Smooth Kernel Density Estimate for Multiple View Reconstruction

Article · November 2010

DOI: 10.1109/CVMP.2010.17

Jonathan Ruttle
Trinity College Dublin
9 PUBLICATIONS **53** CITATIONS
SEE PROFILE

Michael Manzke
Trinity College Dublin
43 PUBLICATIONS **142** CITATIONS
SEE PROFILE

Rozenn Dahyot
National University of Ireland, Maynooth
126 PUBLICATIONS **1,610** CITATIONS
SEE PROFILE

Smooth Kernel Density Estimate for Multiple View Reconstruction

J. Ruttle, M. Manzke and R. Dahyot

School of Computer Science and Statistics, Trinity College Dublin Ireland

Abstract

We present a statistical framework to merge the information from silhouettes segmented in multiple view images to infer the 3D shape of an object. The approach is generalising the robust but discrete modelling of the visual hull by using the concept of averaged likelihoods. One resulting advantage of our framework is that the objective function is continuous and therefore an iterative gradient ascent algorithm can be defined to efficiently search the space. Moreover this results in a method which is less memory demanding and one that is very suitable to a parallel processing architecture. Experimental results shows that this approach is efficient for getting a robust initial guess to the 3D shape of an object in view.

Keywords: Shape from silhouette, Kernel Density estimate, Newton-Raphson

1 Introduction

3D shape estimation from 2D image sequences has been widely carried out in the field of computer vision [18]. Merging information from multiple view images has a wide range of applications such as creating automatically 3D models of real objects or buildings, improving video surveillance [21].

In this article, we focus on improving one of the early method proposed in the domain called the visual hull. This method is modelling a discrete objective function that is memory demanding and while some optimisation techniques like octree spatial data structures can improve its memory requirement and computational cost it can still be considerable. It is very robust and it is still often used as an initial guess in many methods. We propose to reformulate this approach in a smooth continuous statistical framework that allows optimisation by efficient gradient ascent techniques. The paper is organised as follows: we review briefly the domain in section 1.1 and explain our motivation in section 1.2. Paragraph 2 presents our new smooth modelling and optimisation is presented in section 3. Experimental results show promising results (section 4) and further extension of our framework are discussed in the conclusion.

1.1 Review

Methods for inferring 3D volumes or surface from multiple camera views can be classified into two broad groups: the one that uses discrete objective functions in the modelling or in the optimisation strategy, and the one that uses continuous, smooth and/or differential ones. Amongst the discrete class, one of the most popular methods to compute 3D shape of interest objects is silhouette volume intersection [13]. Each 2D silhouette of the object creates a cone to the 3D world and the intersection of these cones from all camera views give an estimation of the 3D object volume and shape. This approach is called Visual Hull [11]. The objective function corresponds to a 3D histogram where each elementary cell (i.e. bin or voxel) is incremented each time a cone is intersecting it. This approach is similar to the Hough transform using 2D histogram for estimating aligned edges in images of contour [6, 9]. Both approaches are connected to the principle of duality [1, 12] and are related to the inverse Radon transform [5, 15]. Moreover both can be understood as the sum (or average) of the likelihoods computed with one observation (one pixel) taken at a time. This type of inference has been shown for being very robust [9]. However, when dealing with a 3D space, this discrete volume-based approach generally suffers from a heavy computation and memory requirement and if more dimensions are added for colour or motion these costs become infeasible.

A way to alleviate the computational load is to infer a surface instead of the whole volume [8]. The surface-based approach focuses on surface representation of the visual hull. The surface vertices and faces are estimated by intersecting the generalized cones from the occluding contours of the silhouettes. This method requires less computation and memory than the volume-based approach. However, the intersection in the 3D space is sensitive to numerical instabilities, especially in complicated objects.

Many methods can be formulated as minimising a functional to find the surface that best explains the consistency of the pixels in different images (that can be understood as a likelihood of the observations), with an added regularisation term (or prior) to ensure the smoothness of the solution [14]. The optimisation can then be performed using Graph cuts [14] that discretizes the continuous objective function which can be memory demanding and therefore is intrinsically not well suited to deal with inference in high dimensional space. Moreover, many algorithms for inferring surfaces or volumes from multiple images require a good initial guess around the object of interest. This can be specified by the user by a bounding box [14]. Alternatively, the discrete visual hull can also provide an efficient initial estimate to be refined [7].

1.2 Motivations

The visual hull is a robust crude estimate of the volume of the object of interest that can be used efficiently as initial guess for more accurate reconstruction methods and is widely used in tracking techniques. However its discreet nature is memory demanding and similarly to the Hough transform [9], the selection of the voxels (i.e. optimisation of the corresponding discrete objective function) that belong to the object is also not computationally efficient.

