# A Semi-supervised Approach to Space Carving

Surya Prakash[1] Antonio Robles-Kelly[1,2*]

[1]ANU, Bldg. 115, Australian National University, Canberra ACT 0200, Australia

[2]NICTA[†], Locked Bag 8001, Canberra ACT 2601 , Australia

**Abstract**

In this paper, we present a semi-supervised approach to space carving by casting the recovery of volumetric data from multiple views into an evidence combining setting. The method presented here is statistical in nature and employs, as a starting point, a manually obtained contour. By making use of this user-provided information, we obtain probabilistic silhouettes of all successive images. These silhouettes provide a prior distribution that is then used to compute the probability of a voxel being carved. This evidence combining setting allows us to make use of background pixel information. As a result, our method combines the advantages of shape-from-silhouette techniques and statistical space carving approaches. For the carving process, we propose a new voxelated space. The proposed space is a projective one that provides a color mapping for the object voxels which is consistent in terms of pixel coverage with their projection onto the image planes for the imagery under consideration. We provide quantitative results and illustrate the utility of the method on real-world imagery.

**Keywords:** space carving, volumetric reconstruction, 3D reconstruction, semi-supervised methods.

## 1 Introduction

One of the areas in computer vision that has attracted considerable interest is the recovery of three-dimensional information from multiple views. This usually involves using images, from a number

*Corresponding author. E-mail: antonio.robles-kelly@anu.edu.au; Tel: +61(2) 6267 6268; Fax: +61(2) 6267 6210

†NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

of cameras placed at different positions, to reconstruct the 3D shape of the object or scene of interest.

There has been various approaches to solving the 3D reconstruction problem. The literature along these lines is vast and spans from the use of stereo vision to volumetric approaches. In stereo vision [14], the aim is to recover the correspondences across frames based on pixel differences so as to use 3D view geometry to compute a point cloud that captures the structure of the scene. Solving the correspondence problem is a demanding task and has received much attention recently [18, 8, 19]. Some techniques also include the usage of bundle adjustment methods [42] or modulus constraints [29] to calibrate from a set of uncalibrated images. Once the calibration is at hand, a metric reconstruction can be effected from the imagery.

In contrast with bundle adjustment, where the rays between each camera centre and the set of 3D points on the object are used, other approaches elsewhere in the literature are based upon volumentric representations. These approaches based on volumetric reconstruction are dominated by shape-from-silhouette techniques [4, 25, 22] and voxel coloring methods [7, 20, 34]. An integral part of shape from silhouettes is the usage of object contours to construct a visual hull by finding the intersection of visual cones formed by the object occluding contours and the camera centers. Thus, a silhouette image is a binary image in which the object becomes an occluder of the background from the observer's view point. As a result, silhouettes determine whether each pixel, projected as a line-of-sight ray from the camera centre, intersects the surface of the object. This approach to object reconstruction using volume intersection was first exploited by Martin et al. [25], who recovered volumetric object representations making use of visual cone intersections. This intersection volume is known as the visual hull by Laurentini [22] and described as the maximal volume that yields the silhouette for the object from any possible viewpoint.

In [9], three binary images are abstracted to quad-tree representations and merged into an octree visual hull. An octree representation of a solid region is obtained by hierarchically decomposing a 3D cube into smaller ones (8 of them in this case). Hierarchical division and cube orientation usually follows the spatial coordinate system. Despite effective, the drawback of this approach is that the input of images is limited and there is a requirement on the orthogonality of the optical axes. Potmesil et al.[30] addresses this problem by reconstructing an octree representation from multiple images with arbitrary viewpoints. This approach first generates conic octree volumes from silhouettes of the object and then combines them so as to obtain a global model. The individual objects are then labelled using a 3D connected component algorithm. In [38], the silhouette is

approximated polygonally after segmentation via thresholding. These polygons are then decomposed into convex components and then efficient octree intersection tests with back-projections are employed. Szeliski et al. [40] build the volumetric models directly from the actual photographs. Laze[23] *et al.* have used polyhedrons to characterise the surface of the object on a set of visual cones. Ilic *et al.* [17] have used implicit surfaces to model 3D shapes using silhouettes in uncontrolled environments. In a related development, Liang and Wong [24] have addressed the identification of the tangent space of an object surface recovered from silhouettes by introducing an epipolar parameterisation of the problem.

Nonetheless the accuracy of the visual hull construction increases with the number of images, the result is dependent on the geometry of the object of interest, being highly sensitive to the genus of the surface object. Thus, one of the drawbacks of shape-from-silhouette algorithms is that not all concavities of the object can be modeled using the visual hull approximation. The shape information from multiple silhouettes only guarantees to provide an enclosing space for the object corresponding to its volumetric upper bound. The other disadvantage is that the silhouettes needed are usually obtained manually [39] or via image differences between consecutive frames [37]. This requires controlled background conditions and illumination so as to enable proper segmentation.

Voxel coloring techniques, on the other hand, start with an array of voxels that must enclose the entire scene. These arrays of voxels are then kept or removed based upon image color values subject to constraints based upon scene reconstruction. A common assumption in voxel colorng methods pertains the Lambertian reflectance of the object under study. If a surface exhibits Lambertian reflectance, light is then scattered such that the apparent brightness and colour of the surface on the image is the same regardless of the camera's angle of view. Voxel coloring approaches were first introduced by Seitz et al. [34]. Their algorithm begins with an array of opaque voxels encompassing the scene. As the algorithm progresses, opaque voxels are tested for color consistency and classified. A point is photo-consistent if it does not project to known background or, alternatively, the light exiting the point, i.e. its radiance, in the direction of the camera is equal to the observed color of the points projection in the image. Moreover, photo-consistency can be defined as the standard deviation of the pixel colors for the set of pixels that can "see" a voxel [36].

In practice, voxel coloring tests consistency of voxels making use of their visibility, i.e. whether or not a given camera can see a voxel. The visibility issue was addressed in [34] by introducing ordinality constraints on the camera locations so as to adapt single scan methods to voxel data. This requires that the camera locations are such that all the voxels can be visited in a single scan in

a near-to-far order relative to all the camera-centres. One of the ways in which this can be achieved is by placing all the cameras on one side of the scene. Then voxels can be scanned in planes whose distance to the camera centre increases monotonically. The drawback of this method hinges in the placement of cameras, which are required to be placed so that no scene points are contained within the convex hull of the camera centers so as to assure ordinal visibility constraints. This implies that the full 3D reconstruction of the scene is not possible because the cameras can not surround the scene.

Kutulakos et al. [20] rectified this limitation by introducing a multi-sweep approach. This algorithm evaluates voxels one plane at a time in a similar fashion to voxel coloring techniques, except that multi-scans are performed typically along the positive and negative directions of each of the three axes. Space Carving forces the scans to be near-to-far relative to cameras by using only the images whose cameras have already been passed by the moving plane. The main argument leveled against the method in [20] is that the method used to determine visibility does not include all the object images as some viewpoints on the moving plane may be visible from particular voxels. This was addressed by Culbertson et al. [10] in the algorithm called Generalized Voxel Coloring (GVC), where the visibility is computed accurately as compared to the approximate visibility utilized in Space Carving.

All the methods above have the common drawback of making hard and irreversible commitments on the removal of voxels. This can lead to large error generation and incorrect 3D reconstruction by creating a hole, even if only one voxel is removed incorrectly. This problem led to the probabilistic approaches to space carving [7, 1, 6] being desirable. The other advantage of probabilistic approaches is that they avoid the need of a global parameter (variance) for color consistency checks. In probabilistic space carving methods, each voxel is assigned a probability determined by computing the likelihoods for the voxel existing or not. Here the voxels are processed starting with the layers closest to the camera. The visibility of each voxel layer is determined by the probabilities of the previous layer. This single sweep algorithm is also dependent on the specific placement of cameras. That is, they have to satisfy the ordinal visibility constraint.

It is worth noting in passing that photo-consistency is a non-trivial task that has drawn research from stereo methods [44]. Esteban and Schmitt [11] have used silhouettes and stereo in an information fusion setting for 3D object modelling. This is somewhat related to the use of implicit surfaces for voxel colouring. Along these lines, Grum and Bors [13] have used implicit surfaces so as to model 3D scenes from multiple views using space carving.

On the voxelisation of the space, the approaches above use cubic voxels whose position is predetermined by the user. The drawback of this cubic voxelated space is that their projections onto the imagery changes in terms of pixel coverage, i.e. the number of pixels "covered" by each voxel, with respect to the distance of the voxels from the camera centre. This leads to inconsistent comparisons of colors or intensities for photo consistency check as the voxels space is carved. Alternatives to the cubic tessellation has been proposed by Saito et al. [31], where epipolar geometry relating two views is used to construct a projective grid space.

# 2 Contributions

In this paper, we present a probabilistic semi-supervised method for space carving. This method makes use of a user provided silhouette and an image sequence at input to deliver, at output the 3D reconstruction of the object under study. To do this, we cast the space carving problem into an evidence combining setting and remove voxels based on the posterior probability of a voxel existing given the silhouette and pixel color information. In our method, the probability of a pixel being foreground or background in a silhouette is recovered making use of a sequential scheme which, departing from the user-supplied object contour, computes the posterior probabilities for subsequent frames. We view the recovery of the probabilities of a voxel existing as a supervised classification setting dependent on the silhouette information. To this end, we make use of a discriminant function which is governed by the average variance of the pixel-values for the voxel across those views in which its visible. Thus, by combining user-provided silhouette information and image data, our method not only provides a means for semi-supervised space carving, but also a link between shape-from-silhouette and unsupervised space carving methods. It exhibits the strengths of voxel coloring methods while having the advantages of probabilistic space carving approaches. Unlike other probabilistic approaches [7, 1], our approach enables us to obtain full 3D reconstruction. This is due to the fact that our method is not restrictive on the camera position and does not require ordinal visibility constraints to be satisfied.

