Automatic 3D Object Segmentation in Multiple Views using Volumetric Graph-Cuts ⋆

N.D.F. Campbell [a,*], G. Vogiatzis [b], C. Hernández [b], R. Cipolla [a]

[a] *University of Cambridge, Department of Engineering, Cambridge, CB2 1PZ, UK.*

[b] *Toshiba Research Europe, 208 Cambridge Science Park, Milton Road, Cambridge, CB4 0GZ, UK.*

Abstract

We propose an algorithm for automatically obtaining a segmentation of a rigid object in a sequence of images that are calibrated for camera pose and intrinsic parameters. Until recently, the best segmentation results have been obtained by interactive methods that require manual labelling of image regions. Our method requires no user input but instead relies on the camera fixating on the object of interest during the sequence. We begin by learning a model of the object's colour, from the image pixels around the fixation points. We then extract image edges and combine these with the object colour information in a volumetric binary MRF model. The globally optimal segmentation of 3D space is obtained by a graph-cut optimisation. From this segmentation an improved colour model is extracted and the whole process is iterated until convergence.

Our first finding is that the fixation constraint, which requires that the object of interest is more or less central in the image, is enough to determine what to segment and initialise an automatic segmentation process. Second, we find that by performing a single segmentation in 3D, we implicitly exploit a 3D rigidity constraint, expressed as *silhouette coherency*, which significantly improves silhouette quality over independent 2D segmentations. We demonstrate the validity of our approach by providing segmentation results on real sequences.

Key words: segmentation, multiple view, graph-cut

⋆ Expanded version of a paper presented at the 18th British Machine Vision Conference (Warwick, September 2007).

* Corresponding author.

Email addresses: ndfc2@cam.ac.uk (N.D.F. Campbell), george.vogiatzis@crl.toshiba.co.uk (G. Vogiatzis),

1 Introduction

Recently there has been growing interest in the development of dense stereo reconstruction techniques [1]. These techniques focus on producing 3D models from a sequence of calibrated images of an object where the intrinsic parameters and pose of the cameras are known. In addition to the multiple views, many of these techniques also require knowledge of the object's silhouette in each view to provide:

(a) an approximate initial reconstruction,
(b) an outer bound for the reconstructed object and
(c) approximate occlusion reasoning [2].

This demand for accurate object silhouettes will influence the acquisition of the image sequence to provide an easy segmentation task, for example sequences are taken on turntables against fixed coloured or known backgrounds. Naturally this places limitations on the subject matter for reconstruction since real world objects will often be immovable and there will be no control over the image background which may well vary considerably with differing views of the object. To extend the applicability of multi-view stereo to real world objects it is necessary to be able to segment a rigid object in a sequence of possibly cluttered images.

The most recent advances in image segmentation adopt an interactive approach [3–5] where the user is required to guide the segmentation by manually segmenting image regions. The information supplied by the user is used to learn the object's colour distribution and the globally optimal segmentation is achieved by the maximum flow algorithm [6]. This approach performs well on simple images, but the extent of user interaction required increases significantly if the object contains multiple distinct colours and if it is found in a cluttered or camouflaged environment.

If we now consider the case of a large number of images, required for high accuracy reconstructions, we can see that even a modest amount of user interaction on an individual image basis will represent a significant task using these interactive tools. In order to reduce the demands placed on the user we aim to exploit the constraints that exist within the image sequence. Note that the sequence contains a series of multiple views of the same rigid 3D object, therefore the segmentations must satisfy a *silhouette coherency* constraint [7]. This constraint follows from the knowledge that the images are formed from projections of the same rigid 3D object with the correct segmentations being the corresponding silhouettes. This suggests that by attempting to perform seg-

carlos.hernandez@crl.toshiba.co.uk (C. Hernández),
cipolla@eng.cam.ac.uk (R. Cipolla).

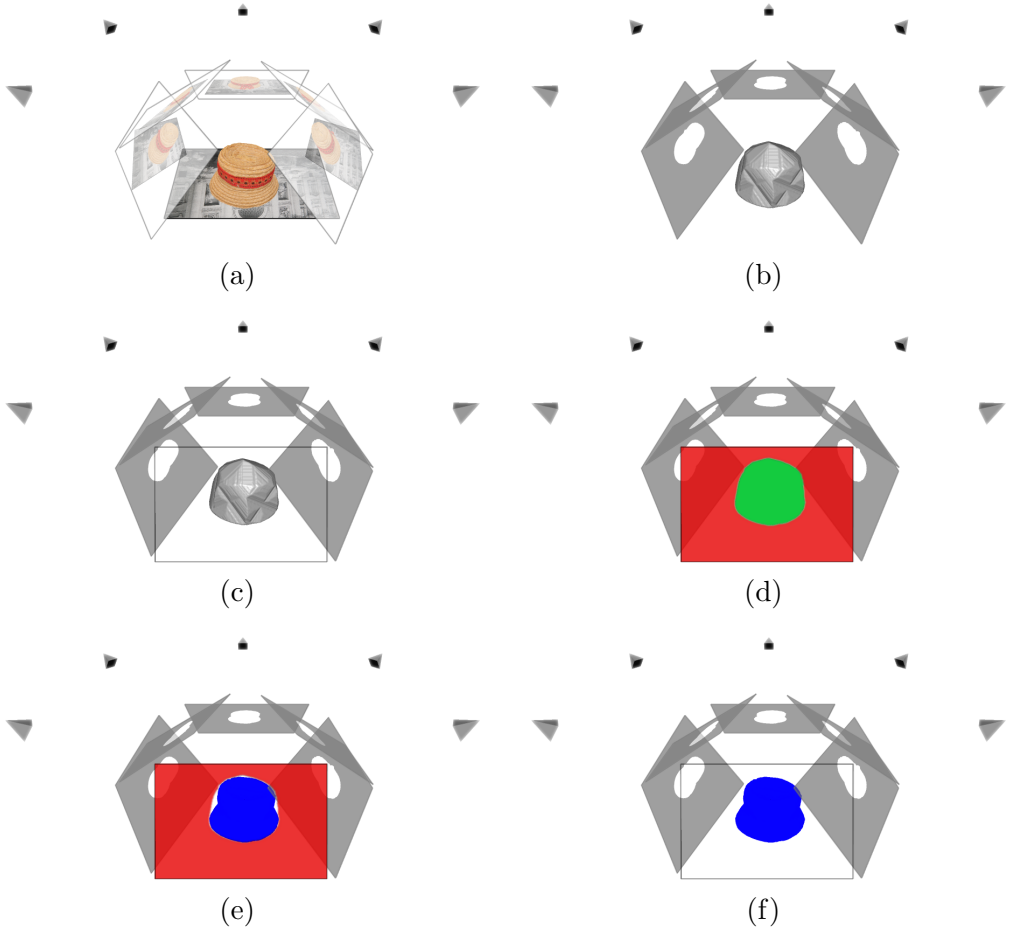


Fig. 1. **Illustration of silhouette coherency.** *If we take a set of views of an object (a) with perfect segmentations then we may form the visual hull of the silhouettes (b). If we then wish to segment a new view (c) the desired silhouette is already constrained to be within the projection of the exising visual hull (d). Thus the range of possible silhouettes is substantially constrained (e) using the knowledge of the other silhouettes making it easier to find the correct silhouette (f).*

mentation on isolated 2D images we are rejecting a great deal of information contained within the sequence. We further observe that, since the intended purpose of the image sequence is to reconstruct the object, we have a *fixation* constraint whereby the object of interest is always being focused upon by the camera and thus likely to be central to the viewpoint.

