Dynamic Probabilistic Volumetric Models

Ali Osman Ulusoy Octavian Biris Joseph L. Mundy
School of Engineering, Brown University
{ali_ulusoy,octavian_biris,joseph_mundy}@brown.edu

Abstract

This paper presents a probabilistic volumetric framework for image based modeling of general dynamic 3-d scenes. The framework is targeted towards high quality modeling of complex scenes evolving over thousands of frames. Extensive storage and computational resources are required in processing large scale space-time (4-d) data. Existing methods typically store separate 3-d models at each time step and do not address such limitations. A novel 4-d representation is proposed that adaptively subdivides in space and time to explain the appearance of 3-d dynamic surfaces. This representation is shown to achieve compression of 4-d data and provide efficient spatio-temporal processing. The advances of the proposed framework is demonstrated on standard datasets using free-viewpoint video and 3-d tracking applications.

1. Introduction

Three dimensional (3-d) dynamic scene modeling from imagery is a central problem in computer vision with a wide range of applications, including 3-d video, feature film production, mapping, surveillance and autonomous navigation. An important aspect of 3-d dynamic scene modeling is developing efficient representations that extend current 3-d models to include temporal information. A common approach is to store a 3-d model at each time step [32, 17, 24]. However, this approach does not yield an integrated space-time (4-d) data structure and does not scale well in dealing with thousands of frames of data. In particular, such representations do not exploit the fact that many 3-d objects, such as buildings, roads and trees, persist through time and need not be stored repeatedly for each time step. This observation suggests storing static parts of the scene only once. In general, compression of 4-d data can be achieved adaptively; static parts of the scene are represented with infrequent updates, while fast-moving objects are dynamically encoded at each time step to accurately describe their motion.

This paper proposes a probabilistic volumetric representation for image based modeling of general dynamic scenes that achieves such compression and allows for efficient spatio-temporal processing. This approach is facilitated by a novel space-time representation that adaptively subdivides in space and time to explain the appearance of dynamic 3-d surfaces. Space is subdivided to represent rapidly varying spatial texture or surface properties, and time is subdivided to represent motion. An intuitive modeling algorithm is proposed, where 3-d models are estimated at each time step and then inserted into the 4-d representation in an online fashion. During this insertion, the algorithm performs 3-d change detection by comparing its prediction with the incoming 3-d data to decide whether motion has occurred. If so, new memory is allocated to explain the changes in the incoming data. Otherwise, if the object is considered static, the current representation is regarded as sufficient. The algorithm provides a natural tradeoff between quality of 4-d data and compression, that can be tuned depending on the application.

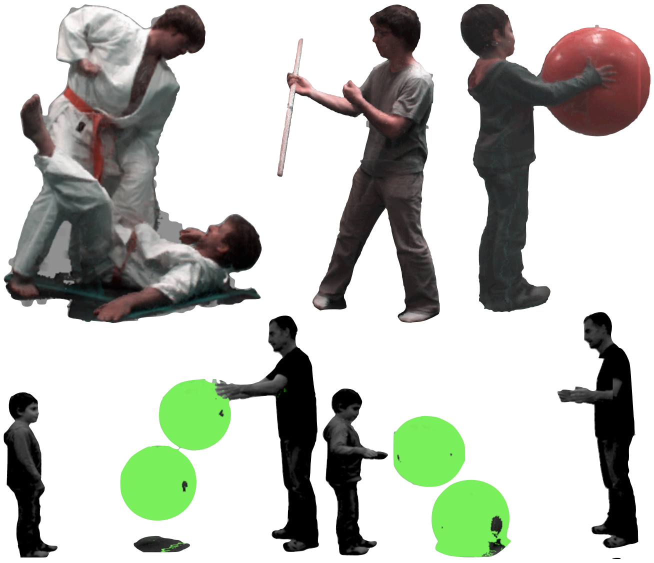


Figure 1: Top: Renderings of various 4-d probabilistic volumetric models. Note the detailed appearance and in particular, specular highlights on the red ball. Bottom: The results of the implemented 3-d tracking algorithm. The object being tracked is highlighted in green.

The resulting 4-d representation encodes probabilistic surface geometry and appearance information that is dense in both space and time, *i.e*. parameter estimates are available at all space-time cells. Appearance is modeled using a novel view-dependent mixture distribution that can explain appearance variations due to non-Lambertian reflectance properties (note the specular reflections in Figure 1 top right image). The 4-d models can be used to synthesize high quality images from novel viewpoints without the need to access the original imagery.