Nonetheless our method can be applied to cubic voxel settings, we also propose here a voxelisation of the carving space which is projective in nature. We formulate our projective space making use of the camera centres to tessellate the space. The proposed projective space offers benefits such as consistent back-projection from any voxel onto the imagery and a more accurate color mapping of surface voxels. Moreover, using this approach, the approximate number of pixels covered by

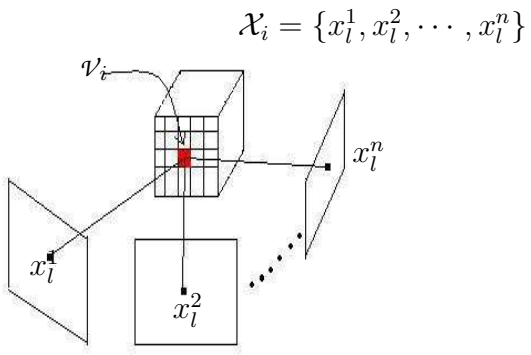



Figure 1. Space carving.

each voxel at back projection can be chosen while populating the space. This can be determined on the basis of the reconstruction setup. The nature of the proposed space ensures that this back projection remains consistent irrespective of the position of voxels with respect to the camera centres. This increases the accuracy of the color mapping on the recovered 3D representation of the object.

# 3 Space Carving

In space carving [34], a voxel array is carved so as to remove those voxels that should be discarded to reveal the volumetric shape of the object under study. This carving process is either a plane sweep one [7] or an operation on the surface voxels being considered [36]. This carving process can be single sweep or iterative in nature and is governed by the intersection of the rays from the surface voxels to the images in the sequence.

Let $x_l^k$ be the $l^{th}$ pixel in the image indexed $k$. Similarly, the set of pixels in the image sequence onto which the voxel $\mathcal{V}_i$ back-projects is denoted $\mathcal{X}_i$ as shown in Figure 1. As the surface voxels are carved, their neighbourhood voxels are revealed either on the same layer or the layer underneath. We, at each iteration, compute the probability of a voxel being discarded, i.e. $\exists_{\mathcal{V}_i} = 0$, based upon the available data corresponding to the pixels in the image sequence onto which it back-projects. With the probability at hand, we perform inference and remove voxels from further consideration. This step sequence is interleaved until no further voxel removal operations are effected.

As mentioned earlier, we aim at posing the space carving process in an evidence combining setting. Our approach is semisupervised in nature and employs a user-provided silhouette which corresponds to the object contour at the initial frame of the image sequence under study. Following this rationale, the decision on whether a voxel should be carved is based upon available pixel attributes, such as colour, brightness, etc. and the silhouette provided by the user at input.

We commence by defining the posterior probability of the $i^{th}$ voxel $\mathcal{V}_i$ being in the object given the pixel-set $\mathcal{X}_i$ and the user provided silhouette $\mathcal{S}$, which we denote by

$$P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S}) = P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i) P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S}) \tag{1}$$

where we have written $\exists_{\mathcal{V}_i} = 1$ to imply that the voxel $\mathcal{V}_i$ is present in the object, i.e. exists, and assumed independence between the pixel data and the user-supplied silhouette.

The expression above opens-up the possibility of employing generative models to recover the posterior probability $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$. These probabilistic models will be used throughout the paper for recovering the shape and pixel probabilities and the optimal cut-off values for the carving process.

## 3.1 Silhouette Information

Since the silhouette $\mathcal{S}$ determines a foreground-background separation in the scene, it can be used to define a shape prior over the image sequence. This observation is important since it allows us to express the probability $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$ in terms of a set of voxel projections onto the input images. Thus, to take our analysis further, we write

$$\begin{aligned} P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S}) &= \frac{P(\mathcal{S} \mid \exists_{\mathcal{V}_i} = 1) P(\exists_{\mathcal{V}_i})}{P(\mathcal{S})} \\ &= \frac{E[\mathbf{1}_{\mathcal{S}} \mid \exists_{\mathcal{V}_i} = 1] P(\exists_{\mathcal{V}_i} = 1)}{\sum_{j \in \{0,1\}} E[\mathbf{1}_{\mathcal{S}} \mid \exists_{\mathcal{V}_i} = j] P(\exists_{\mathcal{V}_i} = j)} \end{aligned} \tag{2}$$

where we have used the fact that $P(\mathcal{S} \mid \exists_{\mathcal{V}_i} = 1) = E(\mathbf{1}_{\mathcal{S}} \mid \exists_{\mathcal{V}_i} = 1)$, $E[\cdot]$ is the expectation operator and $\mathbf{1}_{\mathcal{S}}$ is an indicator variable whose value is unity if the silhouette $\mathcal{S}$ occurs and zero otherwise.

To render $\mathbf{1}_{\mathcal{S}}$ tractable, we consider the $i^{th}$ image $I_i$ in the sequence under study. For the sake of convenience, we assume the user-supplied silhouette corresponds to the image $I_0$. Let the separation given by the silhouette $\mathcal{S}$ between foreground and background be consistent with the probability of a voxel existing or being removed. This is, if $\exists_{\mathcal{V}_i} = 1$, the projection of the voxel $\mathcal{V}_i$ onto the

$l^{th}$ pixel $x_l^k$ in the image $I_k$ denotes a foreground pixel. Otherwise, the pixel $x_l^k$ is a background one.

As a result, and keeping in mind our Bayesian formulation of the problem, we can consider the indicator variable $\mathbf{1}_s$ as the hard limit of the probability $P(\mathcal{C}_F \mid x_l^k)$ of the foreground $C_F$ given a pixel $x_l^k$. Accordingly, the expectation $E(\mathbf{1}_s \mid \exists_{\nu_i} = 1)$ becomes the average over foreground posterior probabilities for the set of pixels $\mathcal{X}_i$, i.e.

$$E[\mathbf{1}_s \mid \exists_{\nu_i} = 1] = \frac{1}{\mid \mathcal{X}_i \mid} \sum_{x_l^k \in \mathcal{X}_i} P(\mathcal{C}_F \mid x_l^k) \tag{3}$$

Thus, the problem reduces itself to recovering the posterior probability $P(\mathcal{C}_F \mid x_l^k)$. Further, note that, since the user is required to provide a silhouette at frame $I_0$, we have, at our disposal, a background-foreground segmentation which we can use to perform inference on the image sequence. Thus, by using a Markovian formulation, we can consider the probability $P(\mathcal{C}_f \mid x_l^k)$ to be governed by the set of foreground labels at the frame indexed $k-1$ for the $n$-order neighbourhood system $\mathcal{N}_l$ centered at pixel coordinates $u_l$. Moreover, since $P(\mathcal{C}_F \mid x_l^k) = E[\mathbf{1}_{\mathcal{C}_F} \mid x_l^k]$, we can use the expectation of the indicator variable $\mathbf{1}_{\mathcal{C}_F}$ given the pixel $x_l^k$ as a means to compute $P(\exists_{\nu_i} = 1 \mid s)$.

Since the expectation $E[\mathbf{1}_{\mathcal{C}_F} \mid x_l^k]$ can be written as

$$P(\mathcal{C}_F \mid x_l^k) = \frac{1}{\mid \mathcal{N}_l \mid} \sum_{x_m^k \in \mathcal{N}_l} P(\mathcal{C}_F \mid x_l^k, x_m^k) \tag{4}$$

we can use M-estimators [41] to express the probability $P(\mathcal{C}_F \mid x_l^k, x_m^k)$ as follows

$$P(\mathcal{C}_F \mid x_l^k, x_m^k) = \frac{\sum_{x_j^k \in \mathcal{N}_m} h_\gamma(x_j^k) P(\mathcal{C}_F \mid x_l^k, \theta_m^{k-1})}{\sum_{x_j^k \in \mathcal{N}_i} h_\gamma(x_j^k, \theta_m^{k-1})} \tag{5}$$

where $\theta_m^{k-1}$ is a vector of hyperparameters that govern the distribution of the foreground pixels in the neighbourhood $\mathcal{N}_m$ and $h_\gamma(x_j^{k-1})$ is a robust weighting function with bandwidth $\gamma$, which, in our approach, is given by a Tukey function [43] of the form

$$h_\gamma(x_j^k) = \begin{cases} \left(1 - \xi_j(\gamma)^2\right)^2 & \text{if } \mid \xi_j(\gamma) \mid \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\xi_j(\gamma) = H[\gamma - (u_j - u_l)^2]$, $H_\gamma[\cdot]$ is a Heviside unit-step function and, as before, $u_j$ are the pixel coordinates of $x_j^k$.