Silhouette coherency has previously been used for camera calibration [7] under the further constraint of circular motion. Here we use the inverse approach and assume that given accurate calibration we can propagate knowledge across multiple views as segmentation proceeds. The intersection of silhouettes from multiple views forms the *visual hull* which must contain the object which cast the silhouettes (the maximal surface which could generate the silhouettes). Fig. 1 demonstrates the concept of the silhouette coherency constraint.

In order to exploit the coherency constraint we perform the segmentation across all images simultaneously. This is achieved by performing a binary segmentation of 3D space where each 3D location (voxel) is labelled as 'object' or 'background'. The 3D segmentation is influenced by: (a) a colour model initially learned from the fixation points and then refined in subsequent iterations and (b) a 3D shape prior based on minimal surface area.

The advantages of our approach over independent 2D interactive segmentations are twofold. Firstly, we replace the user supplied image labelling with the fixation constraint, which is usually satisfied by multi-view stereo sequences. Additionally we employ constraints that link multiple segmentations of the same rigid object to improve segmentation quality.

The rest of the paper is laid out as follows: In Section 2 we review relevant prior work and discuss the differences of our approach. Section 3 begins with an overview of the segmentation algorithm and continues to provide details of the automatic initialisation procedure. A discussion of how the colour models are learnt follows in Section 4 and Section 5 establishes the volumetric graph-cut framework used to perform the optimisation. In Section 6 we provide two sets of results of real world image sequences. The first shows the complete reconstruction of a hat in a fully automatic process with the user only providing the input images. The second provides a much more challenging segmentation task and we compare the performance of our volumetric approach against automatic 2D segmentations. The paper concludes in Section 8 where the main contributions of the paper are reviewed.

2 Previous Work

The work of [4] introduces a framework for adopting an energy minimisation approach to segmentation where the resulting optimisation process is both tractable and unique thanks to the max-flow/min-cut algorithm. Although the paper details how the method may be extended to N-Dimensional graphs, the predominant focus is on the segmentation of 2D images. They also demonstrate a simple histogram based learning process to allow a user to interactively segment a grayscale image. This discussion is continued in more depth in [8] which also provides a taxonomy of segmentation methods based on their representation of segmentation boundaries and the nature of the optimisation (discrete or continuous).

The approach of [4,8] was continued in [5] where an interactive system for the segmentation of 2D colour images was proposed. As with our method, that paper adopts an iterative approach to the segmentation task by learning colour models of the object and the background. Although considered state-

of-the-art for 2D images, the interactive demands placed on the user make the segmentation of a large sequence of images a sizeable task which we avoid by adopting an automatic approach.

In [9], a graph-cut based approach is used to estimate the voxel occupancy of a calibrated volume in space. Their approach is directly aimed at using an energy minimisation framework to regularise the process of combining a series of imperfect silhouettes. The main difference is that they obtain these silhouettes as the result of a background subtraction process from a fixed, calibrated camera rig whereas we adopt an iterative learning approach requiring no prior knowledge of the object or environment.

The task of segmenting objects in multi-views has also been studied in [10]. Whilst also approaching the task in the 3D domain, here the authors use a level set method to evaluate the segmentation based on Lambertian scenes with smooth of constant albedo. Level set methods are known to be susceptible to local minima so [10] relies on smooth albedo variation and a multi-resolution scheme to achieve convergence. In contrast, our method tolerates albedo discontinuities and, due to the graph-cut optimisation scheme, is guaranteed to converge to the globally optimal segmentation.

The concurrent work of [11] also addresses multiple view segmentation. Similarly to our algorithm they make use of the rigid object constraints to propagate information across views. Whilst they use graph-cuts to perform the segmentation, the images are segmented individually (a 2D graph-cut) rather than the volumetric graph-cut we use to segment over all images at once. This means that silhouette coherence is not guaranteed at each iteration of the algorithm although it is the target for the convergence of the algorithm.

Camera fixation has been used to constrain vision problems, for example to aid recovery of camera motion and shape [12], however we have not seen it used previously to assist segmentation across multiple views.

3 Problem Statement

We accept as an input a sequence of M images, $I_1 ... I_M$, of the object with each image made up of a set of pixels $\mathcal{P}_m$. We assume the images are calibrated which allows any location in the voxel volume, $\mathbf{X} \in \mathbb{R}^3$, to be mapped to its corresponding location in image I_m, $\mathbf{x}_m \in \mathbb{R}^2$. We also assume our fixation condition where the object is taken to be fully contained by and centred in each view.

We form an array of N voxels, $\mathcal{V}$, from the intersection of the volumes projected

by each of the images, thus the fixation constraint mandates that this volume contain the visual hull of the object. Each voxel has dimensions $(\delta x, \delta y, \delta z)$ and may be indexed as $v_n \in \mathcal{V}$. We intend to label each voxel as inside the object's visual hull ($\mathcal{O} \subset \mathcal{V}$) or outside ($\mathcal{B} \subset \mathcal{V}$), and thus part of the background in the image segmentation, such that $v_n \in \mathcal{O} \cup \mathcal{B}$. We also define $\mathbf{u}_{m,n} \in \mathcal{P}_m$ where $\mathbf{u}_{m,n} = I_m(\mathbf{Proj}(v_n))$ is the RGB colour of the pixel which v_n projects to in image I_m. When formulating the voxel array as a graph we define a set of edges $\mathcal{E}$ containing neighbouring voxels in a six-connected sense.

Objective

- Given a calibrated sequence of M images, $I_1 ... I_M$, generate the corresponding silhouettes, for each image, of the object fixated upon by the camera.

Algorithm

- **Initialisation**

(a) Generate array of N voxels $\mathcal{V}$ from bounding volume of the sequence
(b) Extract fixation points for each image to initialise the colour models

- **Iterate until Convergence of Visual Hull**

(a) Construct cost function graph using the volume term (8) and the boundary term (11)
(b) Perform volumetric graph-cut to produce optimal segmentation (2)
(c) Extract the silhouettes from the current visual hull estimate
(d) Update the object colour model (5) using the new silhouettes over all the images
(e) Update the background colour models (6) using the new silhouettes for each image individually

Output

- The converged visual hull is used to extract the object silhouettes which are guaranteed to be coherent.

Fig. 2. The iterative segmentation algorithm.

An overview of our segmentation algorithm is given in Fig. 2. The first stage of the algorithm is to establish a location within the voxel array known to be contained within the visual hull of the object. This is required to provide a starting point for the estimation of the colour model of the object. We make use of the fixation condition previously mentioned to automatically generate this seed location from the centroid of intersection of the optical axes of the