The proposed framework is tested on publicly available datasets [1]. Two applications, novel view rendering (for free-viewpoint video) and 3-d tracking, are used to evaluate the quality of the 4-d models as well as the performance of the overall modeling system. Novel view rendering allows quantitative evaluation of the tradeoff between quality of novel view imagery and storage requirements. Experiments indicate a graceful drop in quality with increasing compression. Moreover, the implemented 4-d rendering algorithm is capable of rendering 3-d video in almost real-time, based on space-time ray tracing in the GPU. For tracking surfaces in 4-d space-time, a mutual information (MI) based particle filter algorithm is implemented. The proposed MI measure integrates probabilistic surface and appearance information. The tracking algorithm does not assume an a-priori shape model and can track any initial 3-d region defined by a block of cells. Accurate tracking performance is demonstrated for objects undergoing complex motion including non-rigid deformations.

2. Related Work

Interest in 3-d dynamic scene modeling has been renewed recently, thanks to the advances in 3-d image-based modeling for static scenes. Common 3-d representations such as point clouds, meshes, and patches can be extended to 4-d in naive ways. However, there are difficult issues related to changes in topology and ambiguity that must be taken into account. The desired representation should be able to model arbitrary 4-d scenes, provide continuous (dense) 4-d data for spatio-temporal analysis and scale gracefully with increasing data. A brief overview and discussion of previously proposed representations is provided.

Time varying point clouds or patches are collections of 3-d primitives augmented with temporal information [34, 10, 23]. Such representations can model arbitrary 4-d shapes with a specified number of 4-d primitives. A difficulty, which is also encountered in their 3-d counterparts, is sparsity, which hinders spatio-temporal processing that requires dense association of 4-d model neighborhoods

Polygonal meshes are arguably the most popular representation for 4-d modeling. They have been used extensively in performance capture [4, 15] and free view-point video [11]. Changes in scene topology are difficult to model using meshes. This problem has been acknowledged in 3-d tracking applications, where assumptions such as fixed or known topology are commonly made [9, 8]. However, such assumptions are generally not true for arbitrary 4-d scenes, where little a-priori information is available. Moreover, recovering the topology of dynamic objects is a challenging problem [19, 28]. Although there exist works that can handle changes in topology such as [31, 7], their robustness under ambiguities inherent to image-based model inference due to occlusion and featureless surfaces is yet to be addressed.

Volumetric models provide an alternative to point and mesh based representations. They can be used to model complex 3-d scenes evolving over time while assuming little about the observed scene. Moreover, they encode data that is dense in space and time. This encoding supports scene flow analysis [32, 17] and applications such as temporal interpolation for free-viewpoint video [32].

A well known drawback of volumetric models is the exceedingly large storage requirements. The large storage footprint presents a major obstacle for high resolution 3-d modeling of static scenes, and is even more prohibitive for 4-d scenes with possibly thousands of frames. It is clear that storing 3-d models for each time step individually is not practical, nor efficient for spatio-temporal processing.

Compression of time varying volumes has been studied in the context of real time rendering [20]. In particular, Shen *et al*. propose Time-Space Partitioning (TSP) tree [26], which is a time supplemented octree. Instead of containing spatial information at its nodes, the TSP tree contains a binary time tree that adaptively subdivides to explain the temporal variation in the corresponding node. This adaptive subdivision produces a coarse discretization of time for slowly moving or static objects and a fine discretization of time to accurately describe motion. Hence, the TSP tree achieves compression of time varying volumes due to its adaptive subdivision of both space and time.

To the best of our knowledge, storage limitations and efficient spatio-temporal processing of volumetric dynamic scenes has not been addressed in image-based 4-d modeling works proposed so far [32, 17, 24]. These issues inhibit the processing of real world 4-d scenes learned from imagery. Notable exceptions include [30, 29], where a 3-d model of the static parts of the scene is used to identify and reconstruct only dynamic objects at each time step. However, these approaches do not address scalability nor do they target spatio-temporal processing.

This paper proposes a novel 4-d representation combining the state of the art in compression of time varying volumes [26] and probabilistic 3-d modeling in the GPU [21]. Compared to storing and processing 3-d models at each time step individually, the proposed framework allows for significant reduction in storage requirements as well as efficient spatio-temporal computation. Experiments indicate processing of detailed 3-d models of probabilistic surface and appearance over hundreds of frames is made feasible using the proposed framework. Novel view rendering and 3-d tracking applications are used to demonstrate the high quality of 4-d data learned from imagery as well as the benefit of dense space time data for flow analysis.

3. Dynamic Probabilistic Volumetric Models

This section describes the proposed framework for modeling dynamic 3-d scenes from multi-view video. The proposed 4-d space-time data structure is introduced in Section 3.1. The surface and appearance models encoded in this representation are discussed in Section 3.2. Finally, estimation of the proposed models from multi-view video is described in Section 3.3.

3.1. Representation and Data Structures

The proposed data structure is a time-supplemented hybrid grid-octree optimized for computation in the GPU. It is an extension of the data structure proposed by Miller *et al*. for modeling 3-d static scenes in the GPU [21] to dynamic scenes, based on the TSP tree [26]. This extension is made by supplementing the 3-d data structure with binary time trees that model the temporal variation of each 3-d cell. Rather than working with a single, deep octree and time trees that span the entire time interval as proposed in [26], the key idea is the use of shallow and compact data structures (for both space and time) amenable to GPU processing. The proposed data structure is shown in Figure 2.

