Fast and Accurate Depth Estimation from Sparse Light Fields

Aleksandra Chuchvara, Attila Barsi, and Atanas Gotchev, *Member, IEEE*

***Abstract*— We present a fast and accurate method for dense depth reconstruction, which is specifically tailored to process sparse, wide-baseline light field data captured with camera arrays. In our method, the source images are over-segmented into non-overlapping compact superpixels. We model superpixel as planar patches in the image space and use them as basic primitives for depth estimation. Such superpixel-based representation yields desired reduction in both memory and computation requirements while preserving image geometry with respect to the object contours. The initial depth maps, obtained by plane-sweeping independently for each view, are jointly refined via iterative belief-propagation-like optimization in superpixel domain. During the optimization, smoothness between the neighboring superpixels and geometric consistency between the views are enforced. To ensure rapid information propagation into textureless and occluded regions, together with the immediate superpixel neighbors, candidates from larger neighborhoods are sampled. Additionally, in order to make full use of the parallel graphics hardware a synchronous message update schedule is employed allowing to process all the superpixels of all the images at once. This way, the distribution of the scene geometry becomes distinctive already after the first iterations, facilitating stability and fast convergence of the refinement procedure. We demonstrate that a few refinement iterations result in globally consistent dense depth maps even in the presence of wide textureless regions and occlusions. The experiments show that while the depth reconstruction takes about a second per full high-definition view, the accuracy of the obtained depth maps is comparable with the state-of-the-art results, which otherwise require much longer processing time.**

***Index Terms*—3D reconstruction, depth map, light-field video, multi-view stereo (MVS), superpixel segmentation**

I. Introduction

THE notion of light field [1] is employed to describe full visual information of a scene in terms of individual light rays reflected or emitted by the objects. In the recent years, a number of different light field acquisition techniques have been proposed [2]. Robotic arms, gantries and handheld cameras [1], [3], [4], for example, are used to obtain large collections of multi-view data, providing high-resolution sampling in both spatial and angular domains. However, due to sequential nature of capture with a single camera, the applicability of these techniques is limited to static scenes. Alternatively, new plenoptic technologies [5], [6] using a single high-resolution imaging sensor are capable to capture multiple sub-aperture images of a scene in a single shot. Inevitably, this creates a trade-off: providing comparatively good angular resolution, such cameras suffer from low spatial resolution. In contrast, systems based on camera arrays [7], [8], [9], [10] give the opportunity to record dynamic light-field videos with good spatial resolution. Light field video production is particularly beneficial for many practical applications, e.g. 3D television, free-view television, teleconferencing, virtual and augmented reality. However, using dense camera arrays for direct light field sampling is often restricted on practice due to vast amount of data and associated bandwidth problems. Therefore, the required amount of light field samples has to be rendered from the limited set of available camera views.

Rendering based on reduced number of light field samples requires knowledge of the scene geometry in order to avoid rendering artifacts [11]. The lesser the amount of the available visual information, the more rendering quality depends on the accuracy of the provided geometry. Naturally, fast and accurate 3D reconstruction techniques from a sparse set of light field samples are in great demand. For many years, dense depth estimation has been an active research topic in the context of multi-view 3D reconstruction. Nevertheless, automatic recovery of high-quality dense geometry remains a challenging problem. Whilst many existing 3D reconstruction methods concentrate on accuracy, efficiency in terms of runtime and memory consumption is often undermined, limiting their applicability in cases where real-time processing is required and bandwidth is limited. Such applications include light field teleconferencing enabling eye contact, VR/AR interacting with remote scenes, and light field streaming [12], [13], [14]

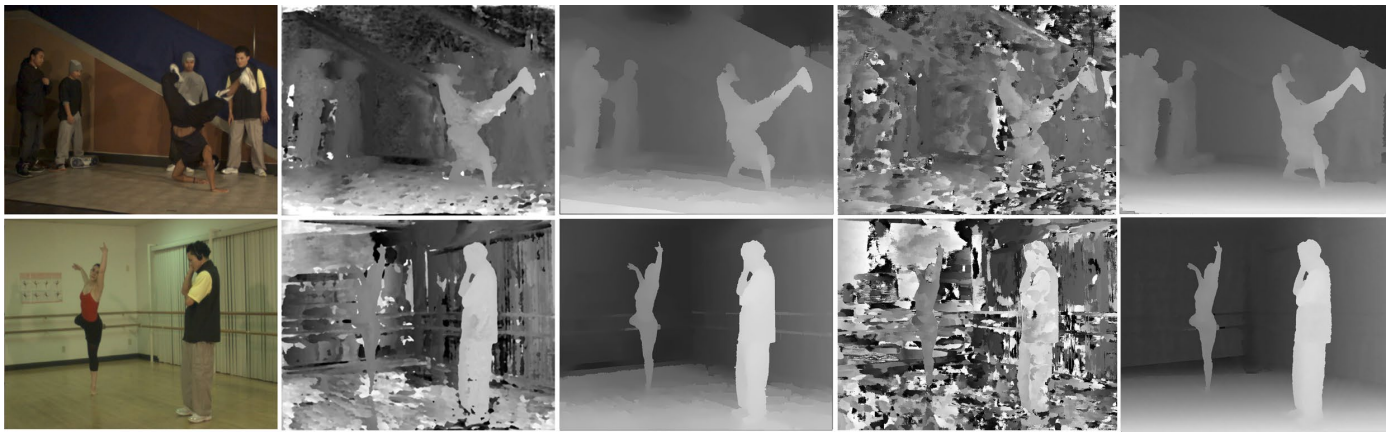
In this paper, we address the challenging task of dense depth reconstruction from sparse light field data. As oppose to angularly dense (the disparity is of the order of one pixel) light fields, obtained e.g. by plenoptic cameras, the proposed method

Manuscript received June 7, 2019; revised October 29, 2019; accepted November 23, 2019. Date of publication December xx, 2019; date of current version December xx, 2019. This work was supported by the Doctoral School of Tampere University of Technology (now Tampere University) and the PROLIGHT-IAPP Marie Curie Action of the European Union's Seventh Framework Programme, REA grant agreement 32449. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sérgio de Faria. (Corresponding author: Aleksandra Chuchvara.)

Aleksandra Chuchvara and Atanas Gotchev are with Tampere University, Finland (e-mail: aleksandra.chuchvara@tuni.fi; atanas.gotchev@tuni.fi). Attila Barsi is with Holografika Kft, Hungary (e-mail: a.barsi@holografika.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes evaluation of temporal consistency for light field video synthesis. Contact aleksandra.chuchvara@tuni.fi for further questions about this work.

Digital Object Identifier …



(a) (b) (c) (d) (e)

Fig. 1. Example depth estimation results on sparse multi-view datasets 'Ballet' and 'Breakdance' [44]. (a) One of the input views. (b) Depth estimation result by the epipolar image analysis [19] as provided in [21]. (c) Depth estimation result by segmentation-based multi-view stereo as provided by [44]. (d) Depth estimation result by multi-view PatchMatch stereo obtained by us using the publicly available code provided by [49]. (e) Depth maps obtained with the proposed method.

is specifically tailored to process wide-baseline light field data captured with camera arrays (see, e.g. Fig. 1 in [10]) . 'Sparse' in this context refers to the minimum disparity between adjustment views, which is considerably higher than one pixel. Our goal is to balance the two key performance aspects, namely efficiency and quality. Efficiency is addressed in terms of the required density of the light field sampling, processing time, and memory consumption. Quality is addressed in terms of accuracy, completeness, and robustness of recovered geometry. To achieve this goal, we use superpixels, which represent regular and compact image regions of homogeneous color, as the basic units for depth estimation. Elevating the representation from raw image pixels to superpixels brings a number of important advantages. First of all, the computational efficiency is improved as the number of the elements to be processed is significantly reduced, while, at the same time, underlying image structures are preserved, allowing for accurate handling of depth discontinuities. Moreover, combining similarly colored pixels leads to improved robustness against noise and intensity bias, and mitigates the ambiguity associated with textureless and occluded regions. We demonstrate that our method, while being faster and simpler than many previous methods, can nevertheless provide very accurate reconstruction results.

The rest of the paper is organized as follows. In Section II, we briefly present the related work and our contributions. Details of our method are provided in Section III. In Section IV, we demonstrate experimental results, and we conclude our work in Section V.

II. Relation with prior art

A. Related work

Typically, light-field depth estimation methods rely on densely sampled narrow-baseline input that exhibits certain structural properties: points at different depths form lines with different slopes in so-called epipolar-plane image (EPI). To infer scene geometry, the EPI lines can be identified, e.g. by applying structure tensor [15] or defocus cue [16]. Usually, such local techniques fail due to noise, occlusions or ambiguities caused by textureless regions (just as stereo methods). Therefore, various methods utilize computationally expensive global optimization techniques to refine the initial depth estimates [15], [17]. Several more efficient methods that avoid global optimization were proposed in [18], [19], [20]. However, while still relying on redundancy and coherence of densely sampled input, the performance of these methods degrades substantially when the disparities between neighboring views grow too large, as in case of sparse light fields, c.f. Fig. 1(b) [21]. Detailed description of different strategies of dense light-field depth estimation is presented in the taxonomy [22]. More recently, several works have been proposed that address: occlusions handling [23], [24], [25], noisy input [24], [26], and reconstruction from sparser set of light field views [27], [28], [29].

For sparse wide-baseline light fields, depth can be estimated using multi-view stereo (MVS) methods [30], [31]. Plane sweeping [32], [33] is the seminal MVS approach, where for each pixel in each view, multiple depth hypotheses are tested and the one that maximizes photo-consistency between the input views is chosen. Although optimization of multi-view photo-consistency provides satisfactory results, it is challenged by the presence of occlusions and textureless image regions, which may lead to ambiguities. To cope with outliers resulting from ambiguous matching, explicit smoothness constraint (based on the assumption that adjacent pixels should have similar depth) is usually imposed on the depth estimates. In order to simultaneously refine smoothness and photo-consistency terms, top-ranked MVS methods utilize global optimization techniques, such as graph cuts [34], [35] and belief propagation [36], [37], [38]. Further, instead of performing optimization for each view independently, some MVS methods utilize so-called bundle optimization, where photo-consistency is combined with visibility or multi-view geometric coherence constraints [37], [39]. Relating depth estimates from different views allows reducing ambiguities and handling occlusions.

However, recovery of textureless regions is still problematic; the optimization procedure in such regions often converges to sub-optimal local solutions as the local pixel-wise smoothness constraints are usually too 'soft' to prevent it. Moreover, while concentrating on accuracy, global MVS methods are memory and time consuming, which leads to bad scalability and low runtime performance and restricts their usage in practical applications.

Using a more rigid form of smoothness constraint, namely a planarity constraint, helps to improve the depth estimation within poorly textured areas. Planarity constraint enforces parametrical relation between pixels that belong to the same planar region. Piecewise planar geometry can be recovered, for example, by fitting planes to a sparse set of 3D feature points and line segments [40], [41]. Alternatively, assuming that depth discontinuities coincide with color boundaries, images are segmented into homogeneous color regions and each segment is modelled as a 3D plane [42], [43]. Although faster and more scalable compared to global MVS, planarity-based methods typically consider man-made environments (such as buildings) and mainly find application in urban scene reconstruction.

