Applying Bayesian segmentation in volumetric silhouette-based reconstruction

Willie Brink

Department of Mathematical Sciences
University of Stellenbosch, South Africa
wbrink@sun.ac.za

Abstract

We present a technique for incorporating probabilistic (or soft) segmentations into the shape-from-silhouette (SFS) technique for volumetric reconstruction from a set of images. In basic SFS points in 3D space are back-projected to all the images and if any of these falls outside the contours of the object, the point is disregarded. We extend the concept to a more probabilistic nature and show that high quality models can be obtained by thresholding the final point probabilities. The method used for determining the soft segmentations is based upon a Bayesian framework, but the idea of combining probabilistic segmentations would remain the same for any soft segmentation method.

1. Introduction

We consider the problem of reconstructing a volumetric 3D model from a set of images depicting some physical object from different viewpoints. Shape-from-silhouette [1] is an obvious first and widely used approach, even just as a simple initialization for a more advanced procedure [2]. The aim in shape-from-silhouette is to determine the intersection of the conic solids defined by the outlining contours of the object in all the images. If an accurate segmentation of the object from the background is available for every image in the set, along with the spatial relationships between the different cameras in world coordinates, the problem is straightforward and easily solved by, for example, space carving or exact polyhedral intersection [3, 4].

Calibrating an array of cameras is a well-studied topic and is usually performed off-line. Segmenting the object from every image, on the other hand, can be far more intricate and deserves some attention. The general problem of image segmentation remains, to this day, an enormous challenge. In a recent survey and evaluation Seitz *et al.* [5] noted that most multi-view volumetric reconstruction algorithms assume that accurate silhouettes are available, and do not consider the crucial problem of obtaining them if they are not.

We attempt to address, at least partially, the segmentation problem in the context of shape-from-silhouette by showing how probabilistic segmentations, i.e. segmentations that carry some uncertainty, can be incorporated in the reconstruction.

The method presented can operate alongside any technique that produces probabilistic segmentations. For the purposes of this paper we decided upon a Bayesian framework and describe in some detail this classic approach to probabilistic image segmentation. It models prior probability distributions over known object and non-object regions and then establishes maximum likelihood estimates for pixels to be classified. A so-called "soft" segmentation results which, unlike traditional "hard" segmentations, do not yield a decisive yes-or-no answer to the question of whether a certain pixel belongs to the object but rather a probability.

A sensible combination of probabilistic segmentations over all the images in the set yields a probabilistic reconstruction in which 3D points are assigned probabilities of being part of the object. A threshold can be applied to these point probabilities in order to generate a deterministic and decisive 3D model (as Fig. 1 below illustrates) if need be. Experimental evaluation of the obtained 3D models suggests both high accuracy and completeness by the measures defined in [5].

2. Bayesian segmentation

The Bayesian approach to image segmentation has its roots in digital matting [6, 7] where the aim is to estimate opacities for object pixels so that objects can be composed with new scenes without visible edge artefacts. Thin wisps of hair and motion blur are notorious difficulties and many improvements have since been explored; see for example [8].

A common assumption in digital matting is that the observed intensity $\mathbf{c}$ of each pixel in the input image is some linear composition of a foreground colour $\mathbf{f}$ and a background colour $\mathbf{b}$, in the following way:

$$\mathbf{c} = \alpha\mathbf{f} + (1 - \alpha)\mathbf{b}. \tag{1}$$

Here α denotes the opacity of the pixel and ranges between 0 (pure background) and 1 (pure foreground). For the rest of this section $\mathbf{c}$, $\mathbf{f}$ and $\mathbf{b}$ are assumed to be 3×1 column vectors containing for example RGB-components, and α is a scalar.

Figure 1: *Images of an alien sculpture from various views, a probabilistic reconstruction and a 3D model obtained by thresholding.*

Of course, the problem of finding $\mathbf{f}$, $\mathbf{b}$ and α that satisfy (1) for a given $\mathbf{c}$ is determinately underconstrained. The idea behind Bayesian segmentation is to use prior probabilities on the unknowns and pick optimal solutions by a maximum-likelihood criterion.