By considering neighbourhoods $\mathcal{N}_j$ of the same size for all $j$ and selecting the bandwidth $\gamma$ so as to be consistent with the neighbourhood-sizes, we can greatly simplify the expression for $P(\mathcal{C}_F \mid x_l^k)$. By substituting Equation 5 into Equation 4, and After some algebra, we get

$$P(\mathcal{C}_F \mid x_l^k) = \frac{1}{\mid \mathcal{N}_m \times \mathcal{N}_l \mid} \sum_{x_m^k \in \mathcal{N}_l} \sum_{x_j^k \in \mathcal{N}_m} P(\mathcal{C}_F \mid x_j^k, \theta_m^{k-1}) \tag{7}$$

To complete our analysis, we turn our attention to the computation of the posterior probabilities $P(\mathcal{C}_F \mid x_j^k, \theta_m^{k-1})$. Making use of the Bayes theorem, the class-conditional probabilities $P(x_l^k, \theta_F \mid \mathcal{C}_F)$ and the priors for the classes $\mathcal{C}_F$ and $\mathcal{C}_B$ we have

$$P(\mathcal{C}_1 \mid x_j^k, \theta_m^{k-1}) = \text{sig}(f_j^k) = \frac{1}{1 + \exp(f_j^k)} \tag{8}$$

where $\text{sig}(\cdot)$ is the logistic sigmoid function and $f_j^k$ is a discriminant function of the form

$$f_j^k = \ln \left( \frac{P(x_j^k, \theta_m^{k-1} \mid \mathcal{C}_B) P(\mathcal{C}_B)}{P(x_j^k, \theta_m^{k-1} \mid \mathcal{C}_F) P(\mathcal{C}_F)} \right) \tag{9}$$

The sigmoid $\text{sig}(f_j^k)$ is, effectively, a "squashing function" that maps the function $f_j^k$ into the interval $[0, 1]$. Furthermore, $f_j^k$ can be related to discriminant analysis by assuming $P(\mathcal{C}_F \mid x_j^k, \theta_m^{k-1})$ to be normal. Let the probability distributions for the background and foreground be governed by the parameter vector $\theta_m^{k-1} = \{\beta_m^{k-1}, \varphi_m^{k-1}\}$, where $\beta_m^{k-1} = \{\mu_F, \mathbf{\Sigma}_F\}$ and $\varphi_m^{k-1} = \{\mu_B, \mathbf{\Sigma}_B\}$ are the parameter vectors, i.e. mean and covariance, for the foreground and background distributions, respectively, in the neighbourhood $\mathcal{N}_m$ at frame $I_{k-1}$.

This is an important observation since the mean and covariance parameters may be computed making use of the foreground and background labels for the frame indexed $k - 1$. This suggests the use of a sequential silhouette extraction scheme in which, at frame $I_0$ we use the labels for the foreground and background pixels, as given by the user-supplied contour, to compute the posterior probabilities for each pixel at frame $I_1$. Once the posterior probabilities are at hand, we can use the probabilities $P(\mathcal{C}_F \mid x_l^1)$ to recover the label-set for frame indexed $1$. We do this by computing the optimal cut-off value so as to separate the distributions for the foreground and background pixels. This process is repeated for the subsequent frames in the sequence using the label-sets corresponding to previous frames.

In Appendix A, we show how the optimal cut-off value $r_k$ can be recovered from the posterior foreground and background $P(\mathcal{C}_F \mid x_l^k)$. For now, we continue with our analysis and proceed assuming that the set of background and foreground pixels are available. Let the set of pixels in the

foreground be given by $\mathcal{X}_F^k = \{x_i^{k-1} \mid P(\mathcal{C}_F \mid x_i^{k-1}) \geq r_k\}$. Similarly, the background pixel-set is given by $\mathcal{X}_B^k = \{x_i^{k-1} \mid P(\mathcal{C}_B \mid x_i^{k-1}) < r_k\}$. Recall that, for the first frame of the sequence, the foreground and background pixel-sets are extracted from the contour information provided by the user. From the pixel sets $\mathcal{X}_F^{k-1}$ and $\mathcal{X}_B^{k-1}$, it becomes a straightforward task to compute the vectors $\beta_m^{k-1} = \{\mu_F, \mathbf{\Sigma}_F\}$ and $\varphi_m^{k-1} = \{\mu_B, \mathbf{\Sigma}_B\}$ for every neighbourhood $\mathcal{N}_m$.

With these ingredients, the discriminant function $f_j^k$ becomes

$$f_j^k = -\frac{1}{2}\ln\left(\frac{\mid \mathbf{\Sigma}_B^{-1} \mid}{\mid \mathbf{\Sigma}_F^{-1} \mid}\right) - \frac{1}{2}(x_l^k - \mu_B)^T \mathbf{\Sigma}_B^{-1}(x_l^k - \mu_B) + \frac{1}{2}(x_l^k - \mu_F)^T \mathbf{\Sigma}_F^{-1}(x_l^k - \mu_F) + \ln(\varrho)$$

where $\varrho$ is the ratio of the number of background to foreground pixels in the neighbourhood $\mathcal{N}_m$ at the view indexed $k-1$.

Therefore, we can compute the discriminant function above for each pixel given a neighbourhood $\mathcal{N}_m$ at frame $I_{k-1}$ and recover the probability $P(\mathcal{C}_1 \mid x_j^k, \theta_m^{k-1})$ making use of Equation 8. With the probability at hand, we can use Equation 7 and compute $P(\mathcal{C}_F \mid x_l^k)$, from which the expectation $E[\mathbf{1}_{\mathcal{S}} \mid \exists_{\nu_i} = 1]$ can be recovered. Moreover, noting that $\sum_{j\in\{0,1\}} E[\mathbf{1}_s \mid \exists_{\nu_i} = j]P(\exists_{\nu_i} = j) = 1$ in Equation 2, we can make use of Equation 3 and write

$$P(\exists_{\nu_i} = 1 \mid s) = \frac{1}{\mid \mathcal{X}_i \mid} \sum_{x_l^k \in \mathcal{X}_i} \left\{ \frac{1}{\mid \mathcal{N}_m \times \mathcal{N}_l \mid} \sum_{x_m^k \in \mathcal{N}_l} \sum_{x_j^k \in \mathcal{N}_m} \frac{1}{1+\exp(f_j^k)} \right\} \tag{10}$$

## 3.2 Pixel Data

In this section, we employ the probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$ to recover the parameters that govern the probability $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i)$ by casting the problem into a supervised classification setting.

So far, we have focus in the probabilities emanating from the user-provided silhouette $\mathcal{S}$. In this section, we turn our attention to the probability $P(\exists_{\mathcal{V}_i} \mid \mathcal{X}_i)$. To commence, we note that the attributes of those pixels in $\mathcal{X}_i$ can be viewed as vectors, for which each entry correspond to a value of brightness, colour, etc. Following this rationale, we treat the pixels as N-dimensional vectors, i.e. $x_l^k = [x_l^k(1), x_l^k(2), \ldots, x_l^k(N)]^T$ and, making use of the formalism introduced in the previous section, we write

$$P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i) = \frac{1}{1+\exp(g_i)} = \text{sig}(g_i) \tag{11}$$

where

$$g_i = \ln \left( \frac{P(\mathcal{X}_i \mid \exists_{\mathcal{V}_i} = 0) P(\exists_{\mathcal{V}_i} = 0)}{P(\mathcal{X}_i \mid \exists_{\mathcal{V}_i} = 1) P(\exists_{\mathcal{V}_i} = 1)} \right)$$

is a discriminant function.

To take our analysis further, we note that, solely on the basis of the probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$, a number of voxels will have a null posterior $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$. This is due to the fact that, for those voxels whose back-projection correspond to pixels that are labelled as background, the probability $P(\mathcal{C}_F \mid x_l^k)$ will be identical to zero. This is consistent with shape-from-silhouette approaches, in which the object contour information is available. As a result, we can start the carving process and remove, in an iterative fashion, voxels with null $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$ until no further removals can be effected.

This is an important observation, since we can employ the removed voxels and those that can not be further carved, i.e. those for which $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S}) \neq 0$, to recover the discriminant function $g_l^k$. This can be done by viewing the logistic sigmoid $\text{sig}(g_i)$ as the probabilistic output of a classifier. This classifier, whose output depends on pixel information, should be consistent with those carving operations effected on silhouette information alone. As a result, we can use the voxels carved using the posterior probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$ to train a classifier in which the function $g_i$ is modelled as follows

$$g_i = a_1 \sum_{n=1}^{N} \alpha_n y_n(\mathcal{X}_i) + a_2 \tag{12}$$

where $y_n(\mathcal{X}_i)$ is a function operating on the $n^{th}$ dimension of the pixels in $\mathcal{X}_i$, $\alpha_n$ are real-valued weights and $a_i$, $i = \{1, 2\}$ are constants.

Here, we make use of a function $y_n(\cdot)$ of the form

$$y_n(\mathcal{X}_i) = \frac{1}{\mid \mathcal{X}_i \mid} \sum_{x_l^k \in \mathcal{X}_i} (\mu_n - x_l^k(n))^2 \tag{13}$$

which yields the average variance of the pixel-values for the voxel $\mathcal{V}_i$ across those views in which its visible. Our choice of function $y_n(\cdot)$ reflects the notion that, for those voxels that exhibit large variation in terms of pixel attributes across different views, the value of the discriminant function should be large.

Here, we follow a two step process to recover the weights $\alpha_n$ and the constants $a_1$ and $a_2$. Firstly, we use AdaBoost [5] to recover the weights. Secondly, we use maximum likelihood to compute the constants $a_1$ and $a_2$ [28]. We do this by making use of a number of voxels for training purposes, whose binary label variables are assigned as follows. We commence by building the set of carved

voxels $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{|\Psi|}\}$, where $\psi_t$ are the set of voxels removed at iteration $t$. Note that $\Psi$ is the set all voxels for which the probability $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$ is null. Let the set of those voxels which, at iteration $\mid \Psi \mid$, can be back-projected onto the views in the image sequence with $y_n(\mathcal{X}_i) \neq 0 \, \forall \, n$ be $\psi_{|\Psi|+1}$. With these ingredients, we set to unity the label variable $\zeta_i$ for the voxel $\mathcal{V}_i$ if $\mathcal{V}_i \in \psi_{|\Psi|-1}$. The label variable is $\zeta_i = -1$ if $\mathcal{V}_i \in \psi_{|\Psi|+1}$.