The 3-d data structure proposed in [21] is based on a uniform grid of shallow (4 levels) octrees. Its shallow nature reduces the number of memory accesses needed to traverse to a cell of interest. A compact bit tree representation (16 bytes) is used instead of a pointer based representation so that once the bit tree is loaded in GPU memory, traversal is free. Experiments indicate this data structure is four times more efficient in terms of memory access compared to the standard octree [21]. The data (surface, appearance, *etc*.) associated with cells of the bit tree are stored contiguously in separate data buffers.

The proposed representation supplements this 3-d data structure with shallow binary time trees as shown in Figure 2. The time trees have a limited depth of 5 and can be stored compactly in 8 bytes using the bit tree representation. Data is stored only for the leaf cells of time trees to save storage. Once the time tree is loaded in the GPU, only a single memory access is needed to traverse to a time query. Overall, two memory accesses are sufficient to traverse to a space-time cell of interest. This representation displays spatio-temporal locality, *i.e*. cells that are close in space and time are inexpensive to query, with possibly zero memory access. This locality is important since spatio-temporal tasks typically require neighborhood accesses. For instance, in 3-d tracking, a cell at time t is frequently compared to a nearby cells at time $t-1$. Note that such locality is not supported if 3-d models are individually stored.

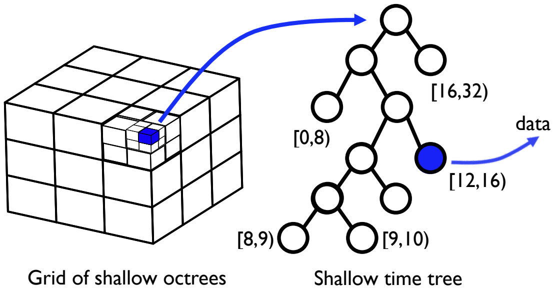


Figure 2: The proposed data structure. The blue cells and arrow depict a typical space-time cell query.

Compression of space and time is naturally achieved using the adaptive subdivision of the octrees and the time trees respectively. Spatially homogeneous regions are represented with coarse subdivisions of the octree. Static (or slowly moving) 3-d objects can be represented with a few subdivisions of their time trees, hence avoiding repeated storage for each time step.

In the proposed data structure, the time trees are limited in their depth, which also limits the extent of the time interval they are associated with. Since such a binary tree can subdivide up to 32 leaves, the time interval of each time tree spans 32 time steps. When the time interval of a time tree ends, a new grid of octrees and associated time trees, ”brick”, is initiated. The next brick's time span begins where the previous brick ends, *e.g.* when the brick in Figure 2 ends, the next brick would start at 32.

For a perfectly static scene, a time tree can represent the 32 time steps it is associated with, in a single root node. Hence, it can achieve *at most* 32 levels of compression. Although this shallow nature is not optimal in terms of compression, it presents a major benefit. It allows for limiting the number of cells (equivalently data items) in each brick, such that they can be transferred to and processed efficiently in the GPU. Note that as GPU memories grow, processing larger bricks will become feasible. This will allow deeper time tree structures, leading to higher levels of compression.

3.2. Surface and Appearance Models

The proposed 4-d data structure is capable of storing various kinds of surface and appearance information. In volumetric 3-d image based modeling, probabilistic models of occupancy and appearance have been proposed [6, 3, 22]. These models explicitly represent ambiguities and uncertainties caused by calibration errors, moving objects, areas of constant appearance and self-occlusions. They also facilitate estimation through Bayesian inference [5, 22, 16]. In particular, Pollard and Mundy propose an online learning algorithm that can update surface and appearance probabilities one image at a time [22]. Initially implemented on a regular grid of voxels, this model has been extended to variable resolution grids by Crispell *et al*. through a continuous representation of occupancy [12].

The 4-d models in this work store Crispell *et al*.'s continuous occupancy representation as well as an appearance distribution. Formally, for a cell X at time t, the surface probability is denoted as $P(X^t \in S)$ and the appearance distribution as $p(I_X^t)$, where I can be intensity or color.

The 4-d surface and appearance information can be used to synthesize images from novel viewpoints at time t. The expected appearance on an arbitrary ray R at time t can be computed as,

$$\mathbb{E}[I_R^t] = \sum_{X \in R} \mathbb{E}[I_X^t]\, P(X^t \in S) P(X^t \text{is visible}) \qquad (1)$$

where $X \in R$ denote the voxels along the ray R. This equation is a direct extension of the expected image equation proposed in [22] to include time.