In general the visual hull method uses a regular sampling on a 3D grid to preform a reconstruction. This is where the world is broken up into small volume elements (voxels). An algorithm which uses data from the input images then determines whether the voxel belongs to the object or lies outside the object. The main issue with this approach is that the resolution of the final object depends on the size of the voxels and decreasing the size of the voxels mean greatly increasing the number of voxels. A two times increase in resolution means an eight time increase in voxels. The computation time and memory requirement are linear with the number of voxels so they too greatly suffer when an increase in resolution is required. For example the Middlebury [18] temple object size 10cm x 16cm x 8cm with a voxel size of 0.5mm requires over 10 million voxels, the dataset is set up so that each pixel scales to about 0.25mm on the object at that resolution you would require over 80 million voxels. If we are calculating a likelihood value for each voxel that say is saved in a double (8 bytes), this mounts up to 625 MB of data storage required.

Granted a number of methods have tried to tackle some of these problems for example some spatial data structures like octree can give considerable improvements, but does have similar problems [20].

As a main contribution, we reformulate the visual hull approach using a smooth continuous objective function for which we can define an optimisation algorithm that converges at a much faster rate [9]. This new modelling remains robust and computationally the advantages are that the resulting algorithm is not memory demanding since the whole 3D space do not need to be stored, and it is very suitable for parallel architecture.

The main advantage of our smooth continuous model is that we can define an iterative gradient ascent algorithm. This will converge a point towards the object of interest. The main difference now is that instead of calculating something for every voxel element we are iterating a number of points to sit on the surface of the object. A visual hull with 80 million voxels would now be equivalent to a 100,000 surface points. This results in about a thousand times memory saving. While each surface point has to be iterated about 10 to 20 times to reach the surface (this depends on how good an initial guess we start off with, with better guesses this can be greatly improved) and each iteration is more costly to computer than a single voxel, there is still a significant saving in computation time.

It can be seen from the Middlebury results that the results of our method are comparable to that of the standard visual hull and while not as good as the state of the art in 3D reconstruction this is because our method is currently only using silhouette information rather than full colour information. Shape from silhouette methods can not reconstruct concavities theoretically, but the concave regions can be estimated using stereo information, measuring texture or photo-consistency of surface patches.

The real potential of this framework will be realised when higher dimensions are explored. For example when searching in a 6D space (consisting of colour and space domain) using a histogram or regular sampling approach becomes extremely impractical due to the exponential increase in memory requirements and computation time. While our solution should continue to work with only a minor increase in complexity.

2 Smooth modelling with Kernel density estimates

We consider a recording system composed by C cameras. P^c represents the camera matrix of camera c and n_c is the total number of pixels in the image recorded by camera c (cameras may have different resolutions). We note $\mathbf{u}_i^c = (u_i^c, v_i^c)$ the ith pixel location in the image recorded by camera c, $\mathbf{m}_i^c$ is the vector of colour intensity values (not yet used in this paper) at the position $\mathbf{u}_i^c$. We consider available the indicator variable π_i^c that is 1 if the pixel is in the object and 0 otherwise: this defines the binary silhouette image.

Note that additional features can be extracted from the images and added to the set of observations available. This includes disparity maps or depth information when stereo-matching is possible between pairs of images, and the normals of the silhouette contours [12].

In this paper, we assume that we only know $\{\pi_i^c, \mathbf{u}_i^c, \mathrm{P}^c\}_{i=1,\cdots,n_c \text{and}, c=1,\cdots,C}$. The latent variable of interest corresponding to the 3D location of the object is noted $\mathbf{x} = (x, y, z)$ and from the information available, we aim at modelling the likelihood of $\mathbf{x}$ to be in the object.

2.1 Averaged likelihoods

To ensure robustness of the modelling, we use the averaged likelihood:

$$\overline{lik}(\mathbf{x}) = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{\sum_{i=1}^{n_c}\pi_i^c}\sum_{i=1}^{n_c} p(\mathbf{u}_i^c|\mathrm{P}^c, \mathbf{x})\ \pi_i^c\right) \tag{1}$$

with $p(\mathbf{u}_i^c|\mathrm{P}^c, \mathbf{x})$ is the conditional of one observation $\mathbf{u}_i^c$ given the corresponding camera matrix P^c and the latent 3D position $\mathbf{x}$. $\overline{lik}(\mathbf{x})$ has the form of a kernel density estimate (KDE). In the same spirit as the Hough Transform or visual hull approach, expression (1) corresponds to the average of all the likelihoods defined using one observation $\mathbf{u}_i^c$ at a time.