As mentioned earlier, we commence by removing those voxels for which the probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{S})$ are null. This process allows us to recover the parameters of the discriminant function $g_i$ and compute the probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$, as given in Equation 1. In practice, after the voxels in $\Psi$ have been carved, the further removal of voxels requires a decision rule. This decision rule is based upon a cutoff value $\tau$. Therefore, a voxel $\mathcal{V}_i$ is removed if its probability $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$ is less or equal to $\tau$. Otherwise, the pixel exists in the object. This variable $\tau$ can be computed using the method in Appendix A from the probabilities $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S})$ corresponding to those voxels in $\psi_{|\Psi|+1}$. The use of $\psi_{|\Psi|+1}$ for computing $\tau$ hinges in the notion that the mixture of both classes, i.e. removed and existing voxels, will be more evident for those voxels $\mathcal{V}_i$ that are close to the boundary of the object.

Hence, after computing the cutoff $\tau$ from the voxels in $\psi_{|\Psi|+1}$ making use of the formalism in the Appendix A, we continue our plane sweep carving process removing those voxels for which $P(\exists_{\mathcal{V}_i} = 1 \mid \mathcal{X}_i, \mathcal{S}) \leq \tau$ until no further voxels are removed.

## 3.3 Projective Voxel Space

As mentioned earlier, we perform our carving algorithm on a projective voxel space which provides a consistent voxel projection with respect to the image planes. This is irrespective of the position of voxels on the object space. Traditionally, the cubic tessellation of voxels used in space carving algorithms provides voxels $\mathcal{V}_i$ which are all of the same size, regardless of their position with respect to the viewpoint. This is an important observation since the projection area onto each image changes with the distance of the voxels from the camera centre under consideration. With this in mind, in this section, we provide a method aimed at recovering a voxel space based upon a Voronoi tessellation procedure which minimises the voxel back projection error.

To commence, let $I_k$ denote the image of the object of interest from the $k$th camera, $k = 1, .., n$, where $n$ is the number of images in the sequence. For each $I_k$, we can view the voxels as being spanned by a cone-like volume, defined by rays passing through the four corners of the image $I_k$

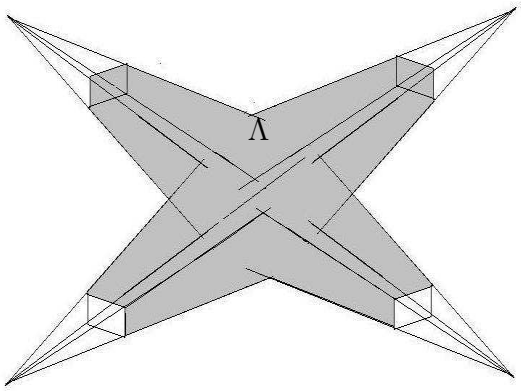
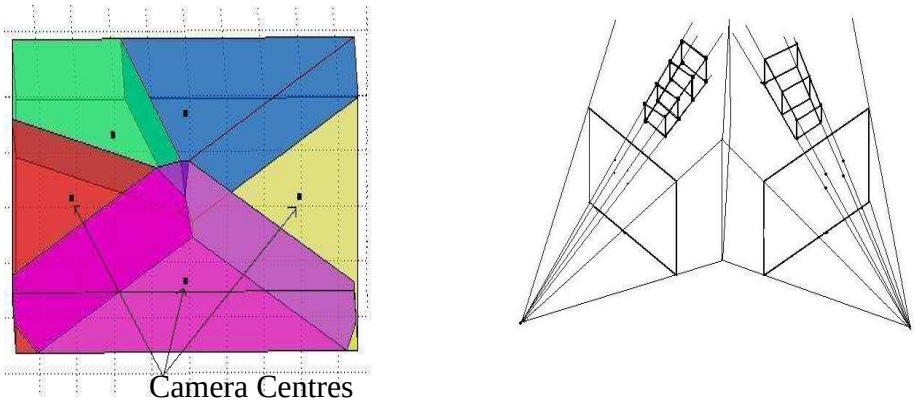



Figure 2. Union of Cone Volumes.

and the camera centre $cc_k$. Given that the object is contained in each image, it is guaranteed, as proved in [20], that the object will be contained in the union $\Lambda$ of these cone volumes, as shown in Figure 2.

The union $\Lambda$ can be voxelised for purposes of space carving by noting that the voxel-centres should lie on the projecting rays passing through the camera centres. Moreover, the voxel-centres should be such that their back-projection onto the image plane is invariant with respect to their distance from the camera centre. Thus, in terms of proximity to reference views, the boundaries of points being projected from each viewpoint are defined using a Voronoi tessellation [2, 26] as shown in Figure 3(a). Our choice of the Voronoi tessellation hinges in the nature of the problem itself, which aims at solving a problem dependent on a proximity geometric relationship so as to divide the 3D carving region into sections according to the geometric position of a number of camera centres. Thus, here we exploit voronoi diagrams so as to divide the plane according to the nearest-neighbour rule by associating each point with the region of the plane closest to it. Thus, our approach divides the carving space into voxels whose centres are the nearest-neighbours to the pixels corresponding to the closest image plane.

This approach has two main advantages. Firstly, it enables us to populate the carving space with voxels generated by points projecting from the nearest camera centre. Secondly, it permits us to associate each voxel $\mathcal{V}_i$ to its nearest color map. Moreover, it can be shown that by using a Voronoi tessellation to subdivide the carving space, the back projection error for the colour is minimum. This is due to the fact that, as a consequence of the tessellation, the colour mapping for any voxel




(a) (b)

Figure 3. (a) Region partition yielded by the Voronoi tessellation; (b) Pixel projection in the tessellated space.

would be drawn from the view corresponding to the nearest camera. The result of the carving space tessellation for the case of two reference views is shown in Figure 3(b). In practice, the tessellation is computed from the voxel-centres making use of Delaunay triangulation [3]. Figure 4 shows the geometric comparison of voxel back-projection using our voxel space and the usual cubic space. Note that, for our Voronoi tessellation, the back-projection of any voxel onto its respective image plane consists of one pixel. In constrast, the back projection for cubic tesselations varies with respect to the distance of the voxel under consideration from the camera centres.

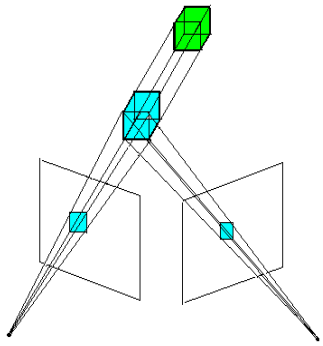
For a more formal error analysis, we follow [21]. We commence by noting that the image noise caused by optic devices is often considered to be white and Gaussian. Consider the case of an $m$-order neighbourhood $|\mathcal{N}_m| = 1$, i.e. the case when the voxel $\mathcal{V}_i$ back projects to a region covering only 1 pixel. Let $\tilde{x}_p^k$ be the irradiance of the pixel under consideration. Then we have

$$\tilde{x}_p^k = x_p^k + I_n^k$$

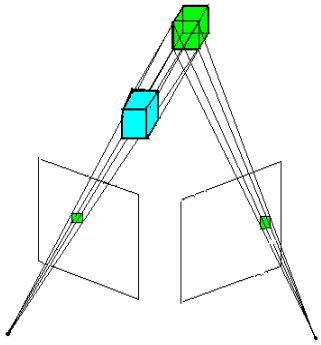
where $I_n^k \approx \mathbf{N}(0, \sigma_I^2)$. Likewise, for projections on multiple pixels, i.e. for $|\mathcal{N}_m| > 1$, we represent the mapping as

$$\begin{aligned} \tilde{x}_p^k &= \frac{1}{|\mathcal{N}_m|} \sum_{\mathcal{N}_m} (x_l^k + I_n^k) \\ &= \frac{1}{|\mathcal{N}_m|} \sum_{\mathcal{N}_m} x_l^k + \frac{1}{|\mathcal{N}_m|} \sum_{\mathcal{N}_m} I_n^k \end{aligned} \tag{14}$$

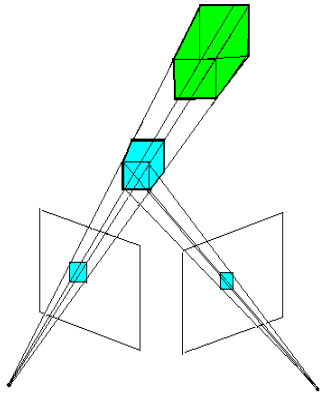
Thus, for our proposed voxel space, the error for the coloring is not accumulated as a consequence of large $|\mathcal{N}_m|$ for those voxels that are further from the image plane. Also, for the recovery



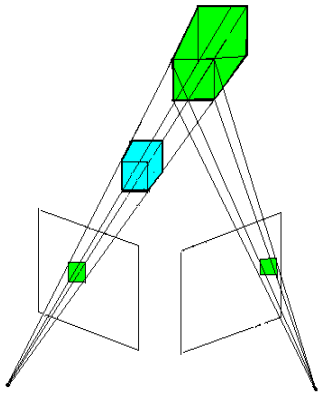
(a)



(b)



(c)



(d)

Figure 4. Comparison of voxel back-projection onto the images in the sequence; (a)&(b) Back projection of a cubic voxel; (c)&(d) Back projection of a voxel drawn from a Voronoi tessellation.

of the colour mapping, the voxel colouring is in close accordance to its respective pixel $\tilde{x}_p^k$. Thus, by using the proposed voxel space, we minimize the error by ensuring that $|\mathcal{N}_m| \approx 1$ and, consequently, obtain a mapping for each voxel corresponding to a single pixel on the respective image.

Further, note that one of the main consequences of our projective voxel space is that the quantity $|\mathcal{N}_m|$ does not depend on the voxel position, but rather can be fixed so as to account for calibration errors and noise corruption. Thus, denoising can be effected by back projecting onto a fixed number of pixels. Thus, the voxel-centres can be obtained making use of rays passing not through every pixel but every other pixel and so on. Our projective voxel space still ensures that there is consistency on the number of pixels being projected onto, irrespective of the size of voxels. The color mapping, for each remaining voxel will thus be determined by taking the mean of the back-projection over an equal number of pixels regardless of their position in the carving space.