The choice of how $P(I_X)$ is modeled can have significant impact on novel view generation and free-viewpoint video applications. The Gaussian distribution has been used extensively for 3-d modeling [6, 3], as well as the more expressive Mixture of Gaussians (MoG) [22, 12]. However, these models are inherently Lambertian and do not capture view-dependent variations that commonly occur in current motion capture datasets. The Lambertian assumption not only degrades the appearance quality in novel view generation, but also leads to lower quality surfaces. This degradation is due to the fact that estimation of appearance and surface probabilities are coupled; an inadequate appearance model cannot explain the fluctuations in appearance due to view point changes, thus lowering the evidence of a surface.

A novel appearance model is proposed to capture view-dependent variations. The model is parametrized by canonical directions $\{V_i\}_{i=1}^N$ and associated Gaussian distributions $\{\mu_i, \mathbf{\Sigma}_i\}_{i=1}^N$ as pictured in Figure 3. The distribution corresponding to direction V_i is used to explain the intensity when the cell is viewed from a camera looking towards V_i. In general, when the cell is viewed from an arbitrary direction R, multiple distributions are used, weighted by their directions' proximity to R. Formally, the probability of an appearance I seen from camera ray R is expressed as,

$$p(I\,;\,R) = \frac{1}{\sum w_i} \sum_{i=1}^{N} w_i \; p(I_X;\, \mu_i, \mathbf{\Sigma}_i) \qquad (2)$$

$$\text{where } \; w_i = \begin{cases} -V_i \cdot R & \text{if } V_i \cdot R < 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Note that distributions only corresponding to directions that lie on the hemisphere facing R have non-zero weights, *i.e.* components facing away from R do not contribute.

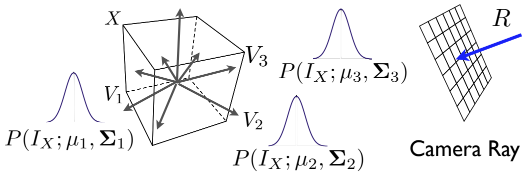

X
V_3
$P(I_X; \mu_3, \mathbf{\Sigma}_3)$
R
V_1
$P(I_X; \mu_1, \mathbf{\Sigma}_1)$
V_2
$P(I_X; \mu_2, \mathbf{\Sigma}_2)$
Camera Ray

Figure 3: Depiction of the proposed view-dependent appearance model.

Similar algorithms have been proposed in the context of view-dependent texture mapping [11, 13]. Most such works use the original imagery to blend novel textures. Instead, the proposed method takes a probabilistic approach by modeling view-dependent distributions. Moreover, it does not require access to the training imagery or precomputed texture atlases during novel view generation.

3.3. 4-d Modeling from Multi-view Video

This section describes the algorithm used to estimate 4-d models that encode the proposed surface and appearance distributions, from multi-view video. The input to the algorithm is imagery and associated cameras calibrated in space and time, *i.e.* known pose and time of capture. The motion capture studio environment provides such data alongside with foreground/background segmentation, *i.e.* silhouettes. The proposed algorithm does *not* require silhouettes but can benefit from them, to remove irrelevant background surfaces.

The algorithm starts by estimating a 3-d volumetric model (surface and appearance distributions) independently for each frame, using the online update algorithm of Pollard and Mundy [22]. The update algorithm is modified to make use of silhouette information by simply discarding background pixels. This scheme is similar to Photo hulls proposed by Slabaugh [27] in that occupancy and appearance of voxels *only* inside the visual hull are estimated.

The next step is inserting the 3-d models into the proposed 4-d representation in an online manner. The insertion algorithm takes as input a 4-d model with time span 0 to $T-1$ and 3-d model at time T. The algorithm performs two major steps, here referred to as *conform* and *compare*. In the *conform* stage, the octree of the current 4-d model is subdivided such that each octree node has same or higher resolution compared to the corresponding node in the incoming octree. This makes sure the 4-d model can match the spatial subdivision of the incoming 3-d model. In the *compare* stage, the 4-d model's prediction for time T is compared against the incoming 3-d model's data. Note that this prediction is obtained simply by traversing the time trees to reach leaf node containing T. If the prediction does not accurately match the incoming data, the corresponding time trees are subdivided to allocate new memory and incoming data is copied into the 4-d representation.

As proposed in Section 3.2, the 4-d models represent probabilistic information. Hence, in the comparison step, the distance between two probability distributions must be evaluated. KL divergence provides an attractive measure for this task. $D_{\mathrm{KL}}(P||Q)$ is a measure of information lost when Q is used to approximate P. This provides the interpretation: P is the true distribution of the incoming 3-d data, and it is being approximated by Q, the prediction. Hence, P is regarded as well-represented if $D_{\mathrm{KL}}(P||Q) < \tau$.

In practice, both surface and appearance distributions are used for comparison, denoted by

$$\begin{aligned} &D_{\mathrm{KL}}(P(X^T)\,||\,Q(X^T)) < \tau_S \wedge \\ &D_{\mathrm{KL}}(p(I_X^T)\,||\,q(I_X^T)) < \tau_A, \end{aligned} \tag{4}$$

where τ_S and τ_A are specified thresholds on surface and appearance distances respectively.