2.2 Pin-hole camera

The pin-hole camera model is used for modelling the conditional $p(\mathbf{u}_i^c|\mathrm{P}^c, \mathbf{x})$. Using the camera projection matrices $\mathrm{P}^c = [\mathrm{P}_{ij}^c]_{0<i\leq 3, 0<j\leq 4}$ the 3×4 matrix for camera c, the projected image coordinates of $\mathbf{x}$ in the image plane of camera c are $\mathbf{u}^c(\mathbf{x}) = (u^c(\mathbf{x}), v^c(\mathbf{x}))^T$ computed by:

$$\mathbf{u}^c(\mathbf{x}) = \left| \begin{array}{ll} u^c(\mathbf{x}) & = \frac{x\ \mathrm{P}_{11}^c + y\ \mathrm{P}_{12}^c + z\ \mathrm{P}_{13}^c + \mathrm{P}_{14}^c}{x\ \mathrm{P}_{31}^c + y\ \mathrm{P}_{32}^c + z\ \mathrm{P}_{33}^c + \mathrm{P}_{34}^c} \\ \\ v^c(\mathbf{x}) & = \frac{x\ \mathrm{P}_{21}^c + y\ \mathrm{P}_{22}^c + z\ \mathrm{P}_{23}^c + \mathrm{P}_{24}^c}{x\ \mathrm{P}_{31}^c + y\ \mathrm{P}_{32}^c + z\ \mathrm{P}_{33}^c + \mathrm{P}_{34}^c} \end{array} \right. \quad (2)$$

We model the conditional as a Normal distribution:

$$p(\mathbf{u}_i^c|\mathrm{P}^c, \mathbf{x}) = \frac{1}{2\pi h^2} \exp\left[\frac{-\|\mathbf{u}_i^c - \mathbf{u}^c(\mathbf{x})\|^2}{2h^2}\right] \quad (3)$$

The shape of the density $p(\mathbf{u}_i^c|\mathrm{P}^c, \mathbf{x})$ looks like as a fuzzy cone in the 3D space (see Figure 1).

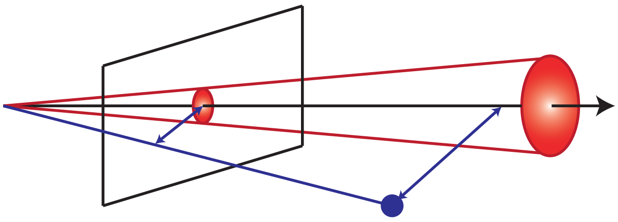


Figure 1: Image plane and pixel ray (black) with cone like distribution (red), the point (blue) $\mathbf{x}$ can be projected back to the image plane using equation (2).

2.3 Remark

Note that our framework considers that binary silhouettes are available:

$$\pi_i^c = \left\{ \begin{array}{ll} 0 & \text{if the pixel } \mathbf{u}_i^c \text{ belongs to the background} \\ 1 & \text{if the pixel } \mathbf{u}_i^c \text{ belongs to the object} \end{array} \right. \quad (4)$$

However our modelling could also accommodate probability maps about the position of the object of interest in each camera view, instead of binary silhouette images. In this case the binary priors defined in equation (4) would be replaced by more fuzzy ones having values between 0 (background) and 1 (object).

By regular sampling we can generate a likelihood map of the object in the scene. Figure 2 shows the kernel density estimate for several slices of a 3D object. High probabilities are in light pixels whereas black pixels indicate low probabilities.

3 Optimization

Gradient methods are well known deterministic numerical approaches for optimisation [16]. Starting with an initial guess, the position in the 3D space is iteratively updated (section 3.1) and the sequence of these positions creates a Markov Chain. In order to speed up convergence and avoid local maxima, the bandwidth h in the kernel can be used as temperature in a simulated annealing like approach [16, 19]. The bandwidth starts large ($h = 10$) which results in a smoother probability density function with less local maxima. As the point approaches the global maximum the bandwidth is decreased ($h = 1$) to achieve the greatest accuracy possible and reflect the uncertainty in the pixel itself.

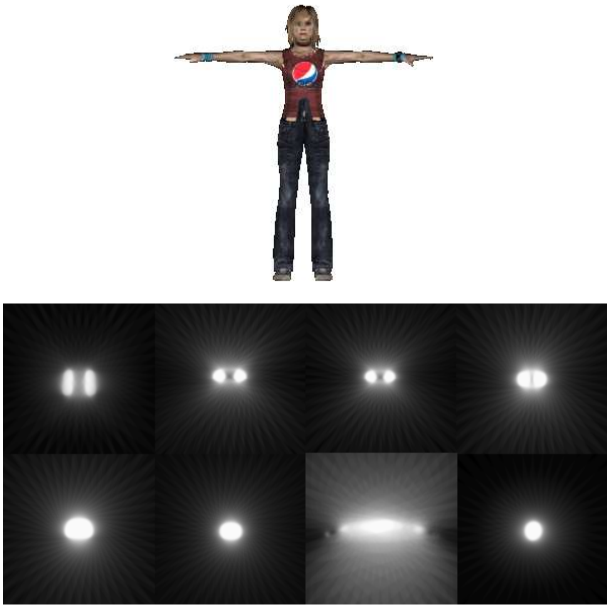


Figure 2: Example of $\overline{lik}(\mathbf{x})$. Top: one view of the original object. Bottom: For visualisation, horizontal slices of $\overline{lik}(\mathbf{x})$ are presented starting at the feet then moving up the legs and body to the shoulders and head (left to right, top to bottom).