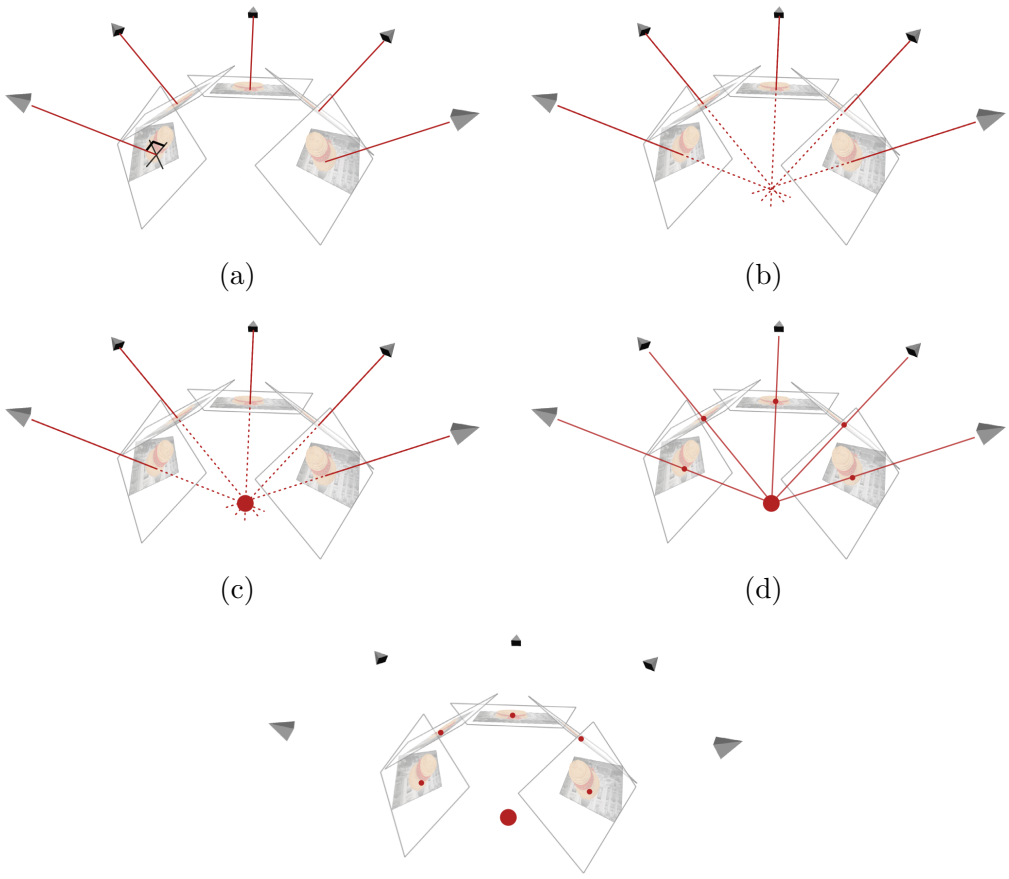


Fig. 3. **Using the fixation constraint for initialisation.** *The optical centres are found in each image (a) and the optical axes projected into the volume (b). The centroid of intersection of these axes is found using a least square method (c) and the centroid back projected into each of the images (d) to find initialisation locations in each image (e).*

individual cameras as shown in Fig. 3. By finding locations which have a large probability of being within the object of interest in the image we can gather sufficient data about the appearance of the object to develop an initial colour model. The initialisation is used to provide the starting point for the second stage of the algorithm, namely the iterative refinement of the visual hull estimation using a volumetric graph-cut scheme. The following two sections discuss the main components of the iterative stage: building the colour models and the volumetric graph-cut.

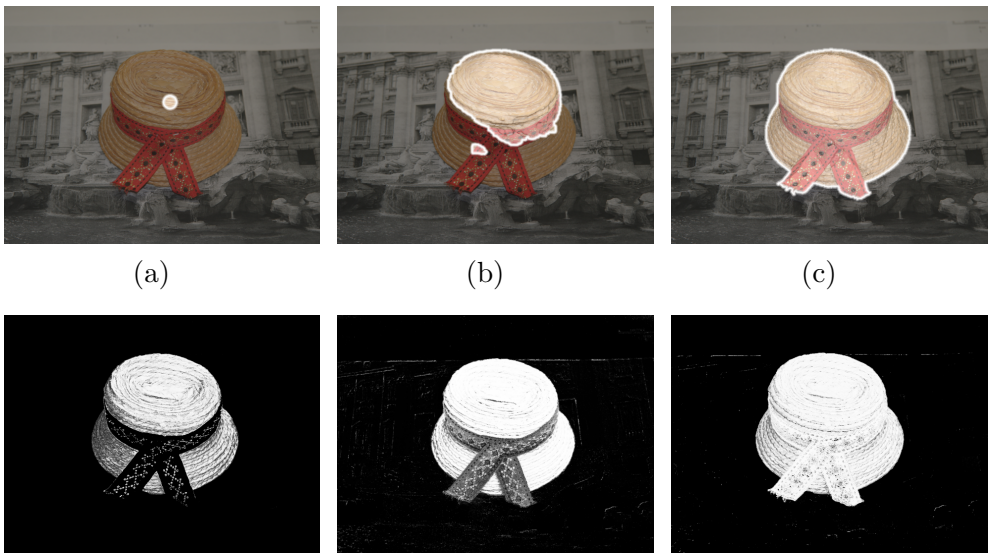
4 Building Colour Models

During the algorithm we develop colour models to provide probabilistic likelihoods for image pixels to be part of the object or background. In common with general practice in this area [5] we use a K component Gaussian Mixture Model (GMM), [13], in 3D colour space (red, green, blue) to model the likelihood. These models take the form of (1) where $\mathbf{u}$ is a vector in colour space. The probability distribution of the colour of the k^{th} component of the mixture is given by a normal distribution with mean μ_k and covariance Σ_k. Each of the individual components is weighted by the marginal probability of the component $p(k)$, termed the mixing coefficient and denoted π_k. The number of mixture components, K, provides a trade-off between computation time and model over-fitting against the discriminative power of the colour model. For our experiments we used $K = 5$.

$$p(\mathbf{u} \,|\, \pi_k, \mu_k, \Sigma_k) \;=\; \sum_{k=1}^{K} p(k)\, p(\mathbf{u} \,|\, \mu_k, \Sigma_k) \;=\; \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{u} \,|\, \mu_k, \Sigma_k) \qquad (1)$$

At each iteration, GMMs are learnt for both the object and the background. The learning process consists of sampling the pixels as a sequence of colour vectors and using the Expectation-Maximisation (EM) algorithm to 'fit' the model parameters of the GMM to the sampled data [13]. We intend to exploit the fact that the object is seen in multiple views, therefore we build a full colour model for the object by sampling pixels from all the views using the silhouettes as a mask. This approach allows us to increase our knowledge about all the colours present in the object even if the initialisation fails to capture all the colours of the object. In this situation an automatic 2D segmentation would most likely fail. However, under our method the visual hull estimation forces a spatial coherency, thus when the graph-cut is performed the segmentation will generate a visual hull whose silhouettes should include a portion of the colour in at least one of the views. Since the colour model is built over all views, we only require one view to register the second colour as object in order to allow the subsequent iterations to add the colour to the model and extend the volume into the other region.

This is demonstrated in the hat sequence of Fig. 4. The initialisation of the colour model, Fig. 4(a), contained only the straw colour in each view, therefore the first iteration object colour model, Fig. 4(d), fails to classify the red ribbon as object. The result of the first graph-cut segmentation, given in Fig. 4(b), attempts to combine the separated straw coloured regions and in the process includes some of the ribbon in the segmentation. The second iteration learns an object model which propagates this knowledge of the ribbon over all the image sequences so that the object colour model includes the ribbon as in



(a) (b) (c)

(d) (e) (f)

Fig. 4. **Iterative learning of the object colour model.** *The fixation condition is used to provide a seed (a) for the first object colour model (d). In all images the seed areas contain only the straw colour. The result of the first graph-cut (b) contains areas of the red ribbon which is incorporated into the colour model (e). The second iteration refines this to produce the correct segmentation (c) with corresponding colour model (f). (d)(e)(f) Show the likelihood of being object* $\hat{\mathrm{L}}_{\mathcal{O}}$ *under the colour model at each iteration (7).*

Fig. 4(e). This is refined in the subsequent iteration to produce the correct segmentation of Fig. 4(c), with the object colour model Fig. 4(f), after the second iteration graph-cut.

In contrast to the object model, we expect the background model to vary over each image, therefore a separate GMM is learnt for the background of each image. Since we are unsure of the final visual hull at the start of the iteration, we conservatively estimate the background by sampling image pixels which aren't currently segmented as the object. In the early stages, particularly during the first iteration, the sampled data will contain many pixels from the object itself. However, the localised object model will necessarily be a better fit to the data in and around the seed location, therefore the more generalised background model will not prevent the initial region from growing.

Since the algorithm is EM based it is possible that convergence will be to a local rather than global optimum. However, since we reject background pixels using spatial consistency, the algorithm is forced to converge to a spatially coherent object.