4. Experiments and Applications

The proposed framework is implemented under the open source VXL project [2]. Experiments are presented on a number of sequences from the publicly available "'4-d repository" from INRIA [1]. All sequences were collected in a motion capture studio and share the same setup with 16 cameras distributed on a hemisphere and 1624x1224 imagery. Foreground segmentation is also provided for each image. The sequences are STICK (280 frames), GUARD PUNCH TWO (255 frames), ADULT CHILD BALL (340 frames) and BOY PLAYING BALL (530 frames), and example renderings of estimated 4-d models is displayed in Figure 1 respectively, from left to right.

First, the effectiveness of the proposed view-dependent appearance model is demonstrated in comparison to a single Gaussian and Mixture of Gaussians (MoG) models. The evaluation is carried out for static 3-d scenes. Five random frames are selected from each of the four sequences. All models are estimated using the online update framework [22]. The learning for the view-dependent model is similar to that of the single Gaussian case, where the parameters of the distribution are updated incrementally with incoming intensities. However, each Gaussian component is updated with a different weight (3), according to its direction's proximity to the training camera.

The resulting models are compared through novel view generation (see eq (1)). Namely, the models are trained with a leave-one-out procedure, where a random viewpoint is left out during training. This viewpoint is used to render the most probable image, given the model, corresponding to the actual left out image. Subsequently, the model-predicted image is compared to the left out (ground truth) image using the SSIM image similarity measure [33], which takes into account human perception.

Sequence	Gaussian	MoG	View-dependent
STICK	0.59	0.62	**0.70**
GUARD PUNCH TWO	0.46	0.53	**0.62**
ADULT CHILD BALL	0.74	0.71	**0.76**
BOY PLAYING BALL	0.79	0.76	**0.80**

Table 1: Average SSIM scores of novel view rendering experiments for three different appearance models.

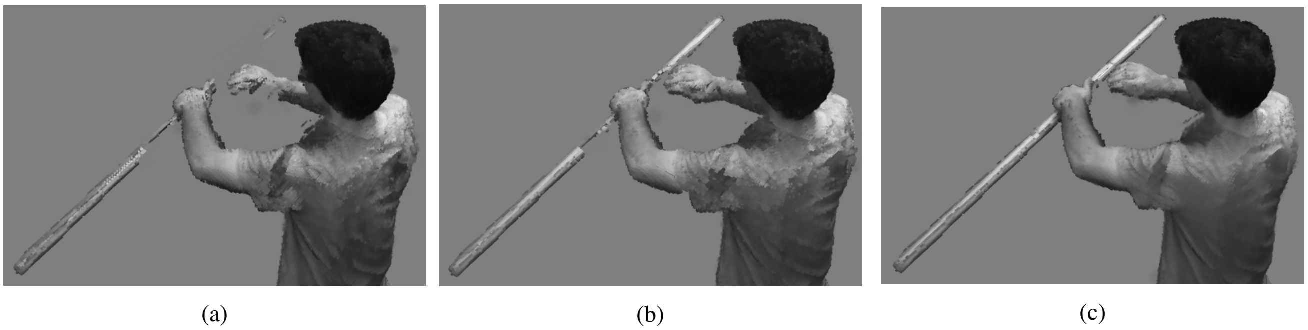
The average SSIM scores for each sequence are presented in Table 1. It can be observed that the proposed view-dependent model scores consistently higher than Gaussian and MoG models. The contrast can be best appreciated in scenes where non-Lambertian effects are abundant. For instance, in GUARD PUNCH TWO, the white karate uniforms are the dominant material in the scene and demonstrate view-dependent variations due to the spot lights above. In STICK, the Taekwando staff as well as the arms and shoulders of the actor also exhibit such effects. Figure 4 presents a visual comparison of the three appearance models on a frame from STICK. It can be observed neither the Gaussian nor the MoG models can explain the variation of intensities on the staff, arms or shoulders. Therefore, these models have difficulty forming the geometry of such surfaces. In contrast, the proposed model is able to explain these complex variations via learning distinct appearance models for different viewpoints. Accurate estimation of surface geometry as well as appearance result in the higher scores achieved by the proposed model.

In sequences where view-dependent variations are not prevalent, the proposed model behaves similarly to Gaussian or MoG models. For instance, in ADULT CHILD BALL or BOY PLAYING BALL, only the upper region of the ball contains highlights, whereas rest of the scene is largely Lambertian. The proposed model attains the highest SSIM scores for these sequences as well, however, the improvements are less significant.

4.1. Free-viewpoint video rendering

Free-viewpoint video is a popular application in 4-d modeling, where the user interactively chooses viewpoints to observe a dynamic scene. This application necessitates realistic synthesis of novel view imagery at interactive rates. The proposed framework readily supports this synthesis through computation of Eq. (1), which provides the expected appearance of a camera ray. This equation is computed efficiently using space-time ray tracing in the GPU.

