}_t$ and previous estimate of the volume $\mathcal{V}_{t-1}$, the volume at frame t is given by

$$v_t^i = v_{t-1}^i f(I_t^{\alpha}(\mathbf{P}_t W(v^i))). \tag{8}$$

This recurrence relationship progressively refines the visual hull beginning with the initial set $\mathcal{V}_0 = \{v_0^i = 1\} \forall i$, representing the superset of the true scene.

The voxel state described by equation (8) is critically dependent on accurate camera estimation and image segmentation because it takes only a single observation of $I_t^{\alpha}(\mathbf{p}) = 0$ for a voxel to be permanently marked empty. In our experience, PTAM occasionally loses track of the camera, leading to errors in the camera pose and consequently errors in the silhouette prediction. The inaccurate silhouette prediction, coupled with potential mis-classification errors from the colour mixture model in $\psi(\cdot)$ due to changes in appearance, means that the object may not be reliably segmented. Furthermore, an inaccurate projection matrix will misalign the back-projection of the silhouette into world-space and is therefore likely

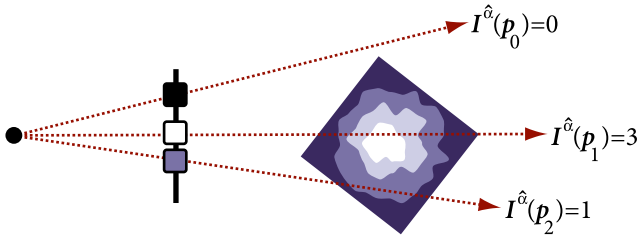


Figure 7: The volume is rendered into predicted image to build a confidence map of the predicted silhouette.

to erroneously remove voxels from the model even if the object has been reliably segmented.

We account for these inaccuracies by requiring consensus over a number of frames before a voxel is pernamently removed. Rather than assigning a binary value to indicate whether or not a voxel is free-space, we associate with each voxel a scalar representing the support for the voxel's inclusion in the volume. In this parameterisation, $v_t^i = 0$ indicates that the voxel is free-space while $v_t^i > 0$ indicates that the voxel is currently believed to be occupied.

Occupancy is modified with each new observation only if the voxel has not been completely removed. Consequently, a voxel that should be part of the 3D volume is able to tolerate a number of votes for its removal without irrevocably destroying the volume, yet a voxel that is consistently rejected will be removed. The modified silhouette-update is given by

$$v_t^i = \mathrm{sgn}(v_{t-1}^i) \left(v_{t-1}^i + \frac{2f(I_t^{\alpha}(\mathbf{P}_t W(v^i))) - 1}{n} \right) \tag{9}$$

where $\mathrm{sgn}(\cdot)$ is the signum function, $v_0^i = 1$, and n is an *a priori* parameter which represents the number of votes required before a voxel is permanently marked as free-space.

2.7 Revised model-guided segmentation

The per-voxel confidence may be used to guide the segmentation prior when the volume has been created, replacing the boolean mask of the initial triangle mesh used in the segmentation prior (7). The new segmented image prior is generated by back-projecting every pixel in the predicted segment image to give a ray for each pixel through the camera's optical centre and the pixel's position in world-space. The predicted silhouette image is given by

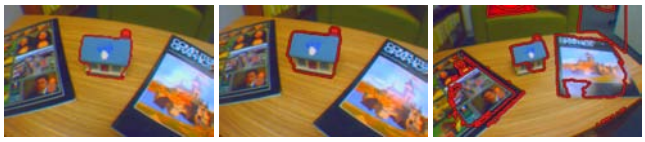
$$I_t^{\hat{\alpha}}(\mathbf{p}) = \max\{v_{t-1}^i \,|\, \mathbf{P}_t W(v^i) = \mathbf{p}\}, \tag{10}$$

where $I_t^{\hat{\alpha}}(\mathbf{p}) = 0$ if the ray corresponding with $\mathbf{p}$ does not intersect the volume. The process of generating the predicted silhouette by traversing along back-projected rays is illustrated in Figure 7. The segmentation prior (7) given the predicted silhouettre remains unchanged, save for replacing the mapping $f(\cdot)$ from foreground/background labels to a scalar with the identity, i.e. $f(x) = x$.

