EUROGRAPHICS 2007 / D. Cohen-Or and P. Slavík
(Guest Editors)

*Volume 26 (2007), Number 3*

# 3D Video Billboard Clouds

Michael Waschbüsch[†1], Stephan Würmlin[‡2], Markus Gross[§1]

[1]Computer Graphics Laboratory, ETH Zurich, Switzerland
[2]LiberoVision Inc., Switzerland

**Abstract**

*3D video billboard clouds reconstruct and represent a dynamic three-dimensional scene using displacement-mapped billboards. They consist of geometric proxy planes augmented with detailed displacement maps and combine the generality of geometry-based 3D video with the regularization properties of image-based 3D video. 3D video billboards are an image-based representation placed in the disparity space of the acquisition cameras and thus provide a regular sampling of the scene with a uniform error model. We propose a general geometry filtering framework which generates time-coherent models and removes reconstruction and quantization noise as well as calibration errors. This replaces the complex and time-consuming sub-pixel matching process in stereo reconstruction with a bilateral filter. Rendering is performed using a GPU-accelerated algorithm which generates consistent view-dependent geometry and textures for each individual frame. In addition, we present a semi-automatic approach for modeling dynamic three-dimensional scenes with a set of multiple 3D video billboards clouds.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems–Video.

## 1. Introduction

Being able to digitally experience real-world locations and events from arbitrary viewpoints and viewing directions has interesting applications in next-generation television, video games, virtual reality and movie production. Imagine watching a sports play on television always from the best perspective by interactively choosing any viewpoint in a sports stadium using your remote control. For such experiences the scene has to be captured first using a set of digital cameras. Then, the multi-view image or video data has to be processed to some three-dimensional representation of the scene which allows virtual views to be synthesized. Such technology is usually referred to as 3D photography if the scene is static, or 3D video if the scene is captured in motion. In recent years, a number of capturing systems and representations have been introduced for 3D video. However, high-quality results are not achieved easily for arbitrary setups. Moreover, the underlying data representation strongly influences the final image quality. Geometry-based representations are typically view-independent and can be used for arbitrary camera configurations. However, they do not allow for production-quality results. Image-based representations allow synthesizing high-quality images even of highly complex scenes. However, the lack of detailed geometry and need for very dense camera setups limit its practical applicability.

In this paper we present 3D video billboard clouds as a representation for three-dimensional video. They combine the generality of geometry-based representations with the regularization properties of image-based representations. As an extension to the original billboard clouds representation [DDSD03], displacement-mapped billboard clouds (DMBBC) [MJW07] have been recently introduced as a new image-based rendering primitive. They represent parts of a geometrically complex scene as a set of geometric proxy planes augmented with detailed displacement maps. We exploit DMBBCs for 3D video by constructing both their proxy planes and displacement maps from depth images of the scene. Those can be acquired using stereo matching algorithms which are usually subject to noise and errors. The billboard planes with their approximate geometry are used to regularize this noisy, detailed geometry. By placing the bill-

[†] waschbuesch@inf.ethz.ch
[‡] wuermlin@liberovision.com
[§] grossm@inf.ethz.ch

board representation in the disparity space of the acquisition cameras, they provide a regular sampling of the scene with a uniform model of acquisition error. This allows applying signal processing algorithms to generate models with space and time coherence. The application of bilateral filters can successfully remove reconstruction and quantization noise, as well as calibration errors, and, thus, allows for higher quality renderings compared to complex and time-consuming sub-pixel stereo matching algorithms. Our GPU-accelerated rendering algorithm is able to further improve the final image quality by generating consistent view-dependent geometry and textures for each individual frame. To handle not only objects with our representation we also present a semi-automatic approach for modeling complete dynamic three-dimensional scenes with a set of multiple 3D video billboards clouds.

## 2. Related Work

Three-dimensional television (3D-TV) and three-dimensional video (3D video) naturally extend their two-dimensional counterparts to the spatio-temporal domain. With 3D-TV, viewers can perceive depth in television and thus achieve deeper immersion into the media. 3D video allows viewers to choose viewpoints and viewing directions in visual media content at will.

To date, there exists a continuum of representations and algorithms suited for different acquisition setups and applications. For applications in 3D-TV [MP04], purely image-based representations [LH96] are well suited but need many densely spaced cameras. Examples include dynamic light field cameras [WJV*05, YEBM02] with camera baselines of a couple of centimeters. Introducing geometry to the representation allows for true navigation in the visual scene. Depth image-based representations [SGHS98] for 3D video applications exploit depth information per pixel which can be computed either by shape-from-silhouettes [MBR*00] or by stereo algorithms [ZKU*04]. While the former is only applicable to stand-alone objects, the latter still requires small baselines and does not integrate multiple cameras easily. Hence, both do not permit practical camera configurations. With our proposed filtering and view-dependent rendering framework, 3D video billboards are able to integrate sparse input views to produce consistent images. This reduces the acquisition system to a hybrid setup combining small baselines for bifocal stereo vision with large baselines between the different input views.

