# Automatic Object Segmentation from Calibrated Images

**Neill D.F. Campbell [1], George Vogiatzis [2], Carlos Hernández [3], Roberto Cipolla [1]**

[1] Department of Engineering, University of Cambridge, Cambridge, UK.
[2] Department of Computer Science, Aston University, Birmingham, UK.
[3] Google Research, Seattle, USA.

**Abstract**

*This paper addresses the problem of automatically obtaining the object/background segmentation of a rigid 3D object observed in a set of images that have been calibrated for camera pose and intrinsics. Such segmentations can be used to obtain a shape representation of a potentially texture-less object by computing a visual hull. We propose an automatic approach where the object to be segmented is identified by the pose of the cameras instead of user input such as 2D bounding rectangles or brush-strokes.*

*The key behind our method is a pairwise MRF framework that combines (a) foreground/background appearance models, (b) epipolar constraints and (c) weak stereo correspondence into a single segmentation cost function that can be efficiently solved by Graph-cuts. The segmentation thus obtained is further improved using silhouette coherency and then used to update the foreground/background appearance models which are fed into the next Graph-cut computation. These two steps are iterated until segmentation convergences.*

*Our method can automatically provide a 3D surface representation even in texture-less scenes where MVS methods might fail. Furthermore, it confers improved performance in images where the object is not readily separable from the background in colour space, an area that previous segmentation approaches have found challenging.*

**Keywords:** Multi View, Foreground Segmentation

## 1 Introduction

Shape-from-Silhouettes is a well established problem in Computer Vision that has seen a lot of interest mainly because in many cases (e.g. textureless objects) it is the only viable option for estimating 3D shape with visual sensors. Even in textured scenes, silhouettes are known to improve reconstruction accuracy in the cases of thin or awkward structures [12, 9]. However the problem of extracting silhouettes of an object from a set of photographic images, that can then be used to infer shape, has received much less attention. Most existing SfS approaches consider simple solutions such as background subtraction or manual segmentation, neither of which is always feasible.

This paper describes a practical method for automatically segmenting a set of images that have been calibrated for camera pose and intrinsic parameters. To remove the need for background subtraction and user input we propose the use of inter-frame epipolar constraints arising from the rigidity of the scene and the known camera motion. This is combined with the established constraints on the object's silhouettes and a *fixation* condition [6, 16] that is provided by the camera pose. To initialise the algorithm we can either start from an automatically computed bounding box visible in all images, or ask the user to provide their own 3D bounding volume.

The main advantage of our approach compared to previous work is that our method still performs well even when foreground and background are not separable in colour-space. Methods that rely solely on generative colour models, for example modelling object and background colour distributions as Gaussian mixture models [3, 6, 16] are not robust when the distributions overlap. In order to separate object and background they must also exploit spatial constraints, either within an image [3] or between multiple viewpoints [6, 16], sometimes termed *silhouette consistency* or *coherency*.

The key to our improvement derives from elegantly combining, in a single framework, geometric constraints, depth information, appearance constraints and spatial consistency both within a single image and over multiple views; we combine this with silhouette coherency to produce a consistent estimate of the object/background segmentation of a rigid 3D object. In this framework, image regions corresponding to spatially consistent scene objects, rather than only similar appearance, are connected across the entire dataset allowing neighbouring images to resolve ambiguity when a particular viewpoint observes the object against a camouflaged backdrop. In order to enforce these constraints in a tractable manner we first simplify the segmentation problem by pre-clustering the scene pixels into superpixels. This allows us to form a single interconnected graph, across all the superpixels in the set of images, and apply the graph-cut algorithm to label each superpixel as object or background.

We should note that, in general, the automatic segmentation problem is ill-posed, even under geometric constraints, since there are often multiple objects or parts of objects that are consistent both in appearance and with respect the multiple view geometry. In this instance the fixation condition and the bounding box (derived from the volume visible by all cameras) will determine the local minimum achieved. It is possible to alter this outcome by editing the initial conditions, for example moving the initial bounding box. Although this does introduce a demand on the user we note that the remaining algorithm is automatic and we have found that the fixation condition is sufficient for a variety of different scenes.

The rest of the paper is laid out as follows. We begin by discussing related prior work in § 2. In § 3 we provide a definition and subsequent analysis of the problem that leads to our proposed solution, presented in § 4. We demonstrate the performance of our algorithm through experiments in § 5 and the paper concludes in § 6.

## 2 Previous Work

This paper is about foreground/background image segmentation and is therefore related to a vast body of previous work (see [3] and references contained therein). However most of this work is concerned with segmenting a single image. Performing the interactive segmentation task for each image in a dataset individually quickly becomes prohibitive as the size of the dataset increases. A simple extension of interactive 2D segmentation on video appeared in [27, 18] where the user labels regions in a 3D space-time volume and the system segments a space-time region corresponding to a potentially deforming object. That method relies too heavily on the continuity of video, specifically optical flow, and hence cannot be applied to a typical wide-baseline MVS images. The specific task of addressing awkward thin-structures has been studied [26] with the use of connectivity priors that achieve good results but rely on specific labelling from the user.

The work most directly related to ours, addressing the problem of segmenting a calibrated set of images, can be found in [6] and [16]. The work of [6] makes use of a *fixation constraint* which assumes that the camera is always fixated on the object of interest. Using this hypothesis, a colour model of the foreground object is built, starting from the centres of the images. This colour model induces a volumetric cost functional which, when optimised, gives a new 3D surface and a corresponding 2D segmentation in all the images. Similarly in [16], viewing volumes of all the images are intersected and from this initial volume, a background colour model is learnt. Because they are highly related, these algorithms are compared to the present work in more detail in § 3.

Our work is related to [23], where sequential multi-view segmentation is achieved by pre-clustering each image using Mean-Shift and then interactively segmenting the clusters using a graph-cut optimisation. The optimisation exploits multi-view constraints by sequentially segmenting a set of images, and using the previous result as a shape prior on the new image segmentation. In comparison, our work proposes a formulation of the multi-view constraints that simultaneously segment all the images.

The idea of over-segmenting a set of images in the context of multi-view stereo has appeared in [13]. The key difference of that paper to the present work is that its aim is a Multi-view stereo algorithm using superpixel over-segmentation to reduce computational load. Here we focus entirely on the segmentation task, using multi-view stereo constraints to propagate pixel labelling.

In [22], 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 our method requires no prior knowledge of the object or environment.

The task of segmenting objects in multi-views has also been studied in [28]. 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 [28] relies on smooth albedo variation and a multi-resolution scheme to achieve convergence. In contrast, our method tolerates albedo discontinuities. This idea was continued in [15] where the authors adopt a probabilistic approach that is more robust to initialisation. This approach is along the same lines as both [6] and [16], without the iterative stage to estimate an appearance model for the object, with a continuous formulation rather than a discrete Graph-Cuts based approach.

The work of [1] uses unsupervised learning techniques to attempt to automatically segment different semantic classes from images. We address a different problem since we require pixel accurate segmentations of a specific object observed in multiple views in order to accurately determine shape. The class specific segmentations of [1] deal with different instances of objects of different types and the accuracy of the individual segmentations is too low to recover 3D shape.

Our work is also related to uncalibrated image co-segmentation [20] which aims at simultaneously segmenting an object out of a set of images, but without access to any geometric rigidity constraints. These methods focus on segmentation regions in different image with common appearance, specifically colour histograms, and will therefore encounter difficulties when faced with different viewpoints, and thus appearances, of an object and the fact that the object is observed in the same setting, making the distinction between object and background unclear; we are able to overcome this limitation at the expense of requiring calibrated views.