In order to define sensible priors on $\mathbf{f}$, $\mathbf{b}$ and α an initial segmentation is needed that divides the image into three regions: foreground, background and unknown. Such a segmentation is normally called a *trimap*. The foreground and background regions provide the information (colour and α-values) needed to process pixels in the unknown region. Some ideas on the generation of trimaps are explored in section 4.1 of this paper.

For a specific pixel in the unknown region we proceed to find values for $\mathbf{f}$, $\mathbf{b}$ and α that maximize the joint posterior probability of $\mathbf{f}$, $\mathbf{b}$ and α, given $\mathbf{c}$. From Bayes' theorem,

$$P(\mathbf{f}, \mathbf{b}, \alpha | \mathbf{c}) \propto P(\mathbf{c}|\mathbf{f}, \mathbf{b}, \alpha) P(\mathbf{f}) P(\mathbf{b}) P(\alpha), \qquad (2)$$

which yields the following unconstrained optimization problem to be solved:

$$\arg\max_{\mathbf{f},\mathbf{b},\alpha} \Big[L(\mathbf{c}|\mathbf{f}, \mathbf{b}, \alpha) + L(\mathbf{f}) + L(\mathbf{b}) + L(\alpha) \Big], \qquad (3)$$

with $L(\cdot) = \log P(\cdot)$. It remains to define the log-likelihoods $L(\mathbf{c}|\mathbf{f}, \mathbf{b}, \alpha)$, $L(\mathbf{f})$, $L(\mathbf{b})$ and $L(\alpha)$.

Following [6] we model $P(\mathbf{c}|\mathbf{f}, \mathbf{b}, \alpha)$ as a Gaussian PDF with mean $\alpha\mathbf{f} + (1-\alpha)\mathbf{b}$ and some specified standard deviation σ_c. Then

$$L(\mathbf{c}|\mathbf{f}, \mathbf{b}, \alpha) = \frac{-||\mathbf{c} - \alpha\mathbf{f} - (1-\alpha)\mathbf{b}||^2}{2\sigma_c^2} + \text{terms}. \qquad (4)$$

The extra "terms" correspond to normalization factors and can be omitted in (3) as they remain constant with respect to the optimization parameters $\mathbf{f}$, $\mathbf{b}$ and α.

The prior probability $P(\mathbf{f})$ is estimated from foreground colours of pixels known and already processed in a neighbourhood around the current pixel. Let N denote this set of pixels. The contribution of every pixel in N can be weighed by its α-value (placing higher confidence in more opaque pixels) and, relying on the spatial coherency of the object, its proximity to the current pixel (stressing nearby pixels over those further away). Supposing that $\mathbf{f}_i$ and w_i denote the foreground colour and weight of pixel i respectively, such that $\sum_{i \in N} w_i = 1$, a weighted mean $\overline{\mathbf{f}}$ and weighted covariance matrix $\Sigma_{\mathbf{f}}$ is determined as

$$\overline{\mathbf{f}} = \sum_{i \in N} w_i \mathbf{f}_i, \quad \Sigma_{\mathbf{f}} = \frac{1}{1-W} \sum_{i \in N} w_i (\mathbf{f}_i - \overline{\mathbf{f}})(\mathbf{f}_i - \overline{\mathbf{f}})^T, \quad (5)$$

with $W = \sum_{i \in N} w_i^2$. These two entities define a Gaussian distribution such that

$$L(\mathbf{f}) = -\tfrac{1}{2}(\mathbf{f} - \overline{\mathbf{f}})^T \Sigma_{\mathbf{f}}^{-1} (\mathbf{f} - \overline{\mathbf{f}}) + \text{terms}. \qquad (6)$$

Again, the extra normalization terms have no effect on the solution to (3) and can be omitted. The log-likelihood of $\mathbf{b}$ is modelled in a similar way:

$$L(\mathbf{b}) = -\tfrac{1}{2}(\mathbf{b} - \overline{\mathbf{b}})^T \Sigma_{\mathbf{b}}^{-1} (\mathbf{b} - \overline{\mathbf{b}}) + \text{terms}. \qquad (7)$$

For now we adopt the assumption in [6] that the likelihood of α is constant and thus omitted from (3). Naturally this is a rather strong assumption that may be marginally applicable at best, but it does simplify the optimization problem and yields reasonable results.