Explicit 3D geometry allows for view-independent representations. Examples include triangular meshes [MBM01, RNK97] and 3D point samples [WWC*05, WLSG02]. Voxel-based reconstructions and representations [VBK02] are also suited for 3D video. They are usually restricted to stand-alone objects and are limited in resolution. The use of a template model geometry [CTMS03] together with video textures achieves high-quality output images for almost all camera configurations. However, complex scenes cannot be modeled since the template has to be known beforehand. In contrary, our technique combines detailed geometry with simple but generic planar priors that allow for regularizing acquisition noise using a signal processing framework.

## 3. 3D Video Billboard Clouds

A 3D video billboard cloud models a single 3D video object and comprises a collection of multiple 3D video billboards. A 3D video billboard represents the 3D structure and texture of an object at a specific point in time as observed from a single viewpoint. It consists of an arbitrarily placed and oriented texture-mapped rectangle or proxy $B$ approximating the real geometry of the object. Its associated textures are a displacement map $\mathbf{D}$ for adding fine scale geometric detail, a color map $\mathbf{C}$ modeling the surface appearance, and an alpha map $\mathbf{A}$ holding a smooth alpha matte representing the object's boundary. The latter is employed for seamless blending with the background of the scene.

Let us first assume that the required input data to generate such a billboard is available. Figure 1 shows the input data consisting of color images, alpha mattes, and displacement maps of a single object, which serve as texture maps on the billboard planes. Color images and alpha mattes can be recorded e.g. by standard cameras in front of a green screen or using segmentation and matting algorithms [WBC*05, LSS05]. In Section 6 we propose a method to construct this data from multi-view recordings of real-world scenes with multiple objects. The displacement maps can be modeled using the depth information in the acquisition cameras by placing them in disparity space as explained in 3.1. The depth values can be reconstructed using computer vision methods or can be acquired with special equipment, e.g. so-called $Z$-cams. Figure 2 illustrates the billboard planes and their composition to a billboard cloud, as well as displacement-mapped billboard planes.

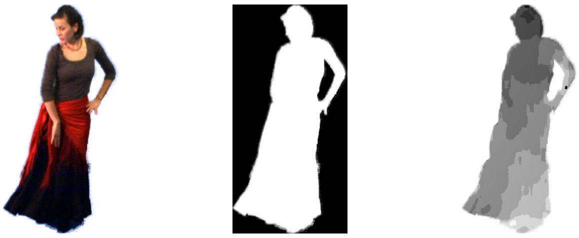

**Figure 1:** *Billboard textures: Colors (left), alpha matte (middle), and displacement map (right).*

We impose a set of requirements for an optimal 3D video billboard clouds representation:

1. **Simple geometric proxy.** The geometric proxy should be as simple as possible, i.e. a rectangle. This permits an easy parameterization for texture mapping.

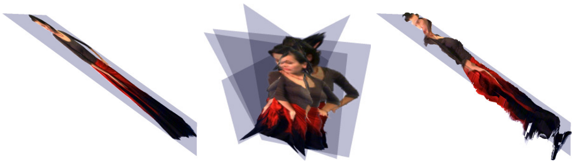

**Figure 2:** *Illustration of the billboard cloud for one object: Billboard plane from one input view (left), composition of planes from multiple input views to a billboard cloud (middle), displacement-mapped billboard plane from one input view (right).*

2. **Regular sampling.** By ensuring a regular sampling we can exploit standard signal processing methods for post-processing of the geometry without the need of resampling. In particular, we would like to directly exploit the existing regular sampling from the acquisition cameras.
3. **Uniform error model.** 3D reconstruction introduces noise which is usually not uniform in world coordinates. The uncertainty of depth values reconstructed by triangulation increases with their absolute value. Our billboard representation should be defined in a space where the reconstruction error is approximately uniform, independent from the distance of a surface from the camera. Thus, uniform, linear filters can be applied for smoothing the acquired geometry.
4. **Minimal displacements.** A minimal displacement of the proxy to the real surface ensures a good approximation of the geometry and can improve future compression and level-of-detail algorithms.

Requirement (1) can be guaranteed by definition. Requirements (2) and (3) are fulfilled by defining the billboards not in conventional 3D space of the scene but in the so-called disparity space of the acquisition camera. This is described in Section 3.1. Finally, a minimization algorithm introduced in Section 3.2, ensures the last requirement.

### 3.1. Scene Sampling and Error Model

Consider an input depth map $D = \{(u_i, v_i, z_i)\}$ which has for each pixel at coordinates $(u_i, v_i)$ a unique depth value $z_i$. The pixels are sampled on a uniform, regular grid. This is a representation of the scene in the ray space of an acquisition camera, as each pixel corresponds to a unique viewing ray. Assume a pinhole model with $3 \times 3$ projection matrix $\mathbf{P}$ and center of projection $\mathbf{c}$. Camera space coordinates $(x_i, y_i, z_i)$ are projected into ray space by

$$\begin{pmatrix} z_i u_i \\ z_i v_i \\ z_i \end{pmatrix} = \mathbf{P} \cdot \left[ \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{c} \right] \tag{1}$$

followed by the division through the homogeneous $z_i$ coordinate. This is a nonlinear transform, i.e. linear functions in camera space are not linear in ray space anymore, as illustrated in Figure 3, and vice versa. Hence, if we defined the billboard plane in ray space and use the depth values as displacements, it would not be planar in world coordinates and thus it would be difficult to use it as an approximation for the real geometry. On the other hand, if we placed it in camera space, the sampling would become irregular.