An analysis of novel view rendering quality with varying compression is presented. The test datasets are trained with a leave-one-out procedure, where a randomly left out viewpoint is used to render a video of the dynamic scene. The rendered video is compared to ground truth by averaging the SSIM scores of each frame in the video. The results are presented in Figure 5a. The datasets yield high SSIM scores



(a) (b) (c)

Figure 4: Novel view renderings using three different appearance models. Please zoom in the images to appreciate the differences. The scene is a frame from GUARD PUNCH TWO. (a) Gaussian. (b) MoG. (c) Proposed view-dependent model.

when compression rate is low, except for GUARD PUNCH TWO, which results in a relatively lower score. GUARD PUNCH TWO presents challenges in terms of 3-d modeling because it contains regions of constant appearance (the karate uniforms) and significant occlusions coupled with limited viewpoints.

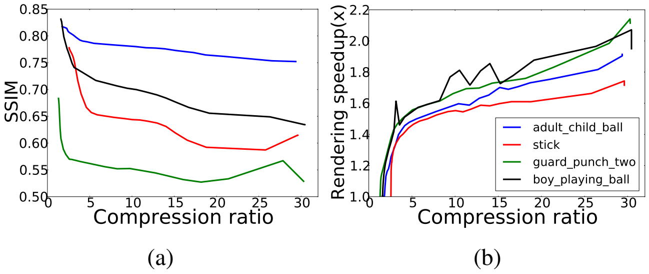
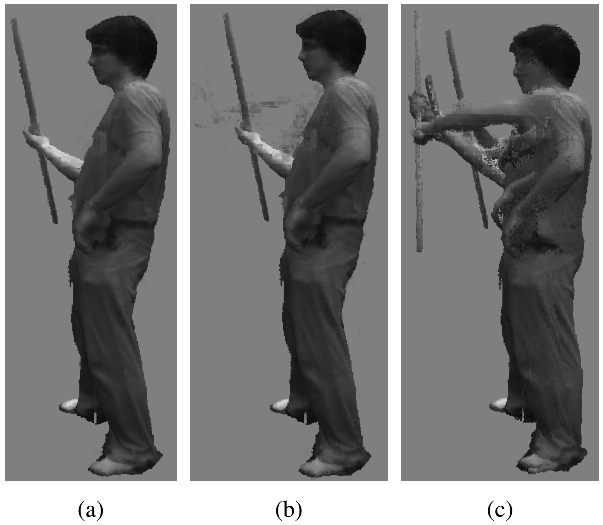


Figure 5: Novel view rendering quality and performance behavior with varying compression. (a) Rendering quality measured by SSIM. The legend for this plot is identical to that of (b). (b) Rendering performance. The baseline is the performance of rendering 3-d models at each time step.

As observed in Figure 5a, video fidelity gradually decreases with increasing compression. Note that compression is controlled by the subdivision of time trees (see Section 3.1). High levels of compression can be achieved by allowing a coarse subdivision of time. However, an object undergoing fast motion may require a fine temporal resolution to be described accurately. Such objects may not be modeled well if the allowed resolution doesn't match the speed of the object. In general, insufficient sampling of time leads to motion artifacts along object trajectories which, in turn, degrade the rendered image quality. An example is provided in Figure 6. Note that due to the high speed of the rotating staff, artifacts begin to appear under 3 fold compression and are more severe under high compression.

It should also be noted that the legs and torso of the actor move at lower speeds and therefore, their motion can be significantly compressed with little effect on visual quality. As seen in Figure 6, they are modeled accurately even under high compression. Overall, the datasets analyzed in the paper are quite challenging in this regard; they contain fast moving large objects. In such scenes, even little compression may prevent achieving the desired temporal resolution and result in image artifacts. Moreover, the objects undergoing motion are quite large and therefore affect substantial image regions on the rendered video. Nonetheless, the proposed framework achieves compression ratios of at least 3 while retaining visually acceptable quality as seen in the supplemental video. Future work will address real world, e.g. outdoor, scenes where static or slowly moving objects are much more frequent and higher levels of compression are anticipated.



(a) (b) (c)

Figure 6: Novel view renderings of STICK with varying compression ratios. (a) No compression. (b) 3 fold compression. (c) 15 fold compression.

The performance of novel view video rendering with varying thresholds is also evaluated and the results are shown in Figure 5b. The baseline against which performance is compared is the performance of rendering when 3-d models are stored at each time step. The performance is evaluated as the GPU execution time, which includes transfer of data as well as computation. It can be observed that the performance increases rapidly with higher levels of compression, due both to the decrease in the total number of cells that need to be transferred to the GPU, as well as reduced traversal of time trees. The baseline performance on average takes 100ms to render a novel view image. Hence, the proposed system can achieve rendering at interactive rates when allowing acceptable degradation of rendering quality.