The use of the GMM provides a good model for object and background when the two are separable in colour space, i.e. they don't contain the same colours.

In our experiments we use $K = 5$ since we found no significant improvement with greater values and increasing K increases the computation time and the risk of over-fitting the colour models to the sampled data. As mentioned in Section 8, we identify the colour models as the main limitation of the algorithm, in particular difficulties are introduced when the same colours are found in the foreground and background as identified by the statue sequence. It is a topic of future work in incorporate more advanced object and background models, possibly involving texture information, to deal with these situations.

5 Volumetric Graph-Cut

The task of segmentation is performed within the voxel array and the resulting silhouettes from the computed visual hull propagated back to the individual images. This ensures the set of image silhouettes are consistent with one another at every iteration. The segmentation operation is one of energy minimisation as in (2). The energy to be minimised, given in (3), is comprised of two terms: a volumetric term and a boundary term. The parameter Θ denotes the collection of colour model parameters , as in (4), which are updated at each iteration using the results of the previous segmentation.

$$\mathcal{O} = \arg\min_{\mathcal{O} \subset \mathcal{V}} \mathrm{E}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) \tag{2}$$

$$\mathrm{E}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) = \lambda\, \mathrm{E}_{\mathrm{vol}}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) + (1 - \lambda)\, \mathrm{E}_{\mathrm{surf}}(\mathcal{O}, \mathcal{B}, \{I_m\}) \tag{3}$$

$$\text{where} \quad \Theta = \left[\{\pi_k^{\mathcal{O}}\}, \{\mu_k^{\mathcal{O}}\}, \{\Sigma_k^{\mathcal{O}}\}, \{\pi_{k,m}^{\mathcal{B}}\}, \{\mu_{k,m}^{\mathcal{B}}\}, \{\Sigma_{k,m}^{\mathcal{B}}\} \right] \tag{4}$$

Fig. 5 displays the voxel graph structure used to perform the volumetric graph-cut. Every voxel in the 3D volume becomes a node in a graph. The nodes are connected to their neighbours by edges (red in Fig. 5) with weights dictated by the boundary cost term, that is the cost of cutting the edge such that one of the voxels is inside the visual hull and the other outside. In addition, every node has an additional edge (black in Fig. 5) to the 'source' and to the 'sink' which correspond to the object (part of the visual hull) and the background (empty voxel) respectively. The weights of these edges are determined by the volume term and are construed as the probability of the voxel being object given the colour models. Since we must partion the graph so that each node is either connected to the source or sink we observe that the object and background links are complimentary which ties in with the probability (each voxel must be either object or background). The graph encodes a Markov random field and it can be solved in polynomial time using the max-flow/min-cut algorithm

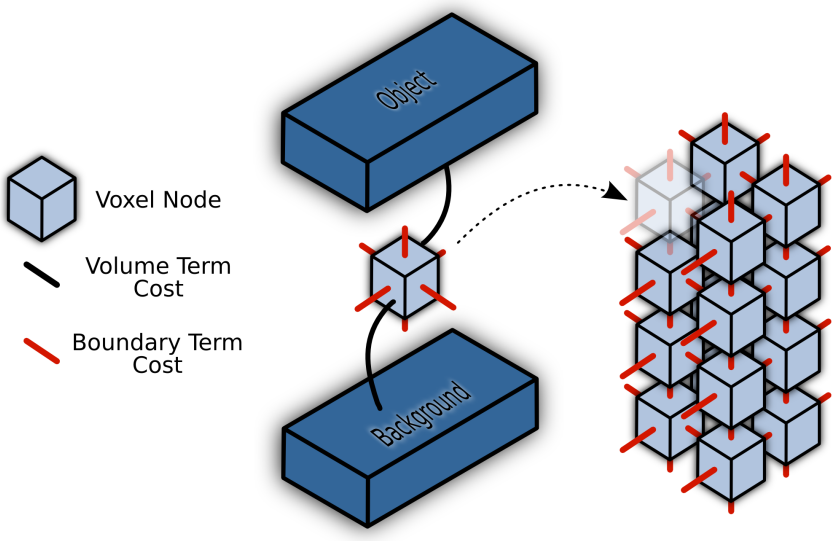


Fig. 5. **The voxel graph structure.**

[6] to produce the globally optimal partition which minimises the energy cost of (2).

5.1 Volume Term

The volume term encodes the preference for a voxel to be classified as inside or outside the object. We therefore construct this term from the colour models of the individual views. For the m^{th} image in the sequence we can evaluate a likelihood term for the projected pixel colour $\mathbf{u}_{m,n}$ of voxel v_n to be part of the object (5) or the background (6) given the current GMM model parameters in the same form as (1).

$$\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m) = p(\mathbf{u}_{m,n} \,|\, \pi_k^{\mathcal{O}}, \mu_k^{\mathcal{O}}, \Sigma_k^{\mathcal{O}}) \tag{5}$$

$$\mathrm{L}_{\mathcal{B}}(v_n, \Theta, m) = p(\mathbf{u}_{m,n} \,|\, \pi_{k,m}^{\mathcal{B}}, \mu_{k,m}^{\mathcal{B}}, \Sigma_{k,m}^{\mathcal{B}}) \tag{6}$$

In the absence of any prior knowledge about the pixels class, we may form a classification probability of being object by normalising the likelihoods. We then sum over all images for a single voxel, normalising by the number of images, as shown by (7). Fig. 6 details the construction of the volume term cost and an example of the final cost is given in Fig. 6(e).

$$\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) = \frac{1}{M} \sum_{m=1}^{M} \frac{\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m)}{\mathrm{L}_{\mathcal{O}}(v_n, \Theta, m) + \mathrm{L}_{\mathcal{B}}(v_n, \Theta, m)} \tag{7}$$

The final volume cost is given by (8). Rather than use a per-image binary voting cost [9] we combine the probabilities from the individual images using an offset parameter $\phi \in [0,1]$. This parameter encodes the threshold level of the algorithm. If we consider binary silhouettes, this parameter would represent the number of images which are required to agree that a particular voxel is part of the visual hull. Therefore in the case of ideal silhouettes we have $\phi \rightarrow 1$ since $\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) \rightarrow 1$ in the event of perfect object classification. Thus ϕ controls the rate of convergence to the true solution against robustness to noise in the event of imperfect colour model classification in each image. Usually a conservative value of around 0.8 - 0.9 results in the best trade-off.

$$\mathrm{E}_{\mathrm{vol}}(\mathcal{O}, \mathcal{B}, \{I_m\}, \Theta) \;=\; \sum_{v_n \in \mathcal{V}} \begin{cases} \left(1 - \left[\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) - \phi\right]\right) & v_n \in \mathcal{O} \\ \left(1 + \left[\hat{\mathrm{L}}_{\mathcal{O}}(v_n, \Theta) - \phi\right]\right) & v_n \in \mathcal{B} \end{cases} \tag{8}$$

5.2 Boundary Term

The boundary term in (3) encodes the energy associated with the surface area of the object. In a similar manner to the 2D examples of [4,5], we use the colour discontinuities within the images to project cutting planes through the voxel array which should form the boundaries of the visual hull. We identify the colour difference as the vector norm of the projected pixels of neighbouring voxels in each image as (9).

$$Z_m(v_i, v_j) \;=\; ||\mathbf{u}_{m,i} - \mathbf{u}_{m,j}||^2 \tag{9}$$

Since we are estimating the boundary of the visual hull, the maximum colour difference across the image sequence is taken and an exponential cost function, with β estimated from the images [4,5], used to provide the standard Gibb's model of (11). Fig. 6 details the construction of the boundary term cost and an example of the final cost is given in Fig. 6(f).

