Efficient 3D Object Segmentation from Densely Sampled Light Fields with Applications to 3D Reconstruction

KAAN YÜCER
ETH Zurich and Disney Research Zurich
ALEXANDER SORKINE-HORNUNG and OLIVER WANG
Disney Research Zurich
and
OLGA SORKINE-HORNUNG
ETH Zurich

Precise object segmentation in image data is a fundamental problem with various applications, including 3D object reconstruction. We present an efficient algorithm to automatically segment a static foreground object from highly cluttered background in light fields. A key insight and contribution of our paper is that a significant increase of the available input data can enable the design of novel, highly efficient approaches. In particular, the central idea of our method is to exploit high spatio-angular sampling on the order of thousands of input frames, e.g. captured as a hand-held video, such that new structures are revealed due to the increased coherence in the data. We first show how purely local gradient information contained in slices of such a dense light field can be combined with information about the camera trajectory to make efficient estimates of the foreground and background. These estimates are then propagated to textureless regions using edge-aware filtering in the epipolar volume. Finally, we enforce global consistency in a gathering step to derive a precise object segmentation both in 2D and 3D space, which captures fine geometric details even in very cluttered scenes. The design of each of these steps is motivated by efficiency and scalability, allowing us to handle large, real-world video datasets on a standard desktop computer. We demonstrate how the results of our method can be used for considerably improving the speed and quality of image-based 3D reconstruction algorithms, and we compare our results to state-of-the-art segmentation and multi-view stereo methods.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Image and Video Processing, Modeling

Additional Key Words and Phrases: 3D object and image segmentation, image-based reconstruction, light field processing, visual hulls

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

1. INTRODUCTION

Automatically segmenting a foreground object from the background in image data is a long-standing and important problem in computer graphics and vision. One common application is the use of segmentation in the context of image-based modeling, e.g., computing geometry proxies, such as visual hulls, for image-based rendering, or limiting the search space and increasing reconstruction quality in multi-view stereo techniques. However, producing such segmentations can be cumbersome and time-consuming; typically, green-screens or other known backgrounds are used to facilitate the process, and manual correction is often required. In the case of cluttered scenes with no a priori knowledge about the background or the camera motion, segmenting images into fore- and background layers becomes very challenging. In this paper, we show how densely captured video sequences of static objects can be efficiently segmented via a joint 2D-3D procedure that requires only a simple yet effective assumption about the depth distribution of the scene as input, and is otherwise fully automatic.

Segmentation of a video sequence can be formulated as separating the pixels that belong to the foreground object from the pixels that belong to the background. Given a static foreground object, the 2D segmentation problem can be lifted into 3D, where it can be interpreted as the estimation of a 3D occupancy volume. Recently, a number of methods for computing a joint 2D-3D segmentation have been proposed (discussed in Section 2), which borrow ideas from correspondence-based 3D scene reconstruction techniques and combine them with color-based segmentation methods. However, when the input images are taken from rather sparsely sampled viewpoints, which is the case for most existing joint segmentation and 3D reconstruction techniques, the lack of coherence between the images due to (self-) occlusions and abrupt changes in parallax make it difficult to reason about complex, detailed object shapes. To compensate for ambiguities in the required estimation of image correspondences and color models, these methods generally have to resort to complex global optimization procedures, which are difficult to scale to higher image- and 3D-resolution.

In contrast, *smooth* parallax changes between successive images make distinguishing between foreground and background simpler, in particular for complex and detailed objects, since the differences in motion parallax at different depths become apparent (just like moving one's head side to side yields a clear depth ordering). Video capture devices are now commonplace, and acquiring appropriate, dense data has become trivial with standard consumer tools. However, most joint segmentation and reconstruction techniques are not

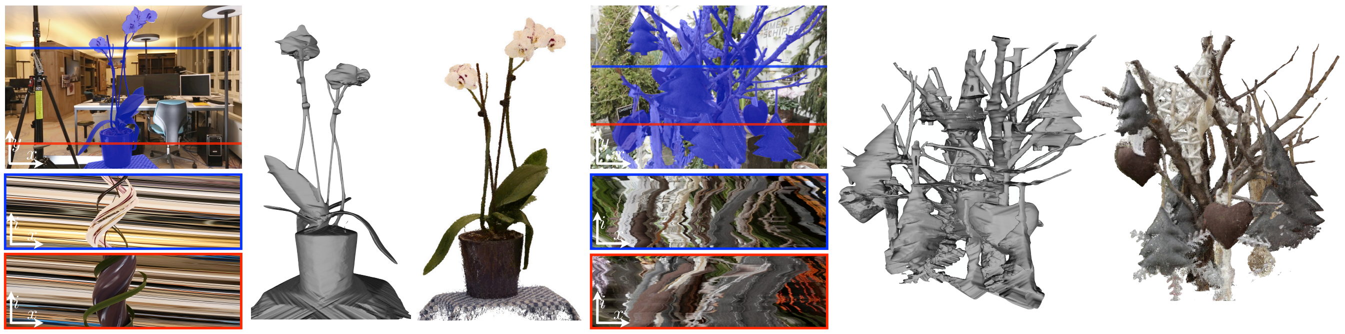
Fig. 1: Example results of our method captured with a rotating (left) and a hand-held (right) camera. Slices of densely sampled light fields reveal continuous structures that arise as a result of motion parallax (blue and red rectangles). Our method leverages the coherency in this data to accurately segment the foreground object in both 2D (blue image mask) and 3D (mesh), which can be used to aid 3D reconstruction algorithms (colored point clouds).

designed to handle these large volumes of data (on the order of thousands of frames), and as a result cannot take advantage of the extra information available. Another fundamental issue with dense light field data arise when using triangulation-based methods for computing correspondences. Small baselines between image pairs lead to low-quality occupancy and shape estimates in 2D as well as 3D. As a result, most existing approaches apply some form of view selection to remove images with small baselines in order to improve accuracy (e.g., [Furukawa et al. 2010]). However, this leads to loss of valuable information about the scene and reintroduces the same correspondence ambiguities mentioned earlier. In contrast, we propose a computationally efficient and *scalable* method where all steps in our pipeline are designed in such a way that they can take advantage of the inherent coherency in dense datasets. As a result, our method can segment thin and complex foreground geometry even with uncontrolled, cluttered backgrounds.

As input, we take densely sampled continuous image sequences, (e.g., video) that observe an object from different angles. Our algorithm is comprised of the following efficient building blocks: first, we create a light field volume by stacking the input images on top of each other. This representation clearly reveals the motion parallax of even single pixels in 2D slices of the volume (see Figure 1), similar to epipolar-plane images produced by certain linear camera motions [Criminisi et al. 2005]. We compute gradients in the light field volume and estimate the likelihood of image edges belonging to the foreground using local gradient filters. Since this step creates reliable estimates only on strong image edges, we use the coherence in the light field and propagate this information to the rest of the image volume using edge-aware filtering. In a final step, the per-image segmentation estimates are aggregated into a single, consistent 3D volume using a Bayesian framework. The result of our complete algorithm is a 3D object probability volume, which can be thresholded to obtain the object segmentation in 3D space. At this point, consistent per-image segmentations can then be extracted by projecting this volume back onto the images.

We show that our approach efficiently and automatically produces precise segmentation results for complex, detailed object shapes in cluttered environments, and these segmentations can be used to improve the quality of existing 3D reconstruction methods. Our method works robustly with different capture scenarios: we demonstrate results on data captured with an uncalibrated circular camera motion, a linear stage, and various hand-held video sequences.

2. RELATED WORK

In the following, we discuss previous works from different areas that are most related to our proposed approach.

Image and video segmentation. Object segmentation from images and videos usually works by learning color models from user input or motion cues. Techniques based on graph-cuts [Boykov and Jolly 2001; Rother et al. 2004] compute single-image segmentations using discrete graph-based optimization; they require input constraints (e.g., scribbles or bounding boxes) that indicate some amount of foreground or background pixels. Learning-based methods exploit similarities between images for co-segmentation [Joulin et al. 2010] or multi-class image segmentation [Krähenbühl and Koltun 2012], but they require training and are not designed to efficiently handle densely sampled input.

For more densely sampled input data, such as video, recent interactive methods use graph-based algorithms on the entire video domain to find the foreground/background boundary [Li et al. 2005; Wang et al. 2005; Grundmann et al. 2010]. While these methods work well in general video-editing applications, they are less suited for creating a consistent segmentation in the presence of complex occlusions and disocclusions. Other methods compute a dense optical flow field to propagate user constraints between different frames in order to segment videos in a temporally coherent manner [Chuang et al. 2002; Lang et al. 2012]. Our method also works on images taken from densely sampled trajectories around the object, but it is able to compute segmentations *without* the user input required by those methods. Recent unsupervised approaches for video segmentation make use of the motion cues in the video volume [Apostoloff and Fitzgibbon 2006; Lezama et al. 2011], but they cannot work with static scenes. These approaches are also prone to cutting away pieces of objects, rendering them impractical for our scenario.

Visual hulls. The intersection of multiple 2D segmentations of different views of an object in 3D space is known as the visual hull of the object [Martin and Aggarwal 1983; Szeliski 1993; Laurentini 1994]. Many techniques have been proposed to robustly compute such hulls, ranging from methods working in 2D image space [Matusik et al. 2000] to 3D voxel-based representations [Snow et al. 2000], with applications to, e.g., image-based rendering using the visual hull as a geometry proxy. More recent work has focused on increasing robustness, e.g., using probabilistic approaches to decrease the influence of errors in silhouette extraction and image calibration [Grauman et al. 2003; Franco and Boyer 2005; Tabb 2013]. All the above methods assume that an existing, often manually generated, precise segmentation of the object in the input images, or that it can be robustly extracted using background subtraction or other color priors. In contrast, we automatically compute a precise object segmentation in cluttered scenes, without prior knowledge about the background or scene structure.

Joint 2D-3D segmentation. Our method is most closely related to 2D-3D co-segmentation methods, which make use of camera calibrations in conjunction with color information for segmenting an object in 3D space. Such techniques usually employ color models, depth hypotheses and probabilistic measures in order to segment the 3D object and its projection in the 2D images. Yezzi et al. [2001], for example, use a variational framework to compute segmentations, which works mostly for objects without texture. Several methods use graph-cuts in voxel space [Campbell et al. 2010] and image space [Campbell et al. 2011] to segment an object in multiple views by assuming that all principle camera axes intersect the foreground object, from which a color model can be learned. As the number of images grows, the computation time of graph-cuts become intractable. Computing depth maps beforehand to help with the segmentation has also been studied [Kowdle et al. 2012], but computing depth maps for thousands of images is inefficient. The most similar works to ours are by Kolev et al. [2006], who compute a 3D surface probabilistically in a voxel grid using color models, and by Lee et al. [2011], who make similar computations in image space by leveraging epipolar geometry. However, these methods also have a very high computational complexity due to their iterative nature, and report several minutes for 8 images of 640×480 pixels [Lee et al. 2011], or 20 to 30 minutes on 20 images of the same size using a 128^3 voxel grid [Kolev et al. 2006]. Our method works by extracting simple depth relationships instead of explicit depth computations, and efficiently leverages datasets consisting of thousands of images.