# 4 Implementation Issues

Having presented the theoretical foundations of the method in the previous section, there are a number of issues that deserve further discussion.

Firstly, we explain how we obtained our projective voxelated space. We divide the carving 3D region using Voronoi tessellations with respect to the camera centres as follows. Let $cc_{1..n}$, be the camera centres with $n$ being the number of cameras to be considered. We insert these points into the space, creating cells in the space closer to each camera center $cc_l$. The regions of space are outlined by vertices, $v_s, s \in \mathbf{I}$. Each added point $cc_i$ is marked with references to its associated vertices. We can now add voxel centres to the partitioned regions. As voxel centres are added, all vertices, $v_d$ closer to $cc_l$ are deleted and the new vertices are formed using the voxel centres for each neighbouring vertices, $v_u$ of $v_d$. The list of this new regions is then updated with respect to point and vertex mappings. For a more detailed algorithm description and further optimizations refer to [16]. We implemented this using list iterators storing separate lists for voxel centres and all resulting vertices $v_s$. Thus, at the end of the algorithm, the list containing $v_s$ is iterated to obtain the resulting boundaries. With the vertices and points at hand, the next step is to associate projected voxel centres to each camera centre by solving the inequalities as outlined by the vertices for each camera field of view. This delivers, at output, a set of projected voxel centres whose nearest camera is the one corresponding to its line-of-sight. Once a voxel is removed, the list is updated through a search on the tessellation list. This is effected by determining the neighbouring voxels. To do

this, we rely on the tessellation list which contains the reference to all those voxels projected to a particular pixel on the image plane.

The carving process itself is implemented using list iterators. At the beginning of the carving process, we build a binary look-up list. This list is used to keep track of those voxels in the surface of the model and their corresponding color value. When a voxel is removed, all the entries corresponding to neighbouring voxels are set to unity. In this manner, only those voxels which are on the surface of the carving need to be processed at each iteration. Finally, when there are no further removals to be effected, each remaining voxel takes on its corresponding stored color value from the look-up list. For visibility of voxels, we adopted the concept of item buffers as used in [10]. The item buffer is used to record, for every pixel in an image, the surface voxel that is visible from the pixel and provides an efficient means to voxel removal.

Note that, for the computation of the posterior probabilities $P(\mathcal{C}_F \mid x_l^k)$, only those regions $\mathcal{N}_k$ whose foreground and background pixel-sets are both non-empty need be processed. This allows the use of iterators across those neighbourhoods for which neither the background nor the foreground label-sets are null. This is due to the fact that, if either of these is empty, the probability $P(\mathcal{C}_F \mid x_{,}^k, \theta_m^{k-1})$ will take the hard-limit values of zero or unity. This is understandable, since the label-sets indicate whether the foreground or the background occur given the pixel $x_l^{k-1}$. It the foreground does not occur, its probability is then null. In the contrary, if the region is all foreground, its probability is one.

Also, since the carving process depends solely on the silhouette information at start-up, we can reduce the computational burden of the algorithm by computing the discriminant function $g_i$ only for those voxels carved after iteration $\mid \Psi \mid -2$. This is also desirable since this assures the existence of a training domain containing in which foreground and background voxels. This also reduces the bias on $\tau$. This is understandable since, by employing the user-provided information as a hard constraint for the recovery of $\tau$ and the discrimant function $g_i$, the voxel boundary of the object employed for training becomes a volumetric upper bound for the carving process.

Also note that, as the number of views increases, it is somewhat expected the accuracy to increase accordingly. The reasons for this are twofold. Firstly, a larger number of views implies additional information will be available for the computation of the discriminant function. Secondly, a larger amount of views also provides, in general, a smaller displacement in camera position between frames in the sequence. This, in turn, implies that the silhouette recovery step across the image sequence is less prone to error.

# 5 Experiments

In this section, we illustrate the utility of our method for purposes of 3D reconstruction. To this end, we compare our results to those yielded by alternative methods and provide robustness test results. We have used three sets of sequentially acquired imagery of real-world objects. At this point, it is worth noting that, to our knowledge, there are no semi-supervised space carving methods elsewhere in the literature. Here, we provide a qualitative comparison of our method to that in [7] and bundle adjustment. Both, the method in [7] and ours are based upon statistical techniques. Bundle adjustment methods, in the other hand, share with our method the use of bundle rays for the 3D recovery and are standard in the community. However, these results must be interpreted with caution since both alternatives are unsupervised in nature. Moreover, our method does not require that every ray must intersect at least one voxel is imposed upon the removal process. In contrast with other probabilistic methods, such as the alternative, we do not assume that if a scene point is occluded in a view, then there must be another surface point along the line-of-sight.

The datasets are comprised of 30 views, acquired in house, for a toy camel, a leather boot and a mannequin head. In our experiments, we have provided the algorithm, at input, the views of the objects under study and a silhouette, which corresponds to the contour of the model at the first frame of the sequence. Note that, in practice, semisupervised segmentation methods such as the Random Walker [35] can be used to recover an input silhouette. Other methods, such as edge detection, can also be employed. Nonetheless, in the case of edge detection methods, a disambiguation between foreground, i.e. the object of interest, and background may be required. Also, note that for the quantitative analysis throughout the section we have set the number of trials to 10 for purposes of computing the mean squared error and its variance.

In Figure 5, we show on the top row, the first view of our datasets. The input silhouettes are shown in the bottom row. In the silhouette panels, the pixels corresponding to the foreground class, i.e. $\mathcal{C}_F$, are shown in black. As outlined in the previous section, our algorithm commences by computing the probabilities $P(\mathcal{C}_F \mid x_l^k)$ and the foreground-background labels for every frame in the sequence. In Figure 6, we show sample frames for the datasets used in our experiments. The probabilities $P(\mathcal{C}_F \mid x_l^k)$ and the labels recovered making use of the cutoff value $r_l$ are shown in the second row of figure. From the panels, its clear that the probabilities recovered by the algorithm are in good accordance with the model outline.

In Figure 7, we show 3D reconstructions obtained using our method. For the method in [7],

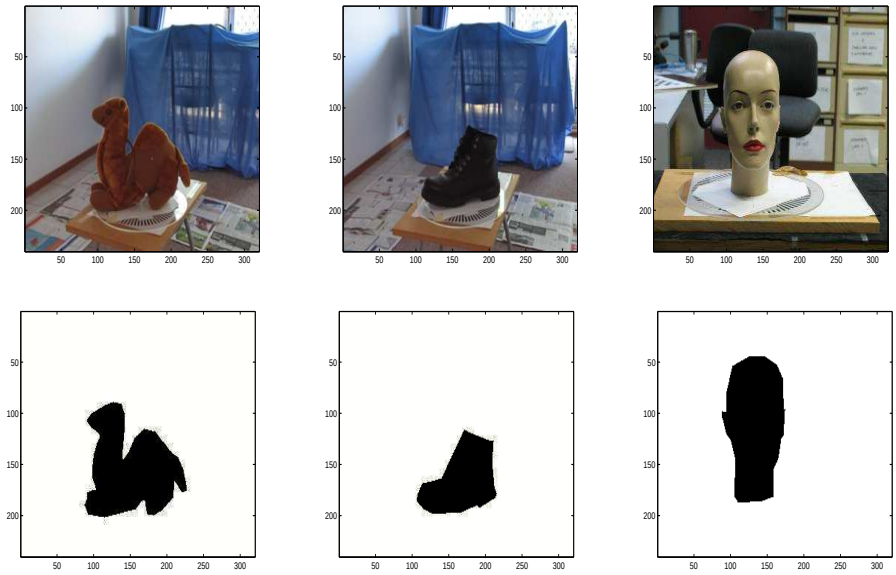


Figure 5: Top row: First frame for the image sequences under study; Bottom row: corresponding input silhouettes.

we have used the GVC algorithm [10] as an alternative to the plane sweep approach. This is so as to relax the ordinal visibility constraints in [7]. For the bundle adjustment, we have used the method in [42]. The 3D point cloud generated by bundle adjustment methods was rendered using VTK [32] after removing outliers via RANSAC. As evidenced by the reconstructions, compared to the alternatives, our method displays better accuracy and detail. The two main reasons for the improved results are the usage of the proposed projective voxelated space and the semi-supervised approach using silhouette information to guide the carving process. The back projection yielded by the projective voxel space provides the optimal color mapping for each remaining uncarved voxel. In the figure, we present both, the 3D shapes rendered using a Lambertian blue shade and the voxel mapping onto the image pixels. We have done this so as to facilitate comparison with the scanned ground truth data in the top row of Figure 7.