$$\hat{Z}(v_i, v_j) \;=\; \max_m Z_m(v_i, v_j) \tag{10}$$

$$\mathrm{E}_{\mathrm{surf}}(\mathcal{O}, \mathcal{B}, \{I_m\}) \;=\; \sum_{(v_i, v_j) \in \mathcal{E},\; \substack{v_i \in \mathcal{O} \\ v_j \in \mathcal{B}}} e^{-\beta \hat{Z}(v_i, v_j)} \tag{11}$$

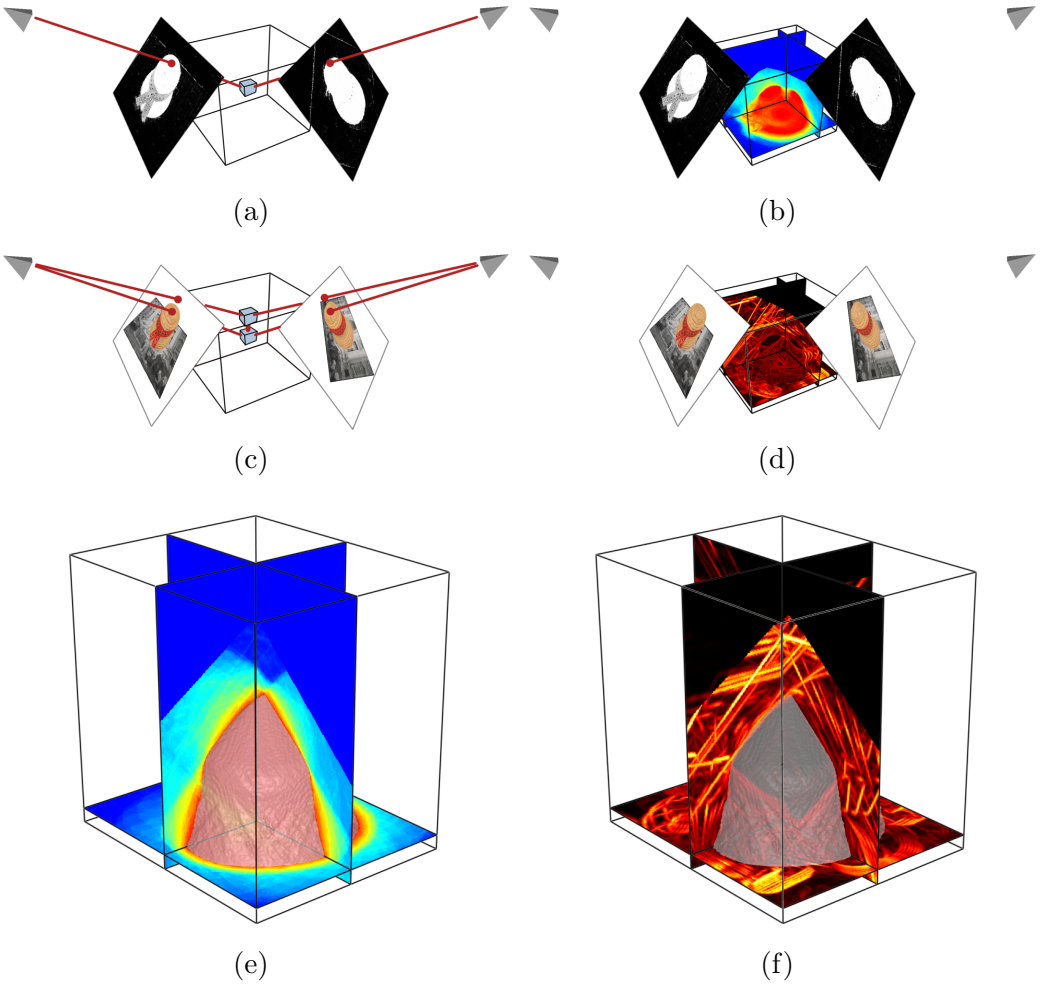


Fig. 6. **The volume and boundary term costs.** Volume Term: *Every voxel (node) in the array is projected into all of the images of which two are shown in (a). The images shown are the likihoods of being object given in (5). Cross-sections through the combined likelihood cost of (7) are shown in (b).* Boundary Term: *Every pair of neighbouring voxels (nodes) are projected into all of the images of which two are shown in (c). Cross-sections through the maximum colour difference volume of (10) are shown in (d).* Final Terms: *The final energy terms are shown with the converged visual hull. The volume term is shown in (e) with low to high* p(object)*. The boundary term is shown in (f) with low to high* p(edge)*.*

6 Experiments

In our experiments we calibrate a sequence of images. For the acquisition of test sequences we derived our own automatic calibration method. We assume the object lies on a textured plane and thus derive correspondences, using interest points and SIFT descriptors [14], and estimate plane-to-plane homographies. A modified version of the linear calibration from planar homographies in [15] then provides a initial input for non-linear optimisation,

via bundle adjustment [16], of the camera intrinsic parameters and individual poses.

For the statue sequence of Fig. 8, planar calibration was not possible and therefore the camera intrinsic parameters were calibrated separately followed by the application of structure from motion techniques [16] to track correspondences on the object itself and thus provide an initialisation for bundle adjustment.

The system used to provide the results was based on an Intel Xeon Processor with 4 GB of RAM running at 2.6 GHz. The unoptimised code required 20 minutes per iteration for the hat sequence with a voxel array size of 230^3. The complexity scales linearly with the number of pixels and the majority of the time is spent fitting and evaluating the colour mixture models. We now present the performance of the method on two 'real world' image sequences.

6.1 Hat Sequence

Fig. 7 shows the segmentation results for a sequence of 30, 4 megapixel images of a hat. The silhouettes and visual hulls shown were obtained after 3 iterations. In addition to the segmentation process being performed without user input, our planar calibration procedure is also fully automatic and thus no user interaction, other than capturing the images themselves, was required to complete this segmentation. Fig. 7(e) shows a 3D model reconstruction of the hat using an implementation of [17].

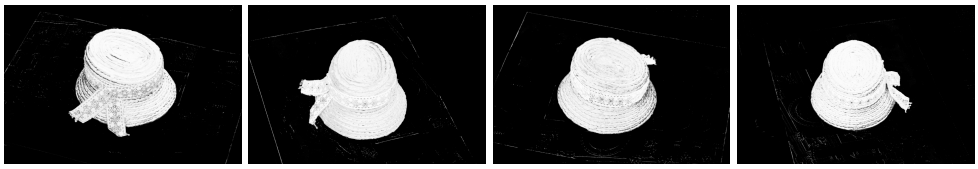
The object colour model probabilities, Fig. 7(b), are observed to have correctly captured all the object colours resulting in the accurate visual hull of Fig. 7(d) with corresponding silhouettes Fig. 7(c).

6.2 Statue Sequence

The method was also tested on a much more challenging image sequence. The statue sequence consists of 69 images of a statue in the British Museum. The images were captured by hand at a resolution of 5 megapixels. As can be seen from the example images of Fig. 8(a), the background continually changes including both the surrounding building and numerous different people and hence a very wide range of colours. The lighting conditions also change dramatically as can be seen by comparing the 3^{rd} and 4^{th} images of Fig. 8(a).