Other methods try to solve the segmentation and 3D reconstruction problems in a joint framework. Goldlücke and Magnor [2003] compute segmentation and depth labeling together in a single optimization, solved by graph-cuts. Another method that makes use of graph-cuts [Guillemaut and Hilton 2011] minimizes an energy functional that includes color models for foreground and background layers, the 3D shape of the foreground object and some smoothness priors on the object shape. However, these methods assume knowledge about the background, such as an image of the static background or a keyframe for each camera, where each background layer is labeled by a user. Häne et al. [2013] use appearance-based cues together with 3D surface orientation priors to segment objects belonging to different classes while reconstructing their shape, but require rigorous training to learn such priors for different object classes. In our work, we do not assume any knowledge about the appearance of the background and do not require training phases, yet can still effectively process the given data.

Sparse data interpolation. One component of our algorithm is based on the propagation of local object estimates in high-gradient areas to less textured regions, which can be formulated as a sparse data interpolation or regularization problem. Optimization-based approaches can make globally optimal decisions [An and Pellacini 2008], but they are computationally expensive for large datasets. When image structure is known, joint-filtering methods have been used for propagating sparse information in many applications [Eisemann and Durand 2004; Petschnigg et al. 2004; Kopf et al. 2007] thanks to their computational efficiency and quality of results. Still, efficient bilateral filtering implementations do not scale well with increased dimensionality. Instead, we use a separable joint-geodesic filter [Gastal and Oliveira 2011], which scales linearly with dimensionality. A complete review of this topic is out of the scope of this paper; we refer the reader to [Paris et al. 2007] for a detailed overview.

Light fields. Fundamental concepts of dense light field capture and rendering have been introduced in the seminal works of Levoy et al. [1996] and Gortler et al. [1996], and have since then been extended to unstructured setups [Buehler et al. 2001; Davis et al. 2012]. A number of works have investigated approaches to exploit light field data, such as recovering shape from silhouettes [Kutulakos 1997], view interpolation [Berent and Dragotti 2007] and, in particular, the analysis of 2D light field slices for 3D object reconstruction [Bolles et al. 1987; Criminisi et al. 2005; Wanner and Goldluecke 2012; Kim et al. 2013; Wanner et al. 2013; Yu et al. 2013]. These works show that the increased coherence in image data with high spatio-angular sampling enables robust depth estimates from local information. Chen et al. [2014] recently showed that the use of image statistics can improve depth estimation near occlusion boundaries. Our work is inspired by these methods and shows how the coherence in light field data can be further exploited to provide a novel solution to automatic joint 2D-3D object segmentation. Moreover, Feldmann et al. [2003a; 2003b] proposed techniques for circular light fields that make use of cylindrical plane sweeping. However, these methods require specific capture scenarios (i.e., perfect circular motion) and cannot be generalized to hand-held video data.

Applications. 3D object segmentation has applications in various areas of computer graphics. A straightforward application is to use the resulting 3D shape as a geometry proxy for image-based rendering and image understanding [Szeliski 1993; Laurentini 1994; Buehler et al. 2001; Eisemann et al. 2008]. In particular, for rendering applications, a faithful representation of fine silhouette detail is essential. Similarly, numerous image-based 3D reconstruction algorithms benefit from or even require object segmentation as a preprocess. Some methods use image segmentations as a starting solution for iterative surface refinement, where a visual hull is first computed by intersecting given image segmentations, and is then carved and optimized using photoconsistency measures and silhouette or contour constraints [Isidoro and Sclaroff 2003; Sinha and Pollefeys 2005; Vogiatzis et al. 2005; Starck et al. 2006; Furukawa and Ponce 2009]. Works that do not explicitly require segmentations [Furukawa and Ponce 2010] also argue that visual hulls increase the quality of results when used as a starting solution. Such segmentations are also useful in estimating and constraining surface patches to lie inside a 3D bounding volume to increase their reliability [Bowen et al. 2007]. Constraining stereo matching to segmented image parts and/or to the interior of visual hulls [Bradley et al. 2008; Oswald and Cremers 2013] has been shown to improve accuracy of the reconstructed objects and speed up the techniques. Methods for reconstructing meshes from oriented point clouds [Shan et al. 2014] can also benefit from segmentations in 3D, where the empty space is used to constrain the final mesh.

With our algorithm, precise, detailed segmentations can be generated automatically for challenging object shapes, which can then be effectively used to increase the accuracy or performance of such methods, for example when merging individual depth maps or providing an initial geometry and visibility proxy. We show how the results of two state-of-the-art stereo techniques [Hirschmüller 2006; Kim et al. 2013] can be improved using our 2D-3D segmentations, and we compare to additional state-of-the-art 3D reconstruction approaches.

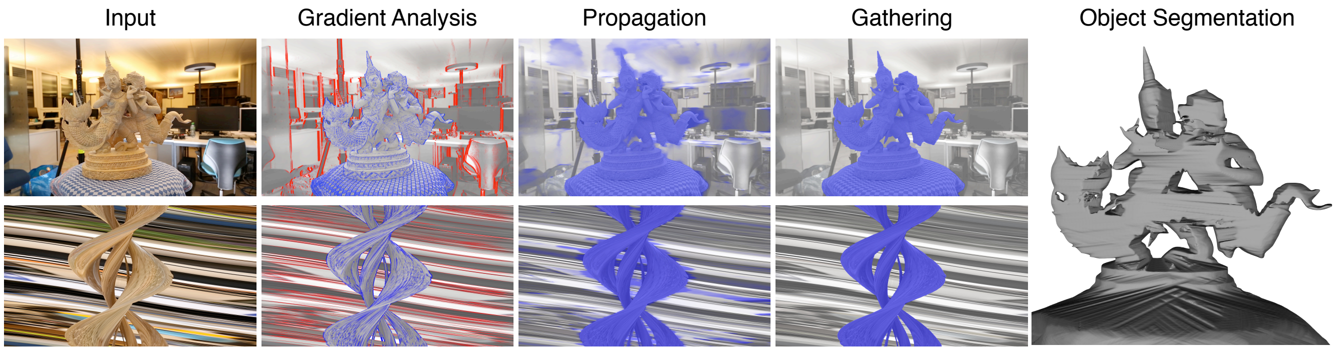

Fig. 2: Components of our local-to-global strategy, showing images in top row and x-i slices in bottom row. From left to right: Input data, local gradient estimates $S_i(\mathbf{p})$ weighted by confidence measures $C_i(\mathbf{p})$ (foreground (S_i) as blue, background $(1 - S_i)$ as red, and confidences (C_i) determine the transparency; see Section 3.1), edge-aware light field filtering (Section 3.2), global gathering (Section 3.3), and the estimated object segmentation in 3D from a slightly different viewpoint. The images were desaturated for improved segmentation visualization. Note how consistency increases at each step of the pipeline.

3. METHOD

Our goal is to segment the foreground object in 2D and 3D from a light field that is represented by a set of images $(I_1, \ldots, I_n)$ (where n is in the order of thousands) with known calibration for each image I_i in the form of a projection matrix $\mathbf{P}_i$. We assume that the images were taken by a camera on a continuous, but otherwise arbitrary trajectory with respect to the captured scene. From the input images, we construct a 3D light field volume L, where $L(x, y, i)$ refers to the pixel $\mathbf{p} = (x, y)$ in image I_i.

Key to our method is the dense spatio-angular sampling of video, which results in smoothly varying parallax between successive frames. The continuous changes intuitively encode the motion parallax (i.e., relative depth) of scene points as differently shaped curves or “traces” in the light field L. These curves live on a 2D manifold in L, which is described by the epipolar geometry between the images. Our input data is comprised of thousands of input images, therefore computational efficiency and tractability are among the key driving factors behind the following algorithm steps.

Our method follows a local-to-global strategy. We first compute a sparse segmentation estimate based on local gradient information (Section 3.1). This segmentation is propagated to the light field volume using edge-aware filtering on L, yielding $S_i : I_i \to [0, 1]$ for each image I_i (Section 3.2). The function S_i describes, for each pixel of I_i, the likelihood of observing the foreground object.

We then gather the individual segmentations using a Bayesian formulation into an object likelihood function H defined over a discretization $\mathcal{V}$ of the 3D volume containing the object. Each voxel $\mathbf{v} \in \mathcal{V}$ is assigned an accumulated likelihood value for being part of the foreground object, aggregated from all the values $\{S_i(\mathbf{p}) : \mathbf{p} = \mathbf{P}_i\,\mathbf{v}\}$ (Section 3.3). This final global gathering step further enhances the per-image segmentations by removing noisy estimates that do not have the support of multiple views, and by enforcing geometric constraints, such that when the values of H are projected back onto the images, the resulting segmentations S_i are globally consistent throughout the light field. This process is illustrated in Figure 2.

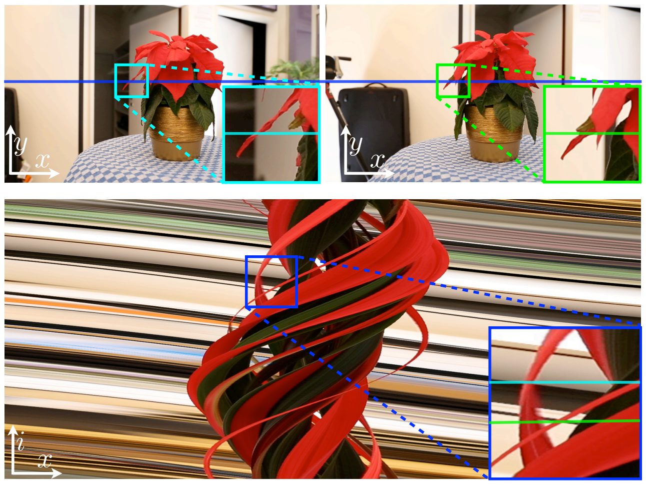

Fig. 3: Top: two frames from the PLANT dataset with close-ups around the blue scanline. Bottom: The corresponding x-i slice; the parts corresponding to the images are highlighted in matching colors. Note how the direction in the slice varies according to the position of the object in 3D space relative to the camera.

3.1 Gradient-based analysis of motion parallax

The fundamental observation to our technique is that scene points follow smooth trajectories inside a light field L with high spatio-angular sampling, which differ substantially for points at different depth values. Even though these trajectories can take any shape depending on the actual camera path, they remain easily distinguishable in their shapes, even in case of hand-held capture scenarios. For illustration purposes, consider the example of a nearly circular camera path: an object located in the center leaves spiral trails in the light field L, whereas background objects' trails move from one end to the other along an almost linear curve (see a 2D x-i slice of a light field L in Figure 3). These clearly visible structures contained in the light fields essentially correspond to 3D trajectories that arise from motion parallax. Pixels that correspond to the background have a specific distribution of trajectory directions that is detectably different from pixels in the foreground. We leverage this fact to compute the foreground object likelihood for each pixel separately.