Note that, for our method, each voxel is mapped onto one pixel on the image. As a consequence of the use of the GVC approach, the mapping obtained by the alternatives corresponds to the average for the set of pixels recovered through the back-projection of the voxels onto the corresponding views. An alternative to GVC in this regard is to take the weighted average of pixels from the back-projection on all visible images. Nonetheless, these methods may compromise detail and accuracy. By comparing the reconstructions of our method and GVC, one can notice the difference in features where the average may not reliably represent the color of surface, such as the eyes, eye brows

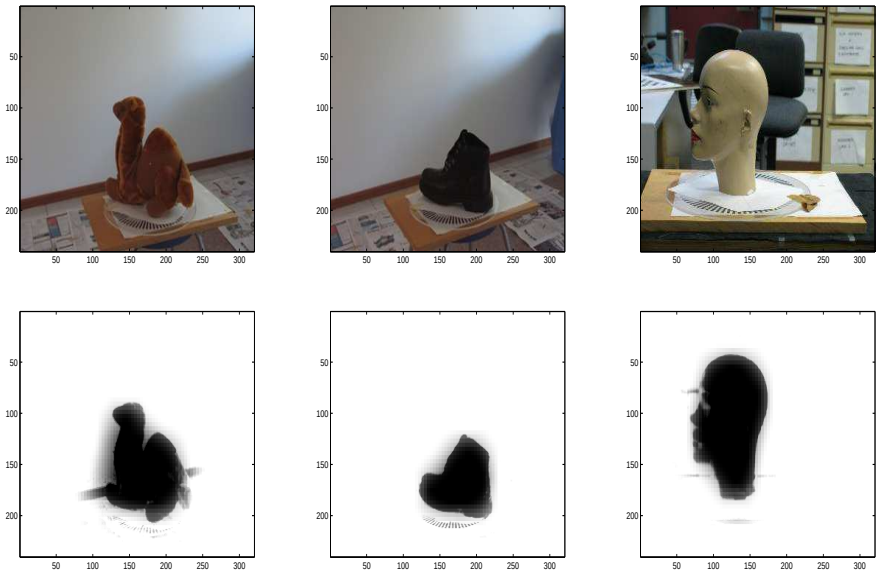


Figure 6: Top row: Sample input views; Bottom row: values of $P(\mathcal{C}_F \mid x_l^k)$ for the views in the top row.

and the nose for the mannequin head. For the reconstruction of the leather boot, we can see the detail of the lace buckle has been preserved by using our method. From the figure, its evident that the results yielded by bundle adjustment methods lack the detail and color mapping accuracy. This is mainly due to the scarce features available for correspondence on some portions of the object. This is also due to the presence of outliers, which degrade the 3D reconstruction result.

We also provide a quantitative analysis for our method making use of 5 real-world images for the toy camel, the leather boot and the mannequin head, taken from novel viewpoints, i.e. views that are not in the dataset used for 3D recovery purposes. To this end, we have added Gaussian noise with zero mean and increasing variance to the input images. For each noise level, we have recovered the model and rendered it at viewpoints equivalent to the novel real-world images. This has been done following [7], where the rendering is done by integrating along each ray and approximating, via maximum likelihood, the marginalised probabilities of a voxel existing. We have then computed the mean least-squared difference between the rendered model and the noise-free, segmented object on the novel views. The plots of the mean least-squared difference, as a function of the noise variance are shown in Figure 8. From the two left-most plots, we can appreciate that the mean-squared error for the renderings exhibit what appears to be a linear dependency with respect to the standard deviation of the added noise. This can be attributed to the use of the silhouette information, which mitigates the impact of pixel-colour corruption due to noise corruption.

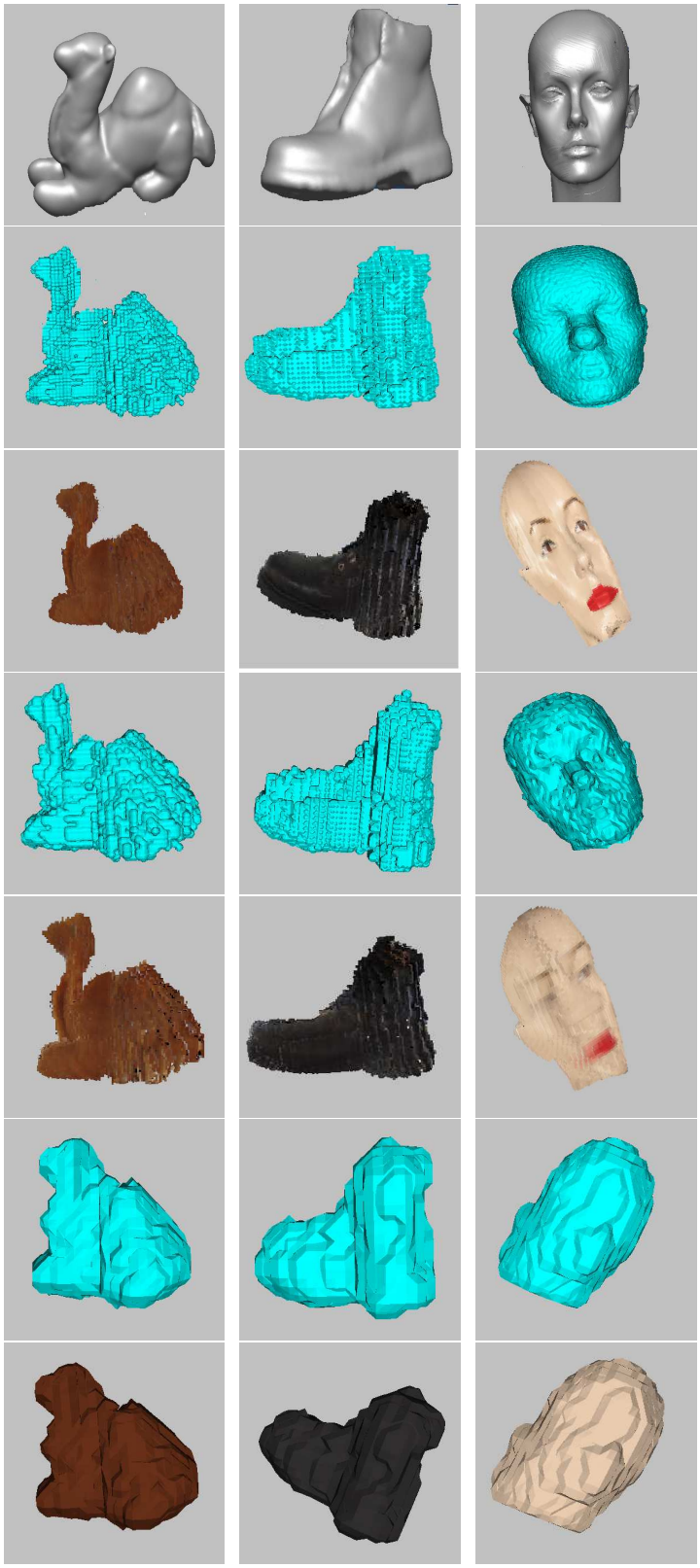


Figure 7: 3D reconstructions. From top-to-bottom: scanned meshes used as ground truth; 3D data recovered by our method; object renderings using our approach; 3D volumes recovered using the method in [7]; renderings for the volumes in the fourth row; 3D meshes recovered using bundle adjustment; renderings for the bundle adjustment results in the sixth row. [42].

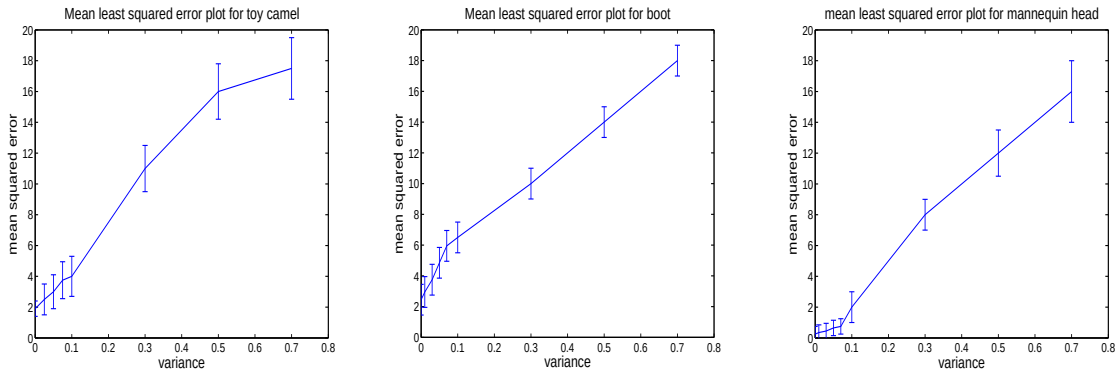



Figure 8: Error analysis of renderings obtained using our method (plot of mean squared-difference vs noise variance). From left-to-right: error plots for the camel toy views, leather shoe data and the mannequin head.

Now, we turn our attention to the quality of the 3D data recovered by our method. To further analyze the accuracy and the robustness of our 3D reconstruction algorithm, we compared the recovered 3D data to the ground truth. The ground truth data was obtained using a Polhemus Scorpion laser scanner at an accuracy of 0.05 inches and is shown in the top row of Figure 7.

Again, as before, we added Gaussian noise with increasing variance to the input images. To conduct our tests, we used the goodness of fit for the reconstruction of the leather boot, toy camel and the mannequin head to the laser-scanned ground truth data. This comparison was effected using Procrustes analysis [33]. Procrustes analysis determines a linear transformation between the two sets of 3D points. Let the centered coordinates of the data point indexed $i$ be $\tilde{p}_i = [x_i - \mu_x, y_i - \mu_y, z_i - \mu_z]^T$, where $\mu_x$, $\mu_y$ and $\mu_z$ are the mean data-coordinate values in the x, y and z axis. With these ingredients, the matrix of normalized 3D point-coordinates is given by $\tilde{\mathbf{D}} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_N]^T$. A Procrustes transformation of the matrix of normalized 3D point-coordinates $\tilde{\mathbf{D}}$ is of the form $\mathbf{Q} = \mathcal{R}\tilde{\mathbf{D}}$ which minimizes the normalized sum of squared errors

$$\mathcal{E} = \frac{\| \mathbf{M} - \mathbf{Q} \|^2}{\| \tilde{\mathbf{D}} \|^2} \tag{15}$$

where $\mathbf{M}$ is a matrix whose $i^{th}$ row corresponds to the coordinates of the ground truth point indexed $i$ and $\mathcal{R}$ is a transformation matrix. In the equation above, we have assumed that the ground-truth point coordinates are centered, i.e. the centroid of the ground-truth point cloud is at the origin.

