Real-time Silhouette Intersection by Maintaining the Distribution of Occupancy

Odest Chadwicke Jenkins, Matthew Maverick Loper

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-05-12
July 2005

Real-time Silhouette Intersection by Maintaining the Distribution of Occupancy

Odest Chadwicke Jenkins Matthew Maverick Loper
Department of Computer Science
Brown University
Providence, RI, USA 02912-1910

Abstract

We present an approach to space carving based on adaptively sampling a propagated estimate of density in the shape of a dynamic environment. Our approach is designed to enable faster computational performance of shape carving, without requiring specialized hardware or code optimization, while maintaining a representative estimate of shape. We present results from comparing Matlab implementations of space carving using voxel grids and adaptive sampling for speed and shape distribution.

1. Introduction

Several techniques have been proposed in computer vision for estimating 3D shape from multiple camera viewpoints. Two popular approaches to volume estimation from image silhouettes are *visual hull construction* [Lau94] and *space (or voxel) carving* [Sze93]. Visual hulls are constructed as a polygonal surface formed from the intersection of viewing cones, silhouetts extruded from the image plane, from multiple calibrated cameras. Space carving culls a set of active 3D points comprising a shape from a set of candidate points (or samples).

As explained by Boyer and Franco [BF03], these approaches are subject to a trade-off of the precision and instability of visual hulls with the robustness and computational cost of space carving. The computational issues for visual hulls and space carving are partially due to artifacts in the silhouettes and the number of candidate points that must be tested, respectively. Boyer and Franco address the trade-off for surfaces via a hybrid approach: a visual hull is used as an underlying distribution for producing samples towards a space carving procedure. This approach could be thought of as using a visual hull as an underlying prior distribution with non-uniform sampling. Space carving then serves as a likelihood function for evaluating a non-uniform and non-regularly spaced set of samples from the prior. The result is a posterior distribution of points on a visual hull surface.

For volumetric space carving, the hybrid approach to shape estimation begs the question if there is a more direct means of representing a prior distribution for samples. In the most naive case, a samples are regularly distributed as voxels in a rectilinear grid. For the experimental setup described by Cheung et al.[CKBH00], this approach would result in the evaluation of $64 \times 64 \times 64 = 262144$ voxels in each frame of motion. Because the cost of evaluating a point sample or a cube voxel can be expensive, our aim is to lower the number of evaluations while retaining the shape distribution. The use of octrees [Sze93] reduces the number of evaluations by recursively evaulating voxels based on occupancy at courser levels of resolution. In the worst case, however, octrees result in the same order of voxel evaluations due to structure of voxels as uniformly distributed axis-aligned samples. Instead, we consider the case where density estimation of the sample distribution and evaluating samples from that distribution are addressed separatly. By addressing these issues separately, we can adaptively select the number of samples to estimate based on user parameters, such as desired frame rate, while maintaining a representative estimate of shape. With this approach, we can fix the number of samples to evaluate and would not incur the 64^3 problem in worst case of the Cheung et al. example.

In this paper, we present an approach to space carving based on adaptively sampling a propagated estimate of density in the shape of a dynamic environment. Our approach is based on the Condensation algorithm [IB98] and primarily intended to enable faster computational performance of shape carving, while maintaining a representative estimate of shape. Our approach is not specific to particular hardware (e.g., [HLS04], platforms or code optimization). We present results from comparing Matlab implementations of space carving using voxel grids and adaptive sampling for speed and shape distribution. The presentation of our method is intended as a preliminary step towards the development of real-time shape estimation for interaction systems (human-computer or human-robot) rather than modeling of shape.

1.1. Previous Work

We propose an approach to accelerate volumetric shape estimation through sampling with respect to a density estimate. This approach exists within the broader line of research known as *shape-from-silhouette* [MA83]. Much previous work exists in estimating shape-from-silhouette. We refer the reader to the description provided by Savarese et al. [SRBP01] for a general overview of shape-from-silhouette approaches. We roughly categorize these approaches into visual hull construction [Lau94], marching cubes [LC87], and space carving [Sze93, KS99]. Visual hulls and marching cubes are geared more towards surface estimation, whereas our interest in volumetric estimation. We focus primarily on volumetric approaches in our summary of previous related work.