For each pixel $\mathbf{p} = (x, y) \in I_i$, we consider the direction $\gamma^{\mathbf{p}}$ corresponding to its trajectory inside L. We model the distribution of the foreground trajectory directions by a Gaussian $N_f^{\mathbf{p}} = N(\gamma_f^{\mathbf{p}}, \sigma_f^{\mathbf{p}})$, and background directions by $N_b^{\mathbf{p}} = N(\gamma_b^{\mathbf{p}}, \sigma_b^{\mathbf{p}})$ around each pixel $\mathbf{p} \in I_i$. We note that the expected trajectory direction is a function of 2D pixel position, depth, and camera motion (see close-ups in Figure 3). For instance, pixel trajectories follow *helical* motions as a function of their depths for circular camera motions [Feldmann et al. 2003a], whereas they form lines of different slope for linear light fields, and more irregular paths for shaky hand-captured video. Our approach does not require the specific set of trajectories be known a priori, but computes them on-demand from camera calibrations.

To compute directions $\gamma_b^{\mathbf{p}}$ and $\gamma_f^{\mathbf{p}}$, we need the expected trajectories for $\mathbf{p}$ inside L. As computing them exactly is a chicken-and-egg problem (the depth determines the trajectory), we instead roughly estimate the directions using two simple proxy geometries, placed at approximate locations of the foreground object and the background. In our experiments, we found that planar proxies were sufficient to robustly model the trajectory directions. We place two planes into the scene representing the foreground and the background objects. The distance R from the camera to the foreground object is set approximately by the user by marking the center of the 3D object. The background distance is then chosen to be $3R$. We can then project $\mathbf{p}$ onto these planes and back onto image I_{i+1}. The expected coordinates in I_{i+1} are denoted by $\mathbf{p}_f = (x_f, y_f)$ if $\mathbf{p}$ belongs to foreground and $\mathbf{p}_b = (x_b, y_b)$ if $\mathbf{p}$ belongs to the background. Note that any other geometric proxy can be used for this estimation; we chose planes due to their simplicity and applicability in various capture scenarios. We found in our experiments that our method is robust to the placement of these geometric proxies: for example, changing the distance of the background proxy between $2R$ and $4R$ did not affect the result quality.

Now that the expected coordinates for the foreground and background proxies are determined, we can estimate the trajectory directions between frames using epipolar geometry. We know that $\mathbf{p}_f$ and $\mathbf{p}_b$ lie on the epipolar line $\mathbf{e}$ in I_{i+1}, which maps $\mathbf{p}$ to I_{i+1}. We also know that the actual scene point at $\mathbf{p}$ will appear on $\mathbf{e}$ in I_{i+1} as well. Hence, we can sample the color values in I_{i+1} along $\mathbf{e}$ to generate the light field cut $s_{i+1}^{\mathbf{p}}$, which is guaranteed to contain the actual mapping of $\mathbf{p}$. Given that the camera motion is small enough, the scene points on $s_{i+1}^{\mathbf{p}}$ will appear on a line with the same direction around $\mathbf{p}$ in I_i, so we generate a cut on I_i along the direction of $\mathbf{e}$ through $\mathbf{p}$, resulting in $s_i^{\mathbf{p}}$. Stacking $s_i^{\mathbf{p}}$ and $s_{i+1}^{\mathbf{p}}$ horizontally, denoted as $s^{\mathbf{p}}$, reveals the trajectory of $\mathbf{p}$ as a motion trail. Specifically, the trajectory direction $\gamma^{\mathbf{p}}$ is orthogonal to the color gradient direction in $s^{\mathbf{p}}$. The expected foreground direction $\gamma_f^{\mathbf{p}}$ can be computed as the direction of the line connecting the coordinates of $\mathbf{p}$ and $\mathbf{p}_f$ on $s^{\mathbf{p}}$. The same method is used for $\gamma_b^{\mathbf{p}}$. Since we only need local information to compute the directions, we only generate the cuts $s_i^{\mathbf{p}}$ and $s_{i+1}^{\mathbf{p}}$ where necessary, i.e., $s_i^{\mathbf{p}}$ is generated around $\mathbf{p}$ using a 7-pixel window, and $s_{i+1}^{\mathbf{p}}$ around $\mathbf{p}'$, which is a point between $\mathbf{p}_f$ and $\mathbf{p}_b$, with the same size window (see Figure 4). In practice, we compute the position of $\mathbf{p}'$ using a separating proxy between the foreground object and the background, such as a cylinder or a plane. We project $\mathbf{p}$ to this proxy and back to I_{i+1}, resulting in $\mathbf{p}'$. Note that this computation is a generalization of checking slopes in linear light fields.

In our experiments, we set the background variance to be fairly narrow and the foreground variance relatively large:

$$\sigma_b^{\mathbf{p}} = (\gamma_f^{\mathbf{p}} - \gamma_b^{\mathbf{p}})/10, \qquad \sigma_f^{\mathbf{p}} = 3\,\sigma_b^{\mathbf{p}}, \tag{1}$$

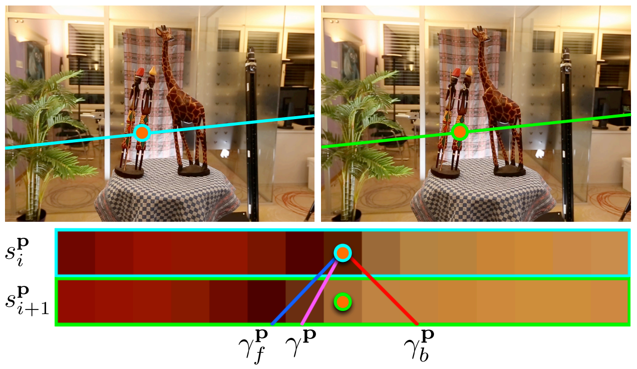

Fig. 4: Top: Two frames from the AFRICA dataset. The orange dots are $\mathbf{p}$ and $\mathbf{p}'$, the green line is the epipolar line $\mathbf{e}$ corresponding to $\mathbf{p}$, and the cyan line is a line through $\mathbf{p}$ with the same direction as $\mathbf{e}$. Bottom: Light field cuts $s_i^{\mathbf{p}}$ and $s_{i+1}^{\mathbf{p}}$ sampled around $\mathbf{p}$ and $\mathbf{p}'$. The expected foreground direction $\gamma_f^{\mathbf{p}}$ is shown in blue, and the background direction $\gamma_b^{\mathbf{p}}$ in red. The actual gradient direction $\gamma^{\mathbf{p}}$ (magenta) is closer to the foreground direction, resulting in a foreground estimate.

because the foreground trajectories have stronger variations in parallax and therefore in trajectory direction. The likelihood $S_i(\mathbf{p})$ of the pixel $\mathbf{p}$ to belong to the foreground can then be estimated using Bayes' theorem:

$$S_i(\mathbf{p}) = \frac{N_f^{\mathbf{p}}(\gamma^{\mathbf{p}})}{N_f^{\mathbf{p}}(\gamma^{\mathbf{p}}) + N_b^{\mathbf{p}}(\gamma^{\mathbf{p}})} . \tag{2}$$

Due to the absence of prior knowledge about the scene geometry, we set the corresponding priors to 0.5.

In order to get robust estimates of motion parallax, we used a window of 5 frames for doing the gradient analysis. Since the gradient computations are done independently for each pixel, this step can be easily parallelized on the CPU or the GPU.

In the special case of regular camera motion, when the relative motion between two consecutive frames remains the same (i.e., linear or circular light fields), $\gamma_f^{\mathbf{p}}$ and $\gamma_b^{\mathbf{p}}$ do not depend on i but only on x and y. In such cases, we precompute and reuse these values, resulting in noticable speed-ups.

Confidence measures. The described foreground/background estimation method is only reliable in high-gradient regions and gives noisy estimates in smooth, homogeneous parts of the light field. Additionally, lines that exist in the input images can produce ambiguities when computing trajectory directions if they are oriented parallel to the motion of the camera, since the resulting trajectory directions in the slices are similar regardless of depth (such as horizontal image edges in horizontal linear light fields). Also, when the trajectories of two separate scene points cross at the same pixel, a reliable gradient direction cannot be estimated. We propose the following measures to detect such cases and attach a confidence value $C_i(\mathbf{p})$ to every $S_i(\mathbf{p})$ value based on these ambiguities:

(1) The first term is based on image gradient magnitude and gives higher confidence values to high-gradient regions:

$$C_i^g(\mathbf{p}) = \|\nabla I_i(\mathbf{p})\|. \tag{3}$$

(2) The second term detects image lines oriented with the camera motion that can create "false trajectories". We define a non-confident area for such image edges, i.e., image gradient directions perpendicular to the camera motion γ_m, using a Gaussian

weighting $N_m = N(\gamma_m, \sigma_m)$. The gradient direction in image I_i is denoted by $\gamma_i^{\mathbf{p}}$:

$$C_i^m(\mathbf{p}) = 1 - N_m(\gamma_i^{\mathbf{p}}). \tag{4}$$

For our experiments, we use $\sigma_m = 5$ degrees.

(3) The final term detects when two edges moving at different speeds w.r.t. the camera cross each other, which may happen at corner features in the input images. To compensate for this ambiguity, we reduce the confidence of all corner regions:

$$C_i^r(\mathbf{p}) = \begin{cases} 1/R_i(\mathbf{p}), & R_i(\mathbf{p}) > 1 \\ 1, & R_i(\mathbf{p}) \leq 1 \end{cases} \tag{5}$$

where $R_i(\mathbf{p})$ is the measure of corner response, as defined by a Harris corner detector.

The final confidence measure is computed by multiplying the individual components:

$$C_i = C_i^g \cdot C_i^m \cdot C_i^r. \tag{6}$$

These measures can be combined using other techniques such as Gaussian models, but we chose the multiplication operation, since it is more conservative and decreases false positives and negatives substantially. See the gradient analysis in Figure 2 for local gradient estimates weighted by confidence values. Note that only high gradient regions have high confidences and are visible due to Eq. (3).

3.2 Confidence weighted image-based propagation

We have now an estimate of foreground probability and associated confidences computed with purely local information. As reliable estimates can be made only for pixels with high gradient magnitude and clear foreground or background motion characteristics, low-confidence estimates greatly outnumber reliable ones, leaving only a sparse set of pixels with dependable information (see Figure 2) that must be propagated to the rest of the light field volume.

To efficiently solve this problem, we again exploit the inherent coherency of the light field, and assume that nearby pixels with similar colors should have similar foreground estimates. To that end, we use a confidence-weighted joint-edge-aware filtering, using the light field volume as the joint-domain. Specifically, we use an approximation of a joint-geodesic filter [Gastal and Oliveira 2011], which is efficient over large kernel sizes, scales linearly with increased dimensionality, and performs well when colors are similar in nearby foreground and background regions. This filter works as follows: Rather than doing the filtering in the image domain by changing the filter weights according to color values, the 2D image is first transformed into a domain where it can be filtered by fixed width Gaussians, and then it is transformed back to the original domain. After the transformation, nearby pixels with similar colors have similar coordinates, whereas pixels on opposite sides of an edge are far away, such that the results exhibit edge-awareness.