Instead, we define a disparity space of a camera as coordinates $(u_i, v_i, z_i')$ with $z_i'$ being inversely proportional to $z_i$. As scaling does not matter in our case, we can use $z_i' = \frac{1}{z_i}$. Using this representation and storing the reciprocal of the $z$-coordinate from ray space, we can observe that planes in disparity space stay planar in camera space (cf. Figure 3). Moreover, sampling in disparity space is identical to the regular sampling of the acquisition cameras. Thus, requirement (2) is fulfilled if we define the billboard planes in these coordinates.

This representation is directly motivated from the fact that most 3D scanners based on triangulation do not compute pixel depths $z_i$ but disparities $z_i'$. For example, a simple depth from stereo algorithm computes disparities as the difference of the $u$ coordinates of two corresponding pixels in two different, rectified camera images. Due to the spatial extent of the pixels, this produces a quantization error of $\Delta z'$ which is constant for each sample. Also for stereo matching techniques with sub-pixel accuracy, there is a remaining uncertainty in the disparity values. In camera space it can be observed that the resulting uncertainty of the geometry is not constant anymore but depending on the absolute value of the disparity. This is illustrated with the brown error bars in Figure 3. By defining the billboards in disparity space we can thus use a uniform model for the reconstruction error and fulfill requirement (3).

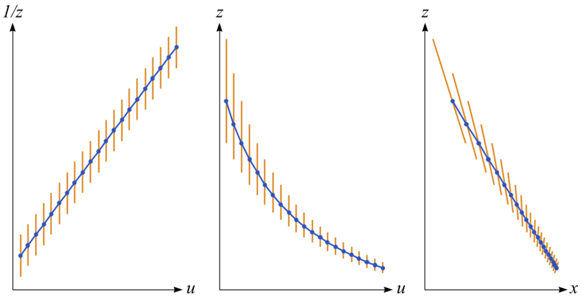

**Figure 3:** *Illustration of the sampling of a plane (blue) and the triangulation error model (brown) in disparity space (left), ray space (middle), and camera space (right).*

In conclusion, we obtain an image space representation of a billboard using pixel disparities by modeling the plane

$$B(u, v) = b_u \cdot u + b_v \cdot v + b_0 \tag{2}$$

as a linear scalar function over the pixel coordinates $u$ and $v$, and the displacement map $D = \{(u_i, v_i, d_i)\}$ with

$$d_i = z_i' - B(u_i, v_i) = \frac{1}{z_i} - B(u_i, v_i). \tag{3}$$

Using stereo vision as input, we can compute the displacements directly from the disparities $z_i'$.

### 3.2. Optimal Billboard Placement

We are still free to choose the position $b_0$ and orientation $b_u$ and $b_v$ of the billboard plane. A bad choice of these values can lead to arbitrarily large displacements in world coordinates. This becomes an important issue as soon as the values of the displacement map should be processed. E.g. by applying filters for improving the surface geometry, already very small errors due to numerical instabilities can become very large in world coordinates and produce large geometric artifacts. Another example is lossy compression of disparity maps, e.g. if they are stored as compressed textures in the GPU. While compression algorithms try to minimize artifacts appearing in the disparity maps it should also be ensured that visible artifacts on the actual surfaces also remain small. In these terms, we are looking for an optimal choice of the plane parameters.

Note that that constructing a least-squares plane in disparity space by computing

$$\arg\min_{b_u, b_v, b_0} \sum_i \left|\left| b_u u_i + b_v v_i + b_0 - z_i' \right|\right|^2 \tag{4}$$

is not sufficient. Although it minimizes the displacements in disparity space they can grow arbitrary large in world coordinates depending on the magnitude of the present disparities. A result of such a minimization in 2D is illustrated on the left side of Figure 4. Instead the optimization has to be carried out directly in world coordinates by solving the non-linear least squares problem

$$\arg\min_{b_u, b_v, b_0} \sum_i \left|\left| \mathbf{P}^{-1} \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \cdot \left( \frac{1}{b_u u_i + b_v v_i + b_0} - \frac{1}{z_i'} \right) \right|\right|^2. \tag{5}$$

Usually, five to ten iterations of the Levenberg-Marquardt algorithm are sufficient for convergence to a relative error below $10^{-6}$. The right side of Figure 4 shows the result of such an optimization.

It may be appropriate to stabilize the billboard planes over time. This can be easily implemented by incorporating additional disparity space coordinates $(u_i, v_i, z_i')$ from previous and successive frames into equation (5), weighted by their temporal distance from the current frame. However, our experiments have shown already a good temporal stability without this approach. This can be explained with the temporal invariance of acquisition noise and the robustness of the least-squares fit against outliers.