It is known that minimizing $\mathcal{E}$ is equivalent to maximizing $\mathrm{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T\mathcal{R}]$ [15]. Let the singular value decomposition (SVD) of $\tilde{\mathbf{D}}\mathbf{M}^T$ be $\mathbf{USV}^T$. The maximum of $\mathrm{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T\mathcal{R}]$ is achieved when

$\mathbf{V}^T \mathcal{R} \mathbf{U} = \mathbf{I}$. As a result, the optimal transformation matrix $\mathcal{R}$ is given by

$$\mathcal{R} = \mathbf{V}\mathbf{U}^T \tag{16}$$

The goodness-of-fit for this transformation is determined using the sum of squared errors $\mathcal{E}$ between the points recovered by our algorithm and the 3D ground-truth data. By using $\tilde{\mathbf{D}}$ as an alternative to the raw data point-coordinates, the error $\mathcal{E}$ is normalized by the sum of squares for the centered 3D point-coordinates. The use of the normalized error makes the quantity $\mathcal{E}$ devoid of scaling and translational components in the 3D data under comparison.

The error plots for the goodness-of-fit as a function of noise variance are shown in Figure 9. Note that, in the figure, we have set, in the sake of consistency, the y-axis range for all plots to the interval $[0, 2]$ and used different line styles for each of the alternatives. From the figure, it is evident that the performance of our algorithm is not overly affected for Gaussian noise variances below $0.6$. This suggests that the corruption on the silhouette information does not play an important role in the accuracy of our space carving for variances below $0.6$. Also notice that, even with the presence of higher level of noise, the output for the mannequin head is noticeably better than the results yielded for the other two objects under study. This is due the the fact that the variation of the occluding contour with respect to successive views is low, which, in turn, improves the performance of the silhouette extraction algorithm and its robustness to noise corruption.

Also, note that, for noise free views, the algorithm in [7] performs well. However, the effect of noise corruption becomes evident after the gaussian noise variance surpasses $0.3$. Moreover, the alternative requires a free parameter to be adjusted. This is related to the voxel removal and its dependent upon image quality. In our experiments, we have set this parameter to its optimal value making use of cross validation. For our algorithm, the cutoff value is determined automatically and does not require empirical setups.

The effects of noise on the reconstruction yielded by bundle adjustment is more noticeable than in the case of the alternatives. This is as a consequence of bundle adjustment depending on image features for purposes of matching. The detection of these features is overly affected by noise corruption. Here, the mean procrustes error increases steadily after the variance exceeds $0.4$. Thus, overall comparison of the 3D data for our method and the two alternatives suggest that, even without the presence of noise, our approach yields a margin of improvement. Moreover, the use of the user-provided input silhouette to guide our 3D reconstructions makes the method devoid of free parameters and noise corruption.

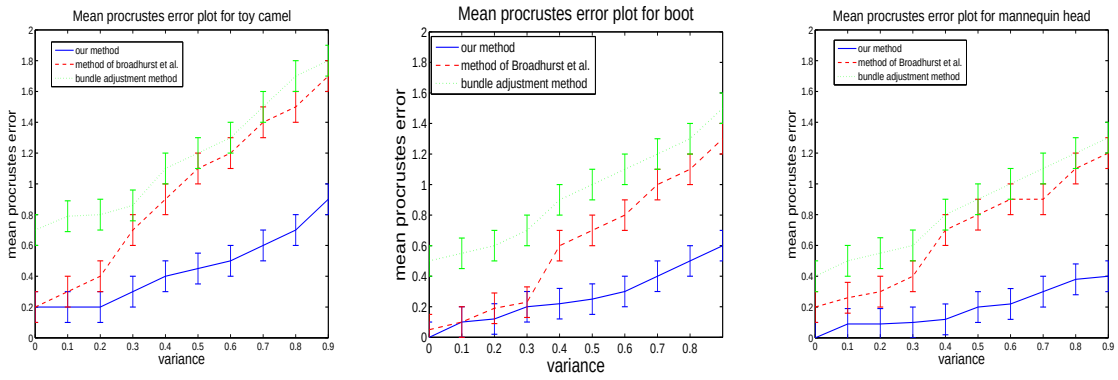



Figure 9: Mean procrustes error for the 3D data recovered using our method (continuous line), the method of Broadhurst et al.(broken line) and bundle adjustment (dotted line). From left-to-right: Plots for the toy camel, leather boot and the mannequin head.

Now, we turn our attention to the effects of silhouette perturbation on our algorithm. To this end, we have blurred the user-provided input silhouette and, simultaneously, added a Gaussian jitter with increasing values of variance to the optimal cut-off value $r_k$ used for the recovery of the posterior foreground and background values. As in our previous comparison, we have separated five randomly selected views from the sequences. Once the 3D data is recovered from the remaining views, we render the objects from the same viewpoints as those corresponding to the excised real-world images. We then compare these renderings with this real-world imagery. In Figure 10 we show the user-provided input silhouettes, the blurred ones and the foreground-background masks after jitter with a variance of 10 has been added to the cut-off value $r_k$. The mean-squared error plots as a function of jitter variance for the rendered novel views are shown in Figure 11. Note that, from the panels, we can appreciate that, despite large amounts of jitter and the blurring of the input silhouette, the error rates in the plots are comparable to those in Figure 8.

As a final comparison, we performed 3D reconstructions using [7] and [42] and compared their output with the results yielded by our method using the set of images [1] provided by the visual geometry group at Oxford University. These images have been widely used in the community and are well suited to the ordinal visibility constraint required in [7]. In contrast with the experiments effected on the toy camel, the boot and the mannequin head, for the Oxford University views we have used the plane sweep algorithm in [7]. The reconstructions are depicted in Figure 12. Note that the reconstruction yielded by [7] misses some details, such as the chimney structure on the house. In contrast, our method preserved detail in the scene. This is as a result of the use of

[1]The sample images are available at http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

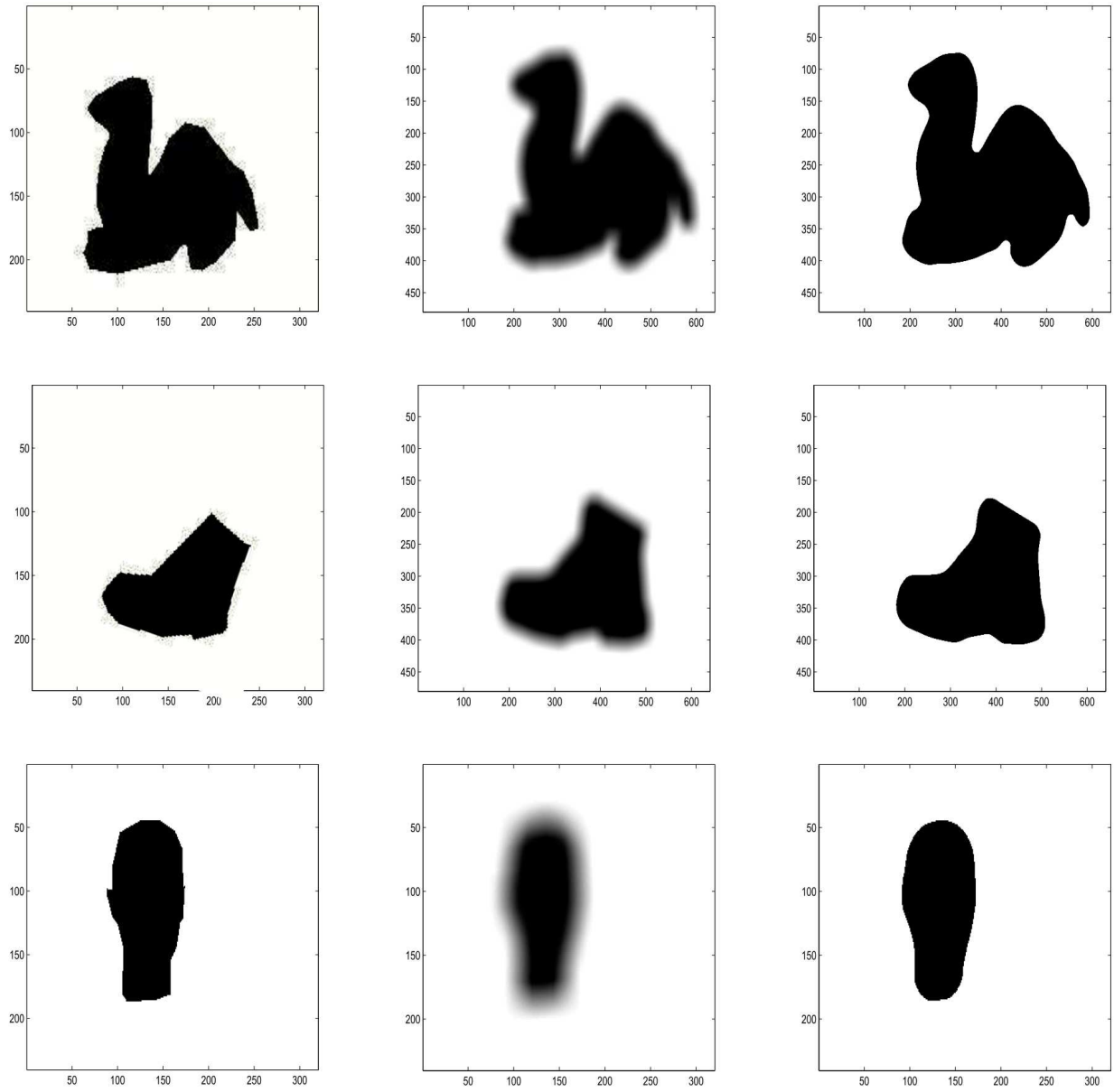


Figure 10: From left-to-right: user-provided silhouettes; silhouettes after the Gaussian blurring operation; foreground-background mask after a jitter of 10 has been introduced into the cut-off value.

prior information in a semi-supervised fashion so as to guide our reconstruction. Thus, yielding relatively more accurate results. The color mapping obtained is also better as compared to the alternatives. In terms of the 3D data recovered, the point cloud obtained from [42] is sparse, with a number of outliers. Again, this is due to the dependence of the method upon features such as edges and corners. This manifests in the accuracy of the reconstructed points.