For the filtering operation, we extend the geodesic filter by a third dimension. In the original work [Gastal and Oliveira 2011], N 1D filtering operations are performed, thereby alternating between the x and y dimensions. Since our domain is a 3D light field, we iterate between x, y and i dimensions, such that information can be propagated inside *and* between images.

We first consider S_i and C_i as slices of the volumes S and C, where $S(x, y, i) = S_i(x, y)$ and $C(x, y, i) = C_i(x, y)$, such that each value in S and C has a corresponding pixel in L, around which the foreground probability and its confidence are computed. Since there is a one-to-one mapping between the three 3D volumes, we can filter S using the image edges from L and confidences from C. To incorporate the confidences C into the filtering process, we adopt the approach from [Lang et al. 2012]: We first multiply C with S element-wise (denoted as $\odot$), and then filter the result using the geodesic filter and the edges from L, producing $(C \odot S)'$. This gives higher weight to more confident regions during the filtering process. The results are then normalized using C', which is generated by filtering C. The final result is calculated as:

$$S'' = \frac{(C \odot S)'}{C'} . \tag{7}$$

For the filtering operations, we used the geodesic filter with the following spatial and range standard deviations (sigmas): 10 and 0.1 for the x and y dimensions, and 5 and 0.1 for the i dimension. For a more detailed discussion on how this step is carried out, we refer the reader to the original papers. Now, S'' contains estimates that are propagated over the whole light field domain and are locally consistent inside the light field among a subset of images (see Figure 2). For simplicity of notation, from now on we refer to S'' as S.

3.3 Global gathering of the image-based estimates

The edge-preserving filtering step results in the segmentations S_i that vary smoothly inside S and are locally consistent. However, as seen in Figure 2, they can still be noisy due to missing image edges in L and incorrect initial estimates. Moreover, the filtering step works solely in image space and does not include geometric information, making the different S_i's geometrically inconsistent. In this global gathering step, we combine the *per-image* segmentation estimates in 3D space using the camera calibrations, such that locally consistent segmentations are aggregated to obtain a globally consistent probabilistic 3D object segmentation H. Then, these results can be reprojected back into the input image to obtain accurate 2D object segmentations. The gathering step constitutes the last stage of our local-to-global framework and ensures that the S_i's become globally consistent.

We discretize the 3D space using a fine voxel grid $\mathcal{V}$. For each $\mathbf{v} \in \mathcal{V}$, we compute $H(\mathbf{v})$, the probability of belonging to the set of foreground elements $\mathcal{F}$ given the per-image segmentations S_i. For that, we project every voxel $\mathbf{v}$ back onto all images, collect the segmentation estimates S_i, and combine them using a probabilistic framework.

Usually, in 3D segmentation or visual hull computations, the per-image segmentations are multiplied. However, this is biased towards favoring background, since lower values tend to pull the result to 0 [Kolev et al. 2006]. So, we accumulate this information using a geometric mean:

$$H(\mathbf{v}) = \left(\prod_{i=1}^{n} P(\mathbf{v} \in \mathcal{F} | S_i(\mathbf{p})) \right)^{1/n} . \tag{8}$$

For each image I_i, we find $P(\mathbf{v} \in \mathcal{F} | S_i(\mathbf{p}))$, i.e., the probability of $\mathbf{v}$ being part of the foreground $\mathcal{F}$ given the segmentation value $S_i(\mathbf{p})$ at the pixel it projects to, using Bayes' rule:

$$P(\mathbf{v} \in \mathcal{F} | S_i(\mathbf{p})) = \frac{P(S_i(\mathbf{p}) | \mathbf{v} \in \mathcal{F}) \cdot P(\mathbf{v} \in \mathcal{F})}{P(S_i(\mathbf{p}))}. \tag{9}$$

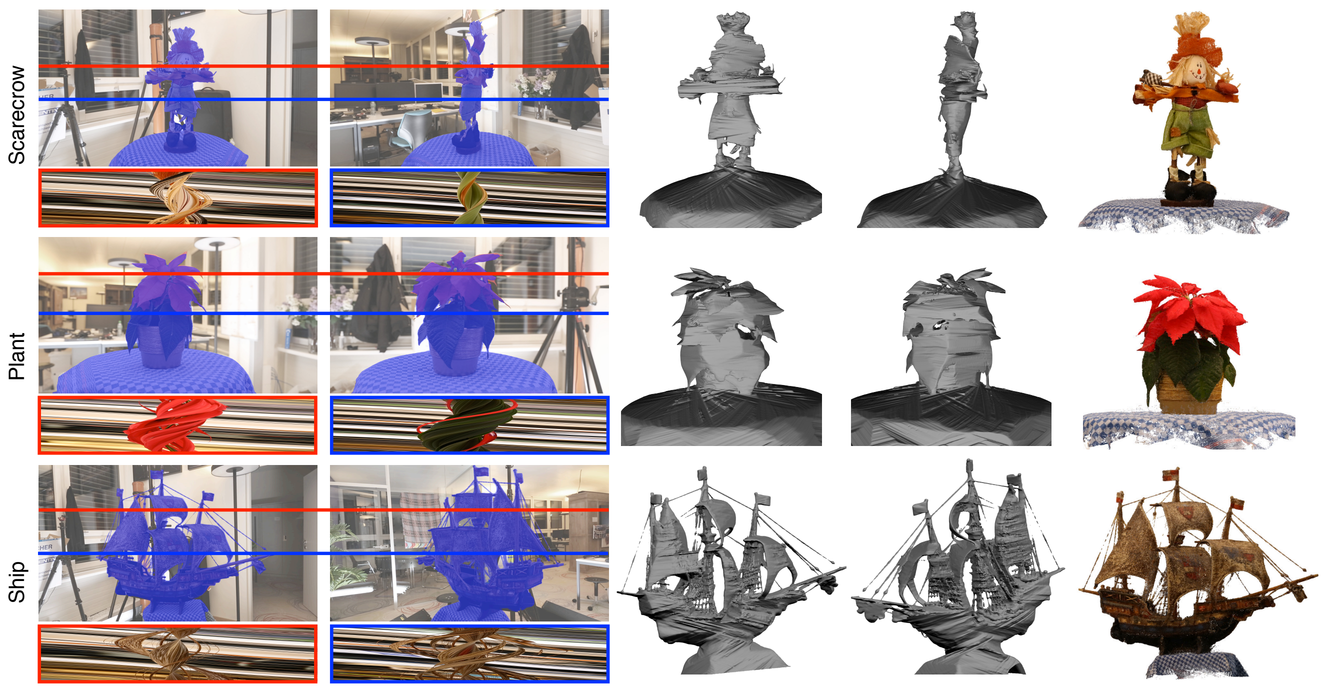

Fig. 5: From left to right: two segmentation examples superimposed against desaturated input images with two light field slices, two views of the meshed object segmentations generated using marching cubes, and a point cloud rendering. Please refer to the accompanying video and supplemental material for further visualizations of our results.

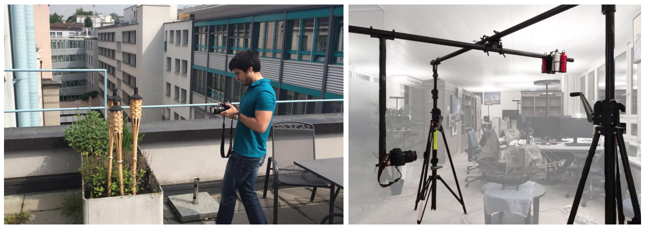

Fig. 6: Our capture setups: hand-held capture (left) and a circular boom that rotates around a fixed object (right).

The term $P(S_i(\mathbf{p})|\mathbf{v} \in \mathcal{F})$ is modeled as follows. The segmentation value $S_i(\mathbf{p})$ has two possible sources: the actual projection of $\mathbf{v}$, which can only get the values 0 and 1, or erroneous segmentation estimates with values in-between. Hence, we model it using a normal distribution $N(1, \sigma_v)$ with mean 1, which is the expected value. Since $S_i(\mathbf{p}) \in [0, 1]$, we truncate the normal distribution to the domain $[0, 1]$:

$$P(S_i(\mathbf{p})|\mathbf{v} \in \mathcal{F}) = \frac{N(1, \sigma_v)}{c}, \tag{10}$$

where c is the normalization constant. By similar arguments, the background probability is defined using a normal distribution with mean 0:

$$P(S_i(\mathbf{p})|\mathbf{v} \in \mathcal{B}) = \frac{N(0, \sigma_v)}{c}. \tag{11}$$

In all our experiments, we used $\sigma_v = 0.1$. The denominator of (9) can then be computed simply as

$$\begin{aligned} P(S_i(\mathbf{p})) = \; & P(S_i(\mathbf{p})|\mathbf{v} \in \mathcal{F}) \cdot P(\mathbf{v} \in \mathcal{F}) + \\ + \; & P(S_i(\mathbf{p})|\mathbf{v} \in \mathcal{B}) \cdot P(\mathbf{v} \in \mathcal{B}). \end{aligned} \tag{12}$$

Similar to before, as we do not assume any priors about the scene, we use a value of 0.5 for the unknown probabilities $P(\mathbf{v} \in \mathcal{F})$ and $P(\mathbf{v} \in \mathcal{B})$. With this, we have all the ingredients required to compute the probabilistic object segmentation in Eq. (8).

In our pipeline, we ask the user to define a bounding box around the foreground object with the help of SfM points. The resolution of the grid is chosen relative to image resolution: We set the voxel size such that when the central voxel is projected onto an image, it does not occupy more than 1.25 pixels. This way, we make sure that the average footprint of a voxel stays comparable to the image resolution. In our examples, the grid resolutions ranged from $400 \times 800 \times 400$ in the ORCHID dataset to $1000 \times 800 \times 1500$ in the DRAGON dataset. This last step of the pipeline operates independently on each voxel $\mathbf{v} \in \mathcal{V}$ without any global regularization requirement, such that this step can also be easily parallelized and computed highly efficiently.

4. EXPERIMENTS AND RESULTS

Our datasets and results will be made publicly available with the paper to facilitate future research.

Acquisition and timing. All datasets were acquired using a Canon 5D Mark III. The hand-held datasets were captured by walking slowly around the objects of interest. For more automated acquisition, we built a setup with the camera mounted on a boom rotating around the object (see Figure 6). This setup is useful in cases where the captured objects cannot be placed on a turntable because they are immobile or heavy, or because they are not rigid (e.g. plants, like the orchid in Figure 1). For all datasets, camera calibrations were computed as a preprocessing step, assuming no prior knowledge of the setup, using standard structure-from-motion techniques [Wu 2013].