3.1 Iterative Gradient Ascent Algorithm

The advantage of generating a continuous kernel density function over a discrete cost function is that now it is possible to implement an iterative gradient ascent algorithm like mean-shift [4]. Due to the properties of a pin hole camera the kernel is non linear which means mean-shift cannot be used. It is however possible to define a Newton-Raphson algorithm that uses the gradient and the hessian of the kernel. Starting from an initial position $\mathbf{x}^{(0)}$, a markov chain is updated from the current position $\mathbf{x}^{(m)}$ to the next $\mathbf{x}^{(m+1)}$ by:

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} - \left[\mathrm{H} \cdot \overline{lik}(\mathbf{x}^{(m)})\right]^{-1} \nabla\overline{lik}(\mathbf{x}^{(m)}) \quad (5)$$

This method of optimisation has been shown to be more efficient (requires fewer steps to converge) than mean shift and makes fewer assumptions on the form of the underlying kernel structure [10]. $\mathrm{H} \cdot \overline{lik}(\mathbf{x})$ and $\nabla\overline{lik}(\mathbf{x})$ are respectively the Hessian matrix and the gradient of the KDE computed at the 3D position $\mathbf{x}$.

Figure 3 represents a slice of the KDE and the magnitude of its gradient. As we are interested in converging towards the edge of the object, the stopping criterion for the Markov chain to converge is a combination of high values for both $\|\nabla\overline{lik}(\mathbf{x}^{(\infty)})\|$ and $\overline{lik}(\mathbf{x}^{(\infty)})$.

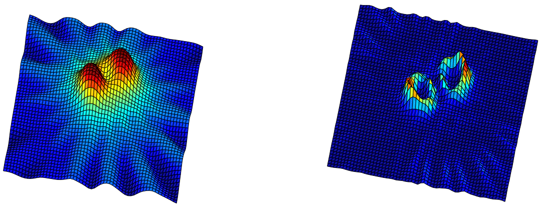

(a) (b)

Figure 3: (a) $\overline{lik}(\mathbf{x})$ in a 2D slice and (b) the magnitude of its gradient $\|\nabla\overline{lik}(\mathbf{x})\|$. This is computed on the probability density function in the top row of figure 2 (second from the left) with 12 cameras, with a bandwidth $h = 4$ and a truncated gaussian kernel.

3.2 Selection of the initial points

J starting positions $\{\mathbf{x}_j^{(0)}\}_{j=1,\cdots,J}$ are randomly selected in the 3D space and iteratively updated until convergence in using our simulated anealing framework to $\{\mathbf{x}_j^{(\infty)}\}_{j=1,\cdots,J}$. Positions $\mathbf{x}_j^{(\infty)}$ with low probability or low gradient can be discarded before their meshing is performed to recover the surface of the object.

To speed-up the process, initial positions $\{\mathbf{x}_j^{(0)}\}_{j=1,\cdots,J}$ can be selected more efficiently using stereo-matching between pairs of recorded images when this is possible. In that case, the starting positions will be closer to the surface of the object allowing a faster convergence. Potential mis-matches (outliers) that would converge towards local maxima, can also be discarded using the value of the probability.

A truncated Gaussian kernel has been used to save computation time and memory allocation. Indeed only the pixels that affect the KDE at the current position $\mathbf{x}^{(m)}$ are used to compute the update $\mathbf{x}^{(m+1)}$. These pixels are found by projecting the current position $\mathbf{x}^{(m)}$ in each camera view using the projection matrices $\{\mathrm{P}^c\}_{c=1,\cdots,C}$ and by selecting neighbours at $\pm 5\ h$.

Finally, this algorithm is very well suited to take advantage of parallel architecture since each Markov Chain can be processed in parallel[3].

4 Experimental results

In section 4.1 we assess the modelling presented in section 2 and section 4.2 illustrates the surface reconstruction using the algorithm presented in section 3.

4.1 Assessment of the modelling with KDE

A bandwidth of $h = 1$ is chosen to match the pixel resolution of the images and a truncated Gaussian kernel has been used. To assess our modelling and visualise the 3D object, the probability is calculated on a regular 3D grid every 1mm, considering all information from every camera. The probability on the 3D grid is thresholded to give solid binary volume of the final object. Alternatively, for surface reconstruction, the boundaries of the horizontal slices can be found using edge detection techniques. These boundaries represent the approximate surface points of the object. A third party meshing algorithm has been used to mesh the points [1].

Three experiments were carried out:

1. Two of the multi-view data sets provided by Middlebury [18] were used; the Temple with 47 camera views and the Dino with 48 camera views. Each of the images were converted to silhouettes using the method described in the data sets supplementary material. The first two images in Figure 4 presents our reconstructions for these two objects.