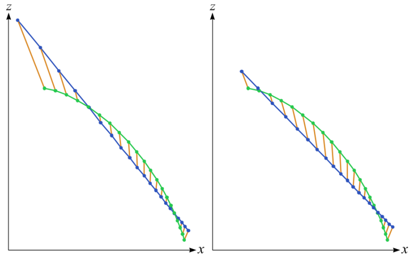

**Figure 4:** *Illustration of surface (green), billboard plane (blue) and displacements (brown) in world space. Left: displacements optimized in disparity space. Right: displacements optimized in world space.*

## 4. Filtering Framework

The displacement values generated by the acquisition system are subject to quantization errors, noise, and calibration inaccuracies, resulting in several kinds of artifacts in the re-rendered image: The object surfaces do not appear smooth and their geometry is very noisy, which is especially visible as flickering over time. Moreover, overlapping parts of surfaces from different scanning directions do not necessarily fit to each other. To improve this, we apply a four-dimensional smoothing filter yielding better spatial coherence within surfaces and between overlapping surfaces, and better coherence over time. This complements window-based stereo matching algorithms like space-time stereo [DRR03, ZCS03]. They apply spatio-temporal filters on the correlation function in order to enhance local maxima before they are extracted [SS02]. Our method filters the locations of these maxima after extraction to smooth errors of the previous optimization step, as well as errors like camera misalignment which stereo matching cannot handle.

Our displacement map representation has some nice properties allowing us to use standard signal processing tools to filter the geometry. The billboard plane serves as a parameter domain for scalar data sampled on a regular grid. Over time, it provides a parameterization along object trajectories because each billboard represents a best fit to the local geometry. Additionally, the invariant error model in disparity space allows for using a uniform filter kernel.

Let $d_i(\mathbf{x})$ be all disparities reconstructed from acquisition view $i$, where $\mathbf{x} = (u, v, t)$ contains the pixel coordinates $u$ and $v$, and the acquisition time $t$. A smoothed version $\tilde{d_i}$ can be computed independently from all other acquisition views by using a bilateral low-pass filter

$$\tilde{d_i}(\mathbf{x}) = \frac{\int d_i(\zeta) \cdot c(\zeta, \mathbf{x}) \cdot s(d_i(\zeta), d_i(\mathbf{x})) \cdot d\zeta}{\int c(\zeta, \mathbf{x}) \cdot s(d_i(\zeta), d_i(\mathbf{x})) \cdot d\zeta}. \tag{6}$$

The domain filter kernel $c$ smoothes the disparities over space and time while the range filter kernel $s$ retains geometric discontinuities.

However, when filtering each acquisition view independently, corresponding surfaces reconstructed from different views will not fit to each other. Thus, we extend the filter by an additional domain by accumulating the data from all acquisition views:

$$\tilde{d}_i(\mathbf{x}) = \frac{\sum_j \int d_j(\zeta^{i\triangleright j}) \cdot c(\zeta^{i\triangleright j}, \mathbf{x}^{i\triangleright j}) \cdot s(d_j(\zeta^{i\triangleright j}), d_i^{i\triangleright j}(\mathbf{x})) \cdot d\zeta}{\sum_j \int c(\zeta^{i\triangleright j}, \mathbf{x}^{i\triangleright j}) \cdot s(d_j(\zeta^{i\triangleright j}), d_i^{i\triangleright j}(\mathbf{x})) \cdot d\zeta}, \qquad (7)$$

where the upper index $i \triangleright j$ symbolizes the application of an image warping function that projects the corresponding pixel coordinate or disparity from view $i$ to view $j$. Intuitively, for computing a disparity $\tilde{d}_i$, this filter does not only compute a convolution in the current view $i$ but it convolves all views with warped versions of the domain filter kernel $c$ and accumulates all values weighted by the range filter kernel $s$, as illustrated in Figure 5. Because the warping depends on unfiltered disparities, we iteratively apply the filter multiple times.

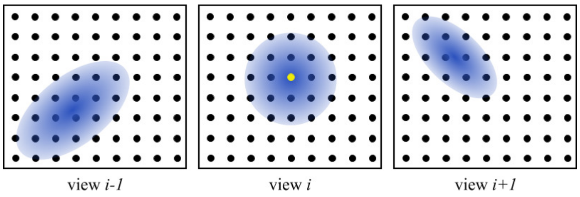

**Figure 5:** *Filtering over multiple views. For computing the value of the yellow pixel in view i, the values all pixels in all views weighted by the warped domain filter kernel (blue) are accumulated.*

For $c$ we use the function

$$c(\zeta, \mathbf{x}) = \alpha(\zeta) \cdot B(\zeta - \mathbf{x}), \qquad (8)$$

i.e. a cubic b-spline low-pass filter kernel $B$ weighted by the alpha $\alpha$ values of the current billboard. This weight ensures that the filter only accumulates points belonging to the current billboard. Moreover, it provides a local extension to our uniform error model by considering points at the surface boundary as less important because they are likely to be more inaccurate. The range filter kernel $s$ does not only maintain discontinuities in the disparity maps but also ensures a correct handling of occlusions that occur during warping. We use a simple step function.

In Figure 6, both filter equations (6) and (7) were applied for comparison. The raw displacements were calculated from disparity maps generated by a simple but fast window-based stereo algorithm without sub-pixel estimation. In contrast to sub-pixel stereo, which can also generate smooth displacements, our second filter can additionally correct for calibration errors. It can even outperform sub-pixel stereo as shown in Section 7.

The implementation of the filter process is done via splatting [Wes90]. For filtering a view $i$, instead of projecting the filter kernel into all other views, we do the inverse and splat all views into view $i$. This has the advantage that we can use the uniform splatting kernel $B$. The new disparities are then accumulated using the weights $\alpha$ and $s$. Splatting can be performed efficiently in the GPU [BHZK05].