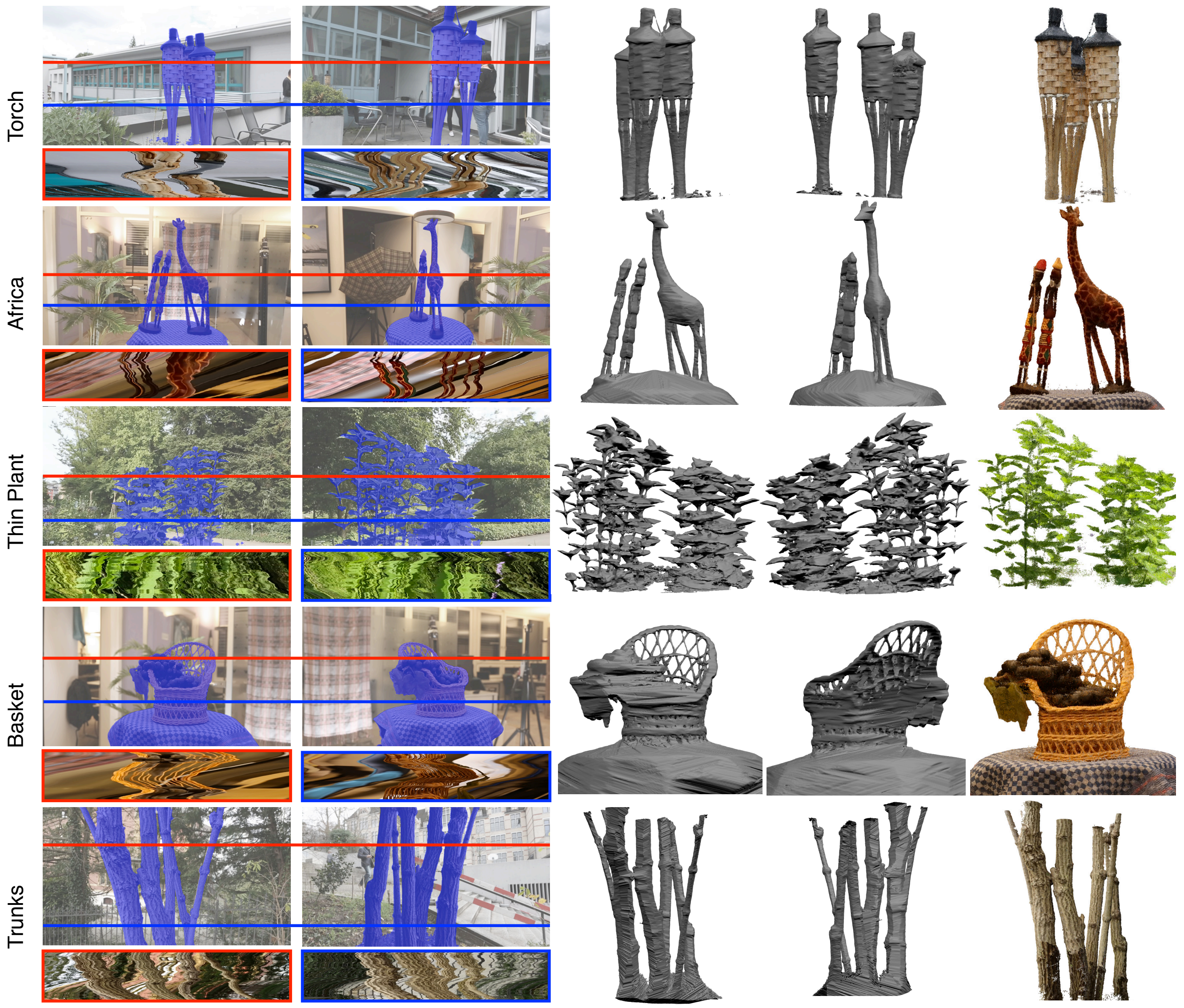

Fig. 7: Reconstruction from hand-held capture. With a sufficiently dense sampling of view points, our method is able to compute segmentations even from hand-held captured video without any modification to the pipeline. For each dataset, we show two segmented input frames, slices of the light fields, two viewpoints of the 3D object segmentation and a point-cloud rendering. Refer to the accompanying video for the inputs.

For each circular dataset, we captured about 3000 to 4500 video frames at 2-mega-pixel resolution, covering a single circle around the object with about 10 images per degree. This amounts to ca. 21 GB of image data. For the hand-held datasets, we captured between 3000 to 4000 video frames at 1-mega-pixel resolution with 60 fps for denser image capture. All results where computed on a desktop with 3.2 GHz Intel Quad-Core and 32 GB RAM.

For a 1 mega-pixel dataset with 3000 frames, the initial segmentation (Section 3.1) took approximately 30 minutes, and the propagation (Section 3.2) took 10 minutes, both of which depend linearly on the image resolution and the number of images. For datasets captured with the circular boom, a speedup of $\times 4$ can be achieved for the initial segmentation step (60 minutes down to 15 minutes using 3000 2-mega-pixel images) by assuming constant motion between successive frames and precomputing the epipolar directions. For all datasets, the global gathering step (Section 3.3) took 20 to 40 minutes; the time depends linearly on the resolution of the voxel grid. Since the filtering step already propagates the fine scale foreground/background information inside the light field and creates locally consistent per-image segmentations, we use a quarter of the input images for the gathering step without sacrificing quality. The total running time of the algorithm is typically 60 to 100 minutes. As there is no global optimization, all computationally expensive steps of the method can be trivially parallelized on the CPU. We expect that with a proper GPU implementation, these running times can be further reduced.

Segmentation results. Figure 5 shows some of our results captured with the rotating boom; we display several final image segmentations for validation. In addition, we show triangle meshes by thresholding the probabilistic 3D segmentations H at a value of $\tau = 0.85$ and

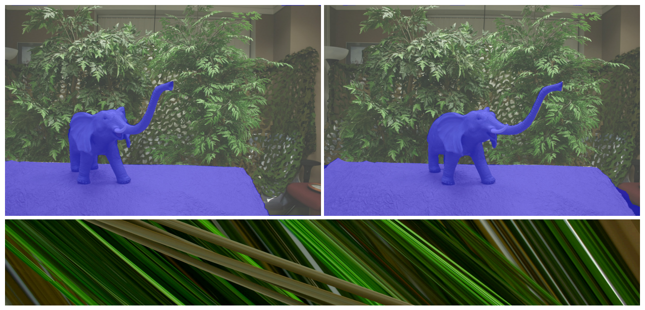

Fig. 8: Top: Our procedure applied to a linear light field from the USCD/MERL light field repository to segment the foreground object. Bottom: One x-i slice of the light field demonstrating the motions of pixels.

using marching cubes to mesh the isosurface. We note that in general, meshing of point clouds and volumetric data is an open problem, and we do not propose a solution here. Additional datasets can be seen in Figures 1 and 2. In some applications, binary image segmentation masks are required. For the segmentation masks shown here, we use the same threshold τ and project the 3D resulting segmentation back to the images. Please see the accompanying video and supplemental material for animated 3D renderings and our resulting models.

We assume that the camera trajectory is continuous, but otherwise unrestricted. This allows us to work with hand-held videos, since our process is robust to the unstructured nature of such light fields as long as the sampling is sufficiently dense. Since our algorithm works directly on epipolar manifolds for computing the gradients, shaky camera motion is not problematic, as long as the camera poses can be correctly estimated using SfM techniques.

Five examples with a hand-held camera are shown in Figure 7 with their corresponding segmentation results, and the results on another hand-held dataset are shown in Figure 1 (right). THINPLANT is a particularly challenging example with fine details, where the foreground object and the background clutter have very similar colors. Moreover, the object slightly moves between frames due to wind. Still, our algorithm manages to construct a reasonable 2D-3D segmentation.

For 3D object captures, circular light fields are especially useful, as they provide a 360° view of objects, such that all sides can be equally carved. However, our method can also generate probabilistic segmentations for other camera trajectories. In the case of linear light fields, scene elements map to lines with different slopes according to their depth values. In such cases, two specific directions can be selected to represent the foreground and the background objects, which can then be used in Eq. (2). As presented in Figure 8, this approach achieves precise segmentations for linear light fields.

Thanks to the dense angular sampling, small-scale features can be resolved, which are challenging for traditional multi-view segmentation methods. In Figure 9, we show a comparison of our technique with our implementation of the work by Campbell et al. [2011] on two datasets. Since their method does not scale well to larger numbers of images, we used, as advised by the authors, 100 equidistantly sampled images to run the algorithm. In addition, the algorithm required manual initialization by selecting some superpixels for the foreground and background, since their assumption that all camera axes intersect on the object does not hold on our datasets. As can be seen, [Campbell et al. 2011] has problems in thin object regions since it works with superpixels. Moreover, since it uses explicit color

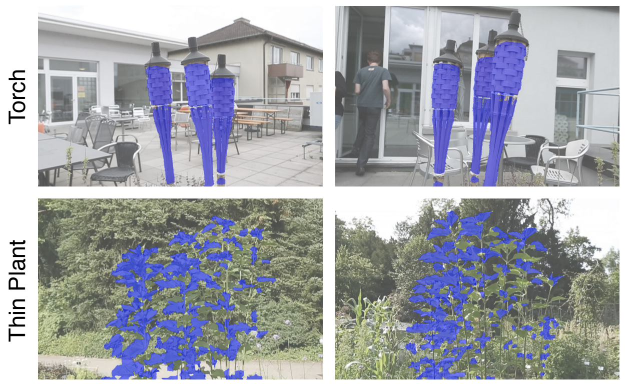

Fig. 9: Results of Campbell et al. [2011] on two datasets. Note the missing features on the foreground and false positives the background, especially around thin object regions.

models for modeling the foreground and background, foreground objects consisting of significantly different colors, and scenes where foreground and background colors are similar become problematic.

We also performed a quantitative analysis of our method, using a synthetically rendered sequence with ground truth segmentations. We asked a trained artist to generate the DRAGON scene and render rendered 3600 images using MAYA, where we rotated a camera around the object in a circle. Then, we used our algorithm to compute the 2D/3D co-segmentations (see Figure 10). In order to measure the quality of our results, we adopted the *intersection-over-union* similarity metric, common for image segmentation evaluations [Li et al. 2013], and used it to compare our 2D segmentations against the ground truth. It can be seen that our method results in very high similarity values to the ground truth even in case of this very challenging dataset, see Figure 11. The same figure shows that the work by Campbell et al. [2011] has a considerably higher reconstruction error, since it cannot make use of the extra images and loses detail due to superpixel-based computations.

Sampling density. The gradient-based analysis and propagation both leverage the dense sampling of the light field volume. The specific density requirements are a function of the texture of the scene in the captured images. Essentially, for our method to work, the motion parallax of an image feature should be smaller than the frequency of the texture around that feature. With insufficient sampling density, the smooth trajectories in light field slices disappear due to aliasing and are replaced by discontinuities and additional structures arising from scene texture (see Figure 12). However, as the sampling rate increases, the structure of the trajectories becomes more apparent. We observe that in real-world datasets, we usually achieve the required coherency in the light field at about 10 images per degree around the object of interest, a rate that is achievable using a standard video camera. If the objects are closer to the camera, the relative speed of the scene points also become faster, resulting in stronger motion parallax and requiring slower camera motion to avoid aliasing. In our experiments, we went as close as 80 centimeters to the object, and saw that the above sampling density was sufficient to observe the fine details. For closer capture scenarios, the sampling rate can be increased for similar quality results.