One of our main contributions is the use of an active point density estimate as a distribution for sampling candidate points. This approach is similar to the hybrid method presented by Boyer and Franco [BF03]. The hybrid approach assumes the visual hull underlying silhouette contours is the appropriate distribution for candidate points. Space carving is applied candidate points (expound) to determine points on the active surface. In contrast, our approach is suited for volumetric shape rather surfaces.

Methods for probabilistic [BFK02, BDC01] and multiple hypothesis [ESG99] space carving address how to determine whether a voxel is occupied and its color. These methods estimate probabilities specific to each voxel in a rectilinear grid. Instead, our approach aims to estimate the density of volume occupancy with limitation to an axis-aligned voxel grid. We can think of this density as the probability of single point being active with respect to all other visible points. Furthermore, our objective is to reduce the number of candidate points necessary to evaluate to produce a representative volume. Bhotika et al. [BFK02] briefly touch on the issue of "fair sampling", but for computational tractability of a photo hull distribution rather than interactive speed (reword). Our approach to sampling candidates does not address space carving explicity and could be used in conjunction with probabilistic space carving.

Kinematic motion capture methods, such as [CKBH00, CBK03, CJM03, MTHC01], use space carving towards pose estimation. The methods use separate procedures for, first, space carving and then pose estimation. In our proposed approach, we would not consider space carving and pose estimation to be independent procedures. Instead, models fit during pose estimation at time t would be used towards a candidate sampling distribution at time t. Work presented by Cheung et al. [CKBH00] is overwheliming similar to our approach with respect to fitting a mixture of Gaussians/ellipsoids in each frame and utilizing the fit from one frame to initialize the fit in the next frame. With our approach, however, we can provide greater precision than 1cm^3 voxels and the ability to trade-off speed and precision. Additionally, our approach is not restricted to human topology.

Given fast space carving, methods multiple hypothesis motion capture, such as [DBR00], will have another mode to evaluate the likelihood of candidate poses.

Other space carving methods have associated information other than color and occupancy with each voxel. Vedula et al. [VBSK00] associates object motion flow with each voxel to provide both occupancy and motion information for a scene. While this approach leads toward more accurate carving, it does not aid in making the carving procedure faster. Pyo and Shin [PS02] associate a density variable with each voxel to carve for volumes with different material properties. This approach, however, requrires a means for estimating material density for object in a scene and is better suited for medical applications.

Several approaches for hardware-accelerated octree-based volume carving have been proposed [HLS04, SBS02, ZKRB04]. These approaches leverage specific hardware, such as a graphics processor, for faster performance. Our method is not specific to any hardware. The Matlab implmentation of our approach carves representative volumes at interactive rates, indicating the potential for greater speed enhancement.

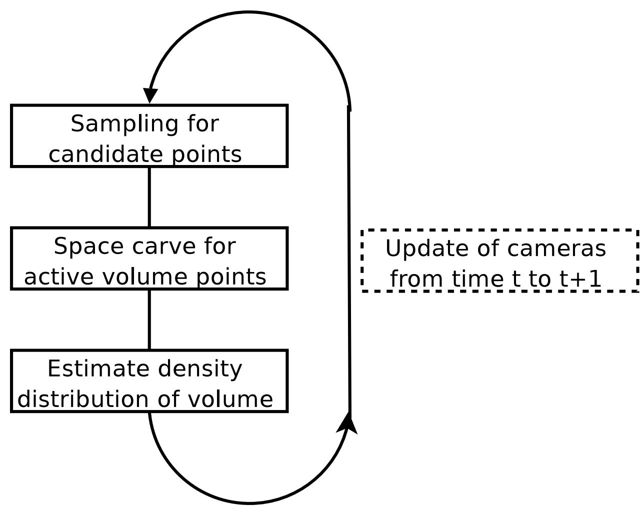

Figure 1: Overview of our algorithm. For each frame, we sample carving candidate from the prior distribution. The samples are given to a space carving procedure to produce the volume at time t. Density estimation is performed to produce a posterior distribution, which is used as the prior at time $t + 1$.