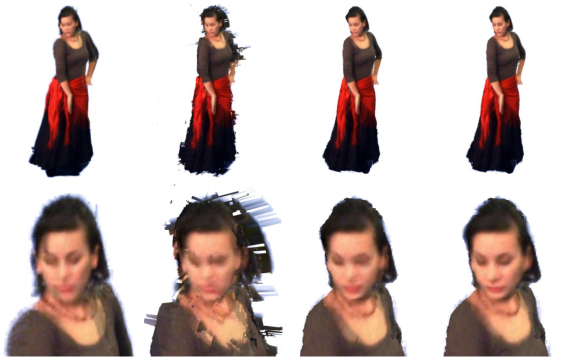

**Figure 6:** *Comparison of disparity filtering methods. Far left: all disparities set to zero. Middle left: unfiltered disparities. Middle right: disparities filtered using equation* (6)*. Far right: disparities filtered using equation* (7)*.*

## 5. View-Dependent Rendering and Blending

The billboards are directly rendered from the disparity space representation. Transformation into world coordinates is done during image generation in the GPU. We implemented a simple displacement mapping technique that stores a tesselated plane as vertex array and uses the CPU to set the $z$-coordinates of all vertices to the disparities. There also exist displacement mapping algorithms for the GPU [Don05, MJW07] that can be directly integrated in our framework.

Consistent images from multiple billboards from different views are generated by our view-dependent rendering approach. Each billboard $i$ is assigned a weight $w_i$ according to the unstructured Lumigraph framework [BBM*01] based on the position and orientation of its corresponding acquisition camera and of the current virtual camera. Thus, billboards closer to the current view have a larger impact on the image generation process.

In contrast to the original unstructured Lumigraph algorithm, our method does not only blend the colors but first reconstructs a consistent, view-dependent geometry of the scene, where each pixel has a uniquely assigned depth value. This results in much crisper renderings. If multiple fragments are rendered at the same pixel, its depth buffer value $d$ is computed in a fragment program by blending all fragment depths $d_i$:

$$d = \frac{1}{\sum_i w_i \alpha_i} \sum_i w_i \alpha_i d_i. \qquad (9)$$

The depths are additionally weighted with the values $\alpha_i$ from the alpha matte of the billboard because they are likely to be more inaccurate at the billboard boundary.

The pixel color $\mathbf{c}$ is assigned afterwards according to the new depth, using projective texturing. It is determined by alpha blending using

$$\mathbf{c} = \frac{1}{\sum_i w_i \alpha_i} \sum_i w_i \alpha_i \mathbf{c}_i. \qquad (10)$$

Furthermore, we compute the alpha value of the pixel as

$$\alpha = \max_i \alpha_i \qquad (11)$$

to ensure that the transparencies of the alpha matte are maintained such that the billboard cloud still smoothly blends with the background.

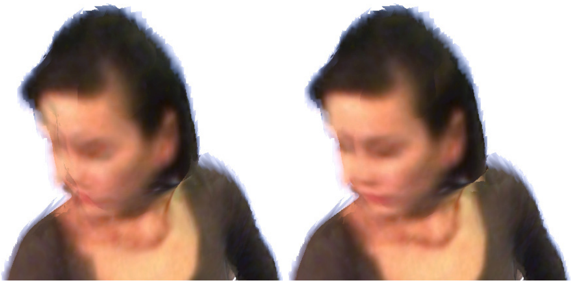

**Figure 7:** *Comparison between unstructured lumigraph blending of colors (left), and colors and depths (right).*

Due to the blending, occlusions cannot be handled using the conventional $z$ buffer algorithm. Instead of doing an expensive back-to-front rendering we implemented a fuzzy $z$-buffer [RPZ02] using two-pass rendering as an approximation.

## 6. Handling Scenes

To represent scenes and not only stand-alone objects, each view has to be decomposed into multiple billboards. We use a semi-automatic video cutout technique to segment the input videos into distinct objects. After the user has marked the objects in a single input frame of each view by applying a few brush strokes, a graph-cut optimization automatically computes the segments over time. The segment boundaries are refined by a Bayesian matting algorithm [CCSS01] yielding alpha mattes for the billboards.

We extend the graph cut segmentation of [WBC*05] by including also the available depth values into the computation. This makes the minimization more robust and needs less user input. Although similar the work by [WWG06], who used surface normals in the graph cut, it still performs the segmentation in image space and is thus much simpler to compute. Furthermore, using depth values is more robust than using normals in our case, considering the coarse input data.

Similarly to [WBC*05], the segmentation algorithm minimizes an energy

$$E(A,C,Z,\Gamma) = \sum_i D(a_i, \mathbf{c}_i, z_i, \gamma_i) + \sum_{(i,j)\in N} L(a_i, a_j, \mathbf{c}_i, \mathbf{c}_j) \qquad (12)$$

over a binary labeling vector $A$ using input colors $C$, depths $Z$, and a user-assigned labeling $\Gamma$, $\gamma_i \in \{F,B,U\}$ of foreground, background and unknown sections. For increasing speed and robustness, the minimization is not done over image pixels but hierarchically over a precomputed color segmentation of the video. In contrast to previous work, our data energy $D$ does not only consider mean segment colors $\mathbf{c}_i$ but also their median depth values $z_i$. Specifically, it uses for each component a negative log likelihood of a Gaussian mixture model. The smoothness energy $L$ describes the difference of neighboring segments $(i,j) \in N$ over space and time. We only use colors in $L$ because of the observation that edges in our input depth maps do not match exactly edges in the color images—a common problem of window-based stereo vision algorithms. Thus, considering depths would cause an inaccurate segmentation of the scene. As shown in Figure 8, inclusion of available depths increases the overall stability of the optimization and requires fewer input brush strokes from the user. As result, the user can quickly generate a segmentation of a complete scene as shown in Figure 9 with a few brush strokes.