For a quantitative evaluation of the sampling rate, we ran our algorithm with the DRAGON data set by changing the sampling density, resulting in different number of images. The resulting error values are shown in Figure 11. Note that we get diminishing returns after 3600 images.

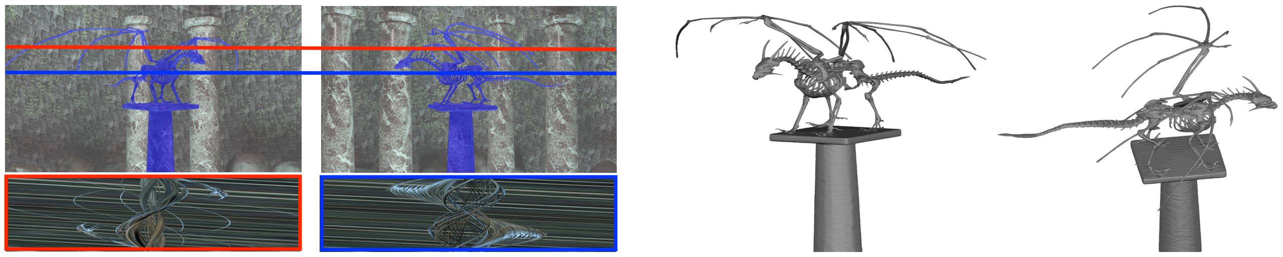

Fig. 10: Our results on the rendered DRAGON dataset with many fine details. Note the reconstruction quality around thin features.

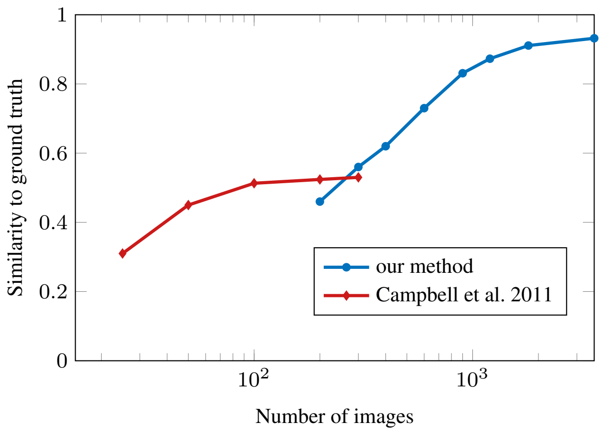

Fig. 11: The change of the similarity measure *intersection-over-union* of our results when compared to the ground truth segmentations (in blue). Note that the similarity to the ground truth improves as the number of images goes up. The same similarity measure on the same dataset is shown for [Campbell et al. 2011] in red. The measure is considerably lower and improvements become marginal after 100 images. Due to scalability issues, we ran the method of Campbell et al. with a maximum of 300 images. Our method needed the same amount of time to compute segmentations for 3600 images as [Campbell et al. 2011] required for 100 images. We used a log scale for the x-axis for a clearer visualization of the trends of both curves.

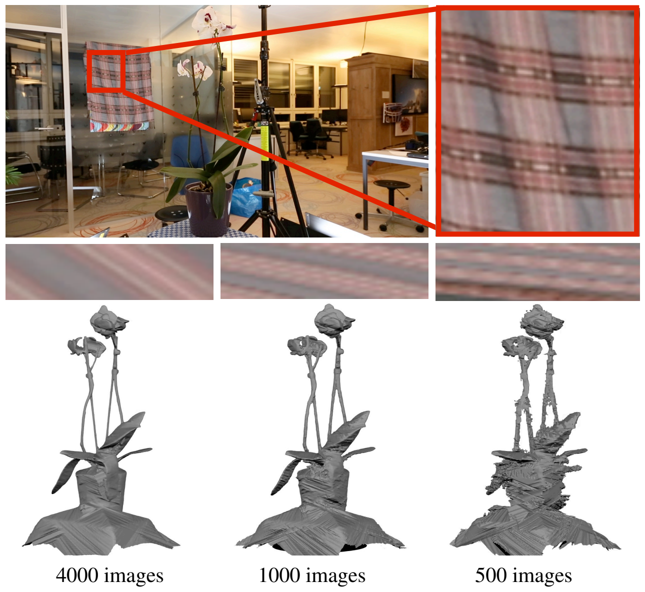

Fig. 12: Reconstruction experiment with different sampling rates, i.e., numbers of input images. Top: input image and close-up. Middle: a crop from the same $x - i$ slice with decreasing sampling rates. Note how angular aliasing starts to occur. Bottom: corresponding 3D segmentations.

Applications. As discussed in Section 2, there are various applications for 3D object segmentations. Here we focus on applications related to improving the result quality and efficiency of methods for image-based 3D reconstruction. We extend the depth-from-light-fields (DFLF) approach [Kim et al. 2013] in two ways. First, we modify the consistency check, such that only pixels lying inside the segmentation masks are tested for depth estimation, which greatly reduces the search space and considerably speeds up the algorithm for 3D object reconstruction. Second, the reconstructed points are only accepted if they are contained within the 3D segmentation, and are otherwise rejected. Using these two extra steps in the DFLF algorithm reduces reconstruction errors and greatly improves efficiency. In our experiments, we observed that our modified DFLF algorithm took **15%** of the computation time of the original method: e.g., the runing time for computing the final point cloud on the full ORCHID dataset decreased from 190 minutes to 30 minutes on the GPU. A comparison of point clouds generated with the original DFLF and our modified method is shown in Figure 13. The outliers in the DFLF results stem from the fact that background points are not captured with sufficient parallax variation due to the rotating camera motion, leading to inaccurate depth estimates. Our modification successfully removes all these artifacts. The cleaned point clouds can be used as input to meshing techniques such as Poisson surface reconstruction [Kazhdan and Hoppe 2013], as shown in Figure 14.

Our segmentations can also be easily incorporated into any off-the-shelf stereo software. In Figure 15, we show how the quality of the depth maps generated by semi-global matching [Hirschmüller 2006] can be improved with the help of our 3D object segmentations. The floating outliers are caused by wrong correspondence estimates around silhouettes, which can become substantial when merging depth maps from different viewing directions around the object. Our approach is able to produce results with fewer outliers, while preserving all details in the objects.

We also compare our method to four additional, publicly available image-based reconstruction methods [Goesele et al. 2007; Zhang et al. 2009; Furukawa and Ponce 2010; Furukawa et al. 2010]. All four methods are state-of-the-art and known for producing excellent-quality results. However, like most existing MVS approaches, these methods are designed to process smaller volumes of image data, typically between 50 to a few hundreds of images (according to personal communication with some of their authors). Our comparison indicates that the amount of reconstructed object detail cannot be easily increased by simply raising the number of input images with existing techniques.

In Figure 16, we compare our method to a publicly available, patch-based MVS method, both without optimal view selection

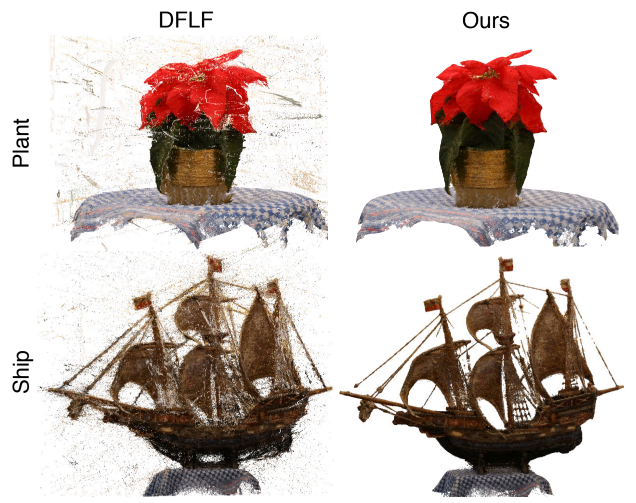

Fig. 13: Left: point clouds generated with DFLF [2013]. Right: Point clouds computed using our enhanced DFLF method that leverages the 2D-3D segmentations.

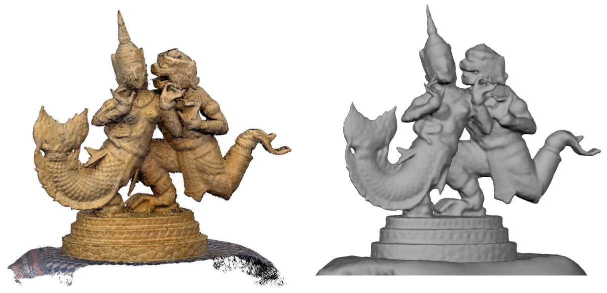

Fig. 14: Left: Oriented point cloud for the STATUE. Right: Results of Poisson surface reconstruction. Note the preserved fine details and accurate silhouettes.

(PMVS [Furukawa and Ponce 2010]) and with view selection (CMVS [Furukawa et al. 2010]). Both methods output oriented point clouds, which we mesh using Poisson surface reconstruction [Kazhdan and Hoppe 2013]. Our meshes are generated as before by running marching cubes on H. Patch-based approaches may struggle with capturing fine details, such as the ropes in the SHIP dataset, and tend to grow object boundaries due to the employed patch-based color consistency computation. The CMVS approach improves upon the quality of the PMVS reconstructions thanks to the optimized view selection, and therefore demonstrates that small camera baselines can indeed be problematic for existing multi-view stereo approaches. In our results, the thin rigging in the SHIP is reconstructed quite well, even though these structures occupy only a few pixels. Similar effects can be observed in the straws and ropes on the SCARECROW.

In Figure 17, we compare our method against two other techniques that first compute per-image depth maps and either merge them in 3D space (MVE [Goesele et al. 2007]) or make them consistent over multiple views (ACTS [Zhang et al. 2009]). MVE uses patch-based image comparisons to generate oriented point clouds, which are meshed with the built-in floating scale surface reconstruction technique [Fuhrmann and Goesele 2014]. ACTS computes per-image depth maps and makes them consistent over multiple views via bundle adjustment. As ACTS only produces point clouds without

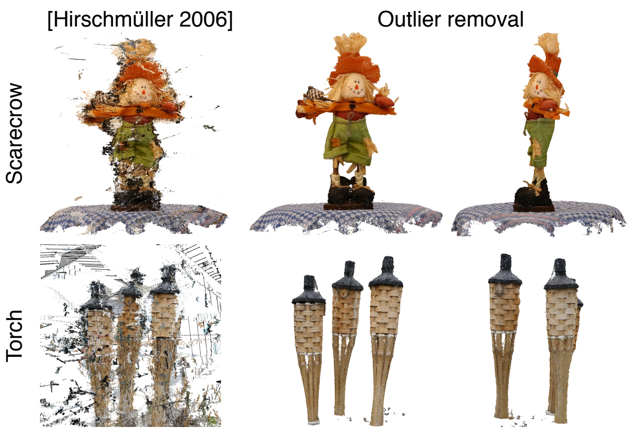

Fig. 15: Point clouds generated with the method of [Hirschmüller 2006] without constraints (left) and with our segmentation constraints (right).

orientation information, we do not generate any meshes for this data. Instead, we show two views of the generated point clouds by projecting the depth maps into 3D space. Since there is no global consistency check between the depth maps, noise can accumulate when multiple views are used together. Problems around object boundaries due to patch based comparisons are observed for both methods, where the background is reconstructed as part of the foreground object, leading to excessive debris around the reconstructed objects. The results by MVE also show that producing meshes from oriented point clouds for objects with thin features can lead to additional noise. Note that most of these outliers could be removed using our segmentations to improve the reconstruction quality.