Note that the partial derivatives of the function to be maximized in (3) are non-linear in $\mathbf{f}$, $\mathbf{b}$ and α, so that the quest for an analytical solution may be an arduous one. We may need to depend upon some numerical optimizer for a solution. However, for the sake of efficiency, it turns out that the problem can be broken up into two easier sub-problems: first fix α and solve for $\mathbf{f}$ and $\mathbf{b}$; then fix $\mathbf{f}$ and $\mathbf{b}$ and solve for α.

With α fixed at some value, setting the partial derivatives of (3) with respect to $\mathbf{f}$ and $\mathbf{b}$ equal to zero leads to the following set of equations:

$$\left.\begin{aligned} \frac{\alpha}{\sigma_c^2}[\mathbf{c} - \alpha\mathbf{f} - (1-\alpha)\mathbf{b}] - \Sigma_{\mathbf{f}}^{-1}(\mathbf{f} - \overline{\mathbf{f}}) = \mathbf{0} \\ \frac{1-\alpha}{\sigma_c^2}[\mathbf{c} - \alpha\mathbf{f} - (1-\alpha)\mathbf{b}] - \Sigma_{\mathbf{b}}^{-1}(\mathbf{b} - \overline{\mathbf{b}}) = \mathbf{0} \end{aligned}\right\} \qquad (8)$$

which is linear in the unknowns $\mathbf{f}$ and $\mathbf{b}$. Values for $\mathbf{f}$ and $\mathbf{b}$ are obtained, and now fixed to find a value for α that maximizes the objective function in (3). Observe that this maximum is reached when $||\mathbf{c} - \alpha\mathbf{f} - (1-\alpha)\mathbf{b}||^2$ is minimum, and that

$$\alpha = \frac{(\mathbf{c} - \mathbf{b}) \cdot (\mathbf{f} - \mathbf{b})}{||\mathbf{f} - \mathbf{b}||^2} \qquad (9)$$

causes $\alpha\mathbf{f} + (1-\alpha)\mathbf{b}$ to be the orthogonal projection of $\mathbf{c}$ onto the line segment between $\mathbf{f}$ and $\mathbf{b}$, thus yielding the desired minimum.

We now alternate between (8) and (9) until convergence or until some maximum number of iterations is reached. For the first iteration α can be initialized to, for example, the mean of the α-values in the neighbourhood N.

Concerning the order in which pixels are processed: since prior probabilities are built from known or already processed pixels, and since the unknown region inside the image is surrounded by known foreground or background pixels, a sensible choice would be to process pixels on the border of the unknown region first. The unknown region is thereby effectively being eroded, a process which can be repeated until no more unknown pixels remain.

An apparent and important shortcoming of the procedure, as it stands, is that it attempts to fit a unimodal Gaussian distribution to neighbouring foreground colours as per (6), and one to neighbouring background colours as per (7). If both the object and the background exhibit a fair degree of homogeneity the model may be adequate but, of course, this is rarely the case. A quick fix would be to cluster the colours in the neighbourhood, solve for each pair of foreground and background clusters and then pick the solution corresponding to the maximum posterior likelihood (2). For future improvements we hope to rather investigate the effectiveness of Gaussian mixture models.

Example output of the described procedure is shown in Fig. 2. The input image depicts two actors in front of a green screen. In this case simple thresholding of the colour channels may yield a fairly good segmentation, but note in particular the ability of Bayesian segmentation to estimate the opacities of the hair strands in the closeup shown.

The probabilistic reconstruction technique described in this paper requires α-values for every pixel in every image, which will be interpreted as the probability of that pixel being part of the foreground (i.e. the object). After the application of Bayesian segmentation on every image in the sequence, let $A_i(r, c)$ denote the calculated α-value associated with the pixel at row r and column c in image i. Also, to simplify matters onwards, we assume that $A_i(r, c) = 0$ for any (r, c) that falls outside the boundaries of image i.