To alleviate piecewise planarity assumption, depth estimation method based on over-segmentation was proposed in [44]. A scene is represented as a collection of small fronto-parallel planar segments that do not correspond to semantically meaningful parts of the scene. The depth values are computed for entire segments rather than individual pixels using segment-based loopy belief propagation. This method makes no assumption about content planarity (i.e. it is not restricted to man-made environments) and is applicable for general scenes (e.g. containing irregular objects such as people, vegetation, etc.), c.f. Fig. 1(c). However, inaccurate estimation results may occur due to initial segmentation errors and violation of the assumption of constant depth within each segment (e.g. in presence of slanted or curved surfaces). To reduce these problems, segmentation prior is used less rigidly: while depth values within a segment are parametrized by a single surface, optimization is performed in a pixel-wise manner [37], [45].

PatchMatch Stereo method [46] demonstrates an effective way to handle slanted planes. Disparity map is over-parametrized by assigning each pixel to a planar surface specified by three parameters. The PatchMatch optimization strategy relies on random search and nearest-neighbor propagation. A random candidate plane is assigned to each pixel, and good guesses are propagated iteratively to the neighborhood maximizing a unary plane-induced photo-consistency term between the views. This allows to quickly find a good solution in the continuous disparity space, while evaluating the pixel-wise matching cost many fewer times than traditional plane-sweeping would require.

While being very efficient in terms of speed and memory, PatchMatch model demonstrates its ability to handle very challenging cases. Therefore, it has become a recent trend among sub-pixel accurate stereo methods. To further improve on speed efficiency, a number of modified propagation schemes that facilitate GPU-based implementation were proposed recently [47], [48], [49]. In [49], original PatchMatch is extended to a multi-view version that is coupled with a GPU-efficient diffusion-like propagation scheme. Although a relatively fast processing time is reported in [49] (~2.7 seconds per depth map), similar to other local stereo methods, PatchMatch methods tend to fail in the presence of textureless areas, Fig. 1(d). Consequently, PatchMatch Stereo was further developed into global models that explicitly incorporate pair-wise smoothness constraints to regularize the local neighborhood of disparity planes [47], [50]. Further, superpixel-based PatchMatch strategy was proposed in [51] that incorporates 'soft' segmentation prior: while the optimal plane for each pixel is estimated independently using unary matching cost, superpixels are used to facilitate random neighbor sampling and efficient collaborative cost aggregation allowing for an extended propagation range and computational speedup.

Several works propose to utilize the coarse-to-fine hierarchical strategy [9], [10], [52] in order to reduce the computational burden of dense depth reconstruction and cope with textureless regions and occlusions. Typically, initial dense depth estimates are calculated for low-resolution down-sampled images. Depth estimates at finer scales are initialized using up-scaled results from the lower resolutions and then refined. Such multi-scale approaches are able to produce denser reconstructions at reduced costs and, usually, large textureless regions can be handled correctly. However, the fine details and small objects are often lost at low resolution levels, and the sharp edges of the object boundaries can be compromised.

B. Contributions

In this work, we combine advantages of superpixel-based segmentation prior and multi-view PatchMatch stereo in order to develop a fast and accurate method that reconstructs dense depth maps simultaneously for all the views of a sparse light field. Namely:

1) We adopt superpixels as the basic primitives for depth estimation. Instead of assigning a depth value to each image pixel, we exploit the beneficial properties of superpixel segmentation and PatchMatch parametrization and model each superpixel as a planar patch in the image space that can be defined by a single depth value at the superpixel centroid and a vector orthogonal to the plane surface. Such representation scales well with the data size, thus facilitating GPU-based implementation. Based on the estimated plane parameters, pixels within the superpixel area are associated with smoothly varying continuous depth values. This improves reconstruction accuracy compared to constant depth quantization.

2) We formulate superpixel-specific smoothness and consistency terms that allow us to refine the plane assignments in superpixel domain. The major advantage of the superpixel-based optimization is that the number of the parameters to be optimized is much smaller than in pixel-wise labeling tasks. Another advantage is that the local smoothness term between superpixels imposes long-range spatial constraints in the pixel domain, thus improving the depth reconstruction robustness, especially in textureless and occluded regions.

3) We refine the plane assignments by propagating plane candidates from the superpixel neighbors and updating the

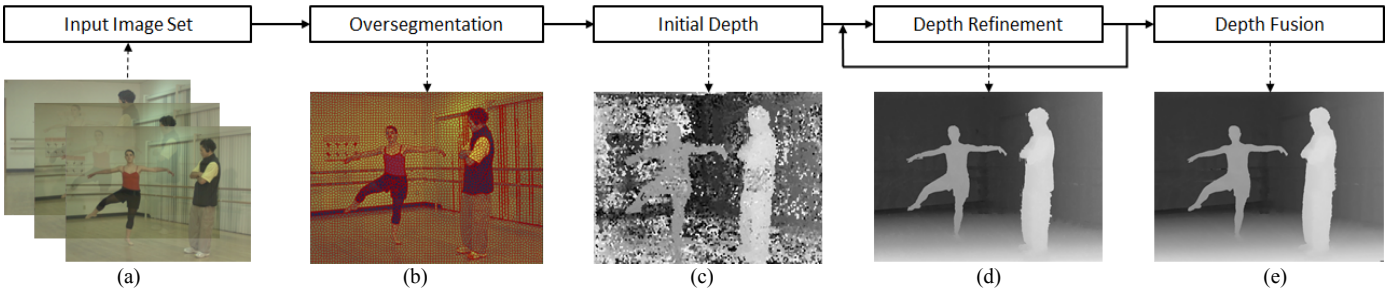


Fig. 2. Fig. 2. Algorithm workflow diagram. (a) Input images. The main stages of the algorithm are: (b) over-segmentation of the input images into superpixels (Section III.A), (c) initial depth estimation for each superpixel by plane sweeping (Section III.B), (d) iterative depth refinement for all the views simultaneously (Section III.C), and (e) final stability-based depth fusion to remove inconsistencies between the recovered depth maps (Section III.D)

current plane parameters whenever the energy function value is improved. To ensure rapid diffusion of the plane candidates across the image, together with the immediate superpixel neighbors, we sample additional plane candidates from more distant neighborhoods. In addition, we employ synchronous update schedule that allows processing of all the superpixels of all the images at once making full use of the parallel graphics hardware. This way, the distribution of the scene geometry becomes distinctive already after the first iterations, facilitating stability and fast convergence of the optimization procedure.

As a result, high-accuracy consistent and dense depth maps are obtained for each input view even in the presence of wide textureless regions and occlusions, c.f. Fig. 1(e).

III. Proposed Method

A. Overview

The workflow of our method and the effects of each processing stage are illustrated in Fig. 2. The input is a set of views from a calibrated camera system, and the output is a set of the corresponding depth maps. The input views are first over-segmented into a relatively high number of regions of homogeneous colors, called superpixels, c.f. Fig. 2(b). The assumption is that depth discontinuities coincide with the color boundaries and pixels within a superpixel area are likely to belong to the same object. The objective is to approximate the depth variation inside each superpixel by a planar patch.

We define a planar patch at each superpixel using 'point-normal' form of planar surface equation, as follows:

$$a(u-u_0)+b(v-v_0)+c(d-d_0)=0, \qquad (1)$$

where (u_0, v_0) are the image coordinates of the superpixel centroid and d_0 is its depth estimate, $\bar{n}=[a,b,c]^T$ is a vector orthogonal to the plane surface (the normal vector). To obtain the initial depth estimates at the superpixel centroids, plane-sweeping across the depth range of the scene is performed for each superpixel independently. The initial depth estimates contain many outliers mainly due to occlusions, ambiguous matching or shading variations, c.f. Fig. 2(c). In order to produce dense globally consistent depth maps, we iteratively refine plane parameters by rotating the planes around superpixel centroids and propagating best fitting planes across the image, so that smoothness between neighboring superpixels and cross-view consistency are maximized, c.f. Fig. 2(d). Superpixels are used as basic data units for the optimization, allowing for the desired speedup. Subsequently, we apply pixel-wise depth fusion in order to relax the planarity assumption and reduce the depth inconsistencies, which may occur e.g. due to initial segmentation errors, c.f. Fig. 2(e).

We assume that unrectified input images captured with a camera rig are provided along with reliable estimates of the camera calibration parameters, e.g. obtained with a typical structure from motion system such as VisualSFM [53]. For a set of rectified images, our method can be used likewise to estimate a set of corresponding disparity maps. The disparity value d_x and depth value z_x of pixel x are inversely proportional as follows: $d_x = f \cdot b/z_x$, where f is the camera's focal length and b is the baseline between the cameras. Due to this relation, we use the terms 'depth' and 'disparity' interchangeably in the following sections. It should be noted that in general different planar models would be recovered depending on whether the optimization is performed in depth or disparity space. However, the optimization procedure itself stays unchanged. The only difference is how the projection is performed: in case of disparity – via horizontal shifting, and in case of depth – via 3D warping using calibration parameters.

B. Initial Depth Estimation

Typically, matching algorithms require a large amount of memory, due to maintaining a cost volume associated with every possible disparity value. For high-resolution wide-baseline data, the disparity range can become prohibitively large in memory-constrained environments, including implementation on GPU. PatchMatch initialization strategy [46] is independent of the disparity range. Each pixel is initialized with a random depth value within the allowed continuous depth range. The assumption is that among a vast amount of depth samples, randomly drawn for each image pixel, there are likely to be good guesses that can be propagated to the neighboring pixels. However, the transition to superpixel-based image representation, where each superpixel is associated with a single depth value, greatly reduces the probability to sample correct depth, as the number of depth samples that can be tested and propagated is much lower compared to pixel-based representation. This motivates us to use a different, more data-driven, strategy in order to assign a potentially good initial depth value to each superpixel rather than follow the fully

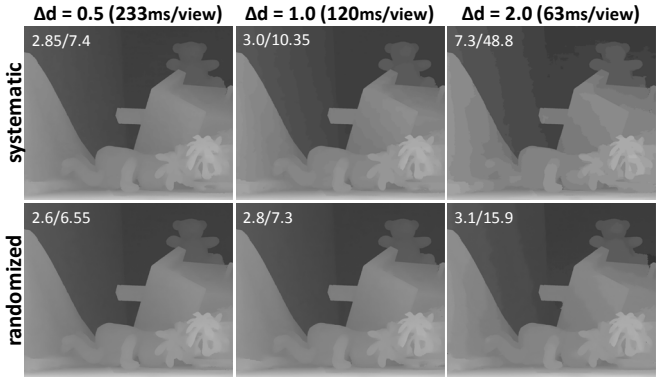


Fig. 3. Effect of plane-sweeping randomization. Top row results obtained using systematic depth hypothesis during initialization; bottom row - randomized depth hypothesis. The percentage of bad pixels for T=1/T=0.5 is marked at the image corners.