2. A second data set was synthesised using 3D studio max. Five objects were chosen; a normal head, a normal human body, the Stanford bunny, the Utah teapot and the Stanford Dragon. The objects were scaled to roughly the same size as the Middlebury objects to allow comparison between the results. A camera was set up in 3D studio max to take 360 equally spaced images horizontally around each object. Each object was then reconstructed using 9 camera configurations (4, 5, 6, 8, 9, 12, 15, 36 and 45) equally spaced around the object. The reconstructions can be seen in the last five images of Figure 4. Two metrics were used to assess the quality of the reconstructions. Both methods come from [18]; an accuracy measurement and a completeness measurement. The accuracy measurement d is computed such that 90% of the points on the reconstruction are within that distance d of the ground truth. The completeness measurement gives the percentage of ground truth points that are less than 1.25mm away from the reconstruction. The results of the accuracy and the completeness measurements can be seen in Figure 5a and 5b. Overall the results show that the accuracy nears 1mm for the head, human body and the teapot which is the same as the resolution of the sampling. The bunny and the dragon approach 3 to 4mm which is good for such complex objects. The completeness does not look as good though due to a poor meshing algorithm used. For the accuracy figures there is a performance boost for odd numbers of cameras (5 and 9). When an even number of cameras is used, each camera has a perfectly opposite camera which provides almost identical silhouette (i.e. same information). Therefore half the cameras are almost redundant, this is not the case with an odd number of cameras.

3. The third experiment was to test the robustness of the method. The human body silhouettes were altered with different levels of salt and pepper noise (5, 10, 15 and

[1] http://www.mathworks.com/matlabcentral/fileexchange/22185-surface-reconstruction-from-scattered-points-cloud-part1

20 percent) to simulate noisy binary silhouettes. The objects were then reconstructed using the same 9 camera configurations as before. The results of the accuracy and completeness measurements can be seen in Figure 5c and 5d. This shows that the method is very robust. Even when 20% salt and pepper noise is added to the original data set it still produces a very accurate reconstruction.

4.2 Assessment of the algorithm

Two main experiments were carried out.

1. Reconstructions were carried out on real images from the Human Eva 2 [2] data set with 4 cameras and an in house dataset [17] with 6 cameras, the reconstructions can be seen and recognised in figure 6 and figure 7.
2. A number of reconstructions were also done on the Middlebury dataset [18] and the reconstruction using our algorithm can be seen in figure 8. Our results are not yet good enough to compete with the leading methods on the Middlebury dataset. However, our approach uses very little information from the silhouettes images: the intensity and colour values of the images are not yet used in our formalism. This means for instance that no concave regions can be recovered. The advantages of this method in terms of memory requirements and computation time can be seen Table 1. Results from the Middlebury website can be seen in Table 2.

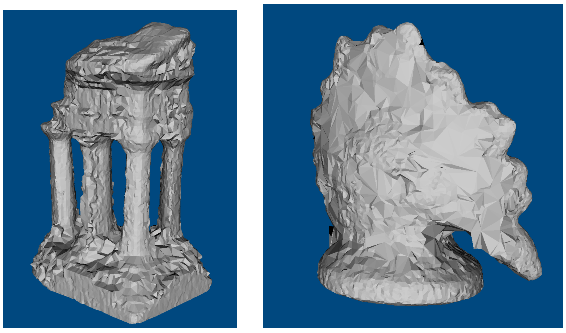


Figure 8: Reconstructions using the Middlebury dataset [18] using our algorithm.

5 Conclusion and Future works

We have introduced a novel approach for modelling and estimating the visual hull using 2D silhouette images recorded from several pin-hole camera views. The resulting iterative gradient ascent algorithm has a better convergence rate [9] and is less memory demanding than histogram-based modelling. It is also very well suited to take advantage of parallel architecture and therefore has an interesting potential for being fast. Experimental results shows that this approach works well considering the limited information used in the modelling (i.e. only silhouette information).

	Visual Hull	Just KDE	IGA
No. of Points	5,292,621	5,292,621	5,000
Memory Requirement	41MB	41MB	120KB
Computation time	10 mins	2.5 hrs	5mins

Table 1: Comparison of standard visual hull method with our KDE using regular sampling and using our iterative gradient ascent algorithm (IGA). Numbers based on reconstruction of the Middlebury [18] dino object. Computation time based on Matlab code implementation that has not yet been optimised.

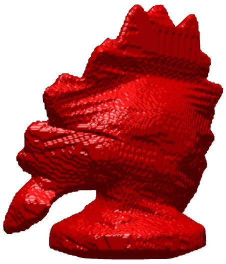
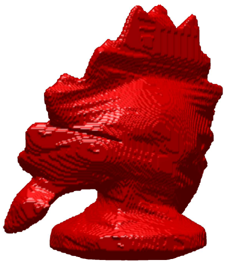
	Visual Hull	Our KDE
		
Accuracy	2.41mm	2.25mm
Completeness	77.0%	75.5%

Table 2: Comparison of standard visual hull method with our KDE of the Dino, the results are calculated by the Middlebury Website [18].