2. Space Carving with Propagating Density

Our aim is to compute a real-time volumetric representation for a dynamic scene. The real-time constraint is essential: we must construct a volume with a representative distribution to a scene given the amount of CPU time allotted. Several factors may reduce the allotted CPU time, including platform speed, desired frame rate, and the need for other per-frame processing.

Fortunately, a dynamic scene exhibits temporal coherence: by knowing the state of the volume at time t, we can make some guesses about its state at time $t+1$. More formally, given a measured volume at time t, we can estimate the density of active points over the space of our volume. This density estimate at t serves a prior distribution at time $t+1$.

We use the prior distribution at time $t+1$ from density estimation at time t to more aggressively sample for candidate points in active areas of the scene. Assuming temporal coherence, active areas in the scene will accord to regions of high density in the prior distribution. This property allows us to concentrate sampling for candidate points in regions of greater density in the distribution. The benefit of this approach is that fewer points are required for shape carving to produce a volume that is a representative estimate of shape. Furthermore, the number of samples can be adaptively modified to adhere to a user constraint, such as desired frame rate.

We express our approach as a shortened version of the formalism presented by Isard and Blake [IB98]:

$$P(x_t|Z_t) = k_t P(z_t|x_t) \int_{x_{t-1}} P(x_{t-1}|Z_{t-1}) \qquad (1)$$

where x_t is the state of the scene at time t, z_t are the silhouette image features at time t, $Z_t = \{z_1, ..., z_t\}$ is the history of image features, and k_t is a scaling constant. This equation expresses the posterior density $P(x_t|Z_t)$ of active samples at time t with respect to the posterior density $P(x_{t-1}|Z_{t-1})$ at the previous instant of time $t-1$ as evaluated by the space carving likelihood $P(z_t|x_t)$. $P(x_{t-1}|Z_{t-1})$ serves as both the prior distribution at time t and the posterior at time $t-1$. Using the factored sampling, described by Isard and Blake, we select a set of candidate samples for voxel carving from $P(x_{t-1}|Z_{t-1})$ and apply space carving to produce both the active volume and posterior density at time t. We assume the presence of any space carving mechanism to evaluate $P(z_t|x_t)$. Space carving is currently assumed to provide a hard 0,1 decision, but a soft decision from probabilistic space carving could also be incorporated. Deviating from Isard and Blake, we do not provide an explict density motion model $P(x_t|x_{t-1})$ because the scene is assume to exhibit temporal coherence between frames.

2.1. Our Algorithm

Our algorithm can first be described in terms of its inputs and outputs.

Inputs: Camera calibration data is required upon initialization. During each frame, silhouettes from our calibrated cameras are required, as is a prior distribution estimate of sample density.

Outputs: A set of points inside the volume, a set of points outside the volume, and a posterior distribution that characterizes the shape of the carved volume.

For the purposes of our implementation, we chose to represent sample distributions as a mixture of Gaussians [Bis95], although any distribution representation could be used.

During the first frame, we assume a uniform likelihood function across the space viewable by all cameras for sampling candiate points. Active points in the initial frame can be found through classic voxel or octree space carving. The result will be a set of points that fall entirely within the visual hull.

In the next step, density estimation for the set of active points is performed. Several procedures could be used for density estimation, such as kernel density estimation [Bis95]. For simplicity, we chose to use k-means clustering to find the modes of the distribution. Singular value decomposition [PTVF92] is applied to the points of each cluster to find the size and orientation of each Gaussian mode. Our approach for active point density estimation is practically the same as work by Cheung et al. [CKBH00], except their approach is specific to distributions with the topology of human subject whereas we make no assumptions about the expected active point distribution.

Considering that we fit a Gaussian to each cluster, we can formulate a likelihood function as a sum of k cluster Gaussians. The probability of a point x being solid would be defined by formula (2). Note that (3) is the standard Mahalanobis distance formula.

$$P(x_{t-1}|Z_{t-1}) \propto \frac{1}{k} \sum_{i=1}^{k} \frac{-\exp\left[d(C_{i,t-1}, x_{t-1})\right]}{(2\pi)^{N/2} \left|S_{i,t-1}\right|^{1/2}} \qquad (2)$$

$$d(C, x) = (x - \mu_C)^T S^{-1} (X - \mu_C) \qquad (3)$$

where $C_{i,t-1}$ and $S_{i,t-1}$ are the centers and sizes of Gaussian i at time $t-i$ and $d(C, x)$ is the Mahalanobis distance between Gaussian C and x.