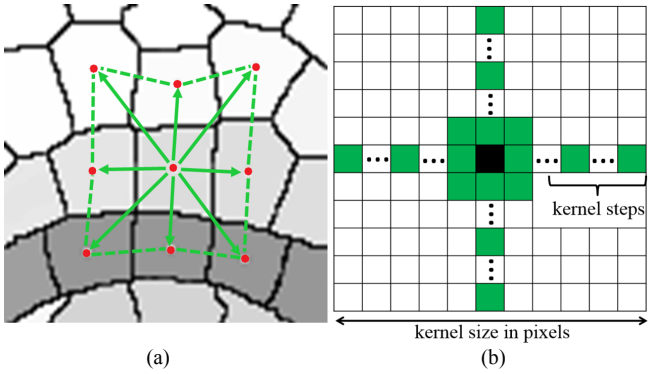


Fig. 4. (a) Candidate normal vectors are formed based on the depth estimates of the neighboring superpixels. (b) The propagation kernel structure.

randomized approach. Hence, unlike [46], we suggest a 'partly randomized' plane-sweeping strategy.

Evaluating multi-view photo-consistency for a large number of depth hypotheses is computationally expensive, therefore we aim at keeping the number of depth tests at each individual superpixel relatively low. On the other hand, the variety of the depth samples collectively tested across the whole image should be high for more accurate depth reconstruction. Therefore, instead of a dense vector of fixed quantized depths in the range $[d_{min}, d_{max}]$, a sparser vector of random values is tested at each superpixel, where each random value is drawn uniformly from the corresponding quantization interval. This way, we can alleviate the computational burden of the depth initialization without losing too much reconstruction accuracy, c.f. Fig. 3.

To find the best depth value in terms of multi-view photo-consistency, a plane is swept through the randomized vector of depth hypothesis for each superpixel independently. Pixels within the superpixel area are mapped to the neighboring views via plane-induced homography defined by the candidate depth d and the normal vector $\bar{n} = [0, 0, 1]^T$ (we consider fronto-parallel planes during the initialization). The visual dissimilarity between the pixels is measured using truncated squared difference of the intensities at corresponding pixel locations. Truncated cost is applied to limit the influence of outliers due to image noise, occlusions, and non-diffuse surfaces. The intensity differences are accumulated over the superpixel area and across the views, a depth candidate that yields the smallest cumulative photo-consistency cost is chosen as the initial depth estimate:

$$d_\Omega = \underset{d}{\operatorname{argmin}} \left(\sum_{i=1}^{N} \sum_{p \in \Omega} \min \left(T, \left(p - p_i(d, \bar{n}) \right)^2 \right) \right), \quad (2)$$

where d_Ω is the depth estimate for the superpixel Ω, N is the number of neighboring views used for the photo-consistency check, p is one of the pixels assigned to the superpixel Ω and $p_i(d, \bar{n})$ is its corresponding projection in the i^{th} view induced by the plane $(d, \bar{n})$, T is the threshold.

Our choice of starting with fronto-parallel planes, (assuming all pixels within each segment having a constant depth value), is driven by computational efficiency aim. However, the constant depth assumption is valid as long as the size of segments is relatively small. As the segment size increases, this assumption may be violated, especially for slanted surfaces, c.f. Fig. 1(c). We thus further estimate plane orientation for each superpixel during the depth refinement stage, where the depth estimates in the local neighborhood of the superpixel can be utilized to better guess possible normal vector candidates (see Section III.C).

C. *Iterative Refinement*

Since the initial depth maps are computed independently for each view, statistical correlation between the depth maps of different views is not exploited. In addition, whereas smoothness is implicitly enforced between the pixels belonging to the same superpixel via the segmentation prior, smoothness between neighboring superpixels that belong to the same object is not considered. In this step, the initial depth maps are jointly refined via iterative optimization, in belief-propagation fashion, in order to maximize depth consistency between the views and enforce smoothness between the neighboring superpixels. The effect of the iterative refinement is illustrated in Fig.5.

Optimization of smoothness and consistency constraints is commonly applied in piece-wise planar scene reconstruction. Examples include [40], where a set of dominant scene planes is defined based on plane-fitting to sparse 3D points and line segments, [42], where only three orthogonal scene directions are considered, and [44], where the scene is modelled as a collection of fronto-parallel planar segments. In our approach, the optimization is not limited to a set of predefined plane candidates. Instead, at each iteration additional plane candidates are detected for each superpixel based on depth estimates of neighboring superpixels. This is different from the PatchMatch methods [46], [49], [50] that rely on random plane generation for initialization and plane refinement.

Each view is modeled as an undirected graph G whose nodes correspond to the superpixels and the state of each node holds the plane parameters assigned to the respective superpixel, namely, depth d at the superpixel centroid and the plane normal vector $\bar{n} = [a, b, c]^T$, and corresponding smoothness and consistency values. We measure the quality of the plane assignment at a superpixel using the following energy function:

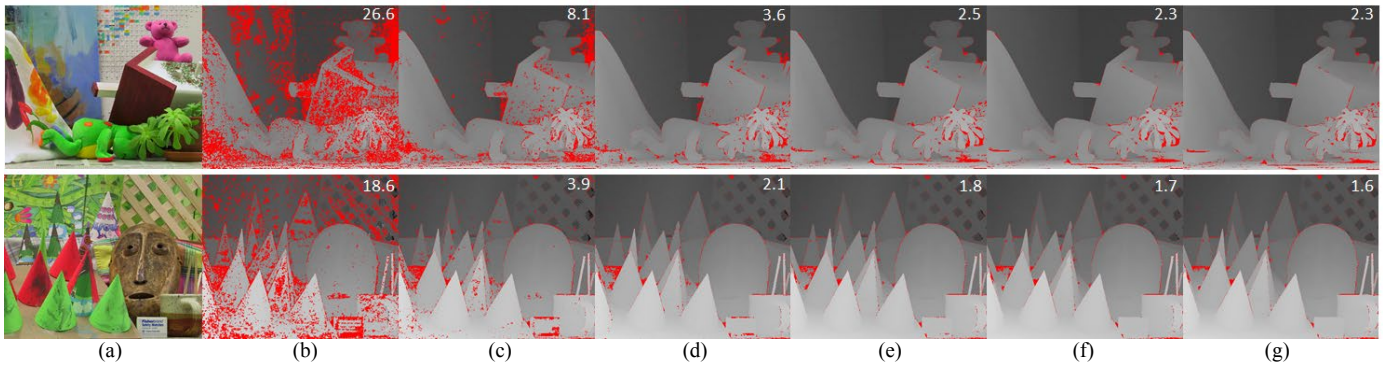


Fig. 5. Refinement effect. Disparity estimation results for Middlebury 'Teddy' and 'Cones' overlaid with the error map (threshold T = 1.0), the percentage of bad pixels is marked at the image corners. (a) One of the input images. (b) Initial disparity estimation by plane sweeping and (c)–(g) after first five refinement iterations.

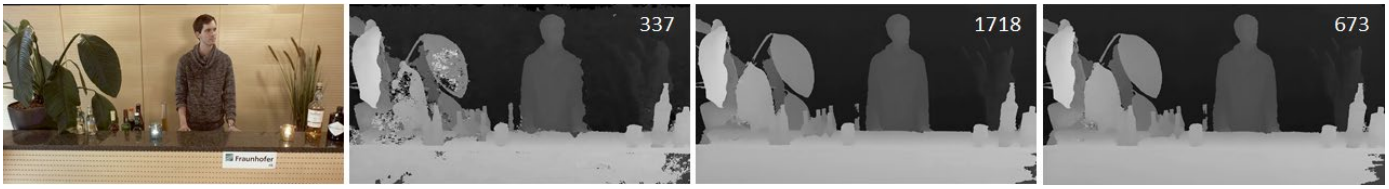


Fig. 6. Propagation kernel. (a) One of the input views. (b) Estimation result when only nearest neighbor superpixels are used for propagation. (c) Estimation result when the propagation kernel with constant size is applied. (d) Estimation result when propagation kernel is shrinking at each iteration. The corresponding runtime of the refinement step in *ms/view* is marked at the image corner. Note, the resulting depth map when using the shrinking kernel is very similar to the depth map produced when the constant size kernel is applied, whereas the runtime is much faster.

$$E(d,\bar{n}) = E_c(d,\bar{n})\, E_s(d,\bar{n}), \tag{3}$$

where $E_c(d,\bar{n})$ is a consistency term and $E_s(d,\bar{n})$ is a smoothness term. Here, the multiplication of the energy terms forces a plane estimate to meet both conditions simultaneously.

At each iteration, two update steps are performed: plane propagation and plane refinement. During the *propagation step*, the plane estimates are propagated across the image in a diffusion-like manner. The candidate plane estimates coming from the neighboring nodes are used to evaluate the energy function at the superpixel location. The current state is updated whenever using candidate plane parameters improves the energy score, $E(d,\bar{n}) < E(d(d',\bar{n}'),\bar{n}')$, where d and $\bar{n}$ are the current plane parameters, and d' and $\bar{n}'$ are the parameters of the candidate plane. Here, $d(d',\bar{n}')$ is a new depth value at the superpixel centroid given by the candidate plane equation.

During the *plane refinement step*, the superpixels are rotated around their centroids in order to introduce slanted planes into the propagation process and mitigate the fronto-parallel bias of the initial setting. For each superpixel we check eight candidate normal vectors. Each vector is defined as a normal vector to a triangle formed by three vertices, including centroid of the reference superpixel and centroids of its two adjacent nearest neighbors, as illustrated in Fig. 4(a). The current normal vector is updated if a candidate vector improves the energy score, $E(d,\bar{n}) < E(d,\bar{n}')$, where $\bar{n}$ is the current and $\bar{n}'$ is the candidate normal vectors. Compared to the PatchMatch strategy [46], this approach is more efficient in finding possible plane candidates.

1) Propagation kernel

The lack of visual cues in textureless regions may lead to ambiguous matching no matter how many images are utilized for depth estimation. Therefore, it is important to ensure rapid plane propagation over wide textureless regions in order to avoid locally optimal solutions corresponding to false depth estimations, c.f. Fig. 6(b). To accelerate the diffusion process, we look beyond the immediate neighbors of the superpixel and sample additional plane candidates from a larger neighborhood. The superpixel nearest neighbors together with more distant candidates form a *propagation kernel* as depicted in Fig. 4(b). We define the propagation kernel by two parameters: 'kernel size' and number of 'kernel steps'. 'Kernel size' defines the size of the image region (in pixels) in the vicinity of the reference superpixel from where additional superpixel neighbors are sampled for propagation. 'Kernel steps' defines the number of additional superpixels to be sampled in each direction from that region. Fig. 4(b) depicts the topology of the propagation kernel schematically to show that: (a) the samples are taken in four directions (up, down, left, right); (b) the nearest 8 superpixels are always considered; and (c) the furthest sample is taken at the edge of the chosen vicinity. Thanks to a larger neighborhood, the distribution of the scene geometry becomes distinctive already after the first few iterations, facilitating stability and fast convergence of the refinement procedure, see Fig. 6(c).