One straight forward extension of our framework is to add colour information to be taken into account by modelling the averaged likelihood $\overline{lik}(\mathbf{x},\mathbf{m})$ where $\mathbf{m}$ represents the colour information encoded as either as a 3 dimensional feature (i.e. RGB values) or 2D one (if keeping only chrominance information for insuring colour consistency between views). The conditional can be modelled for instance by:

$$p(\mathbf{u}_i^c,\mathbf{m}_i^c|\mathrm{P}^c,\mathbf{x},\mathbf{m}) \propto \exp\left[\frac{-\|\mathbf{u}_i^c-\mathbf{u}^c(\mathbf{x})\|^2}{2h^2}\right]\times\exp\left[\frac{-\|\mathbf{m}_i^c-\mathbf{m}^c\|^2}{2h_{\mathbf{m}}^2}\right] \quad (6)$$

Optimisation of $\overline{lik}(\mathbf{x},\mathbf{m})$ by gradient ascent method is again applicable in the latent space of $(\mathbf{x},\mathbf{m})$ and will lead to an estimate of the photohull. This extension should remove the need of segmenting the object from the background to get the silhouette images. Finally, this framework will also be extended by using a prior, leading to the optimisation of a posterior density function.

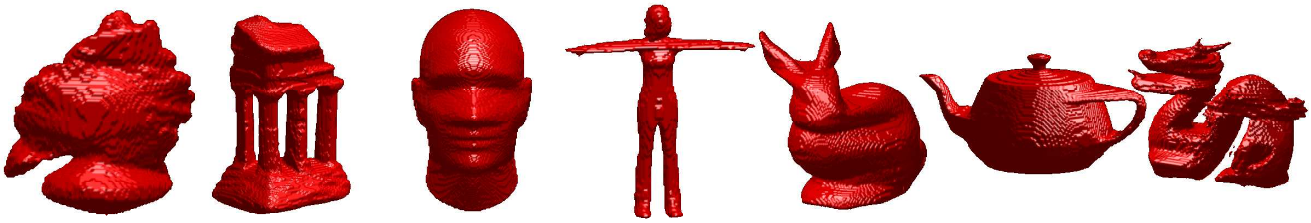


Figure 4: 3D reconstructions (using 48 cameras (dino), 47 cameras (temple), both from the Middlebury dataset [18], and 45 cameras for the other objects). More results can be seen at https://www.cs.tcd.ie/˜ ruttlej/