Our current expression of $P(x_{t-1}|Z_{t-1})$ is a good place to start, but some refinement is required for correct behavior. Three issues are addressed below for the distribution model we actually use, L''. Refer to Figure 2.1 for a visual depiction of these corrections.

First, if a new object enters the scene, the Gaussians (which may be “attached” to other objects) will not sample it well. We’d like to supplement our model by saying that an object might appear anywhere, albeit with low probability. This is easy to implement, as it only requires putting a lower bound on our likelihood function:

$$L'(solid|X) \propto u + (1-u)L(solid|X), \quad 0 < u < 1 \tag{4}$$

Another issue to address concerns the size of our Gaussians. Note that we are fitting a normal Gaussian to a cluster that may be close to uniformly distributed. When a normal Gaussian is fit to a distribution with low kurtosis, the resulting Gaussian represents the extremes of that distribution poorly. Therefore, the probability function should be a sum of enlarged cluster Gaussians, meaning that their standard deviations in S_c should be multiplied by a constant $b > 1$. A larger Gaussian also accounts for potential movement of objects in the scene.

Finally, recall that we wish to sample active areas of the scene evenly. This means that if one cluster's ellipsoid is twice the volume of another cluster's ellipsoid, the Gaussian of the first ellipsoid should account for twice the probability of the second.

With those refinements in place, our probability function can be described as follows:

$$L''(solid|X) \propto u + \frac{1-u}{k\sum a(c)} \sum_{c=1}^{k} \frac{\exp\left[d(c,X)\right] a(c)}{(2\pi)^{N/2}\left|\hat{S}_c\right|^{1/2}} \tag{5}$$

$$d(c,X) = (X-\mu_c)^T \hat{S}_c^{-1}(X-\mu_c) \tag{6}$$

$$\hat{S} = Sb, \quad 1 < b < \infty$$

In practice, there is no need to explicitly evaluate Equation (5). We only need to allocate points in the scene proportionally to (5), in preparation for carving.

As mentioned, speed is essential to our goals. Our frame rate is adjusted continually during run-time, based on the average frame duration of six previous frames. This adjustment controls the number of points allocated for sampling.

In summary, our algorithm achieves our goals in three basic steps: sampling candidates from previous density estimate (prior), carving active points from candidates (measuring likelihood), and active point density estimation (posterior).

3. Results

To evaluate the speed and fidelity of our algorithm, we compared it against a standard voxel-based sampling approach. Both techniques were implemented with MATLAB

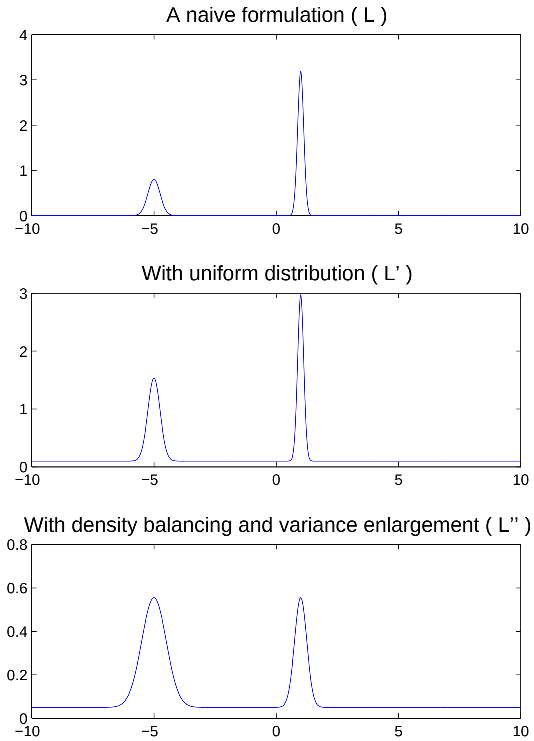

Figure 2: A one dimensional analogue to the improvement of probability function L.

on a 2.8Ghz Pentium IV. The voxel-based sampling used the same dimensions (64^3 voxels) as Cheung et al. [CKBH00].