3 Results

Our experiments are obtained with a Unibrain Fire-I camera capturing images with a resolution of 640×480 and equipped with a 2.1mm lens giving a horizontal field of view of 80.95° degrees. The computation was performed on 2.4Ghz Intel Core2 Quad with an *n*Vidia GeForce 8600 where one thread is used for bundle adjustment and mapping and the second thread for tracking and reconstruction. We recover a volume with resolution of 256^3 voxels and require $n = 8$ (from (9)) consecutive votes before a voxel is permanently deleted. The cut-region described in Section 2.5 is defined by a band of 10 pixels from the predicted silhouette, and we

(a) Grab-Cut segmentation without a foreground prior in successive frames.



(b) Grab-Cut segmentation with a foreground prior defined by a 20×20 region in the centre of each frame.

(c) Our approach where the current 3D shape acts as the foreground prior.

Figure 8: A comparison of our model-guided segmentation process against Grab-Cut over successive frames. Note that Grab-Cut's segmentation progressively degrades as the camera is moved, yet our approach is able to reliably segment the object.

use $\sigma = 3$ for the distance prior in (5). The edge weight in (2) is defined by $\beta = (2v)^{-1}$, where v is the variance of all pairwise colour differences between neighbouring pixels over the current frame.

3.1 Comparison with Grab-Cut

Our model-guided segmentation is able to reliably segment the object even in complicated scenes where other segmentation approaches such as Grab-Cut would fail. An example of this is illustrated in Figure 8, which compares our approach against Grab-Cut and a modified variant of Grab-Cut that uses a foreground heuristic. In all cases, the segmentation begins with a user-supplied scribble to indicate the required object; this scribble is used as a foreground prior to build the colour models $\mathcal{C}_\mathcal{F}$ and $\mathcal{C}_\mathcal{B}$. To test Grab-Cut's ability to segment the object throughout an image-sequence without user intervention, $\mathcal{C}_\mathcal{F}$ and $\mathcal{C}_\mathcal{B}$ are used to segment subsequent frames. An example of Grab-Cut's segmentation with the initial colour models is illustrated in Figure 8(a).

We found that Grab-Cut was crucially dependent on an initial foreground prior; but this is unavailable when the object's projection is displaced as the camera is moved. We modified Grab-Cut to assume that the object is always positioned in the centre of the image. The results in this case are illustrated in Figure 8(b). Note that the segmentation slowly diverges throughout the sequence as the apparent foreground and background colour distributions change.

In contrast, our approach is able to segment the object without requiring user-intervention, or by assuming the object projects within a designated area of the image. Figure 8(c) illustrates that our approach segments the object despite a significantly different camera pose with respect to the initial view. Note that the object is segmented even in the case where the object partially occludes the background with a similar colour-model.

3.2 Inserting synthetic geometry

The estimated 3D volume may be used for a variety of augmented reality applications. For example, the volume may be copied and positioned at other locations in the scene; an example of this effect is illustrated in Figure 9. In both cases the real object is on the left

Figure 9: An example illustrating two views an object that has been copied and transformed. The real gnome is on the left of each image; the gnome on the right is a synthetic copy that has been transformed by the user.

Figure 10: Modelling a reflective object; the copy is on the right.

Figure 11: An example illustrating that the volume may be used to occlude synthetic objects. In this case, the synthetic car is correctly occluded by the ambulance volume model in the foreground.

Stage	**Time**
PTAM Tracking	0.0859
Image segmentation	0.0438
Silhouette prediction	*0.0057*
Updating colour models	*0.0016*
Updating graph	*0.0300*
Computing graph-cut	*0.0065*
Volume update	0.0016
Rendering	0.0690
Total	**0.2004**

Figure 12: Timing results (in seconds) for the key stages of our approach.

of each image and its duplicate is on the right. Note that the gnome from Figure 9 has also been scaled, and rotated.

We do not estimate a colour for each voxel, but instead render the model using view-dependent texturing. For a set of key texture frames, we project the corresponding camera's optical centre onto a unit sphere surrounding the modelled object and link the coordinates via Delaunay triangulation over the sphere. For a given frame, we select a triangle from this map by projecting the camera's optical centre onto the sphere. The object is rendered using the textures corresponding to the vertices of the selected triangles weighted by the Barycentric coordinates of the camera's projected optical centre in the Delaunay triangulation.