The work of [19] makes use of superpixels and epipolar geometry but with the intention of assigning a hard labelling on depth and normal direction for each superpixel whereas we take a different approach, using a soft depth labelling, which may well be multimodal, to influence the segmentation rather than trying to ascertain an accurate depth-map. The authors of [24] adopt a PDE-based approach to estimate depths across multiple images in a MVS dataset simultaneously. Again the focus of this work is a hard estimate of depth rather than our soft depth labelling. We are purposefully using a less precise and multi-modal estimate of depth since the images may not contain sufficient information for precise depth-estimation (e.g. textureless objects) and we will use other constraints to resolve any multi-modalities in the depth-estimates.

Finally, [5] is an example of using multi-view constraints (in the form of Structure-from-motion results) to aid the task of per-pixel scene labelling where the labels have been predefined. Our work also uses multi-view rigidity constraints in a completely unsupervised manner to infer a binary labelling.

## 3 Problem Analysis

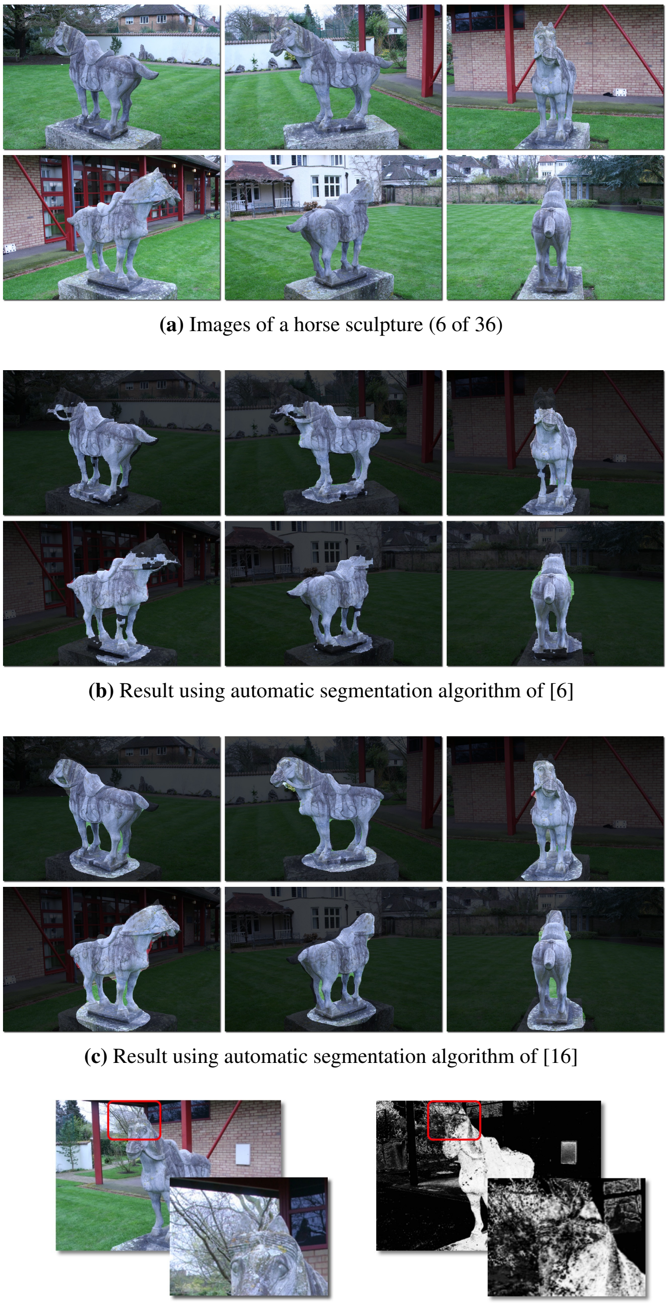
Our task is to obtain the silhouettes of a rigid 3D object observed against an unknown background in a set of $M$ images $I_1 \,..\, I_M$ of known camera calibration (pose and intrinsics). Each image, $I_m$ is composed of a set of pixels $\mathcal{P}_m$ and we wish to assign a label to each pixel $p \in \mathcal{P}$ as belonging to the object, $\mathcal{O}$ or background $\mathcal{B}$. This corresponds to a binary silhouette of the object in each image.

The task of segmentation is a challenging one, many of the latest algorithms adopt an interactive approach allowing feedback from the user to guide the segmentation process [3, 4]. In the case of segmentation in a single image two constraints are typically exploited. Firstly, we expect some form of colour or texture consistency within each segment and variation across segments. Secondly, we also make the assumption that segments are spatially continuous within the image.

If we now consider the case of a calibrated set of images, it has been shown [6, 16] that we may exploit scene rigidity to obtain a silhouette consistency constraint between the images. This arises due to the fact that the corresponding segmentations in each image are projections from the same rigid 3D object. This makes it possible to combine this constraint with the colour coherency and image spatial priors to perform automatic object/background segmentation of an object of interest across multiple images.

Whilst this approach works well for certain datasets, it faces a number of limitations when addressing those that are more challenging. Whilst the enforcement of silhouette consistency does compensate for poor prior information, for example when the object and background are not readily separable in colour space, the resulting segmentation will lose accuracy as seen in Figure 1. Here, both the methods of [6] and [16] fail to converge to the full object. Figure 1(b) shows the silhouettes to be under estimated; the head of the horse is missing. The difficulty faced by the generative model on this dataset is shown in Figures 1(d) and 1(e) where the object likelihood colour model fails to distinguish between the stone horse and background foliage.

Images where the object and background are not readily separable present a difficulty to the models used by the



**(a)** Images of a horse sculpture (6 of 36)

**(b)** Result using automatic segmentation algorithm of [6]

**(c)** Result using automatic segmentation algorithm of [16]

**(d)** Zoomed image

**(e)** Zoomed likelihood

**Figure 1: Limitations of a generative colour model.** (a) 6 of the 36 images calibrated images of a horse sculpture. (b) The automatic segmentation results obtained for the dataset using the method of [6]. We note that the body of the horse is slightly over-estimated whilst the head of the horse is not recovered. (c) The method of [16] captures more of the head of the horse but still fails to recover the extremities (such as the ears and tail) and does so at the expense of capturing portions of the background between the legs. (d) A zoomed region of one of the images that displays the difficulty in separating object and background based on colour or texture. (e) The converged colour model object likelihood of the same region reflects this difficulty by failing to distinguish clearly object and background.

algorithms of [6, 16] that adopt an iterative approach that alternates between updating generative colour models and enforcing spatial consistency across the views. The works of [6, 16] maintain Gaussian Mixture Models (GMMs) which encode probability distributions, one for the object and one for the background in [6] or just the background in [16], offering the advantage that we may take a single colour sample and determine a likelihood that it belongs to the object or background. However, when colours are shared by the object and the background, depending on how we estimate the GMM from the data, we may end up in a situation where the likelihoods are either uninformative (equally likely to be object or background) or incorrect (if the colour is predominantly found in the other category).

The combination of an appearance model with a volumetric approach is advantageous but fails to take into account the location of the surface of the object which is clearly important when similar colours are present in the foreground and background. In contrast to a purely generative colour model, we make use of a kernel based function to compare colours that has the advantage that we can use extra information, such as the epipolar distance or depth, to modulate the colour similarity. This allows two colour samples to have a different similarity score dependent on their spatial location and presents a solution to the problem of the overlapping object and background colour distributions.