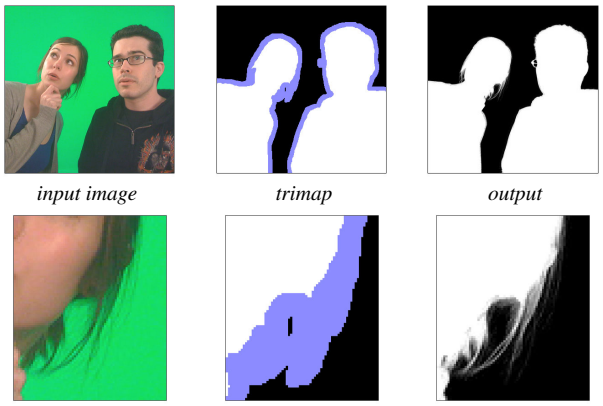


Figure 2: *Result of a Bayesian segmentation. Blue in the trimap indicates the unknown region, and the α-values (corresponding to pixel opacities) from the output are here depicted as greyscale intensities.*

3. Silhouette-based reconstruction

Shape-from-silhouette (SFS) is a well-known technique for volumetric reconstruction [1]. It takes a sequence of images of an object from different viewpoints, extracts the silhouette of the object from each and constructs a 3D model using also knowledge of the relative positions and orientations of the cameras.

The conversion from a colour image to a silhouette image removes a large amount of information and SFS can reconstruct (at best) the so-called *visual hull* of the object. It is essentially the maximal volume capable of reproducing all silhouettes of the object [9]. Certain concavities, such as the inside of a bowl, cannot manifest in any silhouette and therefore will not appear in the reconstructed visual hull. However, because the problem is well defined and generally easy to implement, SFS remains a popular choice even just as a starting point for more advanced reconstruction techniques that, for example, would construct a visual hull and then return to the information in the original images in an effort to chisel detail from it [2].

3.1. Basic shape-from-silhouette

There are chiefly two approaches to SFS for the generation of a 3D model from a sequence of silhouette images. In the first approach a portion of the 3D space, typically one visible by all cameras, is quantized into voxels (that is, volume elements). The center of every voxel is projected onto the image plane of every camera, and removed if it projects outside the silhouette in any of the images; see for example [10]. The second approach, which is slightly more accurate and does not introduce quantization and aliasing artefacts, involves the construction of a polyhedral surface from the intersection of conic solids, with polygonal bases defined by the silhouettes, that extend from the principal points of the corresponding cameras [3, 4].

In this paper we opt for the first approach, that of voxel quantization or space carving as it is often called, as it naturally lends itself to the incorporation of soft segmentations. Before discussing this extension we provide first some detail of the basic procedure.

Let P_i denote the 3×4 projection matrix associated with the image taken by camera i, in some reference frame constant for all cameras. The matrix should therefore contain extrinsic parameters that relate the position and orientation of camera i with all the others. Projection matrices are usually determined during some off-line calibration procedure where objects of known measurements are captured. Here we shall assume that the matrices P_i are already known.

A subspace bounding the region visible to all cameras is specified and divided into voxels, the size of which depending upon the desired output resolution. A possible way to obtain such a bounding region is to determine the intersection of the viewing frusta of all the cameras (obtainable from the projection matrices) by an efficient method such as [11].

The basic idea underlying voxel-based SFS is to project every voxel onto every image plane using the projection matrices. A voxel is believed to form part of the object if and only if it projects onto a silhouette region in every image.

Put more formally, suppose the center of a voxel j is denoted by the column vector $\mathbf{v}_j = [x_j \;\; y_j \;\; z_j]^T$. Its projection onto the image plane of camera i, in homogeneous coordinates, is determined as

$$\mathbf{p}_j^{(i)} = P_i \begin{bmatrix} \mathbf{v}_j \\ 1 \end{bmatrix}, \tag{10}$$

from which pixel coordinates $(r_j^{(i)}, c_j^{(i)})$ can be obtained. The reconstructed visual hull is then the collection of all voxels j such that

$$\sum_{i=1}^{n} S_i(r_j^{(i)}, c_j^{(i)}) = n, \tag{11}$$

with n the number of images and $S_i(r, c)$ the value of a pixel in silhouette image i, either 0 (background) or 1 (object). Here it is also assumed that $S_i(r, c) = 0$ if (r, c) falls outside the boundaries of image i.