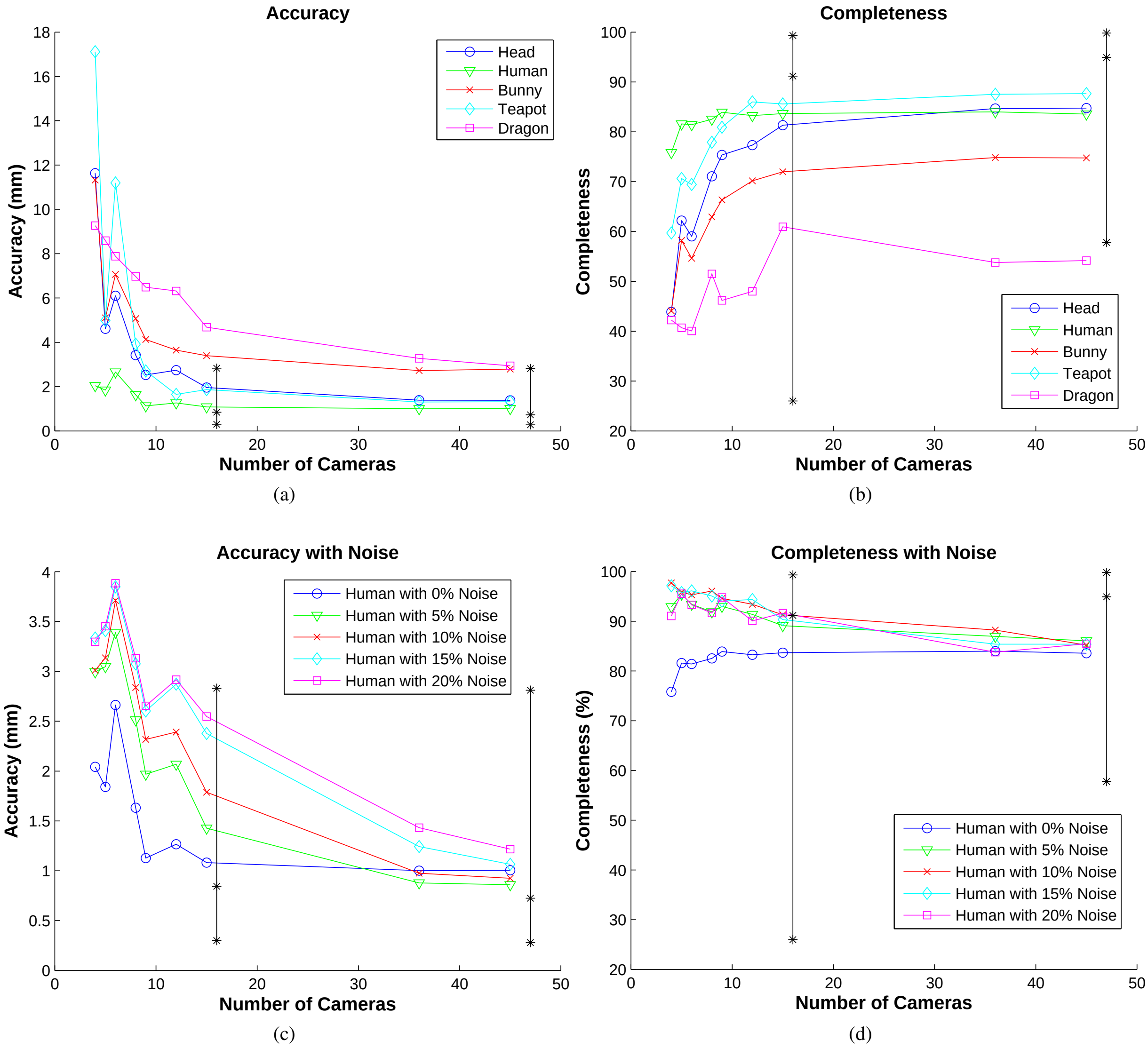


Figure 5: Results of Reconstructions from clean images (top) and from noisy images (bottom). The black vertical lines show the max, mean and min values obtained from all the Middlebury results [18].

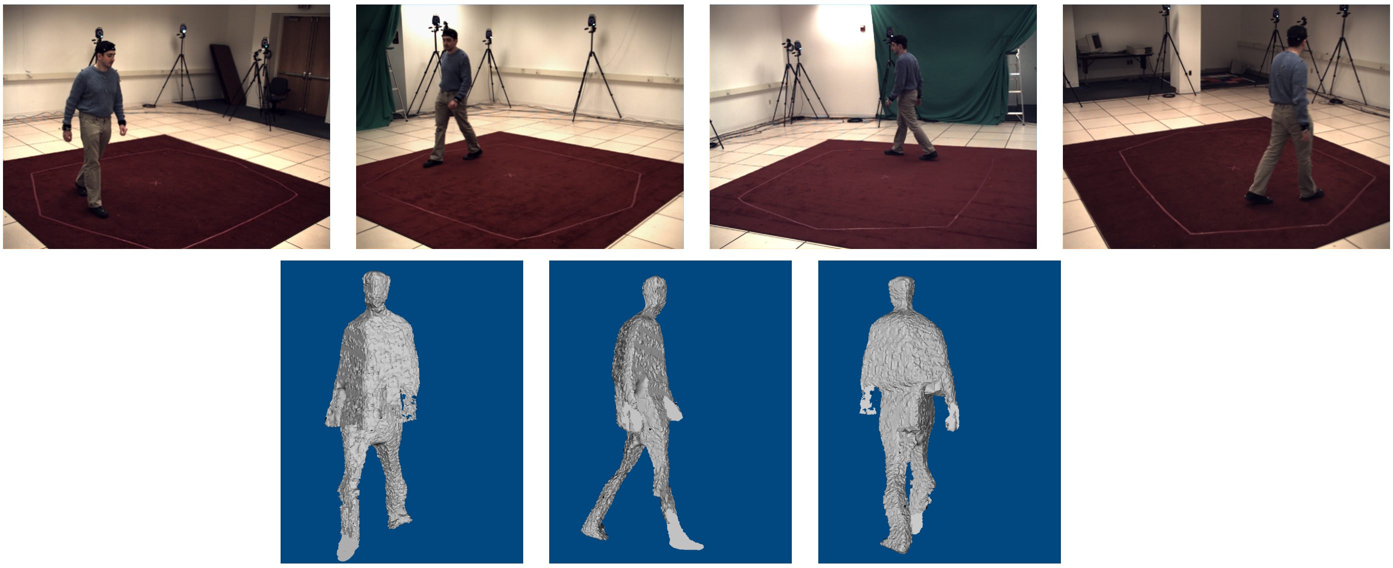


Figure 6: 3D surface reconstruction using 4 cameras (from Human Eva2 data) [2].