This solution does come at the expense of having to set explicitly the kernel parameters for the comparison in colour space. Compared to the previous approaches, the clustering kernel makes use of the Euclidean distance in colour space whereas the GMMs of the generative model effectively translate to a comparison metric using a data-dependent Mahalanobis distance. This distance more accurately reflects colour difference since it is dependent on the colour distribution observed in the object and background. In an attempt to mitigate the impact of the Euclidean distance, we use linear PCA to map the colour samples into a space where the distance is more meaningful [2].

## 4 Algorithm

Our iterative segmentation method is presented in Algorithm 1. We begin with a set of $M$ images $I_1 \,..\, I_M$ with known camera calibration. For the sake of tractability, we begin by over-segmenting each image $I_m$ to obtain a set of superpixels $\{s_i\}$, $i = 1 \,..\, J_m$, with $J_m \sim 4000$, using the algorithm of [17]. Each superpixel represents a cluster of pixels from § 3; therefore, we now wish to label the pixels by assigning each superpixel a label of object or background: $s_i \in \{\mathcal{O}, \mathcal{B}\}$. Each superpixel has an associated average colour $\mathbf{u}_i$ as well as a position $\mathbf{x}_i$ given by the centre of the superpixel. These superpixels then form the vertices $\mathcal{S} = \{s_i\}$ in a graph $\mathcal{G} = (\mathcal{S}, \mathcal{W})$. The edges $\mathcal{W}$ of the graph are represented by an edge adjacency matrix $W$ where $W_{i,j}$ denotes the weight of the edge between the two superpixels $s_i$ and $s_j$, and a value of $W_{i,j} = 0$ indicates the absence of an edge. The construction of this $W$ matrix is described in § 4.2.

**Algorithm 1: The iterative segmentation algorithm.**

```
Input
- A calibrated set of $M$ images $I_1 \,..\, I_M$
Initialisation
- Obtain bounding volume from visibility
foreach image $I_m,\ m = 1 \,..\, M$ do
    - Group pixels into superpixels $\{s_i\}$
    - Extract fixation point
end
- Learn background colour model from outside bounding box
- Learn object colour model from fixation points
- Generate the edge matrix $W$
Main Loop
while visual hull not converged do
    foreach image $I_m,\ m = 1 \,..\, M$ do
        - Evaluate object likelihood
    end
    - Perform graph-cut to label superpixels
    - Enforce silhouette consistency
    - Update object colour model from new silhouettes
end
Output
- The converged object silhouettes and visual hull
```

We maintain two colour models, one for object and one for background, each as $K$ component Gaussian Mixture Models (GMMs) in colour-space. These models are initialised from the visible bounding volume [16]. We assume the object is fully contained in each view and, for the fixation condition [6], we assume the object is centred in each view.

The main loop of the algorithm evaluates the current object likelihood from the colour models and uses it as the unary data term, combined with the pairwise term of the edge matrix $W$, to perform a graph-cut that assigns a binary label to each superpixel as object or background. This labels all the superpixels in the set of images in a single step. The resulting silhouettes are then intersected to form a visual hull, ensuring that the silhouettes are consistent with one another. Finally the new silhouettes are used to update the object colour model and the loop is repeated until convergence.

The vertices of the graph encode the likelihood of object/background given the colour models, and the edges of the graph encode the spatial prior (both within a 2D image and using consistency in 3D space via epipolar transfer). We now discuss the individual stages in further detail.

### 4.1 Colour Models

Comparing the appearance of world surfaces between images is not ideal since we are forced to make quite strong assumptions about Lambertian reflectance, constant illumination of the scene and constant gain in the camera. We improve the situation by using any calibration information, for example a 3D structure point cloud [21], to obtain pixel correspondences between images and subsequently fit an affine colour model to perform colour correction between the images, this is a technique often used in the image mosaicing community, e.g. [8].

We build the colour models in the same manner as [3] and [6]; namely, we maintain two $K$ component GMMs, one for the object ($\mathcal{O}$) and one for the background ($\mathcal{B}$), with corresponding parameter vectors $\Theta^{\mathcal{O}}$ and $\Theta^{\mathcal{B}}$. Thus we have

$$\begin{aligned} \mathrm{L}_{\mathcal{O}}\left(s_i \mid \Theta^{\mathcal{O}}\right) &= p\left(s_i \in \mathcal{O} \mid \{\pi_k^{\mathcal{O}}, \mu_k^{\mathcal{O}}, \Sigma_k^{\mathcal{O}}\}\right) && (1) \\ &= \sum_{k=1}^{K} \pi_k^{\mathcal{O}} \mathcal{N}\left(\mathbf{u}_i \mid \mu_k^{\mathcal{O}}, \Sigma_k^{\mathcal{O}}\right) && (2) \end{aligned}$$

for the object likelihood and a similar form for the background $\mathrm{L}_{\mathcal{B}}\left(s_j \mid \Theta^{\mathcal{B}}\right)$. The probability distribution of the colour of the $k^{\text{th}}$ component of the mixture is given by a normal distribution with mean $\mu_k$ and covariance $\Sigma_k$. Each of the individual components is weighted by the marginal probability of the component $p(k)$, termed the mixing coefficient and denoted $\pi_k$. For all our experiments we used full covariance matrices and $K = 15$ (beyond which we found no further improvement). This parameter may be determined using model selection, if required, at the expense of computation time.

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 ($\Theta^{\mathcal{O}}$ and $\Theta^{\mathcal{B}}$) of the GMM to the sampled data [2]. We exploit the fact that the object is seen in multiple views, and therefore build a full colour model for the object by sampling superpixels from all the views using the current silhouettes as a mask, or the fixation condition [6] at the start. The background model is also built across all the views since the object and the scene are assumed to be rigid and thus the object will be observed in a consistent setting. Following the same logic as [16] we sample the superpixels outside the visibility volume (the region visible from all cameras) since we assume that the object is unoccluded (self-occlusions are not problematic) and visible in all images.

## 4.2 Generating the Edge Matrix $W$

The algorithm that generates the weight matrix $W$ is shown graphically in Figure 2. Taking a superpixel $s_i$ in image $I_m$ we use the camera calibration to determine the corresponding epipolar line $\mathbf{l}_{(i,I_\mu)}$ in a neighbouring image $I_\mu \in \mathrm{N}(I_m)$, where $\mathrm{N}(I_m)$ is the set of neighbouring images to $I_m$. The simplest algorithm would be to take each superpixel $s_i$ and connect it to all the superpixels $\{s_j\}$ in all the neighbouring images that satisfy the epipolar constraint (the perpendicular distance, $\mathrm{dist}(\cdot,\cdot)$, to the epipolar line is within a given threshold $\delta$)

$$\left\{s_j \mid \mathrm{dist}\left(\mathbf{l}_{(i,I_\mu)}, \mathbf{x}_j\right) < \delta\right\} \quad , \qquad (3)$$