Note that, because a voxel has to project to a silhouette region in every single image, we can immediately flag a voxel as soon as it falls outside such a region in one particular image, exclude it from that moment onwards and thus save a substantial amount of computation.

Once a set of object voxels has been identified, a polygonal surface representation is obtained easily by, for example, the marching cubes algorithm [12].

An example to illustrate the shape-from-silhouette technique is given in Fig. 3. The input data set, obtained from [13], consists of 24 images of a *Homo Heidelbergensis* skull replica on a ceramic ring. Accurate silhouettes and camera projection matrices are also provided. We specified a $300\times240\times300$ grid (or voxelation) in 3D space that yielded over 21 million voxels, about 14 million of which satisfy (11) and are therefore included in the reconstruction. A surface representation was also determined, consisting of about 700,000 polygons. Because the basic SFS procedure is so straightforward when silhouettes and camera matrices are available, the entire procedure could be implemented in a few lines of MATLAB-code and executed in less than a minute on a standard PC.

3.2. Incorporating soft segmentations

The procedure described above uses silhouette images that provide a definite yes-or-no answer to the question of whether a pixel belongs to the object, and ultimately if a voxel should be included in the 3D model. However, if probabilities are introduced in the segmentation, a "softer" decision can be incorporated. Moreover, obtaining accurate silhouette images may very well be a non-trivial exercise and it would be sensible to accommodate for some degree of uncertainty in that respect.

Recall that the process of Bayesian segmentation returns, for every image i, pixel opacities $A_i(r, c)$ ranging between 0

4 of the 24 input images and their corresponding silhouette images

3D reconstruction, shaded (left) and texture-mapped (right)

Figure 3: *A shape-from-silhouette reconstruction. Available (ground-truth) silhouettes were used in the construction of the visual hull. Certain concavities, notably the eye sockets, are filled because they do not manifest in any of the silhouettes.*

for pure background and 1 for pure foreground (which in this case is the object). For the sake of brevity we shall henceforth make use of the notation

$$\alpha_j^{(i)} = A_i(r_j^{(i)}, c_j^{(i)}) \tag{12}$$

to denote the α-value associated with the point where voxel j projects onto image i.

Our aim now is to amalgamate the $\alpha_j^{(i)}$'s associated with a particular voxel j, in order to define a probability of that voxel belonging to the object. Consider the following:

$$p(j) = \left(\prod_{i=1}^{n} \alpha_j^{(i)} \right)^{\frac{1}{n}}, \tag{13}$$

i.e. let the probability that voxel j is part of the 3D model be defined as the geometric mean of those α-values to which the voxel projects (a comparable approach was taken by De Bonet and Viola [14]). We chose the geometric mean, over say an arithmetic mean, because it provides a slightly more natural extension to the "and"-operator used in the basic SFS: a voxel must project to a silhouette region in *all* the images, and if *any* of these projections falls outside a silhouette region the voxel is discarded. The probability of voxel j belonging to the object should behave analogously. Note that $p(j)$ as defined in (13) is 1 if *all* the α-values are 1, and 0 if *any* of the α-values is exactly 0. Furthermore if *all* the α-values are close to 1 then $p(j)$ is close to 1, while if *any* of the α-values is close to 0 then $p(j)$ is penalized accordingly.

The calculation in (13) is performed for every voxel in the discretized subspace. Values thus obtained, ranging between 0 and 1, can now be thresholded in order to create a decisive 3D model. Alternatively, the unthresholded voxel probabilities can be carried over to a next phase in some larger system for further analysis. Examples of such larger systems, that could require volumetric reconstruction of objects, include 3D object recognition and robotic grasping systems.