The ORCHID dataset has further challenges, such as the specular reflections on the plant pot (see bottom light field slice in Figure 1) that may present considerable difficulty for most multi-view reconstruction techniques, which generally assume Lambertian reflectance properties. In our results, the pot is faithfully segmented. Moreover, these techniques do not scale well with increased number of images due to their high computational time complexities. For the THIN-PLANT dataset with 4000 frames at 1280×720 resolution, MVE requires 72 hours. This running time decreased to 90 minutes when 200 frames were used instead. ACTS, on the other hand, required 110 minutes on the GPU (or 700 minutes on the CPU) to compute the depth maps for 200 frames. Our algorithm generated the 3D segmentations in 100 minutes using all 4000 frames, which translates to about 1.5 seconds per frame.

5. DISCUSSION

We presented an automatic method that generates precise 2D and 3D object segmentations from light fields. Our method efficiently exploits coherence in densely sampled data and works effectively with thousands of input images.

Our method requires the camera motion between two frames to be slow for robustly estimating segmentation values, as shown in Figure 12. However, in hand-held capture scenarios, the camera can move too fast between two frames, causing wrong estimates. Moreover, if the camera is almost stationary between two frames, the initial estimates become unreliable. In our work, we identified stationary frames using the SfM information and removed them from the final gathering step.

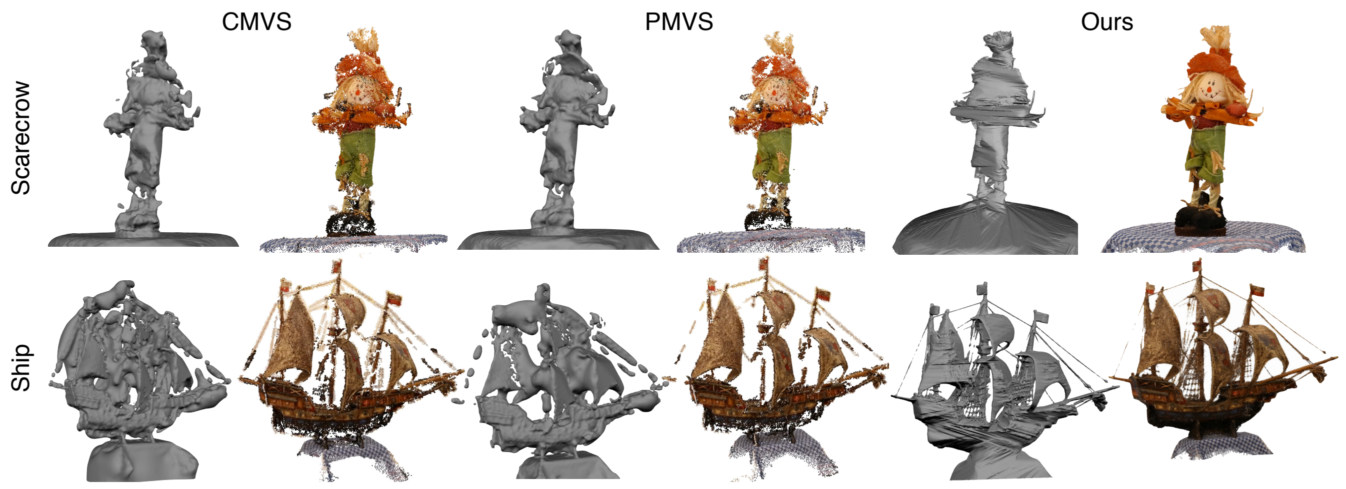

Fig. 16: Comparison of our method to CMVS [Furukawa et al. 2010] and PMVS [Furukawa and Ponce 2010]. For each algorithm, we show the geerated meshes and point clouds from similar viewpoints. Please see Section 4 for a detailed discussion

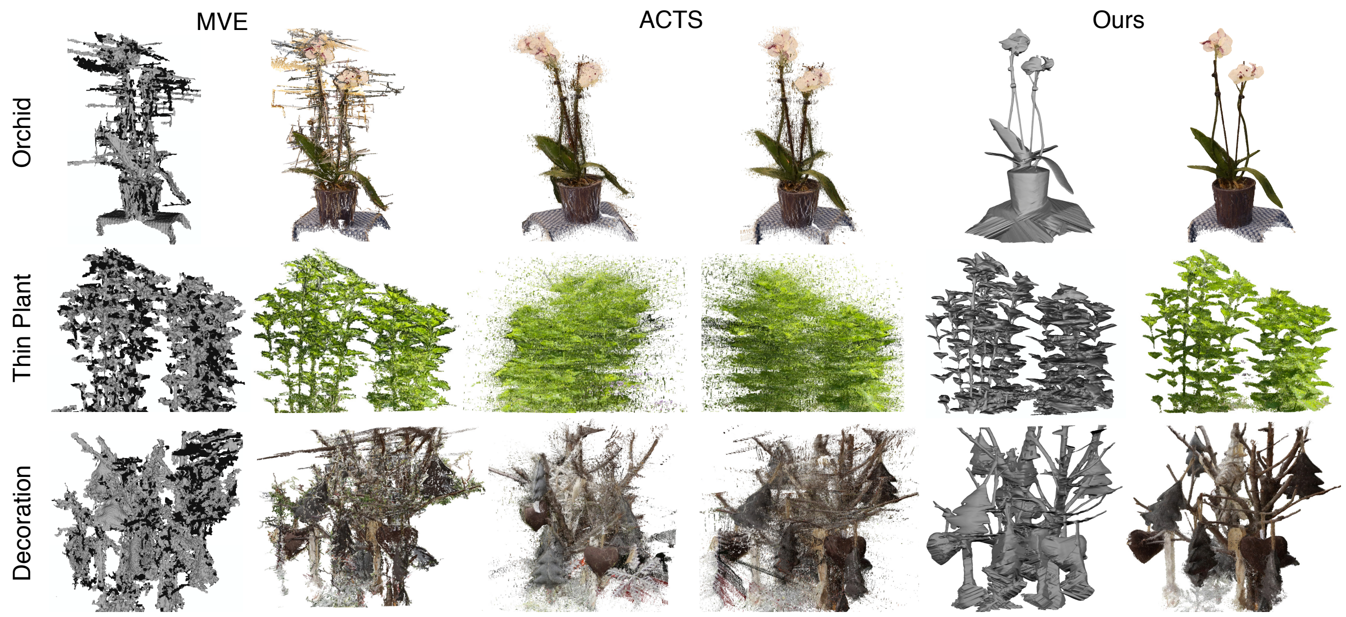

Fig. 17: Comparison of our method to MVE [Goesele et al. 2007] and ACTS [Zhang et al. 2009]. For each algorithm, we show the generated point clouds from similar view points. The point clouds of MVE are meshed with floating scale surface reconstruction [Fuhrmann and Goesele 2014] as part of their pipeline. Since ACTS does not generate meshes, we show the point clouds from two different viewpoints. Please see Section 4 for a detailed discussion

Like for most other methods for generating 3D shapes from images, the quality of our approach depends on the accuracy of the camera calibrations. Due to the multiplicative property of the gathering step, a small number of minority views can remove otherwise well-supported regions. In the PLANT dataset, we see errors in calibration that cause some voxels at the tips of the top leaves to be lost.

We can additionally observe a number of failure cases in Figure 18. In this example, a ground plane is visible, violating our depth-based segmentation assumption. Furthermore, the foreground object moves substantially during capture, the image is saturated (textureless) in brighter regions, the foreground and background have similar color and texture, and the hand-held capture covers only 180° around the object. These factors cause a number of erroneous estimates and missing regions, which can be seen in the resulting segmentations. Nonetheless, the trunk is still clearly visible, and the resulting point cloud of our modified DFLF method shows reasonable structure.

Our trajectory-based parallax segmentation generates high-confidence results in edge regions. If a foreground object is placed on a smooth background, we have to propagate confident estimates to the rest of the volume, which can cause bleeding of the foreground information into the background. If holes in the foreground object are small, and no background edges are visible through them, the method cannot carve these regions away. Such cases can be seen in the TORCH and AFRICA datasets. In practice, however, these cases are well addressed by existing color-based segmentation methods, and combining the two approaches is an interesting future work.

Our method can effectively filter noisy point clouds. However, due to the amount of fine structures in the reconstructed objects, it is still very challenging to mesh such point clouds. As long as the amount of fine structures is relatively low, a good triangle mesh can be extracted using Poisson surface reconstruction [Kazhdan and Hoppe 2013], as in Figure 14. As the amount of fine structures increases, such as for the ropes of the SHIP dataset, current meshing algorithms fail to incorporate this information. Meshing point clouds for objects with intricate details is an interesting future research direction.

As observed in recent works [Criminisi et al. 2005; Wanner and Goldluecke 2012; Kim et al. 2013], dense data capture has enabled entirely novel strategies for image-based reconstruction. In this work, we have extended this line of thought by presenting a novel approach for computing object segmentations in 2D and 3D with applications in 3D scene reconstruction. When faced with big visual data, existing approaches can suffer from scalability issues and cannot effectively make use of the full information available. However, the coherency and redundancy inside such data offers new opportunities, and at the same time requires rethinking existing approaches from a new vantage point. We hope that recent work on dense data will open up various research avenues to take advantage of the ever increasing volume of visual data being generated and stored.

Acknowledgements

We are grateful to Changil Kim for his tremendous help with generating the results and comparisons. We would also like to thank Maurizio Nitti for rendering the DRAGON dataset, Derek Bradley for the discussions on MVS, Joel Bohnes for helping with the figures, Benjamin Resch for helping with the camera calibrations, and Guofeng Zhang for helping with generating the ACTS results.

REFERENCES

AN, X. AND PELLACINI, F. 2008. AppProp: All-pairs appearance-space edit propagation. *ACM Trans. Graph. 27,* 3, 40:1–40:9.

APOSTOLOFF, N. AND FITZGIBBON, A. W. 2006. Automatic video segmentation using spatiotemporal T-junctions. In *Proc. BMVC*. 1089–1098.

BERENT, J. AND DRAGOTTI, P. L. 2007. Plenoptic manifolds – exploiting structure and coherence in multiview images. *IEEE Signal Processing Magazine 24,* 7, 34–44.