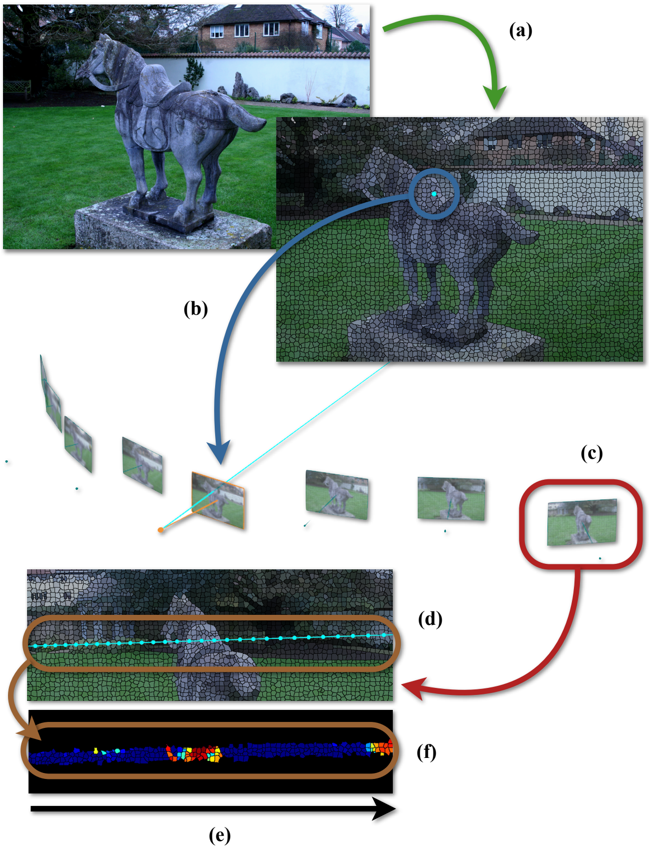


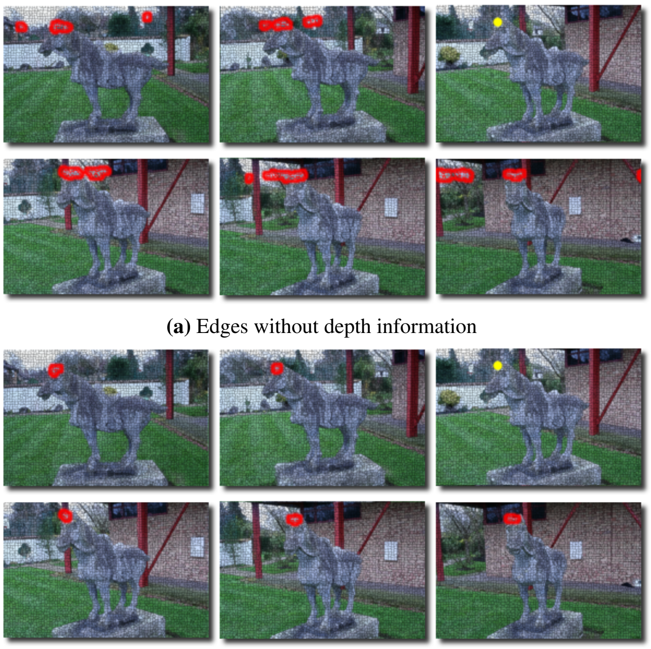
**Figure 2: Illustration of the construction of the edge matrix $W$.** (a) Each of the initial images $I_m$ is over-segmented to produce a superpixel representation $\{s_i\}$. (b) Every superpixel $s_i$ is projected into a set of neighbouring images using epipolar geometry. (c) Each of the neighbouring images $I_\mu \in \mathrm{N}(I_m)$ is selected in turn. (d) The set of superpixels $\{s_j\}$ that lie along the corresponding epipolar line is found. (e) The depth and (f) colour consistency are found for each $s_j$ and used to perform the soft stereo depth binning of Equation (8).

with a weight determined by the colour consistency, $\mathrm{c}(\cdot,\cdot)$, of the two superpixels

$$\mathrm{c}\left(s_i, s_j\right) = \exp\left(-\lambda \left\|\mathbf{u}_i - \mathbf{u}_j\right\|_2^2\right) \ . \qquad (4)$$

This approach suffers from two problems. The first is demonstrated by Figure 3. Constraining neighbouring superpixels to lie on epipolar lines is not sufficient to guarantee that superpixels are matched correctly since image regions of similar colour but belonging to different objects may also lie on the epipolar line, as shown in Figure 3(a).

We enforce a spatial continuity prior within the image by connecting neighbouring superpixels with an edge weighted by their colour consistency as in Equation (4). It makes sense to combine the information from all the different views, subject to the epipolar constraints, as well. We use a weak stereo algorithm to estimate the likely depth of the superpixel and thus identify matches which correspond to a physical object at this location in space. To perform this we quantise the possible



**(a)** Edges without depth information

**(b)** Edges with depth information

**Figure 3: The effect of the depth information.** The neighbouring image superpixels connected to the yellow superpixel (top right image) are shown outlined in red. (a) Shows the edges added without using the soft stereo stage. Whilst the epipolar constraint is satisfied, we observe that a large number of superpixels are incorrectly matched due to loss of depth information. (b) Shows the edges added when using the depth bins. The depth binning rejects almost all the incorrect matches by forming a consensus on the correct depth bin of the original superpixel. In both instances the matches have been thresholded at the same values to produce a sparse matrix.

depth range into a set of $N_B$ depth bins, $\tilde{\mathrm{d}}_n,\ n = [1\,..\,N_B]$, where each bin contains depths in the range $[\mathrm{d}_{n,\min}, \mathrm{d}_{n,\max}]$. This takes account of the ambiguity in depth that occurs due to the size of the superpixels. Noting that we have $s_i$ as a superpixel in a reference image and $s_j$ as a superpixel along the epipolar line in a neighbouring image; we compute the depth for each superpixel $\mathrm{d}\,(s_i, s_j)$ correspondence

$$\mathrm{d}\,(s_i, s_j) = \text{triangulate}\,(\,\mathbf{x}_i, \mathbf{x}_j) \tag{5}$$

and allow each superpixel to vote a particular depth-bin, weighted by its colour consistency. This vote encourages consensus between the neighbouring views whilst accounting for the ambiguity in depth and provides a degree of robustness against occlusion. It is denoted as $\mathrm{h}_i(\,\tilde{\mathrm{d}}_n\,)$ over the set of depth bins $\{\tilde{\mathrm{d}}_n\}$ as

$$\mathrm{h}_i\left(\tilde{\mathrm{d}}_n\right) = \max_{s_j}\left(\left\{\mathrm{c}\,(s_i, s_j)\,\middle|\,\mathrm{d}\,(s_i, s_j) \in [\tilde{\mathrm{d}}_n]\right\}\right) \tag{6}$$

using the (slightly abused) notation

$$\mathrm{d}\,(\cdot,\cdot) \in [\tilde{\mathrm{d}}_n] \iff \mathrm{d}\,(\cdot,\cdot) \in [\mathrm{d}_{n,\min}, \mathrm{d}_{n,\max}]\ . \tag{7}$$

Due to occlusion, the correct depth may be discarded due to an occluded view erroneously registering low colour consistency. To increase the robustness, we include a uniform outlier distribution over the $N_B$ depth bins. The mixing factors are denoted $\alpha$ and $\bar{\alpha} = (1-\alpha)$ for the uniform outlier distribution and normalised histogram distribution respectively. We may then estimate the probability of the true depth of $s_i$ falling within bin $\tilde{\mathrm{d}}_n$ as

$$\mathrm{p}\,(\,\mathrm{depth}(s_i) \in [\mathrm{d}_{n,\min}, \mathrm{d}_{n,\max}]) = \mathrm{p}\left(\tilde{\mathrm{d}}_n\right) = \prod_{I_\mu\,\in\,\mathrm{N}(I_m)}\left[\alpha\left(\frac{1}{N_B}\right) + \bar{\alpha}\left(\frac{\mathrm{h}_i\left(\tilde{\mathrm{d}}_n\right)}{\sum_q \mathrm{h}_i\left(\tilde{\mathrm{d}}_q\right)}\right)\right]\ . \tag{8}$$