4.2. 3-d Tracking

The proposed framework provides dense surface and appearance information in both space and time, *i.e.* estimates are available at all cells. Note that point or mesh based representations are sparse by comparison. This dense nature allows 3-d motion analysis and tracking applications such as scene flow [32, 17]. In particular, motion analysis and tracking algorithms commonly used for video processing can be directly extended to their 3-d counterparts for the proposed framework. As a demonstration, an annealed particle filter tracker [14] is implemented that displays the benefits of dense surface and appearance information, as well as the feasibility of flow analysis in the proposed framework.

The annealed particle filter is a robust Bayesian framework for tracking in high dimensional state spaces [14]. It employs simulated annealing to search for and localize peaks of the observation density or fitness function. A mutual information (MI) measure is proposed as the fitness function, which integrates both surface and appearance distributions in "expected appearance", defined as $\mathbb{E}[I_X^t]\, P(X^t \in S)$. The expected appearance was initially proposed by Restrepo *et al.* as a characterization of volumetric models in the context of 3-d object recognition [25].

The implemented tracker assumes no shape prior and tracks the 3-d positions of objects given an initial labeling. The motion model is chosen to be a two part Gaussian mixture distribution, where the mean of one Gaussian is the velocity estimate and the other Gaussian is zero mean. All experiments were conducted with 5 annealing steps and 128 particles.

A bounding box around the plastic ball is marked as initial 3-d region to be tracked in datasets ADULT CHILD BALL and BOY PLAYING BALL. In ADULT CHILD BALL, the ball is being bounced back and forth between a man and a child. Although the motion of the ball is mostly smooth, there are large velocity changes as well as significant non-rigid deformations when the ball hits the ground. In BOY PLAYING BALL, the boy is bouncing the ball while rotating around himself. The motion is characterized by very high speeds and frequent changes in direction. The deformations are also more pronounced. For both datasets, the track is maintained accurately for the duration of the dataset. Screenshots of the respective tracks are shown in Figures 1 and 7.

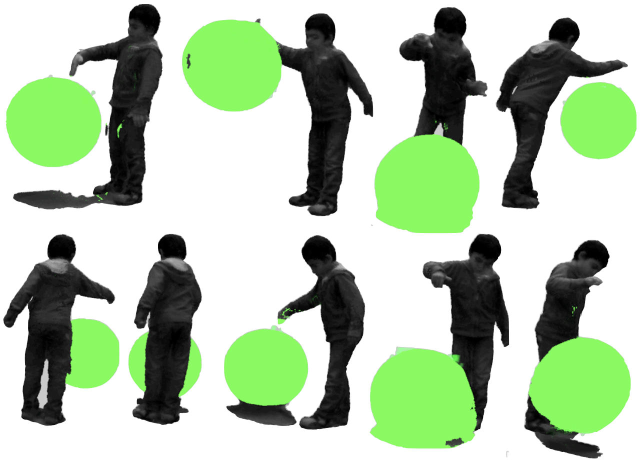


Figure 7: Tracking results of a plastic ball in BOY PLAYING BALL viewed from a training camera.

Please see the supplementary video for the entire result.

The tracker was also extended to include 3-d rotations. An initial experiment is provided for proof of concept. The Taekwondo staff undergoing fast rotation is tracked as shown in Figure 8. The algorithm maintains track until the performer reaches with his second hand and grabs the staff. The second hand disrupts the tracker because it is not contained in the provided initial region and that it encapsulates substantial volume compared to the initial region. This issue will be addressed in future work using adaptive tracking as well as segmentation of objects being tracked.

As mentioned in Section 3.1, the proposed representation possesses spatio-temporal locality, which allows the tracker to evaluate the MI measure very efficiently. The tracker benefits from increasing compression similarly to rendering performance behavior shown in Figure 5b. The current implementation takes roughly thirty seconds (for the ball example) to evaluate the MI function for 128 particles. The ball encapsulates a large 3-d region which slows down the evaluation. Significant speedups are anticipated when tracking smaller regions such as patches. Future work includes analysis of efficiency and quality of dense scene flow in the proposed representation.

5. Conclusion and Future Work

This paper presented a novel probabilistic volumetric representation that addresses the storage and processing limitations of current volumetric image based 4-d modeling works. The proposed framework is shown to achieve high quality modeling of complex 4-d scenes. Experiments were presented to demonstrate the tradeoff between compression and novel view rendering quality, as well as the 3-d tracking capabilities of the system.

Future work will include testing the proposed framework in large scale outdoor 4-d environments such as mountain-

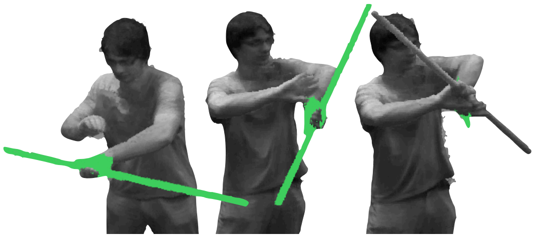


Figure 8: Tracking result of a Taekwondo staff in STICK.

ous regions and urban canyons, where static objects are common. Such scenes present a major source of compression which can be readily exploited by the proposed representation. However, in contrast to motion capture studios, such scenes present significant challenges in terms of data acquisition such as limited number of viewpoints and temporal synchronization of cameras [18, 30]. These challenges highlight a limitation of the current modeling algorithm, that is, in such uncontrolled environments, 3-d models of each time instant may not be acquired reliably. This limitation motivates investigating whether the proposed 4-d models can be estimated directly from imagery.