4. Experiments

Experiments were carried out on three publicly available data sets: *alien*, *temple* and *kunli*. Example images from these sets are shown in Fig. 4 (cropped for display purposes). The table below shows for each the source, number of images, image size, and whether or not projection matrices and ground-truth segmentations are provided with the data. The *alien* set contains images from three cameras and each captured 8 images of sizes 1900×1600, 1600×1400 and 1400×1400 respectively.

data set	# images	image size	P-mats	gr. truth
alien [13]	24	*see text*	yes	yes
temple [5]	16	480×640	yes	no
kunli [15]	20	1024×768	yes	no

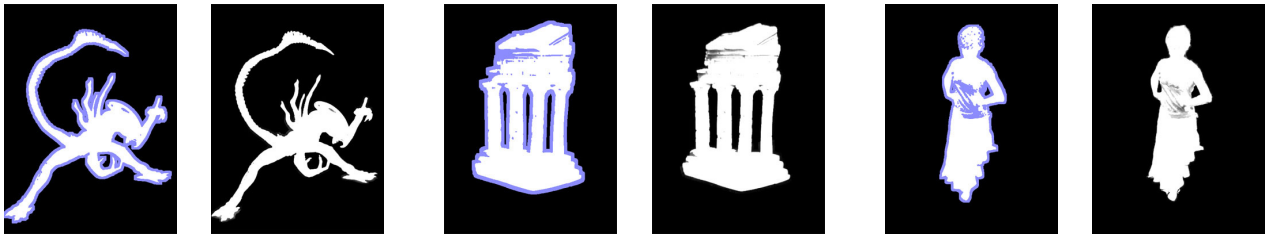
The procedure described in this paper, that of applying soft segmentations in volumetric reconstruction, involves mainly three steps: generating a trimap for each image, performing Bayesian segmentation and creating a probabilistic 3D model.

4.1. Generating trimaps

The Bayesian segmentation method discussed in section 2 requires a trimap. Prior probabilities are estimated from specified foreground and background regions, allowing maximum likelihood estimates to be calculated for the pixels in the remaining ("unknown") regions. Clearly the success of Bayesian segmentation hinges on the accuracy of the specified foreground and background (i.e. the "known") regions of the trimap. Unfortunately an all-purpose segmentation method that requires no prior knowledge of expected objects and/or background, yet successfully copes with the immense diversity in images, eludes researchers in the field to this day. For this very reason we are content with generating trimaps for every data set on a case-by-case basis.

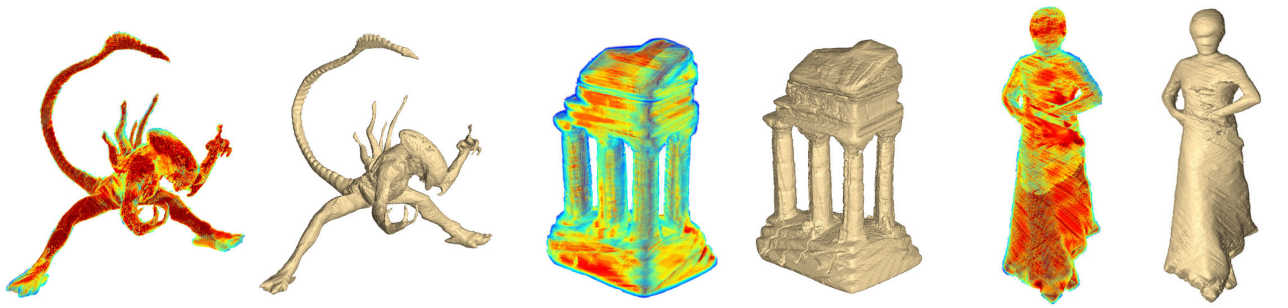
There is, however, a technique for generating trimaps that works quite well in many situations. A deliberately strict segmentation is performed that favours background, say by appropriately thresholding the different colour channels. Many pixels are thus miss-classified as background but those that are classified as foreground should at least be correct. The unknown region is then determined by dilating the foreground region a suitable number of times, hopefully engulfing all miss-classified background pixels in the process. Trimaps created in this way for the example images in Fig. 4 are depicted in Fig. 5. Background regions are represented in black, foreground in white and unknown regions in blue.