Space carving using the two sampling approaches was applied to a calibrated multi-view sequence of human walking. The walking sequence consisted of 200 frames of a human walking on a circular path. The subject was viewed from four calibrated camera. The viewable volume from these cameras is illustrated in Figure 3. Silhouettes in each camera were produced through background subtraction.

We initialized the algorithm to target 10 frames-per-second, using 6 clusters, and using 10% of its sample point allocation for the uniform part of the distribution.

The computational savings of our approach are illustrated in Table 1. Adaptive sampling is clearly faster than voxel-based sampling in this setting. Note that our average fps is lower than our target because we sample heavily in the first few frames.

Next, we address the fidelity of our sampling approach. We consider our volume to be accurate if it closely approximates our voxel-based reconstruction. More specifically, we consider that the distribution of carved points should be very similar between the two techniques. The distri-

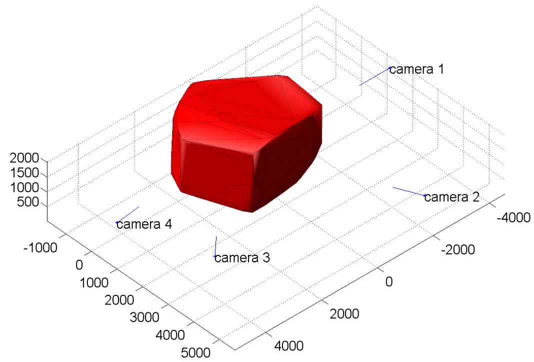

Figure 3: Camera Hull

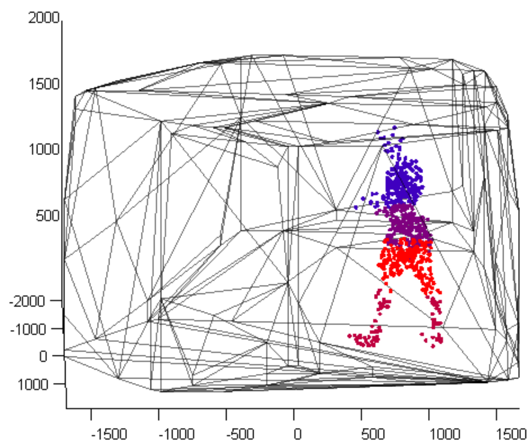

Figure 4: Carved Distribution. Colors indicate cluster membership.

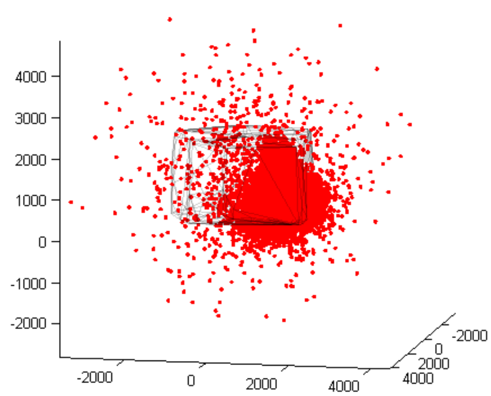

Figure 5: Uncarved Distribution. In practice, points outside the camera hull are eliminated prior to silhouette carving.

200 frames	**Voxel Grid**	**Adaptive** (in desired fps) **2 fps**	**10 fps**	**30 fps**
Total Time	83.5s	107.7s	23.4s	8.9s
Avg # Pts Sampled	262144	26782	8685	2226
Avg # Pts Active	1183.0	4419.2	1434.4	362.9
Actual Avg FPS	2.40	1.86	8.53	22.6

Table 1: Performance Comparison.

butions were compared first with the Kullback-Leibler distance metric, and then by measuring statistical moments.

The Kullback-Leibler divergence was measured with the Kernel Density Estimation toolbox [IM]. A Gaussian kernel was chosen, and the kernel size was configured to be automatically chosen via the "rule of thumb" technique. As can be seen in Figure 6, the distributions are found to be very similar.