BOLLES, R. C., BAKER, H. H., AND MARIMONT, D. H. 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *Internat. J. Comput. Vision 1,* 1, 7–55.

BOWEN, A., MULLINS, A., WILSON, R., AND RAJPOOT, N. 2007. Bayesian surface estimation from multiple cameras using a prior based on the visual hull and its application to image based rendering. In *Proc. BMVC*. 1–8.

BOYKOV, Y. AND JOLLY, M.-P. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*. 105–112.

BRADLEY, D., BOUBEKEUR, T., AND HEIDRICH, W. 2008. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proc. CVPR*.

BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S., AND COHEN, M. 2001. Unstructured lumigraph rendering. In *Proc. ACM SIGGRAPH*. 425–432.

CAMPBELL, N. D. F., VOGIATZIS, G., HERNÁNDEZ, C., AND CIPOLLA, R. 2010. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing 28,* 1, 14–25.

CAMPBELL, N. D. F., VOGIATZIS, G., HERNANDEZ, C., AND CIPOLLA, R. 2011. Automatic object segmentation from calibrated images. In *Proc. CVMP*. 126–137.

CHEN, C., LIN, H., YU, Z., BING KANG, S., AND YU, J. 2014. Light field stereo matching using bilateral statistics of surface cameras. In *Proc. CVPR*.

CHUANG, Y.-Y., AGARWALA, A., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2002. Video matting of complex scenes. *ACM Trans. Graph. 21,* 3, 243–248.

CRIMINISI, A., KANG, S. B., SWAMINATHAN, R., SZELISKI, R., AND ANANDAN, P. 2005. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding 97,* 1, 51–85.

DAVIS, A., LEVOY, M., AND DURAND, F. 2012. Unstructured light fields. *Comput. Graph. Forum 31,* 2, 305–314.

EISEMANN, E. AND DURAND, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. 23,* 3.

EISEMANN, M., DE DECKER, B., MAGNOR, M. A., BEKAERT, P., DE AGUIAR, E., AHMED, N., THEOBALT, C., AND SELLENT, A. 2008. Floating textures. *Comput. Graph. Forum 27,* 2.

FELDMANN, I., KAUFF, P., AND EISERT, P. 2003a. Extension of epipolar image analysis to circular camera movements. In *Proc. International Conference on Image Processing*. 697–700.

FELDMANN, I., KAUFF, P., AND EISERT, P. 2003b. Image cube trajectory analysis for 3D reconstruction of concentric mosaics. In *Proc. Vision, Modeling and Visualization*.

FRANCO, J.-S. AND BOYER, E. 2005. Fusion of multi-view silhouette cues using a space occupancy grid. In *Proc. ICCV*.

FUHRMANN, S. AND GOESELE, M. 2014. Floating scale surface reconstruction. *ACM Trans. Graph. 33,* 4, 46.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2010. Towards internet-scale multi-view stereo. In *Proc. CVPR*.

FURUKAWA, Y. AND PONCE, J. 2009. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision 81,* 1, 53–67.

FURUKAWA, Y. AND PONCE, J. 2010. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32,* 8, 1362–1376.

GASTAL, E. S. L. AND OLIVEIRA, M. M. 2011. Domain transform for edge-aware image and video processing. *ACM Trans. Graph. 30,* 4, 69:1–69:12.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *Proc. ICCV*. 1–8.

GOLDLÜCKE, B. AND MAGNOR, M. A. 2003. Joint 3D-reconstruction and background separation in multiple views using graph cuts. In *Proc. CVPR*. 683–688.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The Lumigraph. In *Proc. ACM SIGGRAPH*. 43–54.

GRAUMAN, K., SHAKHNAROVICH, G., AND DARRELL, T. 2003. A bayesian approach to image-based visual hull reconstruction. In *Proc. CVPR*. 187–194.

GRUNDMANN, M., KWATRA, V., HAN, M., AND ESSA, I. A. 2010. Efficient hierarchical graph-based video segmentation. In *Proc. CVPR*. 2141–2148.

GUILLEMAUT, J.-Y. AND HILTON, A. 2011. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal of Computer Vision 93,* 1, 73–100.

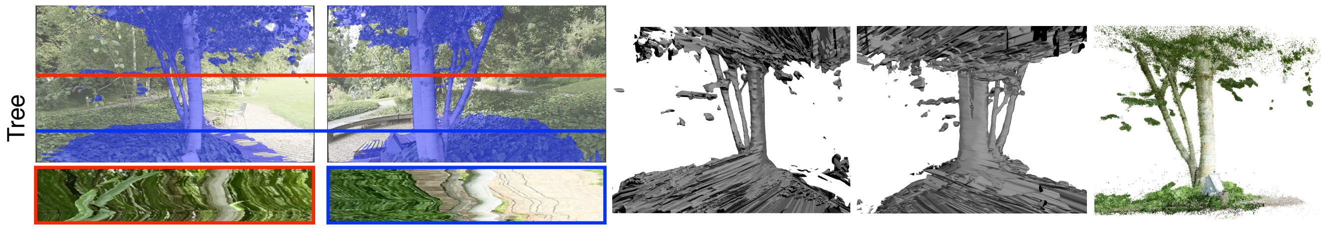

Fig. 18: Example of some failure modes of our approach. Artifacts include: missing branches that were moved by the wind, holes in saturated regions of the sky, a visible ground plane, and extraneous parts of the segmented object, due to only 180^o of the tree being captured in the input video. Despite these challenges, the reconstructed point cloud shows significant detail in the trunk and ground.

HANE, C., ZACH, C., COHEN, A., ANGST, R., AND POLLEFEYS, M. 2013. Joint 3D scene reconstruction and class segmentation. In *Proc. CVPR*. 97–104.

HIRSCHMÜLLER, H. 2006. Stereo vision in structured environments by consistent semi-global matching. In *Proc. CVPR*.

ISIDORO, J. AND SCLAROFF, S. 2003. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Proc. ICCV*. 1335–1342.

JOULIN, A., BACH, F. R., AND PONCE, J. 2010. Discriminative clustering for image co-segmentation. In *Proc. CVPR*.

KAZHDAN, M. M. AND HOPPE, H. 2013. Screened Poisson surface reconstruction. *ACM Trans. Graph. 32,* 3.

KIM, C., ZIMMER, H., PRITCH, Y., SORKINE-HORNUNG, A., AND GROSS, M. 2013. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph. 32,* 4.

KOLEV, K., BROX, T., AND CREMERS, D. 2006. Robust variational segmentation of 3D objects from multiple views. In *Proc. DAGM*.

KOPF, J., COHEN, M. F., LISCHINSKI, D., AND UYTTENDAELE, M. 2007. Joint bilateral upsampling. *ACM Trans. Graph. 26,* 3.

KOWDLE, A., SINHA, S. N., AND SZELISKI, R. 2012. Multiple view object cosegmentation using appearance and stereo cues. In *Proc. ECCV*. 789–803.

KRÄHENBÜHL, P. AND KOLTUN, V. 2012. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. NIPS*. 109–117.

KUTULAKOS, K. N. 1997. Shape from the light field boundary. In *Proc. CVPR*. 53–59.

LANG, M., WANG, O., AYDIN, T., SMOLIC, A., AND GROSS, M. 2012. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph. 31,* 4, 34:1–34:8.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *IEEE TPAMI 16,* 2, 150–162.

LEE, W., WOO, W., AND BOYER, E. 2011. Silhouette segmentation in multiple views. *IEEE TPAMI 33,* 7, 1429–1441.

LEVOY, M. AND HANRAHAN, P. 1996. Light field rendering. In *Proc. ACM SIGGRAPH*. 31–42.

LEZAMA, J., ALAHARI, K., SIVIC, J., AND LAPTEV, I. 2011. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Proc. CVPR*. 3369–3376.

LI, F., KIM, T., HUMAYUN, A., TSAI, D., AND REHG, J. M. 2013. Video segmentation by tracking many figure-ground segments. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2192–2199.

LI, Y., SUN, J., AND SHUM, H.-Y. 2005. Video object cut and paste. *ACM Trans. Graph. 24,* 3, 595–600.

MARTIN, W. N. AND AGGARWAL, J. K. 1983. Volumetric descriptions of objects from multiple views. *IEEE TPAMI 5,* 2.

MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S. J., AND MCMILLAN, L. 2000. Image-based visual hulls. In *Proc. ACM SIGGRAPH*.

OSWALD, M. R. AND CREMERS, D. 2013. A convex relaxation approach to space time multi-view 3D reconstruction. In *ICCV Workshop on Dynamic Shape Capture and Analysis (4DMOD)*.

PARIS, S., KORNPROBST, P., TUMBLIN, J., AND DURAND, F. 2007. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH 2007 Courses*.

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23,* 3.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23,* 3, 309–314.

SHAN, Q., CURLESS, B., FURUKAWA, Y., HERNANDEZ, C., AND SEITZ, S. M. 2014. Occluding contours for multi-view stereo. In *Proc. CVPR*. 4002–4009.

SINHA, S. N. AND POLLEFEYS, M. 2005. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *Proc. ICCV*. 349–356.

SNOW, D., VIOLA, P., AND ZABIH, R. 2000. Exact voxel occupancy with graph cuts. In *Proc. CVPR*.

STARCK, J., MILLER, G., AND HILTON, A. 2006. Volumetric stereo with silhouette and feature constraints. In *Proc. BMVC*. 1189–1198.

SZELISKI, R. 1993. Rapid octree construction from image sequences. *CVGIP: Image Underst. 58,* 1, 23–32.

TABB, A. 2013. Shape from silhouette probability maps: Reconstruction of thin objects in the presence of silhouette extraction and calibration error. In *Proc. CVPR*. 161–168.

VOGIATZIS, G., TORR, P. H. S., AND CIPOLLA, R. 2005. Multi-view stereo via volumetric graph-cuts. In *Proc. CVPR*. 391–398.

WANG, J., BHAT, P., COLBURN, R. A., AGRAWALA, M., AND COHEN, M. F. 2005. Interactive video cutout. *ACM Trans. Graph. 24,* 3, 585–594.

WANNER, S. AND GOLDLUECKE, B. 2012. Globally consistent depth labeling of 4D lightfields. In *Proc. CVPR*.

WANNER, S., STRAEHLE, C. N., AND GOLDLUECKE, B. 2013. Globally consistent multi-label assignment on the ray space of 4D light fields. In *Proc. CVPR*. 1011–1018.

WU, C. 2013. Towards linear-time incremental structure from motion. In *Proc. 3DTV*. 127–134.

YEZZI, A. J. AND SOATTO, S. 2001. Stereoscopic segmentation. In *Proc. ICCV*. 59–66.

YU, Z., GUO, X., LING, H., LUMSDAINE, A., AND YU, J. 2013. Line assisted light field triangulation and stereo matching. In *Proc. ICCV*. 2792–2799.

ZHANG, G., JIA, J., WONG, T., AND BAO, H. 2009. Consistent depth maps recovery from a video sequence. *IEEE Trans. Pattern Anal. Mach. Intell. 31,* 6, 974–988.