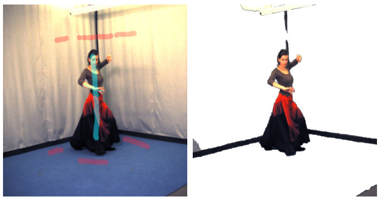

**Figure 8:** *Graph cut segmentation using the input image with user markings on the left. Results of the optimization considering colors only (middle) and additional depths (right).*

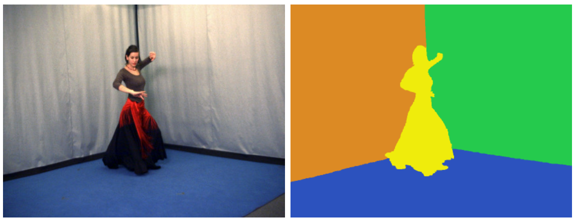

**Figure 9:** *Illustration of a segmentation (right) of a scene (left) into four billboard clouds.*

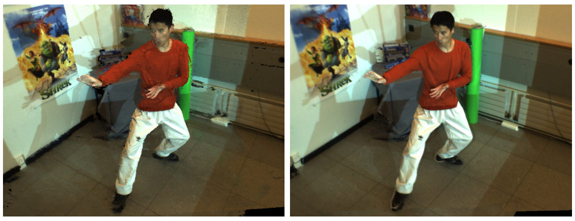

**Figure 10:** *Comparison of the visual quality of the point-based representation of [WWC*05] (left) with our method based on 3D video billboards (right).*

## 7. Results

For evaluating our 3D video billboard cloud representation we used the Teakwondo dataset of [WWC*05] and the Flamenco dataset of [WWG06]. Both were recorded using a system of so-called 3D video bricks that concurrently acquire color textures and structured light images from their respective viewpoints. Teakwondo was recorded at 10 fps with three bricks covering a viewing range of about 70°, Flamenco was captured at 12 fps with four bricks covering 90°. For Teakwondo the structured light projectors were additionally used as illumination source for the texture acquisitions which is visible as white projections and shadows in the scene background. We computed disparity maps using a stereo on structured light algorithm similar to the one described in [WWCG07] but without temporal correlation and sub-pixel estimation. Instead, we used our bilateral filter to get higher quality geometry. Figure 13 shows color and structured light images from one acquisition brick and the computed disparity map.

Figures 14 and 15 show both sequences rendered from novel views. In comparison to previous approaches we achieve better time coherence due to the spatio-temporal filtering. Furthermore, the alpha textures of the billboards smooth the appearance at object silhouettes and nicely blend the different segments of the scene. In comparison to the original point-based representation of those datasets, 3D video billboards yield a more detailed and crisper visual appearance of the renderings, as can be seen in Figure 10.

The proxy geometry of the billboards introduces the possibility of manual user control. The Flamenco dataset is composed of four billboard clouds: one for the actor, two for both walls and one for the floor. For the background billboard, the proxy planes are already a very good approximation to the real geometry. Thus, all disparities can be set to zero. In the Teakwondo dataset the background is much more complex. Nonetheless, because the background billboards are known to have a static geometry they can be stabilized by accumulating their displacements over time. This can be achieved by configuring the bilateral domain filter kernel to have a very long temporal extent.

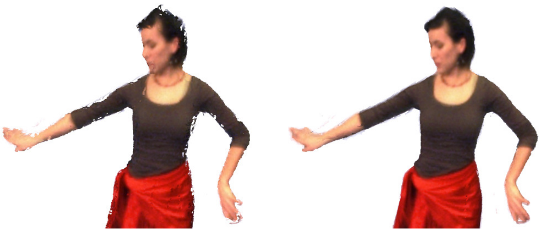

**Figure 11:** *Sub-pixel stereo vs. bilateral filtering. Left: displacement maps generated by sub-pixel stereo matching without filtering. Right: displacement maps generated by simple stereo matching with bilateral filtering.*

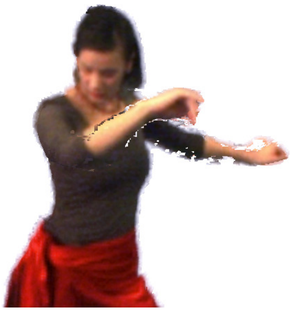

**Figure 12:** *Thin, fast moving structures like the actor's arms cannot provide sufficient spatio-temporal support for high-quality filtering.*

Our filtering framework can also be used as an alternative to time consuming sub-pixel stereo reconstruction algorithms. In the left part of Figure 11 the disparity maps for the four billboard planes have been generated again with the pipeline of [WWCG07], including sub-pixel matching. In contrast, the right part of Figure 11 has been reconstructed with our approach using simple window-based stereo matching with successive bilateral filtering. There is not only the temporal advantage of 1.5 minutes for our approach vs. 28 minutes for the sub-pixel method, but there also is an apparent gain in visual appearance. This is due to a better numerical stability of our filter versus the complex minimization algorithm of the sub-pixel matching, especially at surface boundaries. Moreover, our approach is in principle capable of real-time reconstructions, because efficient GPU implementations for both window-based stereo [YWY*06] and splatting [BHZK05] are available.