It is beneficial to use a bigger propagation kernel in the presence of wide textureless regions; however, this requires more computational time as more candidates are checked during the propagation step. In order to balance between computational cost and diffusion rate, the extents of the

propagation kernel are 'shrinked' at each iteration as follows:

$$\begin{aligned} Size_I &= Size_{init}/I \\ Steps_I &= Steps_{init}/I, \end{aligned} \quad (4)$$

where $Size_{init}$ and $Steps_{init}$ are initial kernel parameters, $Size_I$ and $Steps_I$ are the modified parameters defining the propagation kernel at the I^{th} iteration. The idea is to start with a rather extended kernel that allows to diffuse the information far enough already at the first iteration. Subsequently, kernel size and number of samples are decreased linearly at each iteration, and eventually, only candidates from the nearest neighborhood are sampled. This provides a good compromise between the average cost of each iteration and the number of iterations required for refinement procedure to converge, c.f. Fig. 6(d).

2) Smoothness Term

The smoothness term is used to enforce spatial smoothness of the depth maps by penalizing inconsistencies between neighboring superpixels with a similar color (superpixel color is defined as mean color of pixels assigned to a superpixel). In order to enforce superpixel-based smoothness, we measure how well the plane assignment of a superpixel fits the point cloud formed by the centroids of the neighboring superpixels. The current plane assignment of a superpixel is used to evaluate a depth value at the image coordinates of each neighboring centroid and the difference between the evaluated depth and the current depth assignment at the neighboring centroid is penalized. The smoothness term is computed as a sum of pairwise consistency measurements between the superpixel and its neighbors weighted based on the color similarity, as follows:

$$E_s(d,\bar{n}) = \frac{1}{\sum_{i=1}^{M} \omega(\mu_\Omega,\mu_i)} \sum_{i=1}^{M} \omega(\mu_\Omega,\mu_i)\, s_i(d_i, d_i(d,\bar{n})) \quad (5)$$

$$\omega(\mu_\Omega,\mu_i) = e^{-(\mu_i-\mu_\Omega)^2/2\alpha^2} \quad (6)$$

$$s_i\big(d_i, d_i(d,\bar{n})\big) = e^{-\left(d_i - d_i(d,\bar{n})\right)^2/2\sigma^2}, \quad (7)$$

where M is the number of superpixel neighbors, $\omega(\mu_\Omega,\mu_i)$ is the similarity weight between the superpixel color μ_Ω and the color of its i^{th} neighbor μ_i, and $s_i\big(d_i, d_i(d,\bar{n})\big)$ is the consistency measurement between the current estimated depth d_i at the centroid of the i^{th} neighbor and the value $d_i(d,\bar{n})$ obtained at the centroid of the i^{th} neighbor given the current plane assignment of the superpixel. Color similarity $\omega(\mu_\Omega,\mu_i)$ is evaluated in the form of a Gaussian function, where α denotes the standard deviation and controls the influence of the color difference between superpixels. The similarity weight, thus, varies from zero to one and is equal one when the two colors are the same. Likewise, the consistency measurement $s_i\big(d_i, d_i(d,\bar{n})\big)$ is evaluated using a Gaussian function.

Compared to the local pixel-wise smoothness constraints [50], a superpixel neighborhood covers a larger image area imposing a more rigid spatial constraint. This helps to improve the depth reconstruction robustness in presence of wide textureless and occluded regions.

3) Consistency Term

The consistency term enforces cross-view consistency of the depth maps based on the projection relationships between the input views. In case of pixel-wise depth estimation, the consistency value is usually accumulated over a small patch of pixels surrounding the reference pixel in order to limit the influence of the outliers and image noise. For a superpixel-based representation, it is reasonable to exploit a superpixel itself as a patch since it naturally integrates pixels that likely correspond to the same surface and, therefore, facilitates more accurate handling of depth discontinuities. Thus, we define the consistency term of a superpixel as the sum of the consistency measurements for the pixels in the superpixel.

Given the current plane estimates of the superpixels, for each pixel, the corresponding projection in the other view is found via plane-induced homography. To enforce the geometric consistency, the difference between the depth that is used to project the pixel and the depth value at the corresponding projection location is penalized. This reflects the fact that if the point is visible in multiple images, it should have the same depth value assigned in the respective depth maps. However, due to occlusions, the visibility assumption does not always hold true. Some pixels with correctly recovered depths may be projected onto an occluding surface, which results in an undue penalty. Similar to [44], to reduce the influence of occlusions, we explicitly account for possible occluded regions and formulate the consistency term as a sum of two terms:

$$E_c(d,\bar{n}) = \frac{1}{N}\sum_{i=1}^{N}\big(V_{\Omega\to i}(d,\bar{n}) + O_{\Omega\to i}(d,\bar{n})\big), \quad (8)$$

where N is the number of neighboring views, Ω is the superpixel area, $V_{\Omega\to i}(d,\bar{n})$ is the visibility term that accounts for the matching of visible pixels, and $O_{\Omega\to i}(d,\bar{n})$ is the occlusion term that accounts for possible occlusions.

If a superpixel from the reference view is visible in the i^{th} neighboring view, both intensity similarity and geometric consistency constraints should be satisfied simultaneously. Thus, we define the visibility term as follows:

$$V_{\Omega\to i}(d,\bar{n}) = S_{\Omega\to i}(d,\bar{n})\, C_{\Omega\to i}(d,\bar{n}), \quad (9)$$

where $S_{\Omega\to i}(d,\bar{n})$ is the intensity similarity between the superpixel color and the corresponding projection area in the i^{th} view, and $C_{\Omega\to i}(d,\bar{n})$ is the geometric consistency between visible pixels. We estimate the intensity similarity as follows:

$$S_{\Omega\to i}(d,\bar{n}) = \frac{1}{|\Omega|}\sum_{p\in\Omega}\omega\left(\mu_\Omega, \mu_{\Omega\to i}\big(p_i(d,\bar{n})\big)\right), \quad (10)$$

where $|\Omega|$ is the number of pixels in the superpixel, $\omega(\mu_\Omega,\mu_{\Omega\to i})$ is defined as in (6) and represents the similarity weight between the reference superpixel color μ_Ω and the corresponding superpixel color $\mu_{\Omega\to i}$ in the i^{th} view, which is defined by the projection $p_i(d,\bar{n})$ of the pixel p.

Let d_p be the depth value at the pixel p in the reference view, P be the 3D point corresponding to the pixel p, and $d_{p\to i}$ be the

depth value of the corresponding projection of P to the i^{th} view. If $d_p \leq d_{p\to i}$, point P observed by the reference camera is closer than the corresponding point in the i^{th} view, thus P should be also visible in the i^{th} view and the difference between the two depth values should be penalized to enforce the geometric consistency. Denoting $X = \{p \in \Omega \,|\, d_p \leq d_{p\to i}\}$, a set of pixels that should be visible in the i^{th} view, we estimate the geometric consistency as follows:

$$C_{\Omega\to i}(d,\vec{n}) = \frac{1}{|X|}\sum_{p\in X} e^{-\left(d_p - d_{p\to i}\right)^2/2\sigma^2}, \qquad (11)$$

where $|X|$ is the number of pixels that should be visible in the i^{th} view, σ is the standard deviation of the Gaussian function.

If $d_p > d_{p\to i}$, i.e. point P observed by the reference camera is behind the corresponding point in the i^{th} view, it can either indicate that the current depth estimate d_p is inconsistent between the two views or that point P is occluded in the i^{th} view.

To account for the occlusion case, we estimate the likelihood of a superpixel to be occluded in the other views and increase the consistency term accordingly. Since occlusions usually occur due to depth discontinuities, we utilize the local color gradient of superpixels to identify those superpixels that might be located at object boundaries and, thus, are more likely to be occluded in other views. Denoting $Y = \{p \in \Omega \,|\, d_p > d_{p\to i}\}$, a set of possibly occluded pixels, and using $\eta = 0.5$ as a constant regularizer, we define occlusion term as follows:

$$O_{\Omega\to i}(d,\vec{n}) = \begin{cases} \eta\left(1 - \min\limits_{0\leq i\leq M} \omega(\mu_\Omega,\mu_i)\right), & Y \neq \emptyset \\ 0, & Y = \emptyset, \end{cases} \qquad (12)$$

where M is the number of neighboring superpixels, $\omega(\mu_\Omega,\mu_i)$ is defined as in (6) and represents the similarity weight between the superpixel Ω and its i^{th} neighbor.

D. Depth Fusion

After the depth refinement step, the recovered depth maps may still contain some inconsistencies. Inconsistent estimates mainly occur e.g. due to initial segmentation errors, violation of the planarity assumption, or regions with a view-dependent appearance, such as shadows and reflections. Accumulating evidences from multiple views allows to detect and fix most of these cases. Thus, we apply pixel-wise depth fusion in order to further reduce the depth inconsistencies and relax the planarity constraint. As the geometric consistency between the views is properly exploited during the refinement stage, a rather simple fusion scheme can be applied to merge the depth maps into a consistent 3D point cloud. We use a stability-based fusion method proposed in [54]. Each image in turn is declared as a reference view. Pixels from the rest of the views are projected to the reference camera viewport. As a result, each pixel of the reference view is associated with one or more depth candidates. For each non-zero depth candidate, a stability value is obtained by counting the number of depth candidates that agree with the current candidate (increasing stability value) and the number of those that do not (decreasing stability). In the end, the closest depth with non-negative stability is retained, c.f. Fig. 2(e).

E. Implementation Details

The inherent parallelism of our method as well as linear storage requirements enable its efficient and scalable GPU-based implementation. To segment the images, we use Simple Linear Iterative Clustering (SLIC) algorithm proposed in [55]. In particular, we use the GPU-based SLIC version provided by [56], which we also have re-implemented using GLSL. SLIC segmentation begins with sampling K regularly spaced cluster centers that form a regular 2D grid. After a few clustering iterations, compact and roughly equally-sized superpixels are produced thanks to compactness constraint. We observe that the final superpixel grid stays fairly regular in superpixel domain, i.e. in most cases a superpixel has eight adjacent neighbor segments corresponding to the adjacent seeds. However, such regularity is not guaranteed and there are regions (e.g. containing structure/texture) where default neighbor connectivity is not preserved. In [51] an adjacency graph is created to determine superpixel connectivity. In contrast to this, in our approach, we ignore the fact that the initial regularity of superpixel grid might have changed and utilize the initial 2D indices to fetch the samples during the propagation stage. This way, each superpixel is associated with a unique 2D index corresponding to its seed, which prevents any indexing error. The price is that superpixels with adjacent indices might not always be spatially adjacent. Consequently, the shape of the propagation kernel, c.f. Fig. 4(b), might be slightly distorted in some places. This, however, does not dramatically affect the reconstruction results, whereas such simplification allows us to avoid creating and storing superpixel adjacency graph, which is especially beneficial in case of GPU-based implementation.