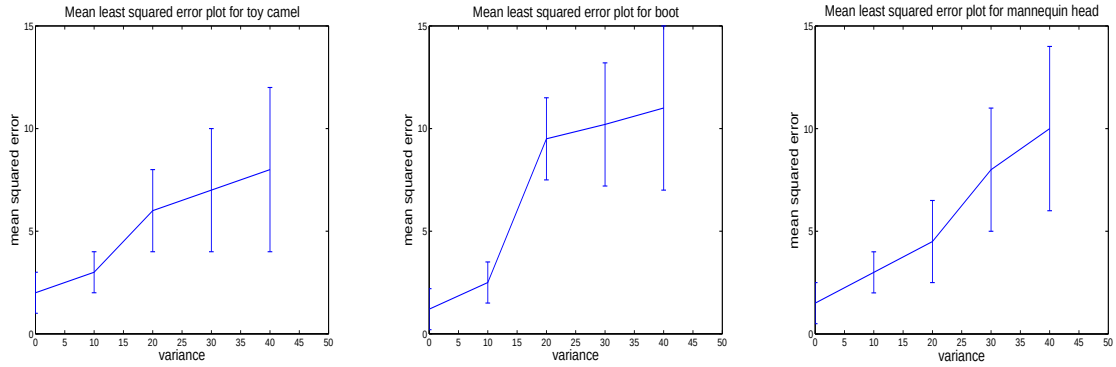



Figure 11: Error analysis of renderings obtained using our method when jitter is added to the cut-off value employed for the silhouette prior computation step. From left-to-right: Plots for the toy camel, leather boot and the mannequin head.

# 6 Conclusions

We have presented a semi-supervised approach to space carving which casts voxel removal in an evidence combining setting. Our method is statistical in nature and combines the information from a user-supplied silhouette and the pixel-variance across those views in which a voxel is visible. In this manner, the posterior probability of a voxel existing can be computed and inference upon its removal can be effected. Note that our proposed method is effectively able to isolate the object of interest from the background and is devoid of free parameters. Moreover, the approach presented here can be further extended to multiple input silhouettes in a straightforward fashion making use of a Markovian formulation. This is due to the fact that the approach taken here can be viewed as a propagation across views in the scene which makes no assumption regarding whether this is "forward" of "backward" in the sequence. In other words, multiple silhouettes can be used so as to correct error propagation across views at the silhouette prior probability computation step. We have also presented an approach to obtain a voxelated carving space which is projective in nature. This projective voxelated space ensures consistency over the projection of voxels onto images irrespective of its position in space. This yields better color mapping and decreases rendering error. We have illustrated the utility of the method to recover volumetric data by performing experiments using real-world imagery and provided a quantitative analysis. We have also provided comparison of our method to alternatives elsewhere in the literature.

## A. Cut-off Value Recovery

Here we describe the method used in this paper to recover the values of $r_k$ and $\tau$. The method

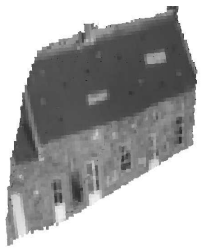
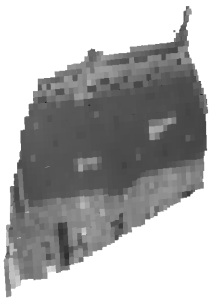
(a)  (b)  (c)  (d) 

Figure 12: (a) Sample image from the Oxford Univerisy image set; (b) 3D reconstruction using our method our method; (c) Result yielded by the method of Broadhurst et al. [7]; (d) Reconstruction yielded by bundle adjustment [42].

presented here builds on that in [27] and its aimed at recovering the optimal cut off value of a binomially distributed set of univariate random variables $z_i \in \mathcal{Z}$. To do this, we maximise Fisher's linear discriminant [12] separability measure. This measure is given by

$$\lambda = \frac{S_b^2}{S_w^2} \tag{17}$$

where $S_b$, $S_w$ are between and within class variances given by

$$\begin{aligned} S_w^2 &= \omega_1 S_1^2 + \omega_2 S_2^2 \\ S_b^2 &= \omega_1 \omega_2 (\mu_1 - \mu_2)^2 \end{aligned} \tag{18}$$

where $\mu_i$ and $S_i$ are the mean and variance of the class indexed $i$ and $\omega_1$, $\omega_2$ are real-valued class weights.

To take our analysis further, we note that the maximum of $\lambda$ is given by $\omega^* = \omega_1 = \omega_1$, where $\omega^*$ is the optimum value of the weights, which can be computed making use of the expression

$$\omega^* = \frac{\mu_1 - \mu_2}{S_1^2 + S_2^2} \tag{19}$$

Moreover, making use of $\omega^*$, it can be shown that the optimum cut-off value is given by

$$\vartheta = \{\eta | (\omega^*)^2 = \omega_\eta (1 - \omega_\eta)\} > 0 \tag{20}$$

where $\omega_\eta$ is a real-valued function of the univariate random variables defined as follows

$$\omega_\eta = \frac{1}{\mid \Omega_\eta \mid} \sum_{z_i \in \Omega_\eta} z_i \tag{21}$$

and $\Omega_\eta$ is the set of variables whose value is less or equal than $\eta$, i.e. $\Omega_\eta = \{x_i \mid z_i \leq \eta\}$. Thus, in practice, we can recover $\vartheta$ making use of a linear search governed by the condition in Equation 20.

# References


[1] M. Agrawal and L. S. Davis. A probabilistic framework for surface reconstruction from multiple images. *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2:II–470–II–476, 2001.

[2] Franz Aurenhammer. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[3] C. B Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.

[4] B.G. Baumgart. *Geometric Modeling for Computer Vision.* PhD thesis, Stanford University, 1974.

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

[6] Jeremy S. De Bonet and Paul A. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *ICCV (1)*, pages 418–425, 1999.

[7] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for the space carving algorithm. *ICCV01*, pages 388–393, July 2001.

[8] N.L. Chang and A. Zakhor. Constructing a multivalued representation for view synthesis. *International Journal of Computer Vision*, 2(45):157–190, 2001.

[9] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *Comput. Vision Graph. Image Process.*, 36(1):100–113, 1986.

[10] W. Bruce Culbertson, Thomas Malzbender, and Gregory G. Slabaugh. Generalized voxel coloring. In *Workshop on Vision Algorithms*, pages 100–115, 1999.

[11] C.H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[12] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[13] M. Grum and A.G. Bors. Refining implicit function representations of 3-d scenes. In *British Machine Vision Conference*, pages II:710–719, 2004.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in Computer Vision*. Prentice Hall, 2000.

[15] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[16] Philip M. Hubbard. Improving accuracy in a robust algorithm for three-Dimensional voronoi diagrams. *Journal of Graphics Tools*, 1(1):33–45, 1996.

[17] S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *Intl. Journal of Computer Vision*, 72(2):159–178, 2007.

[18] S.B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. *Proc. Conference on Computer Vision and Pattern Recognition*, 1:I103–I110, 2001.

[19] R. Koch, M. Pollefeys, and L.V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *European Conference on Computer Vision*, pages 55–71, 1998.

[20] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. Technical Report TR692, 1998.

[21] Musik Kwon, Kyoung Mu Lee, and Sang Uk Lee. A statistical error analysis for voxel coloring. In *ICIP (1)*, pages 425–428, 2003.

[22] A. Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.

[23] Z. Lazebnik, Y. Furukawa, and J. Ponce. Projective visual hulls. *Intl. Journal of Computer Vision*, 74(2):137–165, 2007.

[24] C. Liang and K.-Y. Wong. Robust recovery of shapes with unknown topology from the dual space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12).

[25] W. Martin and J. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[26] Atsuyuki Okabe, Barry Boots and0 Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams*. John Wiley, 2000.

[27] N. Otsu. A thresholding selection method from gray-level histobrams. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[28] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[29] M. Pollefeys, L.J. Van Gool, and A. Oosterlinck. The modulus constraint: A new constraint for self-calibration. In *International Conference of Pattern Recognition (ICPR2006)*, pages I: 349–353, 1996.

[30] Michael Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Comput. Vision Graph. Image Process.*, 40(1):1–29, 1987.

[31] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. *Proc. Conference on Computer Vision and Pattern Recognition*, 2:49–54, 1990.

[32] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*. Kitware, Inc. Publishers, 2004.

[33] G. A. F. Seber. *Multivariate Observations*. Wiley, 1984.

[34] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. of Computer Vision*, 35(2):151–173, 1999.

[35] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and ramdom walker which yields a new algorithm. In *Proc. of ICCV*, 2007.

[36] Gregory G. Slabaugh, W. Bruce Culbertson, Thomas Malzbender, and Ronald W. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *International Workshop on Volume Graphics*, 2001.

[37] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *In Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 345–352, 2000.

[38] Partha Srinivasan, Ping Liang, and Susan Hackwood. Computational geometric methods in volumetric intersection for 3d reconstruction. *Pattern Recogn.*, 23(8):843–857, 1990.

[39] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Pattern Analysis and Machine Intelligence*, 20(10):1091–1097, October 1998.

[40] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing,*, 58(1):23–32, July 1993.

[41] S.Z.Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2001.

[42] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag.

[43] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

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