The spike that occurs around frame twenty is difficult to miss; the reason for this divergence is that the voxel-based carving technique found a transient shadow on the ground, while the adaptive sampling technique carved it away. This might suggest that adaptive sampling is more impervious to transient silhouette artifacts, given its dependence on temporal coherence; but it might equally well be argued that adaptive sampling is slower to account for a quickly changing environment.

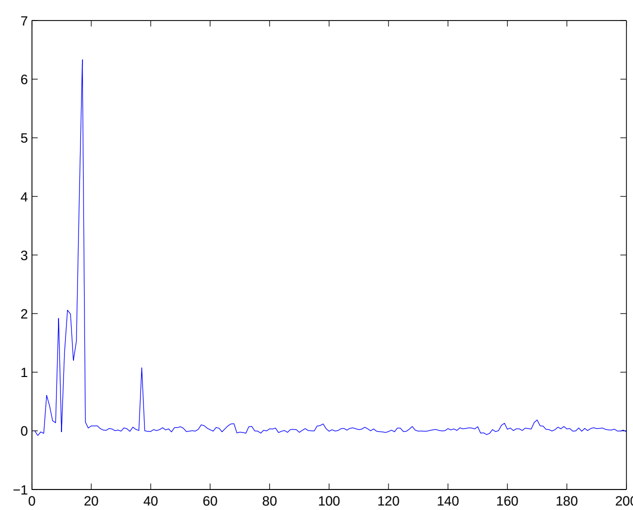

Figure 6: Kullback-Leibler divergence

We also compared statistical moments (up to order 3) between the two carved distributions at each time step. Consider that all the moments of a shape can be encapsulated as a single multidimensional point. We can measure the Euclidean distance between the moments for our voxel representation, and for our adaptive representation, and chart

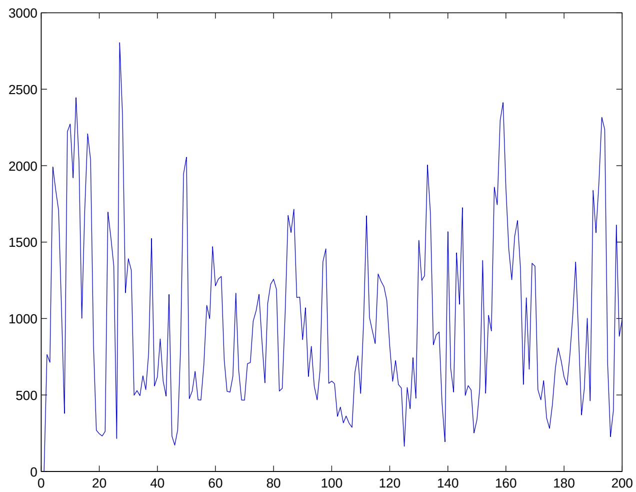

Figure 7: Euclidean distance between moments order 0-3.

them, as we have in figure 7.

4 Discussion

In our estimation, this algorithm has a nice combination of attributes: it is fast, frame-rate adjustable, relatively simple, and utilizes an accessible ellipsoidal representation for the distribution of the scene. However, application of this algorithm is not without pitfalls, and there is certainly room for improvement. In this section, we will discuss computational speed, distribution uniformity, motion models, and occupancy querying.

Computation time is a primary concern for this algorithm. As seen in Table 1, we are able to sample important parts of the viewable volume, both densely and quickly. At around eight fps, we find more solid points than a standard voxel carving approach, at over three times the frame rate.

If precision is a higher priority than computation time, our algorithm should also be considered. Because we sample active areas of the scene more heavily, more precision is granted with the same number of samples. The precision of our algorithm is especially noticeable if active areas of the scene are small in proportion to the entire scene, and if our assumption of temporal continuity remains valid.

One potential drawback of our algorithm concerns uniformity. A voxel-based space carving approach provides an almost ideal point distribution, whereas ours dictates a heavier sampling towards the middle of an actively sampled object. In the future, we will experiment with fitting a transformed Gaussian to a cluster, using calculated statistical moments of the cluster to dictate the sample distribution.