This addition is not computationally intensive but results in a marked improvement in obtaining correct edge matches, as shown in Figure 3(b). The outlier mixing factor $\alpha$ is determined by $\varepsilon$, the expected number of neighbouring images which will occluded, and should be set to allow at least one of the neighbouring images to be inconsistent as

$$\alpha = \frac{\varepsilon}{|\mathrm{N}(I_m)|} > \frac{1}{|\mathrm{N}(I_m)|}\ . \tag{9}$$

Finally, we allocate edges from each superpixel $s_i$ to its neighbours within the image and the superpixels $\{s_j\}$ matched in neighbouring images, under the epipolar geometry. We set the edge weight as

$$W_{i,j} = \begin{cases} \mathrm{p}\left(\tilde{\mathrm{d}}_n\right)\mathrm{c}(s_i, s_j) & \begin{array}{c} s_i \in I_m, s_j \in \mathrm{N}(I_m) \\ \mathrm{d}(s_i, s_j) \in [\tilde{\mathrm{d}}_n] \end{array} \\ \\ \mathrm{c}(s_i, s_j) & s_i \in I_m, s_j \in I_m \end{cases} \tag{10}$$

which differs for neighbouring superpixels within and across images.

The second issue with connecting each superpixel to all its possible neighbours is one of tractability. Even using the superpixels from over-segmenting the image, we still have a large problem size. The horse dataset of Figure 1(a), for example, contains $J = \sum_m J_m \sim 160{,}000$ superpixels and, potentially, a large number of edges. In order to ensure that we may solve the graph labelling problem efficiently the $W$ matrix must be sparse. The epipolar constraint already promotes a degree of sparsity in the matrix; however, we can reduce the computational demand if we can increase sparsity without loss of useful information. The depth binning process encourages this since the incorrect matches will be given a very low weight and may thus be safely thresholded from $W$ without affecting the resulting clusters. This is indicated by the reduction in matches found in Figure 3(b) vs. Figure 3(a) that were both thresholded at the same level.

The number of neighbouring images to use, $|\mathrm{N}(I_m)|$, is dependent on the camera positions in the scene (as well as the availability of computational resources since increasing the number of neighbours reduces the sparsity). For our experiments we used a visibility angle of 45°, resulting in $|\mathrm{N}(I_m)| \approx 6$.

### 4.3 Graph-Cut

The most significant stage in the main loop of the algorithm is the segmentation task performed by the graph-cut process each iteration. Due to the initial over-segmentation into superpixels, we may use the graph-cut algorithm to perform a tractable *st-mincut* on a graph containing all the superpixels from all the images to obtain a global solution to the binary labelling problem. We formulate the task as an energy model

$$E\left(\{s_i\}\right) = E_{\mathrm{d}}\left(\{s_i\}\right) + \psi\, E_{\mathrm{s}}\left(\{s_i, s_j\}\right) \tag{11}$$

that assigns a binary label to each superpixel as object, $s_j \in \mathcal{O}$, or background, $s_j \in \mathcal{B}$. The energy comprises two terms, the data (or unary) term and the smoothness (or pairwise) term. The data term follows along the lines of the proposal in [6] and is simply the likelihood of being object under the current object and background models and thus changes at each iteration

$$E_{\mathrm{d}}\left(\{s_i\}\right) = \sum_m \sum_i d_p(s_i) \tag{12}$$

$$d_p(s_i) = \frac{\mathrm{L}_{\mathcal{O}}\left(s_i \,|\, \Theta^{\mathcal{O}}\right)}{\mathrm{L}_{\mathcal{O}}\left(s_j \,|\, \Theta^{\mathcal{O}}\right) + \mathrm{L}_{\mathcal{B}}\left(s_i \,|\, \Theta^{\mathcal{B}}\right)} \;. \tag{13}$$

The pairwise term is given by the appropriate element of the $W$ matrix, which favours grouping superpixels that are both similar in appearance and spatially consistent

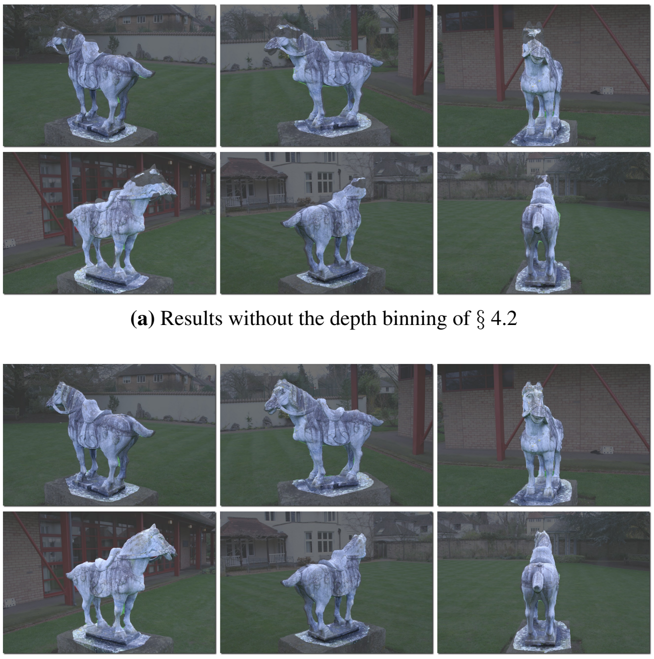
$$E_{\mathrm{s}}\left(\{s_i, s_j\}\right) = \sum_{\{i,j \;|\; W_{i,j} \neq 0\}} W_{i,j} \;. \tag{14}$$

This cost function may be solved exactly, i.e. the global minimum found, in polynomial time using the graph-cut algorithm [11].

### 4.4 Silhouette Consistency and Convergence

The final step of each iteration is to enforce silhouette coherency by projecting the 2D silhouettes into the visible volume and extracting the silhouettes of the intersected volume for each viewpoint to give the final silhouette. In practice this does little to alter the silhouettes but does improve accuracy and ensure complete silhouette coherency in the face of the inconsistent superpixel labellings — especially in the case where the original over-segmentation leads to superpixels that are not consistent between views. The criterion for convergence is that the superpixel labelling fails to change (or the number of changes are below a threshold) upon subsequent iterations.

We identify an area for future work as finding a method of producing a consistent superpixel segmentation which will allow for exact computation of the silhouette consistency from the superpixels themselves and therefore included directly in the optimisation at each iteration. This would also allow us to produce stronger and theoretically derived bounds on the convergence of the algorithm.



**(a)** Results without the depth binning of § 4.2

**(b)** Result using the full edge matrix $W$

**Figure 4: Results for the horse dataset.** (a) The results obtained without the depth binning process of § 4.2 that encodes weak stereo information. As we might expect from Figure 3(a), the edges in the $W$ cannot help separate the horse's head from the foliage so whilst the result is improved compared to the result of[6], in Figure 1(b), the head is still not recovered. (b) The automatic segmentation results obtained using the complete algorithm successfully recover the head of the horse from the background despite the difficulty in separating the two in colour space.