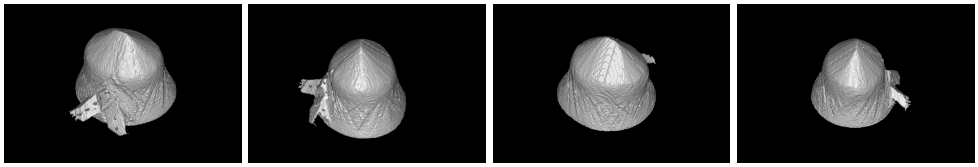
We have compared our results with an implementation of [5] applied independently to each image. Instead of user interaction, we generously supply the

(a) Input images

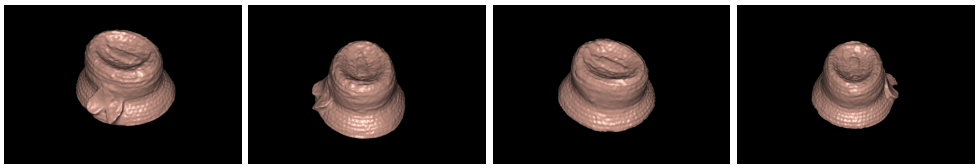


(b) Converged object likelihoods

(c) Converged silhouettes



(d) Converged visual hull



(e) Automatic reconstruction

(f) Reconstruction with texture mapping

Fig. 7. **Fully automatic calibration, segmentation and reconstruction of the hat sequence (4 out of 30 images).** *The fixation condition is applied to the image sequence (a) and used to initialise the colour models. The colour models converge to a capture all the object colours (b) and the graph-cut at convergence produces the correct silhouettes (c) and visual hull (d) which may be used to produce a reconstruction of the object (e) and also perform texture mapping (f).*

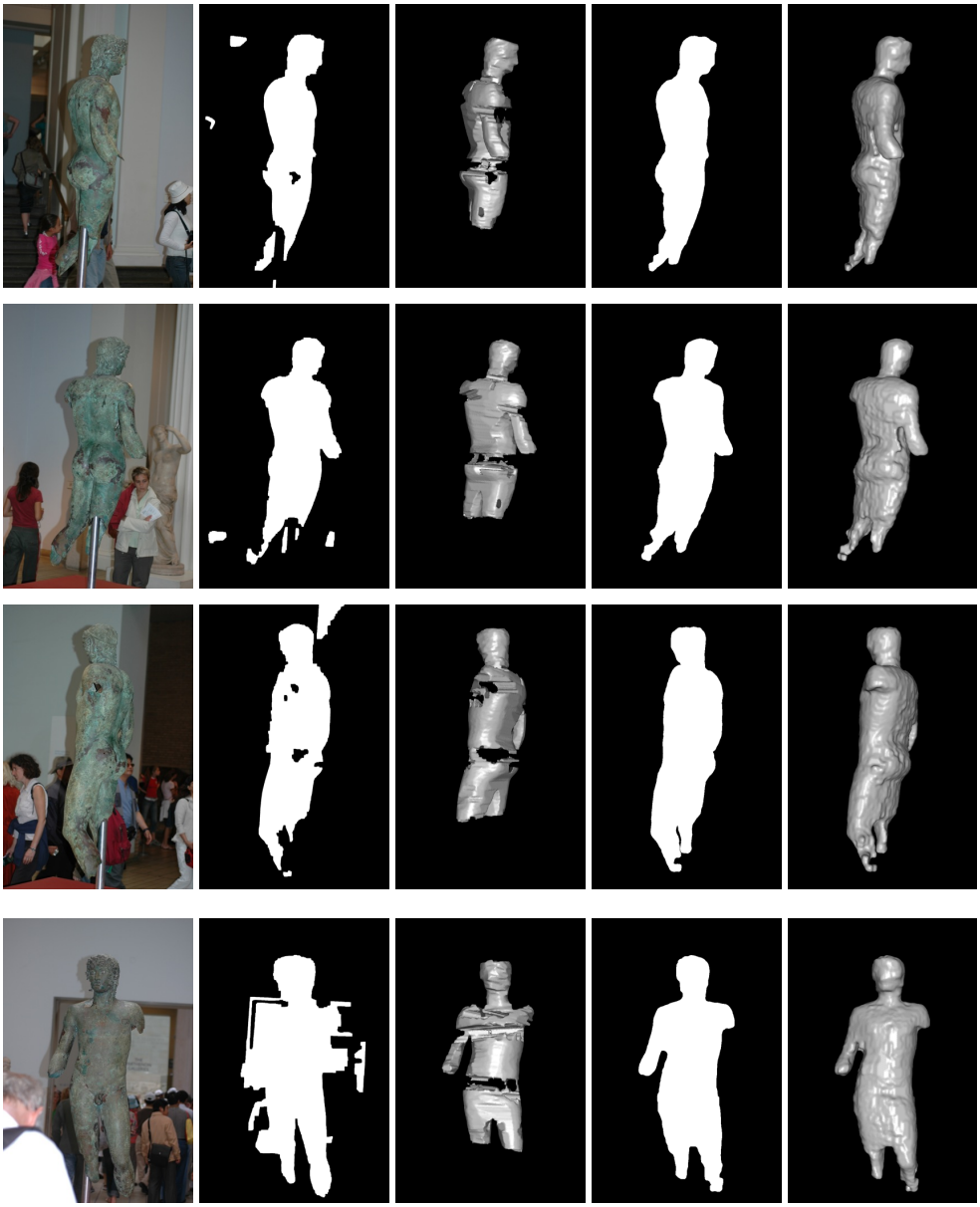


Fig. 8. **A single 3D segmentation improves multiple independent 2D segmentations.** *The statue sequence contains 69 images of a statue observed in a continually changing background environment (a). Independent 2D segmentation results in incoherent silhouettes (b) and a poor visual hull (c). After convergence, our method produces silhouettes (d) and a visual hull (e) with the correct topology.*

2D method with the converged colour models of the 3D segmentation algorithm rather than just the fixation points. Fig. 8(b) provides the silhouettes and Fig. 8(c) shows the visual hull resulting from the combination of the 2D segmentation results.

The segmentation results after 9 iterations of our method are given in Fig. 8(d).

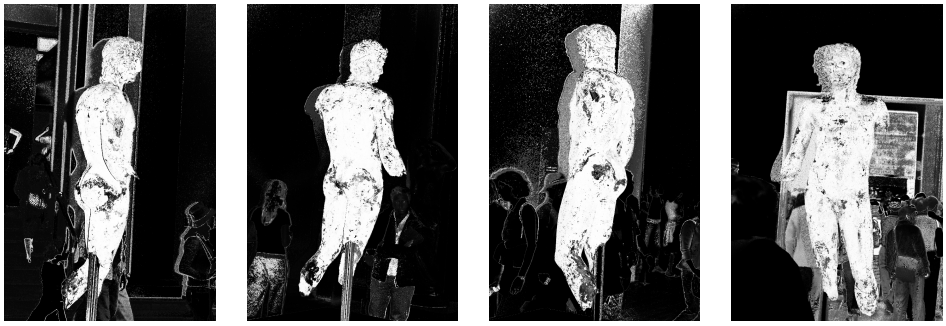


Fig. 9. **Converged object likelihoods from the statue sequence.** *The images show the final likelihood of object of (7) for the statue sequence of Fig. 8.*

The majority of the silhouettes are registered accurately offering significant improvements over the independent 2D segmentations. We observe that the left leg of the statue is not completely recovered. Since the leg structure is hollow, roughly half of the views see the inside of the sculpture which is different in colour and has failed to be modelled by the priors. Whilst the intersection of 2D binary silhouettes results in removing the whole leg, Fig. 8(c), our 3D segmentation algorithm tries to resolve the discrepancy between the silhouette coherency which results in capturing half the leg as object.

It may also be observed that the regularisation of the reconstruction has underestimated the concavity between the legs. Again this is mostly due to the colour model failing to provide a sharp prior classification in the presence of a cluttered background. Since the gap is only seen in a couple of images the resulting surface falls back on the coherency constraints of the remaining views. Thus the failures of the colour models (for example the same colour in object and background in the statue sequence) are compensated for, to an extent, by the enforcement of spatial coherency. Further improvements could be made by improving the object and background models, for example by including texture information.