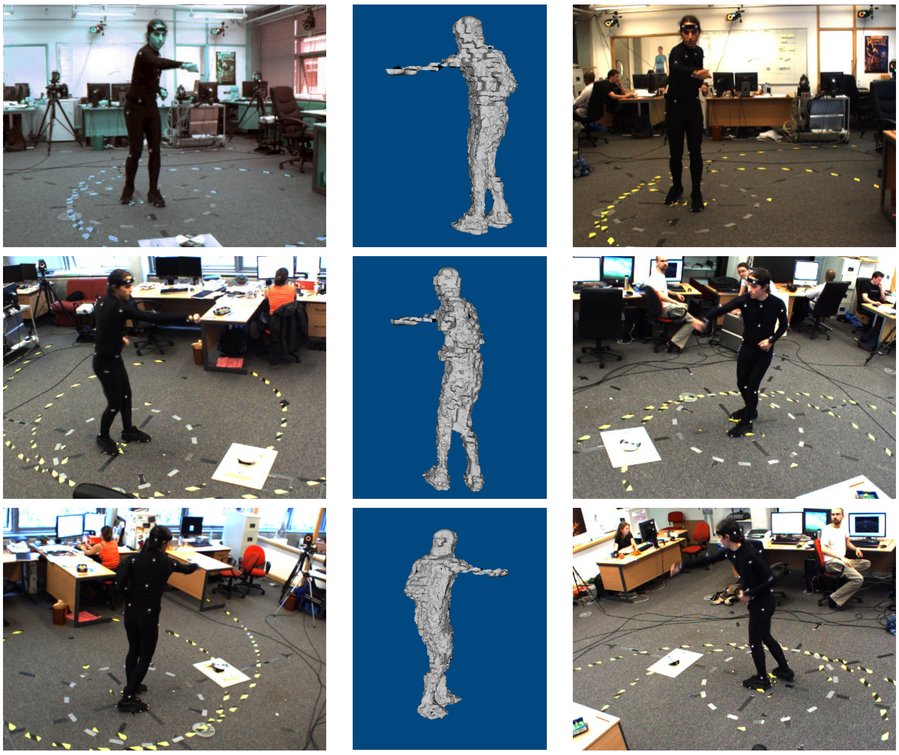


Figure 7: Images from in house dataset with 3D surface reconstructions using 6 cameras [17].

Acknowledgements

This work has been supported by an Innovation Partnership between Sony-Toshiba-IBM and Enterprise Ireland (IP-2007-505) and forms part of Trinity's Sony-Toshiba-IBM European Cell/B.E. Centre of Competence.

References

[1] Alberto S. Aguado, Eugenia Montiel, and Mark S. Nixon. On the intimate relationship between the principle of duality and the hough transform. *Proceedings: Mathematical, Physical and Engineering Sciences*, 456(1995):503–526, 2000.

[2] M. J. Black and L. Sigal. HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion, technical report CS–06–08. http://vision.cs.brown.edu/humaneva/index.html, 2006.

[3] A. E. Brockwell. Parallel markov chain monte carlo simulation by pre-fetching. *Journal of Computational and Graphical Statistics*, 15, 2006.

[4] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[5] S. R. Deans. Hough transform from the radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2), March 1981.

[6] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15, January 1972.

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

[8] J.S. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):414–427, March, 2009.

[9] A. Goldenshluger and A. Zeevi. The hough transform estimator. *The Annals of Statistics*, 32(5), October 2004.

[10] G.D. Hager, M. Dewan, and C.V. Stewart. Multiple kernel tracking with SSD. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:790–797, 2004.

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

[12] S. Liu, K. Kang, J.-P. Tarel, and D. B. Cooper. Free-form object reconstruction from silhouettes, occluding edges, and texture edges: A unified and robust operator based on duality. *IEEE transactions on Pattern Analysis and Machince Intelligence*, 30:131–146, January 2008.

[13] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:150–158, January 1983.

[14] S. Paris, F. X. Sillion, and L. Quan. A surface reconstruction method using global graph cut optimization. *International Journal on Computer Vision*, 66(2):141–161, February 2006.

[15] C. Pintavirooj and M. Sangworasil. 3d shape reconstruction based on radon transform with application in volume measurement. *10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2002.

[16] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, 1999.

[17] Jonathan Ruttle, Michael Manzke, Martin Prazak, and Rozenn Dahyot. Synchronized real-time multi-sensor motion capture system. In *SIGGRAPH Asia, sketch Poster*, Yokohama, Japan, 16 - 19 December 2009.

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, June 2006.

[19] C. Shen, M. Brooks, and A. van den Hengel. Fast global kernel density mode seeking: Applications to localization and tracking. *IEEE Transactions on Image Processing*, 16, May 2007.

[20] R. Szeliski. Rapid octree construction from image sequences. *CVGIP:Image Understanding*, 58(1), 1993.

[21] K. Zimmermann, T. Svoboda, and J. Matas. Multiview 3d tracking with an incrementally constructed 3d model. In *Third international Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2006.