## 5 Experiments

Our experiments were performed on a 2.6 GHz Core 2 machine with 4 GB of RAM. The majority of the code runs under MATLAB, the over-segmentation taking 60 s per image for 4000 superpixels and a further 120 s to construct the $W$ matrix. For the horse dataset, containing $J = 160,000$ superpixels, the graph-cut usually ran in under 2 s with a further 5 s to complete the iteration. However, these are lower bounds since much of the method, with the exception of the graph-cut, can be run in parallel and may be computed on the GPU. The graph-cut can also be speed up by taking advantage of the fact that the $W$, and consequently the pairwise terms, don't change and thus we can save computation time by starting from a previous result as in [14]. We used the following parameter settings: $\psi = 1, \lambda = 1, \alpha = 0.1, |\mathrm{N}(I_m)| = 6, \delta =$ mean superpixel radius and $N_B$ as the mean number of superpixels along the epipolar line ($\approx 50$).

Figure 4 provides the results obtained for the horse dataset. Initially, the algorithm is run without the depth binning of § 4.2 and the segmentation obtained is shown in Figure 4(a). We

| **Algorithm** | Object Pixels Labelled |  | Background Pixels Labelled |  | $p$ (correct) |
|---|---|---|---|---|---|
|  | Correctly | Incorrectly | Correctly | Incorrectly |  |
| **Horse Dataset** |  |  |  |  |  |
| Result of [6] | 82.4% | 17.6% | 99.4% | 0.6% | 95.4% |
| Result of [16] | 93.4% | 6.6% | 98.5% | 1.5% | 97.3% |
| Our Result | 98.9% | 1.1% | 98.3% | 1.7% | 98.4% |
| **Plant Dataset** |  |  |  |  |  |
| Result of [6] | 65.5% | 34.5% | 96.2% | 3.8% | 87.9% |
| Result of [7] | 86.8% | 13.2% | 94.2% | 5.8% | 92.2% |
| Result of [10] | 81.9% | 18.1% | 93.6% | 6.4% | 90.4% |
| Result of [16] | 98.1% | 1.9% | 97.7% | 3.3% | 97.1% |
| Our Result | 94.4% | 5.6% | 98.4% | 1.6% | 97.4% |
| **Fountain Dataset** |  |  |  |  |  |
| Result of [6] | 98.2% | 1.8% | 91.3% | 8.7% | 94.7% |
| Our Result | 98.3% | 1.7% | 97.9% | 2.1% | 98.1% |
| **Vase Dataset** |  |  |  |  |  |
| Result of [6] | - | - | - | - | - |
| Our Result | 97.0% | 3.0% | 99.9% | 0.1% | 98.1% |
| **Table Top Dataset** |  |  |  |  |  |
| Result of [16] | 99.4% | 0.6% | 42.1% | 57.9% | 65.7% |
| Result of [7] | 65.2% | 34.8% | 93.5% | 6.5% | 81.9% |
| Our Result | 96.5% | 3.5% | 99.5% | 0.5% | 98.2% |

**Table 1: Comparison of quantitative segmentation errors.** Values given as percentages of pixels (relative to the bounding box area) that are correctly labelled and we also provide the naive probability that a pixel will be correctly labelled if picked (uniformly) at random from within the 2D projection of the 3D bounding box.

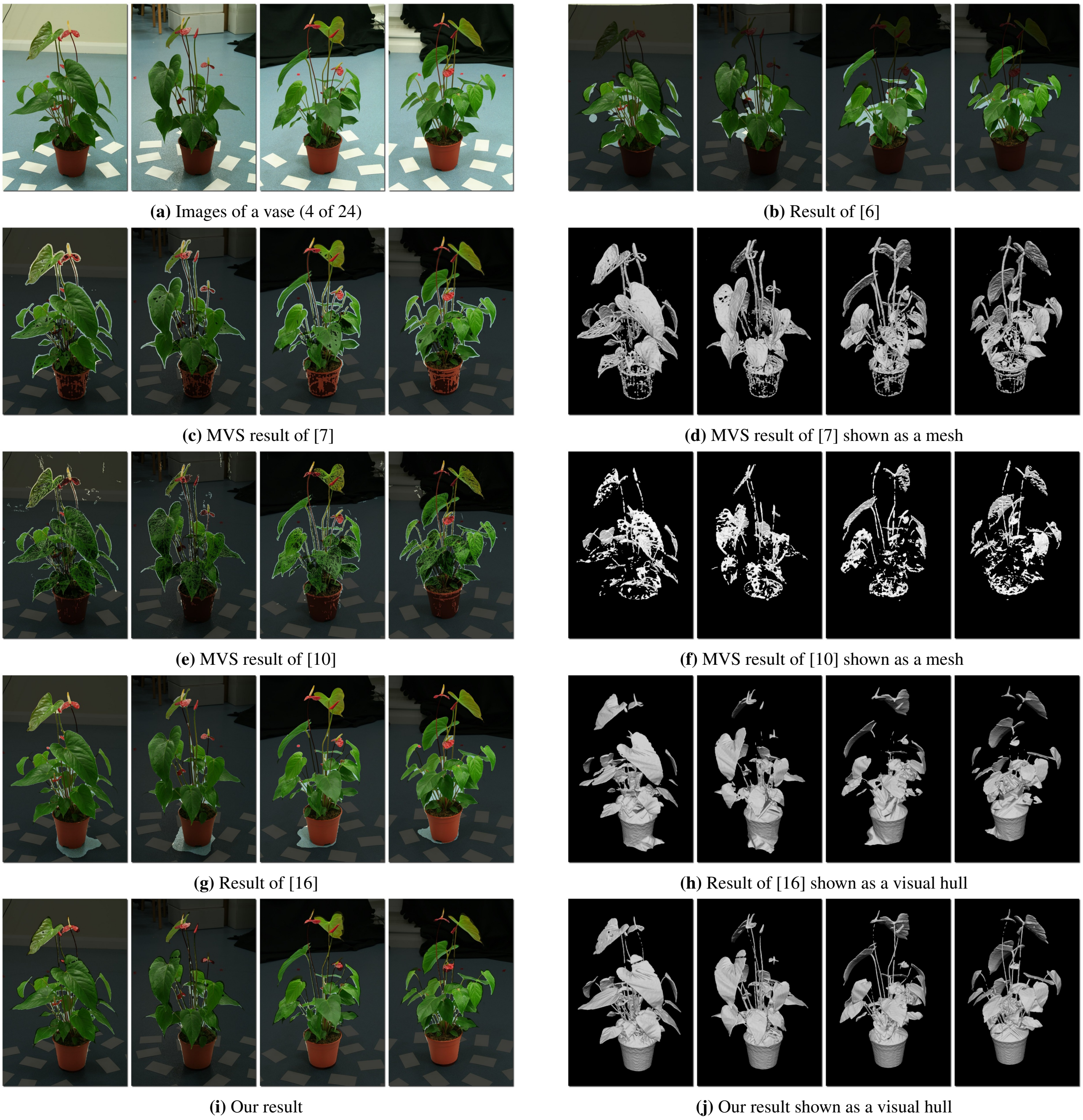
can see that, whilst this is an improvement on the result of [6], the algorithm has still failed to correctly recover the head. Considering Figure 3(a) we might not be particularly surprised since the colour information alone, in the pairwise term, is not really providing any further information until we fully exploit the scene geometry. Figure 4(b) shows the final result of the entire algorithm and the head has been successfully segmented.