In addition to SLIC, there are other segmentation algorithms, which might provide certain preferable properties over SLIC superpixels. The method proposed in [57] guarantees regularity of the produced superpixel grid. According to the benchmark study [58], it provides the best quantitative results among 28 state-of-the-art superpixel algorithms. Furthermore, its CPU-based implementation is fast [57]. However, it relies on priority queue formulation, which impedes its implementation on a GPU. The superray segmentation algorithm proposed in [59] is specifically tailored for light field processing. It is the counterpart of SLIC segmentation for light fields. Its major advance is that perceptually similar and corresponding pixels are grouped across several views. To perform such grouping, depth estimates for the superray centroids are utilized. Further clustering steps substantially rely on the accuracy of the initial depth estimates. In general, sparse accurate depth estimates are hard to acquire due to e.g. wide baseline, occlusions or textureless regions. Although cross-view correspondence of the segments would be beneficial for dense depth estimation, in our case, utilizing this segmentation approach would result in a 'Chicken and Egg' problem: segmentation is needed for accurate depth estimation, while accurate depth estimates are needed to perform segmentation. Therefore, we opted using SLIC segmentation as it suits our task best. Furthermore, the accurate depth estimates produced by our method can be

TABLE I
DATASETS, SETTINGS AND TIMINGS MEASURED ON NVIDIA QUADRO M1000M GRAPHICS CARD (TIME IN MILLISECONDS/VIEW)

Dataset	Number of Views	View Resolution	Disparity Quantization Levels	Superpixel Size	Segmentation Time (ms/view)	Plane Sweeping Time (ms/view)	Refinement Time (ms/view)	Fusion Time (ms/view)	Total Time (ms/view)
Teddy/Cones	9	1800×1500	80	10	168	350	652	99	1265
Truck	3×3	1280×960	80	8	77	104	368	39	588
Bracelet	3×3	1024×640	70	8	49	39	178	21	278
Jelly Beans	3×3	1024×512	55	10	49	32	107	16	204
Unicorn	5×5	1920×1080	150	8	153	154	812	465	1584
Bar	3×5	1920×1080	45	8	120	102	673	148	1023
Beer Garden	3×3	1920×1080	30	8	127	112	614	80	933

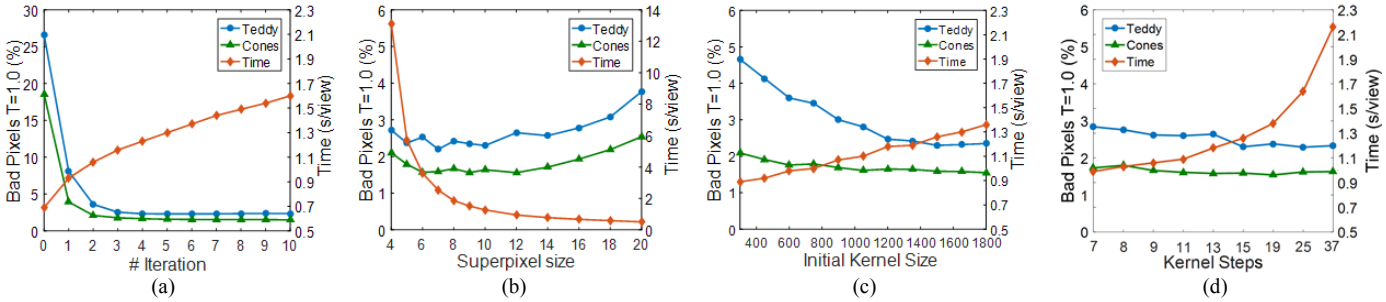


Fig. 7. Sensitivity to parameter settings as time/accuracy trade-off. Each plot depicts the error rate (blue and green) and the corresponding computation time (red): (a) the number of iterations; (b) varying superpixel size; (c) initial size of the propagation kernel; and (d) initial number of kernel steps.

utilized to perform superray segmentation for further light field processing.

All the steps that have been introduced in the previous subsections are implemented on GPU using GLSL compute shaders and texture arrays. The initial plane-sweeping is performed in parallel for all superpixels in all views. The refinement step is designed as a single compute render pass, which is called in a loop for a specified number of iterations. A diffusion-like propagation scheme, where half of the image pixels are updated simultaneously, was proposed in [49]. Instead, we employ synchronous update schedule in order to fully parallelize the message passing during the plane propagation. At the first iteration, a read-only texture holds the initial depth and normal estimates for each superpixel. During the refinement step, an additional write-only texture is allocated to hold the updated parameters, whereas the energy function is evaluated according to the initial plane estimates. After each iteration, these two textures are interchanged in a 'ping-pong' manner: the output texture holding updated estimates from the previous iteration is used as the input read-only texture for the current iteration, while the texture holding old parameters is overwritten by the new estimates of the current iteration. This way, at each iteration all the superpixels in all the images are refined independently in parallel. Thus, the computation time of the refinement step is linear with respect to the total number of superpixels and number of refinement iterations, and inversely proportional to the number of parallel GPU threads. Finally, the depth maps fusion consists of two compute render passes, which are executed for each view in turn. The first pass performs the projection, whereas the second pass performs the stability computation.

IV. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed depth reconstruction method, we have performed experiments on several publicly available light field and multi-view datasets:

1) Middlebury multi-view 'Teddy' and 'Cones', c.f. Fig. 4, each containing 9 rectified views [60], [61];
2) Stanford light fields 'Truck', 'Bracelet', and 'Jelly Beans', c.f. Fig. 11, each containing 17×17 rectified views [62];
3) ULB 'Unicorn', c.f. Fig. 12, containing 5×5 views from a calibrated camera array [63];
4) Fraunhofer 'Bar' and 'Beer Garden', c.f. Fig. 15, containing dynamic light field video sequences taken with 3×5 and 3×3 camera array respectively [9].

Some specifications of the test datasets, such as number of views and spatial resolution, are summarized in Table I.

A. Parameter Settings

Several parameters in the proposed method allow tuning of the time/accuracy trade-off for a particular application. In order to assess the sensitivity of the reconstruction results to parameter variations, we run multiple tests on the 'Teddy' and 'Cones' datasets. As a quality measurement, we use the percentage of bad pixels with the error threshold $T = 1.0$. For various parameter settings, the time/accuracy trade-off results are shown in Fig.7.

Fig. 7(a) illustrates the convergence of the refinement process with an increasing number of iterations. It can be observed that the error rate quickly drops after a few iterations. The biggest drop occurs already after the first iteration and continue decreasing steadily as the iterations go on. After 3-4 iterations, the changes are marginal. This also holds true for

other datasets tested in this section. Fast convergence rate, especially at the first iteration, indicates that our improved propagation strategy (Section III-C.1) ensures rapid plane propagation over the textureless and occluded regions and facilitates reconstruction of true scene geometry.

For a fixed number of iterations, the influence of increasing superpixel size is shown in Fig. 7(b). Based on the error rate curves, we can conclude that, in general, a smaller superpixel size corresponds to a better accuracy due to the improved adherence to object boundaries and approximation of curved surfaces. However, very small superpixels may be easily influenced by noise (see a slight error rate increase for superpixel size smaller than 7×7) and are associated with a longer runtime due to increased number of primitives. However, with increasing superpixel size the increase in error rate is rather slow.

We further study how the shape of the propagation kernel affects the reconstruction results. Parameter 'kernel steps' define the number of candidate planes that are sampled in each propagation direction (sampling density), whereas the 'kernel size' defines the spatial extent of the sampling area. For a fixed sampling density (samples are taken at every five superpixels), the effect of expanding propagation kernel is shown in Fig. 7(c). As can be observed, correlation between the reconstruction accuracy and the kernel size exhibit different behavior depending on a dataset. While increasing kernel size has little effect for 'Cones' dataset, bigger propagation kernel noticeably improves accuracy in case of 'Teddy' dataset. With increasing size of the propagation kernel, the information is propagated faster and further which hinders the optimization procedure from converging to a false locally optimal solutions. This is especially important in presence of low-textured areas, such as in 'Teddy' scene. Whereas for cluttered and highly textured scenes, such as 'Cones', a smaller propagation range is enough.

Finally, we examine the influence of the increasing number of kernel steps on the reconstruction quality. Intuitively, the more candidates that are checked during each iteration, the higher the chances are to find a good match. However, as can be seen in Fig. 7(d), even few additional samples in each propagation direction already yield good reconstruction results due to fast diffusion rate of the parallel execution. While denser sampling does not result in a substantial gain in accuracy, it comes at a price of a longer runtime per iteration.

To summarize, we can conclude that the performance of our method is rather stable; different parameter settings in reasonable ranges do not drastically worsen the reconstruction accuracy. The selection of all parameters can be based on the following rules of thumb. Our algorithm converges within 3-4 refinement iterations. Therefore, we use five iterations in all our further experiments. The superpixel size is driven by two factors: performance and speed. A size within the range of 7-16 pixels provides good depth reconstruction results. In our experiments, we use size of 8-10 to achieve the target accuracy level in a reasonable time. When the time constraint is more important, one can use a bigger superpixel size and still obtain quite accurate reconstruction. For example, using superpixel size 18 for the Middlebury datasets, the depth reconstruction is performed in less than 0.5 second per view (for full size 1800×1500), while the bad pixel percentage rate stays under 3%. The size of the propagation kernel is sensitive to the image size and content of the scene. We suggest adopting a big kernel to ensure a good propagation rate at the first iteration. In all our experiments, we set 'kernel size' equal to the smallest dimension of the input image and 'kernel steps' such that samples are taken every five superpixels. This is a 'conservative' setting from a speed perspective, and still provides a good time/accuracy trade-off: as the kernel size and the number of kernel steps are decreasing linearly at each iteration, the runtime is affected only moderately.

B. Performance

All our experiments were conducted on a laptop equipped with an Intel Core i7 2.6 GHz CPU and an Nvidia Quadro M1000M graphics card. As shown in Table I and Fig.7, the runtime of our algorithm depends on the number and resolution of the input views and the following parameter settings: number of refinement iterations, superpixel size, and number of kernel steps. These parameters were set following the discussion in the previous subsection and are summarized in Table I along with the runtime of each processing step. To generate depth maps for a multi-view frame containing 15 full HD views (1920×1080), it takes about 15.5 seconds, i.e. about 1 second per 2 megapixel view. To generate depth maps for 9 views (1024×512), it takes about 1.3 seconds, or about 0.15 seconds per 0.5 megapixel view. For comparison, in [48], the authors evaluated the performance of several GPU-based PatchMatch methods (results were obtained with Nvidia GTX 280). As reported, the best runtime was achieved by a GPU-based version of the original PatchMatch stereo method, which takes 1.8, 2.4, and 3.5 seconds to process 0.1, 0.2, and 0.3 megapixel data respectively. To produce a 2 megapixel depth map the GPU-based multi-view PatchMatch method proposed in [49] takes about 50 seconds with the settings tuned for accuracy and about 2.7 seconds with the settings tuned for speed (using an Nvidia GTX 980 graphics card). Different settings for our algorithm allow the estimation of high-resolution depth maps in less than a second. For example, as can be seen in Fig.7(b), using superpixel size 18, the depth maps are estimated in about 0.5 seconds per view while the error rate stays low (within 3%).