Figure 4: *Example images from the three data sets: alien, temple and kunli. The images were cropped for display purposes.*



trimap *segmentation* *trimap* *segmentation* *trimap* *segmentation*

Figure 5: *Trimaps generated for the example images, and results from Bayesian segmentation. Object boundaries are well delineated. However note the segmentation uncertainty in shadow regions of the temple image and in detail on kunli's T-shirt.*



voxel probabilities *thresholded* ($t = 0.85$) *voxel probabilities* *thresholded* ($t = 0.8$) *voxel prob.* *thresh.* ($t = 0.8$)

Figure 6: *Voxel probabilities for each data set, and decisive 3D models obtained by thresholding with the indicated t-values. Voxels were rendered with opacities proportional to their probabilities so that regions of high probability are observable.*

4.2. Bayesian segmentation

Bayesian segmentation is performed on every image in a data set. A parameter that needs to be decided upon is the size of the neighbourhood around an unknown pixel from which prior probabilities are estimated. In our implementation we allow this parameter to vary, depending on the amount of known information contained in the neighbourhood. Example output is shown in Fig. 5. It seems that object boundaries are found rather accurately. Note however that shadows in the *temple* image and the green design on *kunli*'s shirt cause some confusion.

In an attempt to quantify the accuracy of segmentations we compare output α-values with ground-truth hard segmentations (which are available for the *alien* set; for the *temple* and *kunli* sets we took up the meticulous task of manual segmentation). The ground-truth assigns either 0 or 1 to every pixel and the calculated α-values range between 0 and 1.

A possible indication of accuracy is the absolute difference between the ground-truth and the α-values averaged over all unknown pixels, i.e. over the pixels on which the segmentation was applied. It produces an average accuracy of about 92% for the *alien* set, about 81% for the *temple* set, and just under 80% for *kunli*. Keep in mind that the accuracy is measured over only the unknown regions which are likely to be "difficult" parts of an image. The higher accuracy for the *alien* set may be ascribed to the fact that those images contain a more-or-less uniformly dark object against a light background. The greater level of detail in *temple* and *kunli* results in less accurate segmentation.

4.3. Building a probabilistic 3D model

Voxel probabilities are determined by combining all associated α-values by means of equation (13). Visualizations of these probabilities are given in Fig. 6, using the standard "jet" colourmap (red indicates high probability, blue indicates low probability). In order to observe regions of high probability each voxel was rendered with an opacity proportional to its probability. We see that there is higher confidence in the reconstruction of *alien*, again due to the undemanding separability of the object and the background in those images.

The figure also shows some examples of decisive 3D models obtained by thresholding the voxel probabilities as indicated. We evaluated the quality of reconstructions resulting from different thresholds t by measuring accuracy and completeness as defined by Seitz *et al.* [5]. Results are displayed below.

	alien		*temple*		*kunli*	
t	acc	comp	acc	comp	acc	comp
0.00	3.58	45.2%	3.46	25.8%	2.26	63.7%
0.50	2.31	81.0%	2.31	69.2%	1.61	90.8%
0.75	1.64	90.1%	1.71	94.7%	1.16	96.8%
0.80	1.60	92.3%	2.86	97.2%	1.05	97.9%
0.85	1.38	95.3%	5.94	97.3%	0.90	98.9%
0.90	1.33	97.5%	8.71	93.7%	1.99	99.1%
0.95	1.98	86.6%	10.12	83.8%	5.05	96.3%
0.99	5.12	60.8%	10.38	63.9%	4.78	86.0%

The accuracy (**acc**) is the distance d such that 90% of the reconstruction is within d of the ground-truth mesh (GTM, obtained from the ground-truth segmentations), shown in *mm* for the first two and in *cm* for the third data set (the object in the *kunli* set is about 10 times larger than those in *alien* and *temple*). The completeness (**comp**) is the percentage of points on the GTM that are within a specified distance of the reconstruction, 1.5*mm* for *alien* and *temple*, and 1.5*cm* for *kunli*.

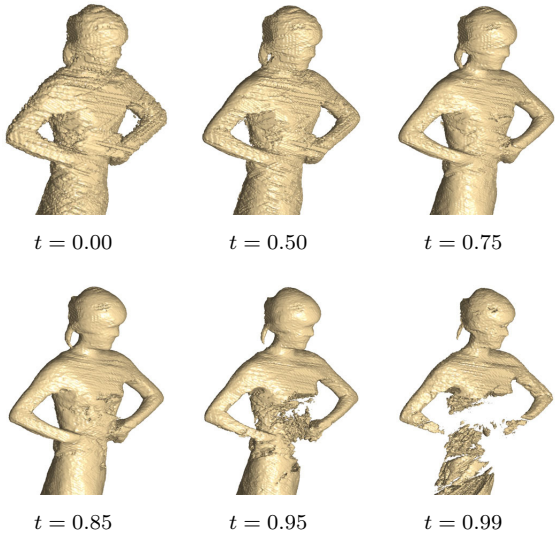


Figure 7: *Some thresholds applied to the probabilistic reconstruction of kunli. Smaller thresholds may produce unwanted volume, while larger thresholds may deny true positives.*