Table 1 details the relative performance of our algorithm for the horse dataset.We can see that the algorithm confers an improvement in performance, both qualitatively (recovering the full head and tail) and quantitatively.

Figure 5 provides some results of the algorithm running on the plant dataset. This dataset is particularly challenging due to the very thin stems and leaves of the plant and the lack of texture. The result of [6], in Figure 5(b), is relatively poor due since the volumetric graph-cut is unable to handle the very fine structures. MVS systems will also struggle with the thin features and lack of texture as shown by the results from two top performing MVS algorithms [7, 10] in Table 1. Our approach yields better results, Figure 5(i), however we are reaching the limits of the superpixel approach since some of the thin structures are lost due to inaccuracies in the original superpixels. Figure 5(j) shows this result as a visual hull that demonstrates the extent to which the fine structures are segmented.

Figure 6 provides some results of the algorithm running on a standard MVS dataset from [25], thus representing a typical MVS dataset. A quantitative analysis is given in Table 1. The results are obtained automatically and again the dataset is challenging since the wall and the fountain colours display significant overlap. In Figure 6(b), we observe that the superpixel labelling has led to some errors at the segmentation borders although this has been reduced by the silhouette intersection stage. We also show an example where the user might wish to change the initial bounding box to alter the segmentation result, e.g. including the wall in Figure 6(c). We may further improve the situation, if required, by performing a boundary graph-cut around the borders of the silhouette on each image individually, similar to the approach taken in [23].

Figure 7 provides the results for a series of images of a vase observed in a room with walls coloured similarly to the vase, again a quantitative analysis is given in Table 1. For this dataset we were unable to find any parameter setting that allowed the method of [6] to converge. Changing the value of $\phi$ in [6] resulted in either the solution collapsing to nothing or exploding to fill the whole bounding box.



**(a)** Images of a vase (4 of 24)

**(b)** Result of [6]

**(c)** MVS result of [7]

**(d)** MVS result of [7] shown as a mesh

**(e)** MVS result of [10]

**(f)** MVS result of [10] shown as a mesh

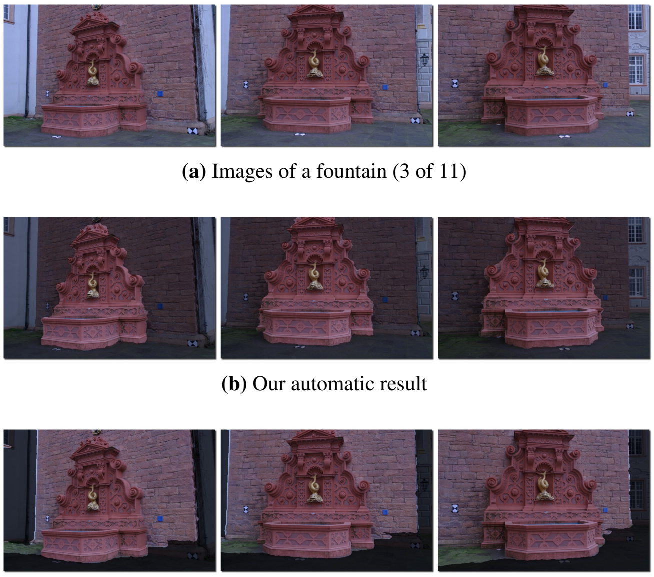
**(g)** Result of [16]

**(h)** Result of [16] shown as a visual hull

**(i)** Our result

**(j)** Our result shown as a visual hull

**Figure 5: Results for the plant dataset.** (a) 4 of the 24 images of the plant dataset. (b) The method of [6] performs poorly since the volumetric graph-cut cannot handle thin structures. (c)-(f) The MVS algorithms suffer due to the lack of texture and specularities in the scene, particularly in areas such as the flower pot. (g)-(h) Whilst the method of [16] achieves a good numerical result, qualitative inspection shows that thin structures have not been well recovered and the algorithm has over estimated the object's silhouettes in many areas. (i) The automatic segmentation results obtained using the complete algorithm improves performance but still fails to reconstruct the finest features due to superpixel boundary errors. (j) The result shown as a visual hull to emphasise the detail recovered for comparison with (h).

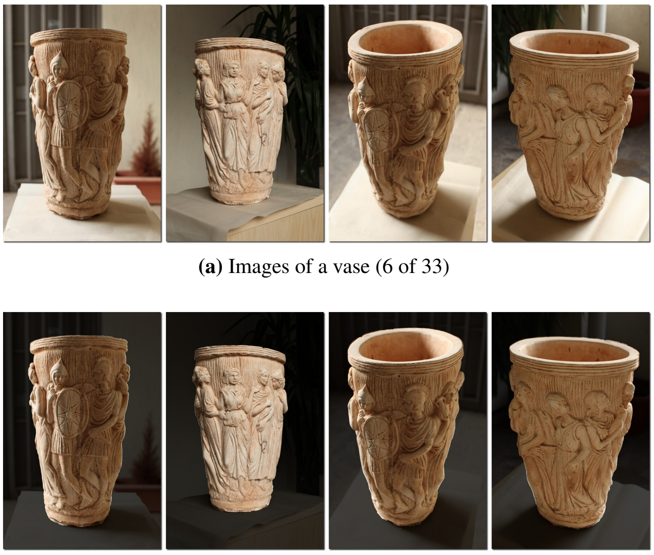


**(a)** Images of a fountain (3 of 11)

**(b)** Our automatic result

**(c)** Result with user edited bounding box

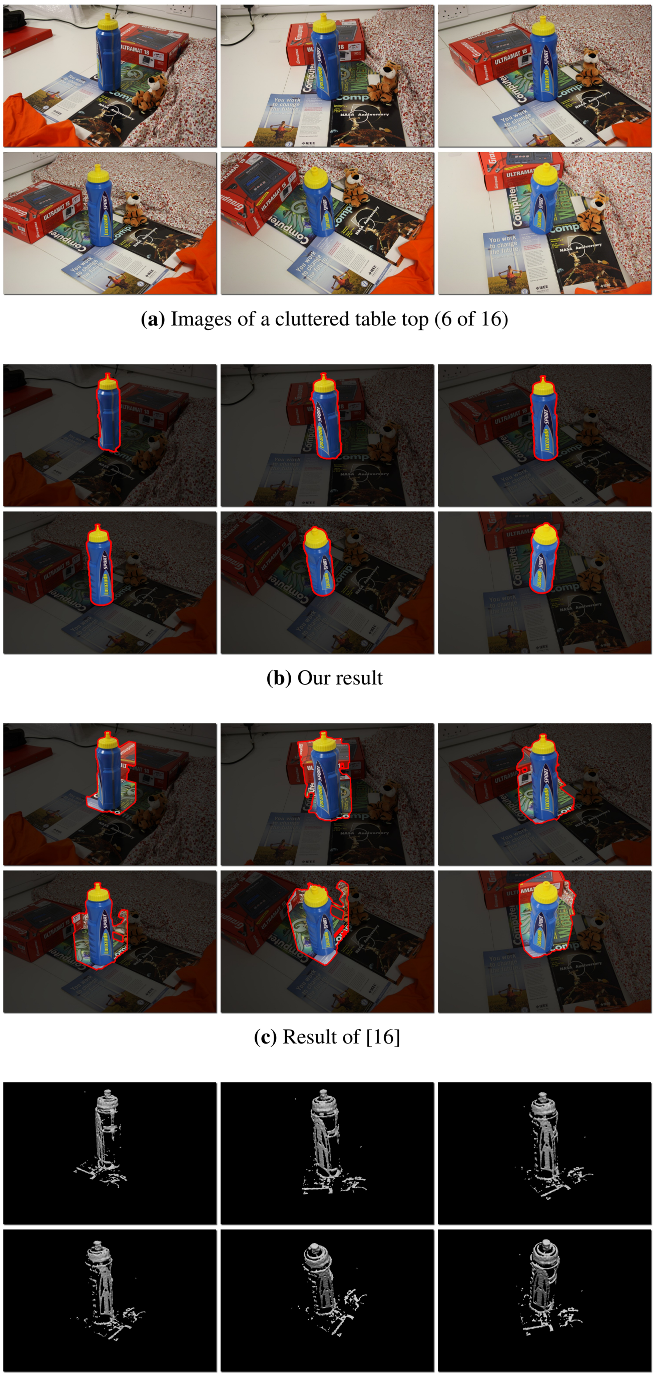
**Figure 6: Results for the fountain dataset.** (a) 4 of the 11 images of the fountain dataset from a standard MVS evaluation data-set [25]. (b) The automatic segmentation results obtained using the complete algorithm successfully recover the fountain from the background. The automatic bounding box does not fully contain the wall, hence it is not recovered. (c) The user is able to enlarge the initial bounding box, in 3D, resulting in silhouettes that containing the wall as well as the fountain.