C. Energy Function Analysis

To evaluate the impact of the energy function terms on the reconstruction accuracy, we obtain the depth reconstruction results by omitting a single term from the energy function formulation, (3) and (8). First, the smoothness term is omitted and only the consistency measurement between the views is optimized, c.f. Fig. 8(a). Without the smoothness term, the refinement process fails to resolve the matching ambiguities and converges to a local sub-optimal solution, leading to a significant degradation in the depth reconstruction quality. Second, the consistency term is left out and refinement is performed purely based on the spatial smoothness term, c.f. Fig. 8(b). The resulting disparity maps are of much better quality, but not as good as that obtained by the full energy optimization.

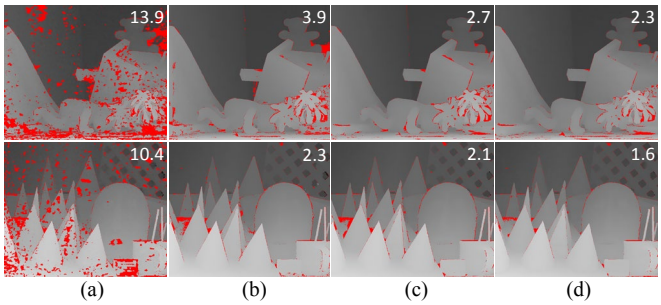


Fig. 8. Impact of the energy function terms. Omitting a single term from the energy function: (a) without smoothness; (b) without consistency; (c) without occlusion term; (d) proposed energy function, where all terms are included.

This shows that the smoothness term is very effective in resolving the difficulties associated with textureless regions and repetitive patterns; however, it cannot fully eliminate the errors in the occluded areas near object edges and image boundaries. In these cases, incorporating the geometric consistency term can help to pinpoint the right solution by exploiting depth-matching cues from multiple views, leading to improved accuracy, c.f. Fig. 8(c). Finally, the occlusion term further improves the overall accuracy of result by balancing out undue consistency penalty at the objects boundaries, c.f. Fig. 8(d).

D. Reconstruction Accuracy

We evaluate the reconstruction accuracy against the ground truth disparity maps of the Middlebury stereo benchmark [60], [61]. The quantitative results are summarized in Table II, providing percentage of bad pixels over non-occluded pixels (*'nocc'*), all pixels (*'all'*), and pixels near discontinuities (*'disc'*). The reconstructed disparity maps overlaid with the error maps are shown in Fig. 9(b). To emphasize the accuracy of our results, we also created 3D point clouds, Fig. 9(c): both planar and curved surfaces are recovered faithfully; the depth discontinuities are well aligned with the object boundaries.

As opposed to stereo PatchMatch methods, where only two views are used for disparity estimation, our method is designed for fast and accurate multi-view reconstruction (sparse light field data). We use all 9 views as an input, and 9 depth maps are produced as an output. Nevertheless, we use PatchMatch stereo methods as a reference due to their high accuracy. Specifically designed to tackle slanted surfaces with sub-pixel precision, these methods provide the state-of-the-art reconstruction results on the sub-pixel accuracy level. For the error threshold $T = 1.0$, our method achieves the disparity accuracy better than the reference methods in all cases. Moreover, for the sub-pixel accuracy level, $T = 0.5$, our results are comparable to or better than the reference results. Since more views are used as an input, our method is able to recover the occluded areas and, thus, significantly outperform the PatchMatch methods on *'all'* measurement. This demonstrates that the multi-view information is successfully utilized to handle occlusions.

E. Baseline Effect

We use the Stanford light field dataset [62], providing a number of dense light fields of 17×17 views, to test the effect of increasing baseline between the views on the performance of our algorithm. In our experiment, we sub-sample the 17×17 image array by skipping 1, 3, and 5 views in horizontal and vertical directions, obtaining 9×9, 5×5, and 3×3 image arrays respectively, and compute three sets of disparity maps with increasingly wider baselines between the views, c.f. Fig. 10.

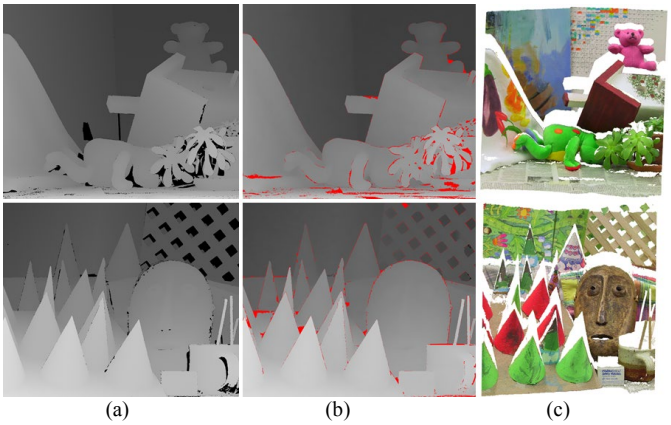


Fig. 9. Qualitative evaluation on the Middlebury Teddy and Cones datasets. (a) Ground-truth disparity map. (b) Our result overlaid with the error map (threshold T = 1.0). (c) Colored 3D point cloud.

TABLE II
PERCENTAGE OF BAD PIXELS RESULTS ON THE MIDDLEBURY DATASET

THRESHOLD T = 1.0

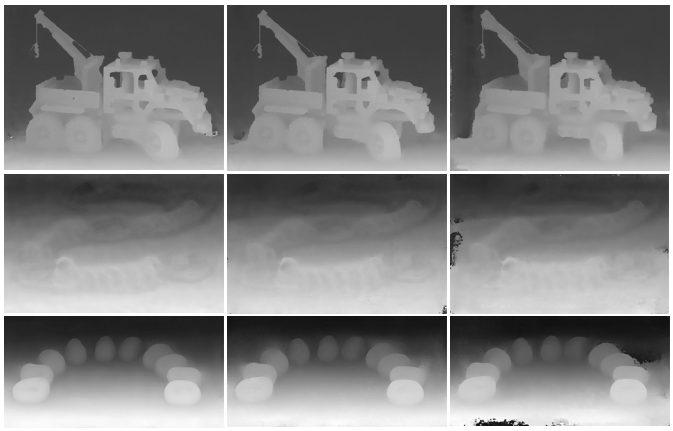
Method	Teddy			Cones		
	nocc	all	disc	nocc	all	disc
PM Stereo [37]	2.99	8.16	9.62	2.47	7.80	7.11
PMBP [41]	2.88	8.57	8.99	2.22	6.64	6.48
PMF [42]	2.52	5.87	8.30	2.13	6.80	6.32
PM-Huber [38]	3.38	5.56	10.70	2.15	6.69	6.40
PM-PM [39]	3.00	8.27	9.88	2.18	6.43	6.73
Ours	**1.96**	**2.56**	**6.55**	**1.93**	**2.72**	**5.61**

THRESHOLD T = 0.5

Method	Teddy			Cones		
	nocc	all	disc	nocc	all	disc
PM Stereo [37]	5.66	11.80	16.50	3.80	10.2	10.2
PMBP [41]	5.60	12.00	15.50	3.48	8.88	9.41
PMF [42]	**4.45**	9.44	13.70	2.89	8.31	8.22
PM-Huber [38]	5.53	9.36	15.90	**2.70**	7.90	**7.77**
PM-PM [39]	5.21	11.90	15.90	3.51	8.86	9.58
Ours	4.49	**5.33**	**12.77**	3.09	**4.62**	7.89

As the ground truth disparity is not available, we evaluate the reconstruction quality by comparing synthesized virtual views with unused intermediate views. Namely, we use the positions of every second view of the central row of the original dense light field, i.e. 8 views overall. We synthesize virtual views by projecting all the input views of the dataset onto the target image plane. At each pixel of the target image, a simple blending procedure of samples is performed. We use the distance between the views to derive blending weights and the reconstructed disparity maps to resolve the occlusions. Fig. 11 shows the rendering results for the 7th view of the central row.

The synthesized views are compared to the corresponding original views using structural similarity index (SSIM) [64]. Table III provides average SSIM scores over the 8 synthesized views. For comparison, we also provide the average SSIM score over the 8 views that were synthesized without disparity (i.e. all disparities were set to zero). As can be seen, utilizing disparity



(a) (b) (c)

Fig. 10. Baseline effect experiment on the Stanford 4D light field dataset. Disparity map of the central view obtained using (a) 9×9, (b) 5×5 and (c) 3×3 image array.

data significantly improves the rendering quality, especially in the case of the sparse 3×3 image array. When disparity maps are used, the SSIM values obtained using sparser (5×5 and 3×3) image arrays are fairly high and very close to those obtained using 9×9 image array. This demonstrates accuracy of the reconstructed disparities as well as the robustness of our method against the varying baseline of the input data.

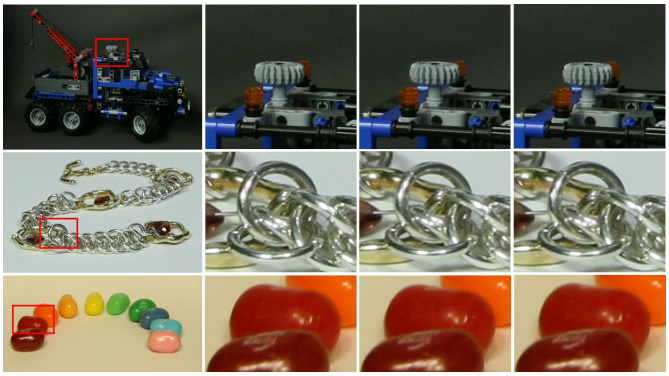
F. Comparison with Other Methods

We compare the proposed method against several established depth reconstruction approaches, including the MPEG depth estimation reference software (DERS) [65], a state-of-the-art semi-global-matching based method (SGM) [66], and an efficient multi-scale correspondence algorithm proposed in [9].

1) Comparison with Reference Software

We first compare our results with DERS version 6.1. DERS is the state-of-the-art depth estimation technique based on Graph Cut optimization. We assess the depth reconstruction quality trough virtual view synthesis. We use the ULB 'Unicorn' light field dataset that contains views from a 5x5 camera array with additional intermediate views between each pair of cameras. The intermediate views are used as ground truths for the view synthesis quality evaluation. We estimate DERS depth maps using the general reconstruction mode with quarter-pixel precision. Examples of the depth maps generated using DERS and our method are given in Fig. 12. Subjectively, the depth maps reconstructed by our method look more accurate and detailed, and even when DERS fails to find correct solution (e.g. the regions where the objects and the background have very similar colors), our method can produce reasonable results.

With the depth maps obtained by DERS and by our method, virtual views are rendered at the intermediate positions of each row of the camera array (20 views overall). We use view synthesis reference software (VSRS) [67] version 4.2., where two reference views, left and right, and two corresponding reference depth maps are used to synthesize a virtual view. We synthesize virtual views in the general synthesis mode, applying quarter-pixel precision and boundary noise removal. Magnified



(a) (b) (c) (d)

Fig. 11. View synthesis results. (a) Reference image. (b) Selected magnified detail. View synthesis results with underlying depth maps obtained using (c) 5×5 image array, and (d) using 3×3 image array.