Synthesised views of the object are rendered by displacing the set of input images via the estimated 3D shape. Selecting view-dependent textures that are closely aligned to the current camera position has the advantage that it minimises distortion between the estimated visual hull and the real object. The appearance of holes in the object, for example, are also rendered back onto the volume as texture. We use the set of PTAM key-frames as view-dependent textures in order to texture the model with a diverse, but not overly dense, set of views. These projective textures have the advantage that their projection matrices are continually improved by bundle adjustment. This process is also able to convincingly render reflective objects such as the bottle in Figure 10. Note that this object could not be modelled by feature- or photoconsistency-based approaches such as those by Pan [13] and Newcombe [12].

The volume may also be used as an occlusion mask. This effect is illustrated in Figure 11 where a synthetic car is occluded by the modelled ambulance. Note that because the ambulance volume is positioned in world-space, the synthetic car is only occluded if it is behind the volumetric model.

3.3 Timing Results

Our timing results for the four key stages of our algorithm are tabled in Figure 12. These results were averaged over 10 frames from the sequence illustrated in Figure 9. The silhouette prediction stage estimates $I^{\hat{\alpha}}$ from Section 2.7 by ray-casting the voxel occupancy weights $\{v^i_{t-1}\}$ on the GPU. The image segmentation stage involves estimating I^{α} for the current frame and includes the time taken to update $\mathcal{C}_{\mathcal{F}}$ and $\mathcal{C}_{\mathcal{B}}$ from (4) and the edge potential β from (5). Note that the time taken to update the graph and compute the optimal partitioning depends on the size of the graph, which in turn depends on the area of the predicted silhouette. In this experiment, the predicted silhouette averaged 12,815 pixels per image. The volume update stage is implemented by a GPU shader over planes swept through the volume to update the occupancy term in (9). Finally, the rendering stage includes the time to generate the view-dependent texture map described in Section 3.2 and composite the volume $\mathcal{V}$ over the input frame.

4 Conclusion

We have described a method for reliably segmenting an object from video by accumulating its projections from previous frames into a view-dependent prior. This approach improves the accuracy and robustness over naïve Grab-Cut without a foreground initialisation. We have shown that both the segmentations and shape estimates generated through this process may be used to copy and paste objects within live video. Work remains to be done in improving the reliability of the SLAM system and improving the 3D through the inclusion of a photoconsistency-based voxel removal process.

References

[1] Y. Baillot, D. Brown, and S. Julier. Authoring of physical models using mobile computers. In *Proc. 5th IEEE International Symposium on Wearable Computers*, 2001.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[3] P. Bunnun and W. Mayol-Cuevas. Outlinar: an assisted interactive model building system with reduced computational effort. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, September 2008.

[4] D. Chekhlov, A. P. Gee, A. Calway, and W. Mayol-Cuevas. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4, Washington, DC, USA, 2007. IEEE Computer Society.

[5] J. Isidoro and S. Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *International Conference on Computer Vision*, pages 1335–1342, 2003.

[6] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07, Nara)*, 2007.

[7] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[8] G. A. Lee, G. J. Kim, and M. Billinghurst. Immersive authoring: What you experience is what you get (wyxiwyg). *Commun. ACM*, 48(7):76–81, 2005.

[9] J. Lee, G. Hirota, and A. State. Modeling real objects using video see-through augmented reality. *Presence: Teleoper. Virtual Environ.*, 11(2):144–157, 2002.

[10] V. Lepetit and M.-O. Berger. A semi automatic method for resolving occlusions in augmented reality. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR '00, Hilton Head Island)*, 2000.

[11] J. Mooser, S. You, and U. Neumann. Real-time object tracking for augmented reality combining graph cuts and optical flow. In *Proc. International Symposium on Mixed and Augmented Reality*, 2007.

[12] R. A. . Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 13-18 June 2010 2010.

[13] Q. Pan, G. Reitmayr, and T. Drummond. ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In *Proc. 20th British Machine Vision Conference (BMVC)*, London, September 2009.

[14] W. Piekarski and B. H. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In *Proc. 5th IEEE International Symposium on Wearable Computers*, 2001.

[15] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[16] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 25(3):151–173, November 1999.

[17] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 345–353, 2000.

[18] A. van den Hengel, R. Hill, B. Ward, and A. Dick. In situ image-based modelling. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando, Florida, USA, October 2009.

[19] G. Vogiatzis, C. Hernández, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246, 2007.