**(a)** Images of a vase (6 of 33)

**(b)** Our result

**Figure 7: Results for the vase dataset.** (a) 6 of the 33 images of the vase dataset. (b) The automatic segmentation results obtained using the complete algorithm successfully recover the vase from the background. Again the background and object colours overlap and for this dataset we were unable to find parameter settings that allowed the algorithm of [6] to converge.



**(a)** Images of a cluttered table top (6 of 16)

**(b)** Our result

**(c)** Result of [16]

**(d)** MVS result of [7]

**Figure 8: Results for the table top dataset.** (a) 6 of the 16 images of the table top dataset. (b) The automatic segmentation results obtained using the complete algorithm successfully recover the bottle from the background. (c) The cluttered background is too complex (many edges) for the method of [16] to converge to a reasonable solution (algorithm remains in a loop of inconsistent solutions); making use of depth information is a distinct advantage for this data. This is a challenging scene for MVS since there is little texture and many specularities, resulting in the poor result of (d). Clearly in such a cluttered scene there will be local minima for spatially consistent objects, here the focus of the camera determines the 3D bounding box and hence the extraction of the bottle as opposed to other items.

Figure 8 shows the results obtained for a cluttered table top scene. The method of [16] is unable to converge to a reasonable solution (remaining in a loop of inconsistent solutions) due to the large quantity of background clutter; the addition of depth information helps our method resolve the ambiguities. Our segmentation result for the bottle is also much superior to the MVS surface obtained from the same bounding box due to the lack of texture on the bottle. The variety of objects in the scene show that there will be many local minima corresponding to spatially consistent objects (demonstrating the ill-posed nature of the problem). In this example, the fixation of the camera and the bounding box (determined as the volume visible from all camera positions) provides sufficient information to pick the bottle as opposed to any of the other objects.

## 6 Conclusion

We have shown that an existing approach to automatic object segmentation can be significantly improved by combining a generative appearance model and silhouette consistency with a more advanced smoothness term that takes into account viewpoint pose as well as appearance. This connects neighbouring images in the dataset and allows the segmentation process to resolve the ambiguities that exist when considering separability in colour space with appearance and silhouette constraints alone.

Our approach is not without its limitations and there are still avenues for further work. In a similar manner to approach such as [3], we cannot show proofs of convergence, plus the algorithm will be susceptible to local minima if there are strong changes in object appearance between images. Whilst the fixation condition has be shown to be very useful there will be situations where it will be insufficient to initialise colour models. We may be able to overcome these situations by allowing user interaction, either by specifying the initial 3D bounding box or indicating errors in the result and then updating the solution. Our formulation lends itself to this form of interaction since the segmentation is performed on the images, via the superpixels.

## Acknowledgements

This work was funded through the Schiff Foundation and Toshiba Research Europe.

## References

[1] B. Alexen, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *Proc. 11th Europ. Conf. on Computer Vision*, pages 380–393, 2010.

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

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P.H.S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. 8th Europ. Conf. on Computer Vision*, pages 428–441, 2004.

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. 8th Intl. Conf. on Computer Vision*, pages 105–112, 2001.

[5] G.J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc. 10th Europ. Conf. on Computer Vision*, pages 44–57, 2008.

[6] N.D.F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. In *Proc. 18th British Machine Vision Conference*, pages 530–539, 2007.

[7] N.D.F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proc. 10th Europ. Conf. on Computer Vision*, pages 766–779, 2008.

[8] D. Capel. *Image Mosaicing and Super-Resolution*. Springer-Verlag, 2004.

[9] D. Cremers and K. Kolev. Multiview stereo and silhouette consistency via convex functionals over convex domains. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6):1161–1174, 2011.

[10] Y. Furukawa and J.-P. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[11] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society*, 51(2):271–279, 1989.

[12] C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3D object modelling. *Computer Vision and Image Understanding*, 96(3):367–392, December 2004.

[13] M. Jancosek and T. Pajdla. Segmentation based multi-view stereo. In *Computer Vision Winter Workshop*, 2009. Paper 9.

[14] P. Kohli and P.H.S. Torr. Effciently solving dynamic markov random fields using graph-cuts. In *Proc. 10th Intl. Conf. on Computer Vision*, pages 922–929, 2005.

[15] K. Kolev, T. Brox, and D. Cremers. Robust variational segmentation of 3D objects from multiple views. In *Pattern Recognition (Proc. DAGM)*, volume 4174 of *LNCS*, pages 688–697, September 2006.

[16] W.W. Lee, W.T. Woo, and E. Boyer. Identifying foreground from multiple images. In *Proc. 8th Asian Conf. on Computer Vision*, pages 580–589, 2007.

[17] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2290–2297, 2009.

[18] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. In *Proc. of the ACM SIGGRAPH*, volume 24, pages 595–600, 2005.

[19] B. Micusik and J. Kosecka. Multi-view superpixel stereo in man-made environments. Technical report, George Mason University, 2008.

[20] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching — incorporating a global constraint into MRFs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 993–1000, 2006.

[21] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. In *Proc. of the ACM SIGGRAPH*, pages 835–846, 2006.

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

[23] M. Sormann, C. Zach, and K. Karner. Graph cut based multiple view segmentation for 3d reconstruction. In *Intl. Symp. on 3D Data Processing Visualization and Transmission*, pages 1085–1092, 2006.

[24] C. Strecha and L. Van Gool. Pde-based multi-view depth estimation. In *Intl. Symp. on 3D Data Processing Visualization and Transmission*, pages 416–425, 2002.

[25] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[26] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[27] J. Wang, P. Bhat, R.A. Colburn, M. Agrawala, and M.F. Cohen. Interactive video cutout. In *Proc. of the ACM SIGGRAPH*, pages 585–594, 2005.

[28] A. Yezzi and S. Soatto. Stereoscopic segmentation. *Intl. Journal of Computer Vision*, 53(1):31–43, January 2003.