A current limitation of our method is shown Figure 12. Thin, fast moving structures in the video like the arms or legs of an actor are difficult to handle. In such cases, the domain kernel of the bilateral filter does not have enough support by the data, neither in space nor in time. As a result, smoothing of those structures is not as good as in other surface regions.

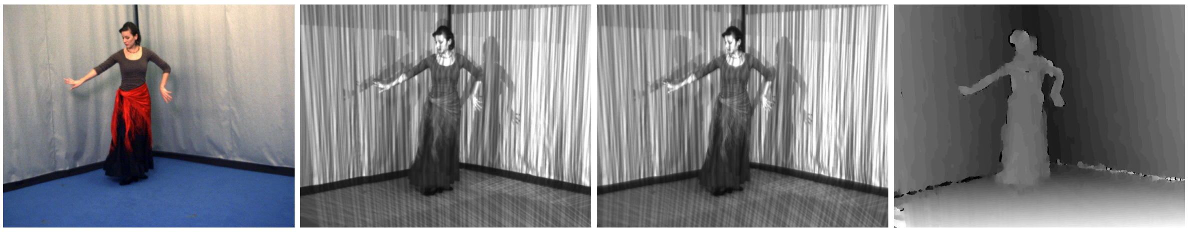

**Figure 13:** *Input color and structured light images of the Flamenco dataset [WWG06], and the resulting disparity map reconstructed by stereo vision on structured light without sub-pixel estimation.*

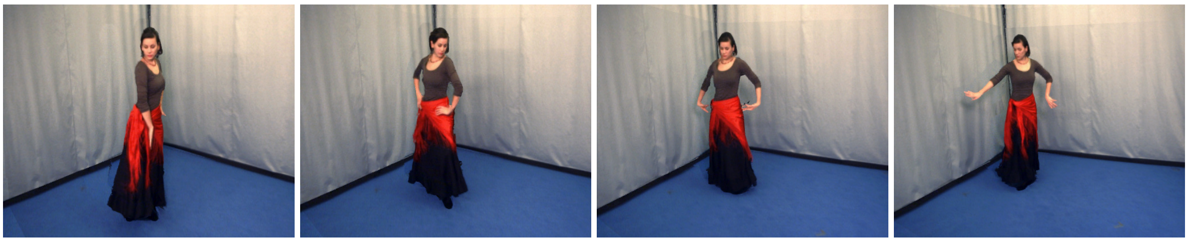

**Figure 14:** *Image sequence of the Flamenco dataset rendered from novel views.*

## 8. Conclusion and Future Work

We introduced 3D video billboard clouds as an enabling representation for 3D video applications. We construct both the proxy billboard planes and displacement maps from depth images of the scene acquired by a standard stereo matching algorithm. They provide a regular, uniform sampling of the scene in space and time which makes it suitable for standard signal processing methods. We apply a four-dimensional bilateral filter to achieve geometry with higher spatial and temporal coherence. We render novel views using a GPU-accelerated method which also generates consistent view-dependent geometry and textures for each individual frame. Modeling dynamic three-dimensional scenes is performed using a semi-automatic approach which generates a collection of 3D video billboard clouds.

Possible extensions include fully automatic segmentation of complex scenes. This may be implemented by searching for planar patches in the input geometry using a 3D Hough transform similar to [DDSD03]. In contrast to the current system, this might lead to an over-segmentation that decomposes the scene into many very small billboards. Moreover, we would like to look for efficient compression methods for our representation and explore its potential for level-of-detail rendering.

## Acknowledgements

We would like to thank Vanessa Stadler and Doo Young Kwon for acting in the 3D video, and Daniel Cotting for fruitful discussions. This work is carried out in the context of the blue-c-II project, funded by ETH grant No. 0-21020-04 as an internal poly-project.

## References

[BBM*01] Buehler C., Bosse M., McMillan L., Gortler S., Cohen M.: Unstructured lumigraph rendering. In *SIGGRAPH '01* (2001), pp. 425–432.

[BHZK05] Botsch M., Hornung A., Zwicker M., Kobbelt L.: High-quality surface splatting on today's gpus. In *Eurographics Symposium on Point-Based Graphics '05* (2005), pp. 17–24.

[CCSS01] Chuang Y.-Y., Curless B., Salesin D. H., Szeliski R.: A bayesian approach to digital matting. In *CVPR '01* (2001), pp. 264–271.

[CTMS03] Carranza J., Theobalt C., Magnor M., Seidel H.-P.: Free-viewpoint video of human actors. In *SIGGRAPH '03* (2003), pp. 569–577.

[DDSD03] Décoret X., Durand F., Sillion F., Dorsey J.: Billboard clouds for extreme model simplification. In *SIGGRAPH '03* (2003), pp. 689–696.

[Don05] Donnelly W.: Per-pixel displacement mapping with distance functions. In *GPU Gems 2*, Pharr M., (Ed.). Addison Wesley, Mar. 2005, ch. 8, pp. 123–136.