Based on these results there appears to be a threshold for each set that yields more-or-less optimal accuracy and completeness simultaneously. In general a smaller threshold would produce an excessively voluminous model, while a larger threshold may deny many voxels that should in actual fact be part of the model. These effects are illustrated in Fig. 7 where the probabilistic reconstruction of the *kunli* set is thresholded by a number of values. It seems that, in accordance with the table above, a threshold of about 0.85 might give the best model for this data set.

5. Conclusions

We showed how soft probabilistic segmentations can be incorporated into a shape-from-silhouette procedure for volumetric reconstruction. The Bayesian approach was taken to segment objects from background and a geometric mean of back-projected probabilities yielded a probabilistic reconstruction. Thresholding this reconstruction, to arrive at a decisive 3D model, can produce high quality models in terms of accuracy and completeness if the threshold is chosen carefully.

A possible direction for further work may involve the inclusion of calibration uncertainty. In this paper we assumed accurate calibration but situations can occur where such an assumption holds little validity. Another interesting venture would be to apply the proposed method to a set of images depicting an object with "fuzzy" boundaries like hair or wafts of smoke. We hope that in doing so the true value of Bayesian segmentation will shine through.

6. Acknowledgments

The author thanks Morné Galanos for his contributions during the early stages of this work. Gratitude is also extended to those people who make their data sets freely available: Lazebnik *et al.* [13] for the *skull* and *alien* sets, Seitz *et al.* [5] for the *temple* set, Liu *et al.* [15] for the *kunli* set and Chris Pirillo for the input image in Fig. 2.

7. References

[1] R. Szeliski, "Shape from rotation", IEEE Computer Vision and Pattern Recognition, 2:625–630, 1991.

[2] G. Vogiatzis, C.H. Esteban, P.H.S. Torr, 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.

[3] J.-S. Franco and E. Boyer, "Exact polyhedral visual hulls", British Machine Vision Conference, 1:329–338, 2003.

[4] S. Lazebnik, Y. Furukawa and J. Ponce, "Projective visual hulls", International Journal of Computer Vision, 74(2):137–165, 2007.

[5] S. Seitz, B. Curless, J. Diebel, D. Scharstein and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms", IEEE Computer Vision and Pattern Recognition, 1:519–526, 2006.

[6] Y.-Y. Chuang, B. Curless, D.H. Salesin and R. Szeliski, "A Bayesian approach to digital matting", IEEE Conference on Computer Vision and Pattern Recognition, 2:264–271, 2001.

[7] M.A. Ruzon and C. Tomasi, "Alpha estimation in natural images", IEEE Conference on Computer Vision and Pattern Recognition, 1:18–25, 2000.

[8] M. Sindeyev, V. Konushin, V. Vezhnevets, "Improvements of Bayesian matting", GraphiCon, 88–95, 2007.

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

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

[11] B. Chazelle, "An optimal algorithm for intersecting three-dimensional convex polyhedra", SIAM Journal on Computing, 21:671–696, 1992.

[12] W.E. Lorensen and H.E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm", Proceedings of ACM SIGGRAPH, 163–169, 1987.

[13] S. Lazebnik, Y. Furukawa and J. Ponce, "Visual hull data sets", `http://www-cvr.ai.uiuc.edu/ponce_grp/data/visual_hull`, last accessed October 2009.

[14] J. S. De Bonet and P. Viola, "Poxels: probabilistic voxelized volume reconstruction", International Conference on Computer Vision, 418–425, 1999.

[15] Y. Liu, Q. Dai and W. Xu, "A point cloud based multi-view stereo algorithm for free-viewpoint video", IEEE Transactions on Visualization and Computer Graphics, *to appear*, 2009.