References

[1] 4-d repository. http://4drepository.inrialpes.fr/, 2012. 2, 5

[2] Vxl. http://http://sourceforge.net/projects/vxl/, 2012. 5

[3] M. Agrawal and L. Davis. A probabilistic framework for surface reconstruction from multiple images. *CVPR*, 2001. 3, 4

[4] E. D. Aguiar, C. Stoll, and C. Theobalt. Performance capture from sparse multi-view video. *ACM SIGGRAPH*, 2008. 2

[5] R. Bhotika, D. J. Fleet, and K. N. Kutulakos. A Probabilistic Theory of Occupancy and Emptiness. In *ECCV*, 2002. 3

[6] J. D. Bonet and P. Viola. Poxels: Probabilistic voxelized volume reconstruction. *ICCV*, 1999. 3, 4

[7] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global Nonrigid Alignment of Surface Sequences. *IJCV*, 2012. 2

[8] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: A patch-based approach. *CVPR*, 2010. 2

[9] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic Deformable Surface Tracking. *ECCV*, 2010. 2

[10] R. L. Carceroni and K. N. Kutalakos. Multi-view scene capture by surfel sampling: from video streams to non-rigid 3D motion, shape and reflectance. *ICCV*, 2001. 2

[11] J. Carranza, C. Theobalt, M. a. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM TOG*, 2003. 2, 4

[12] D. Crispell, J. Mundy, and G. Taubin. A Variable-Resolution Probabilistic Three-Dimensional Model for Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2012. 4

[13] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *ACM SIGGRAPH*, 1996. 4

[14] J. Deutscher, a. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *CVPR*, 2000. 7

[15] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. *CVPR*, 2008. 2

[16] L. Guan, J. Franco, and M. Pollefeys. 3d occlusion inference from silhouette cues. *CVPR*, 2007. 3

[17] L. Guan and M. Pollefeys. Probabilistic 3D Occupancy Flow with Latent Silhouette Cues. *CVPR*, 2010. 1, 2, 7

[18] N. Hasler and B. Rosenhahn. Markerless motion capture with unsynchronized moving cameras. *CVPR*, 2009. 8

[19] a. Letouzey and E. Boyer. Progressive shape models. *CVPR*, 2012. 2

[20] K.-L. MA. Visualizing Time-varying Volume Data. *Computing in Science and Engineering*, March 2003. 2

[21] A. Miller, V. Jain, and J. L. Mundy. Real-time rendering and dynamic updating of 3-d volumetric data. In *Proceedings of the Fourth Workshop on GPGPU*, 2011. 2, 3

[22] T. Pollard and J. L. Mundy. Change Detection in a 3-d World. In *CVPR*, June 2007. 3, 4, 5

[23] T. Popham. *Tracking 3D Surfaces Using Multiple Cameras : A Probabilistic Approach by*. PhD thesis, University of Warwick, 2010. 2

[24] A. Prock and C. Dyer. Towards real-time voxel coloring. *Proceedings of the DARPA Image Understanding Workshop*, 1998. 1, 2

[25] M. I. Restrepo, B. Mayer, A. O. Ulusoy, and J. L. Mundy. Characterization of 3-d Volumetric Probabilistic Scenes for Object Recognition. *Journal of Selected Topics in Signal Processing*, 2012. 7

[26] H.-W. Shen, L.-J. Chiang, and K.-L. Ma. A fast volume rendering algorithm for time-varying fields using a time-space partitioning (TSP) tree. In *IEEE Proceedings Visualization*, 1999. 2, 3

[27] G. Slabaugh, R. Schafer, and M. Hans. Image-based photo hulls. *3DPVT*, 2002. 4

[28] T. Popa and I. South-Dickinson and D. Bradley and A. Sheffer and W. Heidrich. Globally Consistent Space-Time Reconstruction. *Computer Graphics Forum*, 2010. 2

[29] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, 2011. 2

[30] A. Taneja, L. Ballan, and M. Pollefeys. Modeling dynamic scenes recorded with freely moving cameras. *ACCV*, 2011. 2, 8

[31] K. Varanasi and A. Zaharescu. Temporal surface tracking using mesh evolution. *ECCV*, 2008. 2

[32] S. Vedula. *Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events*. PhD thesis, 2001. 1, 2, 7

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *ICIP*, 2004. 5

[34] S. Würmlin, E. Lamboray, and M. Gross. 3D video fragments: dynamic point samples for real-time free-viewpoint video. *Computers & Graphics*, 2004. 2