[DRR03] Davis J., Ramamoorthi R., Rusinkiewicz S.: Spacetime stereo: A unifying framework for depth from triangulation. In *CVPR '03* (2003), pp. 359–366.

[LH96] Levoy M., Hanrahan P.: Light field rendering. In *SIGGRAPH '96* (1996), pp. 31–42.

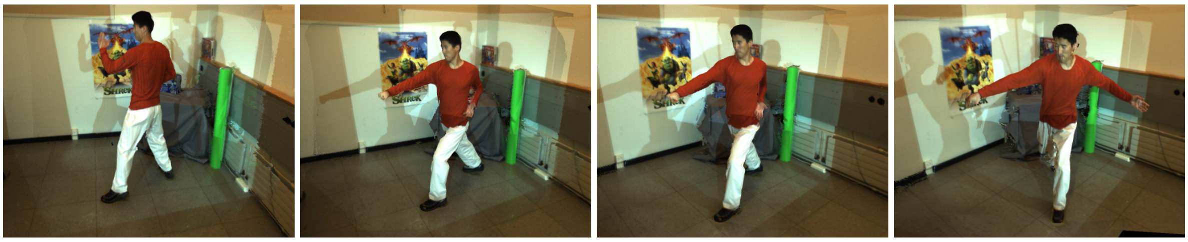

**Figure 15:** *Image sequence of the Teakwondo dataset rendered from novel views.*

[LSS05] Li Y., Sun J., Shum H.-Y.: Video object cut and paste. In *SIGGRAPH '05* (2005), pp. 595–600.

[MBM01] Matusik W., Buehler C., McMillan L.: Polyhedral visual hulls for real-time rendering. In *Eurographics Workshop on Rendering '01* (2001), pp. 115–125.

[MBR*00] Matusik W., Buehler C., Raskar R., Gortler S. J., McMillan L.: Image-based visual hulls. In *SIGGRAPH '00* (2000), pp. 369–374.

[MJW07] Mantler S., Jeschke S., Wimmer M.: *Displacement Mapped Billboard Clouds*. Tech. Rep. TR-186-2-07-01, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria, Jan. 2007.

[MP04] Matusik W., Pfister H.: 3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *SIGGRAPH '04* (2004), pp. 814–824.

[RNK97] Rander P., Narayanan P., Kanade T.: Virtualized reality: Constructing time-varying virtual worlds from real events. In *IEEE Visualization '97* (1997), pp. 277–283.

[RPZ02] Ren L., Pfister H., Zwicker M.: Object space EWA surface splatting: A hardware accelerated approach to high quality point rendering. *Computer Graphics Forum 21*, 3 (2002), 461–470.

[SGHS98] Shade J., Gortler S., He L.-W., Szeliski R.: Layered depth images. In *SIGGRAPH '98* (1998), pp. 231–242.

[SS02] Scharstein D., Szeliski R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision 47*, 1-3 (2002), 7–42.

[VBK02] Vedula S., Baker S., Kanade T.: Spatio-temporal view interpolation. In *Eurographics Workshop on Rendering '02* (2002), pp. 65–76.

[WBC*05] Wang J., Bhat P., Colburn A., Agrawala M., , Cohen M.: Interactive video cutout. In *SIGGRAPH '05* (2005), pp. 585–594.

[Wes90] Westover L.: Footprint evaluation for volume rendering. In *SIGGRAPH '90* (1990), pp. 367–376.

[WJV*05] Wilburn B., Joshi N., Vaish V., Talvala E.-V., Antunez E., Barth A., Adams A., Horowitz M., Levoy M.: High performance imaging using large camera arrays. In *SIGGRAPH '05* (2005), pp. 765–776.

[WLSG02] Würmlin S., Lamboray E., Staadt O. G., Gross M. H.: 3D video recorder. In *Pacific Graphics '02* (2002), pp. 325–334.

[WWC*05] Waschbüsch M., Würmlin S., Cotting D., Sadlo F., Gross M.: Scalable 3D video of dynamic scenes. *The Visual Computer 21*, 8–10 (2005), 629–638.

[WWCG07] Waschbüsch M., Würmlin S., Cotting D., Gross M.: Point-sampled 3D video of real-world scenes. *Signal Processing: Image Communication 22*, 203–216 (2007).

[WWG06] Waschbüsch M., Würmlin S., , Gross M.: Interactive 3D video editing. *The Visual Computer 22*, 9–11 (2006), 631–641.

[YEBM02] Yang J. C., Everett M., Buehler C., McMillan L.: A real-time distributed light field camera. In *Eurographics Workshop on Rendering '02* (2002), pp. 77–86.

[YWY*06] Yang Q., Wang L., Yang R., Wang S., Liao M., Nistér D.: Real-time global stereo matching using hierarchical belief propagation. In *BMVC '06* (2006).

[ZCS03] Zhang L., Curless B., Seitz S. M.: Spacetime stereo: Shape recovery for dynamic scenes. In *CVPR '03* (2003), pp. 367–374.

[ZKU*04] Zitnick C. L., Kang S. B., Uyttendaele M., Winder S., Szeliski R.: High-quality video view interpolation using a layered representation. In *SIGGRAPH '04* (2004), pp. 600–608.