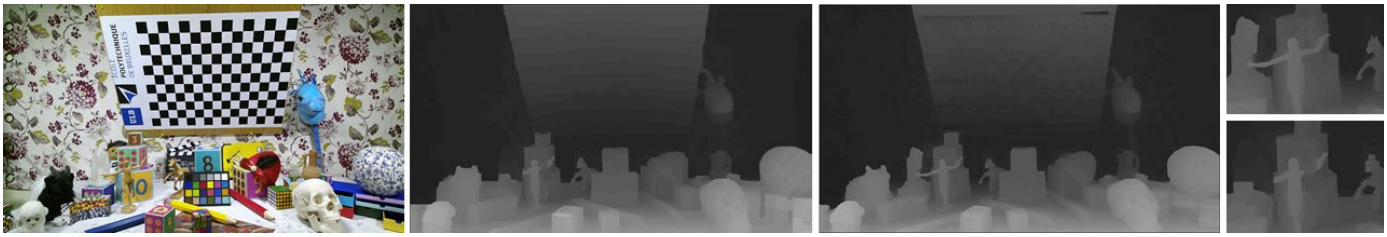
TABLE III
SSIM IMAGE SIMILARITY FOR DIFFERENT LIGHT FIELD SAMPLING DENSITY

Scene	Number of Views		
	9×9 (d = 0)	5×5 (d = 0)	3×3 (d = 0)
Truck	0.980 (0.949)	0.978 (0.924)	0.975 (0.888)
Bracelet	0.985 (0.866)	0.980 (0.810)	0.974 (0.763)
Jelly Beans	0.985 (0.969)	0.985 (0.956)	0.982 (0.942)

details of the synthesized views at two viewpoints can be seen in Fig. 13 for subjective evaluation. Comparing them reveals that the synthesized views generated with our depth maps have a competitive visual quality with those generated with DERS depth maps. We also measure the objective quality of synthesized views in terms of peak signal-to-noise ratio (PSNR) and SSIM (Fig. 13 right side of each subfigure) with respect to the original camera views. The plots with the PSNR and SSIM results corresponding to each synthesized view are depicted in Fig.14. As can be seen, our method exhibits a more stable performance over the views whereas DERS performs better or worse depending on the view, Fig. 14(a). In general, structural similarity results of our method are slightly better than DERS, Fig 14(b), due to more accurate depth reconstruction. However, our results exhibit more boundary artifacts that mainly occur due to violation of the assumption that similar pixels belong to the same object, e.g. the edges of the board and cubes in the scene, c.f. Fig.13(c). Here, the thin boundary of the object has a very different color from the rest of the object, while similar colors are present in the background. Thus, due to the superpixel segmentation prior, some boundary pixels are assigned to semantically wrong areas, which leads to errors in depth maps and, as a result, to rendering artifacts. The average PSNR results over all synthesized views are 32.33dB for our method and 32.17dB for DERS. The average SSIM results are almost the same, around 0.97.

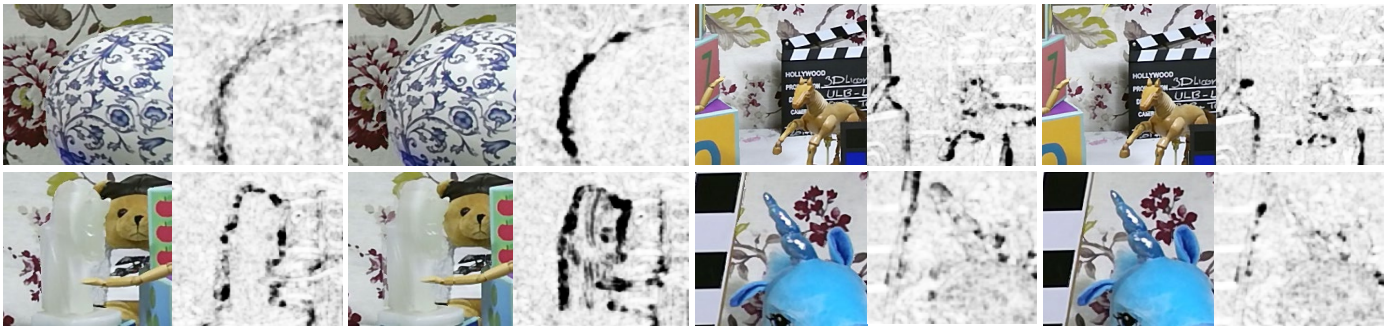
2) Evaluation on Sparse Light Field Videos

We conduct more tests using sparse light field videos 'Bar' and 'Beer Garden' [9] in order to evaluate the performance of our method in a more practical setting. These light field videos provide scenes containing static and dynamic objects (e.g. humans). Apart from the sparsity of the light field data, there are multiple challenging aspects present in the scenes, such as



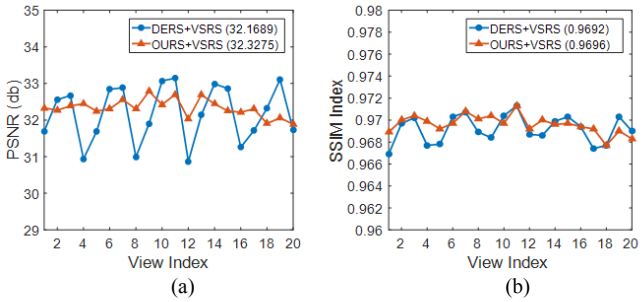
(a) (b) (c) (d)

Fig. 12. Comparison with DERS on ‘Unicorn’ dataset. (a) Central view. (b) DERS depth map. (c) Depth map obtained with our method. (d) Magnified detail.



(a) (b) (c) (d)

Fig. 13. View synthesis results: magnified details along with the scaled SSIM maps. (a) and (c) results obtained using our depth maps. (b) and (d) results obtained using DERS depth maps.



(a) (b)

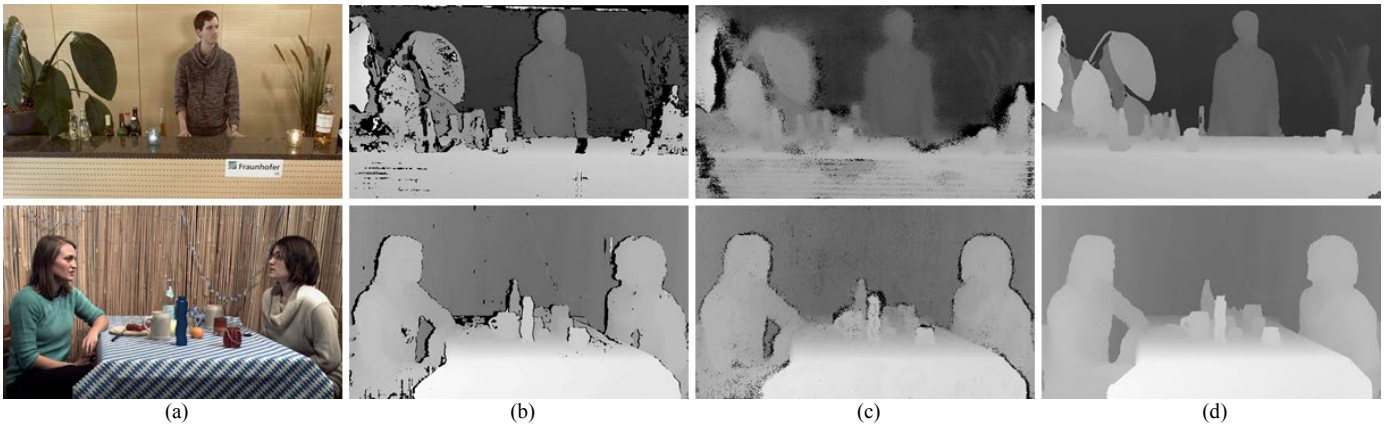
Fig. 14. (a) PSNR and (b) SSIM distribution for ‘Unicorn’ dataset using depth maps obtained by reference software (DERS+VSRS) and by our method (OURS+VSRS).

transparent and reflective objects, big regions with repetitive textures, wide-baseline occlusions, and motion blur. These aspects, however, are frequently encountered in real-world data. We compare our method with the state-of-the-art stereo method (SGM) [66] and an efficient multi-scale method for sparse light field correspondence proposed in [9] (further referred as ‘FH’). In case of SGM, to obtain disparity maps for every view we run this algorithm on each horizontally adjacent stereo pair independently (e.g. for 3×3 setup we consider six stereo pairs). Fig. 15 presents the comparative results for the disparity maps generated by SGM, FH (disparity maps are provided by the authors) and our method. Despite the above-mentioned challenges present in the test datasets, our method demonstrates robustness. By incorporating smoothness and geometric consistency constraints in the propagation process, textured and textureless regions, occlusions and moderate reflections can be handled. As can be seen, our reconstructed disparity maps are denser and visibly more accurate than the results obtained by the reference methods. Some inaccuracies, however, are present, e.g. in ‘Bar’ sequence due to non-Lambertian surfaces (transparent bottle and reflective table).

In case of video synthesis, depth inconsistencies between the frames may lead to uncomfortable flickering artifacts in the static regions of the scene. Currently, we do not explicitly enforce the temporal consistency in our method, and each frame of the video sequences is processed independently. However, as the cross-view geometric consistency is properly exploited during the refinement stage, our recovered depth maps are not only consistent across the views but also rather consistent across the frames. We provide evaluation of temporal consistency for video synthesis in the supplementary materials.

V. Conclusion

We have presented a GPU-based method for fast and accurate depth maps reconstruction from sparse light fields. Our method compares favorably against several state-of-the-art methods in terms of both runtime and accuracy. Whereas the reconstruction time is about one second per full HD view, we are able to obtain accurate and dense depth maps comparable to the reference methods results even on sub-pixel level. We have experimentally demonstrated the potential of our approach in application for sparse light field depth reconstruction and show that our method can successfully and robustly handle difficult wide-baseline video sequences. There are cases, however, when the assumptions of our method do not hold (e.g. non-Lambertian surfaces and violation of segmentation prior) leading to erroneous results. We believe that a greater accuracy can be achieved by applying advanced post-processing methods and incorporating more complex occlusion handling schemes. We are also interested in improving the speed of our method to possibly work at interactive or even real-time frame rates.



(a) (b) (c) (d)

Fig. 15. Comparative results for sparse light field reconstruction. (a) sample frames from 'Bar' and 'Beer Garden' sequences and corresponding disparity maps obtained (b) by SGM, (c) by FH, and (d) by our method.

REFERENCES

[1] M. Levoy, P. Hanrahan, "Light field rendering", *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, pp. 31-42, 1996.

[2] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu., "Light field image processing: An overview", *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 926-954, 2017.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, M. F Cohen, "The Lumigraph", *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, pp. 43-54, 1996.

[4] A. Davis, M. Levoy, F. Durand, "Unstructured light fields", *Comput. Graph. Forum*, vol. 31, no. 2, pp. 305-314, 2012.

[5] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," Tech. Rep. CSTR 2005-02, Stanford University, 2005.

[6] C. Perwass, L. Wietzke, "Single lens 3D-camera with extended depth-of-field", *Proc. IS&T/SPIE Electron. Imaging*, vol. 8291, pp. 1-15, 2012.