Fig. 9 shows the converged object likelihoods for the statue sequence. We observe that there are many views where the object is not completely separable from the background in colour space. This is a limitation of the colour models used by the 2D and 3D algorithms and explains the poor 2D result when no further information, other than the object priors, can be used to perform the segmentation which results in the incoherent silhouettes.

Fig. 10 studies the limitations of the 3D algorithm, more specifically the effect of regularisation, in more detail. The over-estimation of the silhouette is due to the dominance of the spatial regularisation in regions where the colour model provides conflicting information (i.e. the object likelihoods are inconsistent across different images). The under-estimation at the extremities is due to the fact that the statue is hollow with a different inside which adds a confusion to the graph-cut which is trying to estimate a solid body. In fact, if we enlarge

the boundary size used for the final 2D graph-cut we can recover the correct segmentation in this case since we have strong edges in the original image.

7 Hand and House Sequences

Fig. 11 shows the segmentation and reconstruction process for a sequence of images of a hand. Again, automatic calibration was used, this time a newspaper provided a suitable textured plane. Slight modifications were made to the algorithm to cut the voxel array since the arm extends out of the image vol-

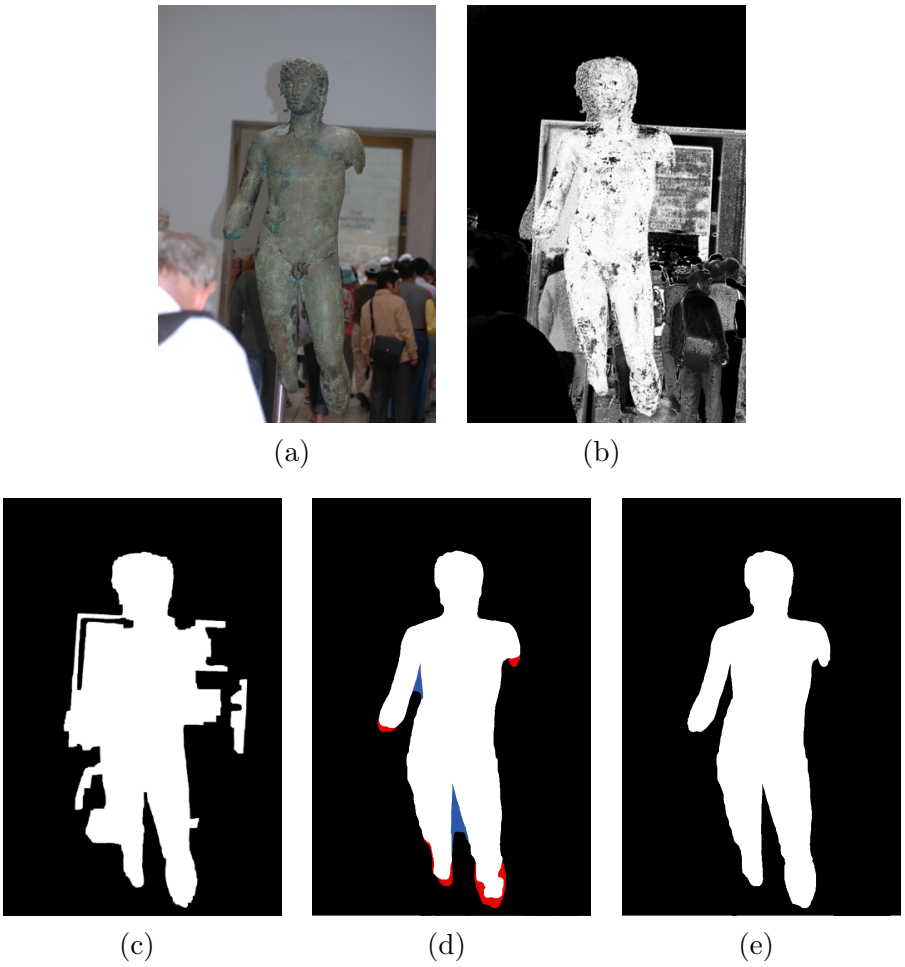
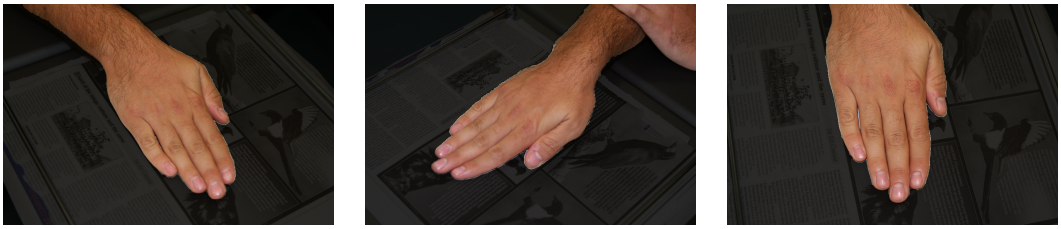
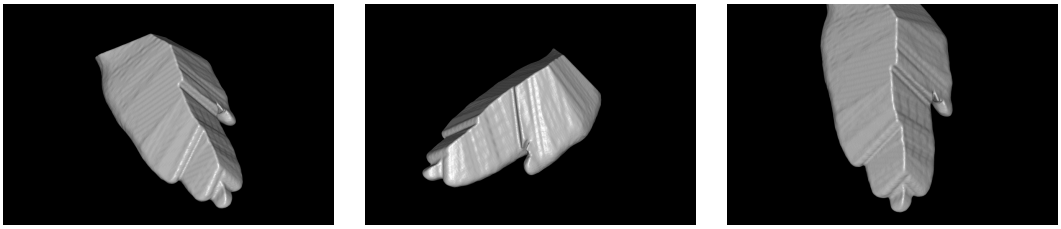


Fig. 10. **Limitations of the segmentation algorithm.** *An example of one of the statue sequence images (a) where the statue is not separable in colour space (b). The regularisation of the 3D algorithm improves upon the results of the 2D algorithm (c) but also results in the over estimation of the silhouette shown in blue in (d). The segmentation result also shows regions of under-estimation, shown in red in (d), but the full silhouette of (e) may be recovered using a final boundary 2D graph-cut around the silhouette from the converged visual hull.*

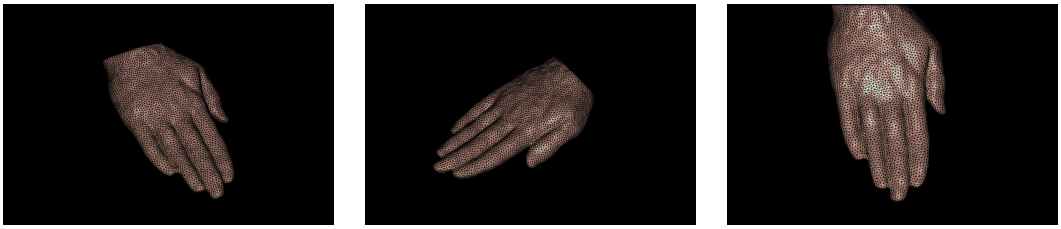
(a) Input images



(b) Silhouettes



(c) Visual hull



(d) 3D reconstruction

Fig. 11. **Fully automatic calibration, segmentation and reconstruction of the hand sequence.**