Another important direction for future work involves developing a motion model for our clusters. In the most simple formulation, we could measure the velocity vector of each centroid at each timestep, and update that centroid of the corresponding cluster Gaussian accordingly. This would improve our prior, in addition to making k-means faster (given that it would have a better starting condition).

One aspect to compare concerns querying a carved distribution for state. Given our algorithm or a voxel carving approach, one might ask the question, "is the point at (x, y, z) solid?" With voxels, it is trivial to look up this information: if the voxels which surround that point are solid, the answer is yes. Our algorithm, however, outputs an unordered set of points and a set of ellipsoids; finding the best answer to this question would probably mean computing a kernel density estimate, which (considering that our points are unordered) would require more sophistication. Using a better density estimator would certainly be another interesting area to explore.

5. Summary and Conclusions

In this paper, we have presented a space carving approach which provides both computational savings and a compact scene representation. Adaptive sampling was used to sample active areas of a scene more aggressively, given our sum of Gaussians likelihood function which was updated every frame. Finally, our technique was compared with a standard voxel-based space carving approach, on the basis of both speed and fidelity.

References

[BDC01] A. Broadhurst, T.W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *IEEE International Conference on Computer Vision*, pages I: 388–393, 2001.

[BF03] E. Boyer and J.S. Franco. A hybrid approach for computing visual hulls of complex objects. In *IEEE Computer Vision and Pattern Recoginition*, pages I: 695–701, 2003.

[BFK02] R. Bhotika, D.J. Fleet, and K.N. Kutulakos. A probabilistic theory of occupancy and emptiness. In *European Conference on Computer Vision*, page III: 112 ff., 2002.

[Bis95] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, England, 1995.

[CBK03] G.K.M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR03)*, pages I: 77–84, Madison, Wisconsin, USA, 2003.

[CJM03] Chi-Wei Chu, Odest Chadwicke Jenkins, and Maja J Matarić. Markerless kinematic model and motion capture from volume sequences. In *IEEE Computer*

Vision and Pattern Recognition (CVPR 2003), pages II: 475–482, Madison, Wisconsin, USA, June 2003.

[CKBH00] Kong Man Cheung, Takeo Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, pages 714 – 720, Hilton Head, SC, USA, June 2000.

[DBR00] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, Hilton Head, SC, USA, 2000.

[ESG99] P. Eisert, E. Steinbach, and B. Girod. Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views. In *International Conference on Acoustics Speech and Signal Processing*, pages 3509–3512, Mar 1999.

[HLS04] Jean-Marc Hasenfratz, Marc Lapierre, and François Sillion. A real-time system for full body interaction. *Virtual Environments*, pages 147–156, 2004.

[IB98] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[IM] A. Ihler and M. Mandel. The matlab kernel density estimation toolbox.

[KS99] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *IEEE International Conference on Computer Vision*, pages 307–314, 1999.

[Lau94] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2), Feb 1994.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM Press, 1987.

[MA83] W.N. Martin and J.K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, Mar 1983.

[MTHC01] Ivana Mikić, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 455–460, Kauai, HI, USA, December 2001.

[PS02] Soon Hyoung Pyo and Yeong Gil Shin. Fast volume carving. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 105–112, 2002.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[SBS02] M. Sainz, N. Bagherzadeh, and A. Susin. Hardware accelerated voxel carving. In *Ibero-American Symposium in Computer Graphics (SIACG 2002)*, pages 289–297, Guimaraes, Portugal, Jul 2002.

[SRBP01] S. Savarese, H. Rushmeier, F. Bernardini, and P. Perona. Shadow carving. In *IEEE International Conference on Computer Vision*, pages I: 190–197, 2001.

[Sze93] Richard Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(1):23–32, July 1993.

[VBSK00] S. Vedula, S. Baker, S.M. Seitz, and T. Kanade. Shape and motion carving in 6d. In *IEEE Computer Vision and Pattern Recoginition*, pages II: 592–598, 2000.

[ZKRB04] C. Zach, K. Karner, B. Reitinger, and H. Bischof. Space carving on 3d graphics hardware. Technical Report VRVis Technical Report 2004-014, Institute for Computer Graphics and Vision, TU-Graz/Austria, 2004.