[7] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, "High performance imaging using large camera arrays", *ACM Trans. Graph*, vol. 24, no. 3, pp. 765-776, 2005.

[8] Y. Liu, Q. Dai, and W. Xu, "A real time interactive dynamic light field transmission system," in *Proc. IEEE Int. Conf. Multimedia Expo.*, pp. 2173–2176, 2006.

[9] L. Dabala, M. Ziegler, P. Didyk, F. Zilly, J. Keinert, K. Myszkowski, H.-P. Seidel, P. Rokita, T. Ritschel, "Efficient Multi-image Correspondences for On-line Light Field Video Processing", *Computer Graphics Forum*, vol. 35, no. 7, pp. 401-410, 2016.

[10] N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, R. Gendrot, T. Langlois, O. Bureller, A. Schubert, and V. Allie, " Dataset and pipeline for multi-view lightfield video", *IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pp. 1743–1753, 2017.

[11] J.-X. Chai, X. Tong, S.-C. Chan, H.-Y. Shum, "Plenoptic sampling", *Proc. ACM Annu. Computer Graphics Conf.*, pp. 307-318, 2000.

[12] Y. Pan, O. Oyekoya and A. Steed, "A Surround Video Capture and Presentation System for Preservation of Eye-Gaze in Teleconferencing Applications", *Presence*, vol. 24, no. 1, pp. 24-43, 2015.

[13] J. Urbach, "The Future of GPU Rendering: Real-Time Raytracing, Holographic Displays and Light Field Media", *Siggraph preview*, 2019.

[14] M. Alain, C. Ozcinar, A. Smolic, "A Study of Light Field Streaming for an Interactive Refocusing Application", *IEEE International Conference on Image Processing*, 2019.

[15] S. Wanner, B. Goldluecke, "Globally consistent depth labeling of 4D light fields", *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 41-48, 2012.

[16] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," *Proc. IEEE Int. Conf. Comput. Vis*., pp. 673–680, 2013.

[17] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 3487–3495, 2015.

[18] S. Wanner, B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 606-619, 2014.

[19] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, M. Gross, "Scene reconstruction from high spatio-angular resolution light fields", *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[20] K. Yücer, C. Kim, A. Sorkine-Hornung, O. Sorkine-Hornung, "Depth from gradients in dense light fields for object reconstruction", *Proc. Int. Conf. 3D Vis.*, pp. 249-257, 2016.

[21] C. Kim, "3D Reconstruction and Rendering from High Resolution Light Fields", Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2015.

[22] O. Johannsen, K. Honauer, B. Goldluecke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, et al., "A taxonomy and evaluation of dense light field depth estimation algorithms", *IEEE Conf. Comp. Vision and Pattern Recognition Workshops*, pp. 1795-1812, 2017.

[23] J. Chen, J. Hou, Y. Ni, and L.P. Chau, "Accurate light field depth estimation with superpixel regularization over partially occluded regions", *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4889–4900, 2018.

[24] W. Williem, I. K. Park, and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.10, pp. 2484–2497, 2018.

[25] H. Schilling, M. Diebold, C. Rother, and B. Jahne, "Trust your model: Light field depth estimation with inline occlusion handling", *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4530–4538, 2018.

[26] Y. Mo, J. Yang, C. Xiao, and W. An, "Toward Real-World Light Field Depth Estimation: A Noise-Aware Paradigm Using Multi-Stereo Disparity Integration", *IEEE Access*, vol. 7, pp. 94391 – 94399, 2019.

[27] X. Jiang, M. Le Pendu, and C. Guillemot, "Depth estimation with occlusion handling from a sparse set of light field views," *IEEE Int. Conf. Image Process. (ICIP)*, pp. 634–638, 2018.

[28] H. W. F. Yeung, J. Hou, J. Chen, Y. Y. Chung, and X. Chen, "Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues," *European Conference on Computer Vision*, 2018.

[29] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 40, no. 1, pp. 133–147, 2018.

[30] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 519-528, 2006.

[31] Y. Furukawa, C. Hernández, "Multi-view stereo: A tutorial", *Foundations and Trends in Computer Graphics and Vision*, vol. 9, no. 1–2, pp. 1-148, 2015.

[32] R. Collins, "A Space-Sweep Approach to True Multi-image Matching", *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 358-363, 1996.

[33] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions", *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1-8, 2007.

[34] V. Kolmogorov, R. Zabih, "Multi-camera scene reconstruction via graph cuts", *Proc. IEEE 7th Eur. Conf. Comput. Vis.*, pp. 82-96, 2002.

[35] J. Y. Chang, H. Park, I. K. Park, K. M. Lee, S. U. Lee, "GPU-friendly multi-view stereo reconstruction using surfel representation and graph cuts", *Comput. Vis. Image Underst.*, vol. 115, no. 5, pp. 620-634, 2011.
[36] P. Felzenszwalb, D. Huttenlocher, "Efficient Belief Propagation for Early Vision", *Int'l J. Computer Vision*, vol. 70, no. 1, pp. 41-54, 2006.
[37] G. Zhang, J. Jia, T. T. Wong, H. Bao, "Consistent depth maps recovery from a video sequence", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974-988, 2009.
[38] A. O. Ulusoy, A. Geiger, M. J. Black, "Towards probabilistic volumetric reconstruction using ray potentials", *Proc. Int. Conf. 3D Vision*, 2015.
[39] Y. Furukawa, J. Ponce, "Accurate Dense and Robust Multiview Stereopsis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362-1376, 2010.
[40] S. Sinha, D. Steedly, R. Szeliski, "Piecewise Planar Stereo for Image-Based Rendering", *IEEE Int. Conf. Comput. Vis.*, pp. 1881-1888, 2009.
[41] D. Gallup, J.-M. Frahm, M. Pollefeys, "Piecewise Planar and Non-Planar Stereo for Urban Scene Reconstruction", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1418-1425, 2010.
[42] B B. Micusik, J. Kosecka, "Piecewise planar city 3D modeling from street view panoramic sequences", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2906-2912, 2009.
[43] A. Bodis-Szomoru, H. Riemenschneider, L. V. Gool, "Fast approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels", *Conf. Comput. Vis. Pattern Recognit.*, pp. 469-476, 2014.
[44] C.L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and r. Szeliski, "High-Quality Video View Interpolation Using a Layered Representation", *ACM Trans. Graphics*, vol. 23, no. 3, pp. 600-608, 2004.
[45] C. Zhang, Z. Li, R. Cai, H. Chao, and Y. Rui, "As-rigid-as-possible stereo under second order smoothness priors", *Proc. IEEE Eur. Conf. Comput. Vis*., pp. 112–126, 2014.
[46] M. Bleyer, C. Rhemann, C. Rother, "Patchmatch stereo—stereo matching with slanted support windows", *Brit. Mach. Vis. Conf.*, pp. 1-11, 2011.
[47] P. Heise, S. Klose, B. Jensen, A. Knoll, "PM-Huber: PatchMatch with Huber regularization for stereo matching", *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2360-2367, 2013.
[48] S. Xu, F. Zhang, X. He, X. Shen, X. Zhang, "PM-PM: PatchMatch with potts model for object segmentation and stereo matching", *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2182-2196, 2015.
[49] S. Galliani, K. Lasinger, and K. Schindler. "Massively parallel multiview stereopsis by surface normal diffusion," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 873-881, 2015.
[50] F. Besse, C. Rother, A. Fitzgibbon, J. Kautz, "PMBP: PatchMatch belief propagation for correspondence field estimation", *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 2-13, 2014.
[51] J. Lu, H. Yang, D. Min, M. N. Do, "Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation", *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1854-1861, 2013.
[52] J. Wei, B. Resch, and H. Lensch, "Multi-View Depth Map Estimation With Cross-View Consistency", *Proc. Brit. Mach. Vis. Conf.*, 2014.
[53] C. Wu, "VisualSFM: A visual structure from motion system," http://ccwu.me/vsfm/, 2019.
[54] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, M. Pollefeys, "Real-time visibility-based fusion of depth maps", *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
[55] R R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274-2282, 2012.
[56] C.Y. Ren and I. Reid, "gSLIC: a real-time implementation of SLIC superpixel segmentation", Dep. of Engineering, Univ. of Oxford, Tech. Rep., 2011.
[57] J. Yao, M. Boben, S. Fidler, and R. Urtasun, "Real-time coarse-to-fine topologically preserving segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2947–2955, 2015.
[58] D. Stutz, A. Hermans, and B. Leibe. Superpixels, "An evaluation of the state-of-the-art," *Computer Vision and Image Understanding*, vol. 166(C), pp. 1–27, 2018.
[59] M. Hog, N. Sabater, and C. Guillemot, "Super-rays for Efficient Light Field Processing", *IEEE Journal of Selected Topics in Signal Processing*, 2017.
[60] Middlebury stereo benchmark, 2003, [online] Available: http://vision. middlebury.edu/stereo/
[61] D. Scharstein, R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 195-202, 2003.
[62] "Stanford (New) Light Field Archive", 2008, [Online] Available: http://lightfield.stanford.edu/
[63] D. Bonatto, A. Schenkel, T. Lenertz, Y. Li, G. Lafruit, "ULB High Density 2D/3D Camera Array data set, version 2, " ISO/IEC JTC1/SC29/WG11 MPEG2017/M41083, July 2017, Torino, Italy.
[64] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity", *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, 2004.
[65] K. Wegner and O. Stankiewicz, "DERS Software Manual", in ISO/IEC JTC1/SC29/WG11 M34302, Sapporo, 2014.
[66] H. Hirschmuller, "Stereo processing by semi-global matching and mutual information", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.
[67] T. Senoh, J. Kenji, T. Nobuji, Y.Hiroshi, K. Wegner, "View Synthesis Reference Software (VSRS) 4.2 with improved inpainting and hole filing", ISO/IEC JTC1/SC29/WG11, n. MPEG2017/M40657, 2017.

Aleksandra Chuchvara received her B.S. degree in Applied Mathematics and Informatics from the Lomonosov Moscow State University (2009), her M.Sc. degree in Information Technology from the Tampere University of Technology (2014), and pursuing Ph.D. degree in Computing and Electrical Engineering at the Tampere University of Technology. Her current research interests include image-based 3D scene reconstruction and rendering.

Attila Barsi completed his M.Sc. degree in computer science from the Budapest University of Technology (2004). From 2005 to 2006, he was a Software Engineer in DSS, Hungary. Since 2006, he has been a Software Engineer, then a Lead Software Engineer of Holografika. He is the author and co-author of several conference and journal papers. His research interests include light fields, real-time rendering, global illumination, ray tracing, 3D computing.

Atanas Gotchev (Member, IEEE) received his M.Sc. degrees in radio and television engineering (1990) and applied mathematics (1992), his Ph.D. degree in telecommunications (1996) from the Technical University of Sofia, and the D.Sc.(Tech.) degree in information technologies from the Tampere University of Technology (2003). He is a Professor of Signal Processing and Director of the Centre for Immersive Visual Technologies at Tampere University of Technology. His recent work concentrates on the algorithms for multi-sensor 3-D scene capture, transform-domain light-field reconstruction, and Fourier analysis of 3-D displays.