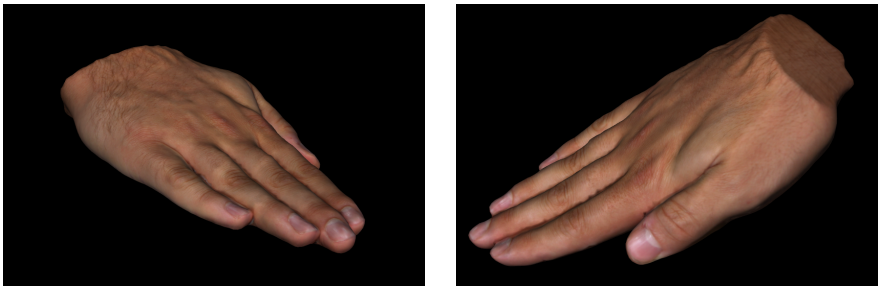
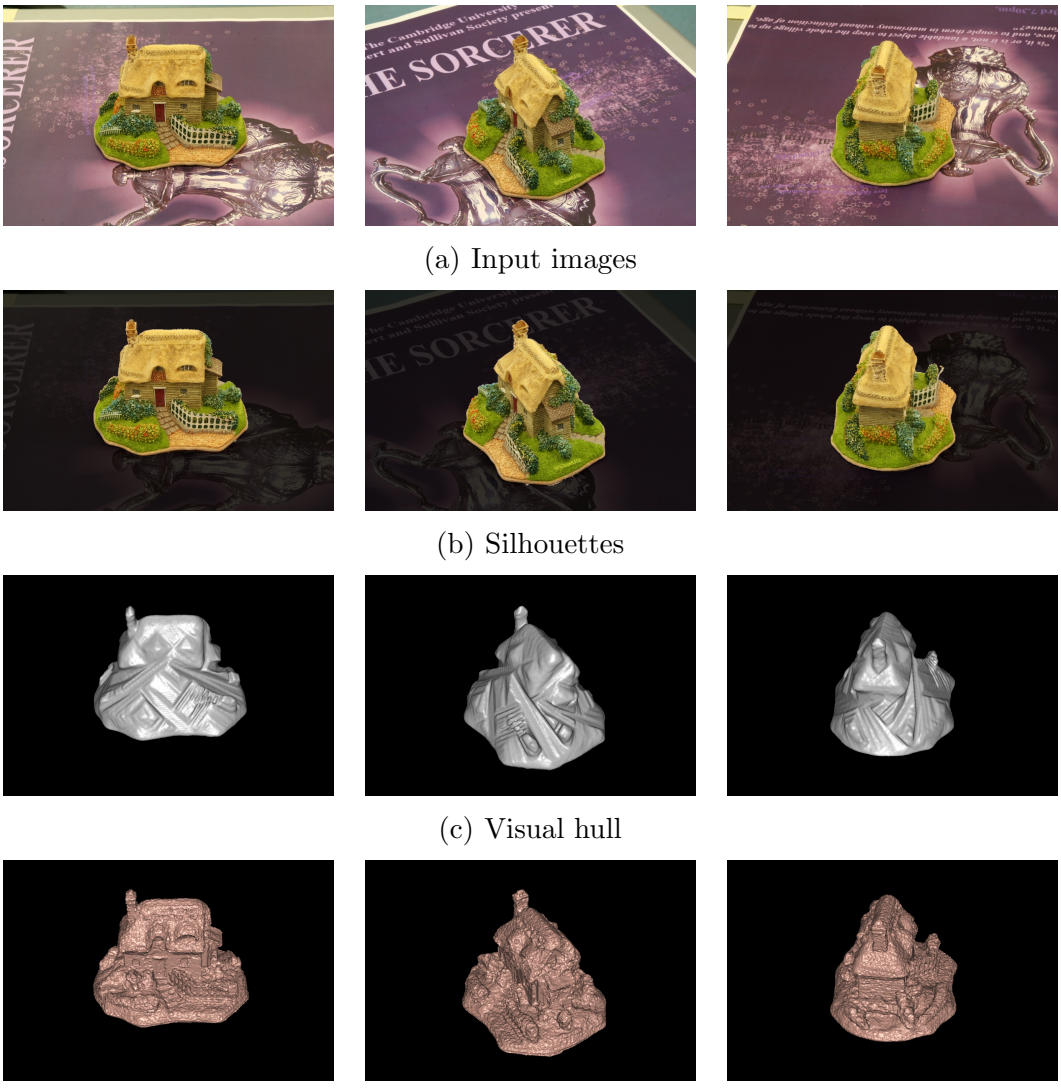


Fig. 12. **Texture mapped reconstruction of the hand.** *Recovery of the texture map is also an automatic process and alllows the hand to be rendered from novel viewpoints not present in the input image sequence.*

ume and the segmentation algorithm is designed for objects to be completely contained by the visible volume of the cameras. Automatic reconstruction was performed using the algorithm of [2] which also performs a volumetric graph-cut to produce the final surface. Fig. 12 shows some novel views of the texture mapped hand reconstruction.



(a) Input images

(b) Silhouettes

(c) Visual hull

(d) 3D reconstruction

Fig. 13. **Fully automatic calibration, segmentation and reconstruction of the house sequence.**

Fig. 13 shows the results of segmentation and reconstruction for a sequence of images of a toy house with a theatre poster providing the textured plane for calibration.

8 Conclusions

The results of our experiments indicate that our method confers considerable advantages for automatic object segmentation over the best 2D algorithms. The volumetric approach makes use of the silhouette coherency constraint to perform the segmentation in 3D, allowing the object to be segmented in all images simultaneously. This allows us to combine the learnt colour model, containing information from all views of the object, with a 3D shape prior to produce a more accurate result. We have also shown that it is possible to exploit a fixation constraint in order to initialise an iterative estimation algorithm to converge to the visual hull of an object observed in multiple views and thus avoid the need for any interaction from the user, thus making the whole process automatic.

The colour models used to segment the object are the main limitation of the algorithm, as may be observed by the statue sequence. In future work we intend to make use of more advanced image models to improve the object and background likelihood terms which should improve the performance of the algorithm.

9 Acknowledgements

This work is supported by the Schiff Foundation and Toshiba Research Europe.

References

[1] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 1, 2006, pp. 519–528.

[2] G. Vogiatzis, P. Torr, R. Cipolla, Multi-view stereo via volumetric graph-cuts., in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 391–398.

[3] A. Blake, C. Rother, M. Brown, P. Perez, P. Torr, Interactive image segmentation using an adaptive GMMRF model., in: Proc. 8^{th} Europ. Conf. on Computer Vision, 2004, pp. 428–441.

[4] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images., in: Proc. 8^{th} Intl. Conf. on Computer Vision, 2001, pp. 105–112.

[5] C. Rother, V. Kolmogorov, A. Blake, "grabcut": interactive foreground extraction using iterated graph cuts, in: SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, 2004, pp. 309–314.

[6] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts., IEEE Trans. Pattern Anal. Mach. Intell. 26 (2) (2004) 147–159.

[7] C. Hernández, F. Schmitt, R. Cipolla, Silhouette coherence for camera calibration under circular motion., IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 343–349.

[8] Y. Boykov, G. Funka-Lea, Graph cuts and efficient n-d image segmentation, Intl. Journal of Computer Vision 70 (2) (2006) 109–131.

[9] D. Snow, P. Viola, R. Zabih, Exact voxel occupancy with graph cuts, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2000, pp. 345–353.

[10] A. Yezzi, S. Soatto, Stereoscopic segmentation, Intl. J. of Computer Vision 53(1) (2003) 31–43.

[11] W. W. W. Lee, E. Boyer, Identifying foreground from multiple images, in: 8th Asian Conference on Computer Vision, 2007, pp. 580–589.

[12] M. Taalebinezhaad, Direct recovery of motion and shape in the general case by fixation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 847–853.

[13] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[14] D. Lowe, Distinctive image features from scale-invariant keypoints, Intl. J. of Computer Vision 60 (2) (2004) 91–110.

[15] Z. Zhang, A flexible new technique for camera calibration., IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.

[16] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, 2004.

[17] C. Hernández, F. Schmitt, Silhouette and stereo fusion for 3d object modeling, Computer Vision and Image Understanding 96 (3) (2004) 367–392.