Scene Reconstruction from High Spatio-Angular Resolution Light Fields

Changil Kim[1,2] Henning Zimmer[1,2] Yael Pritch[1] Alexander Sorkine-Hornung[1] Markus Gross[1,2]

[1]Disney Research Zurich [2]ETH Zurich

Figure 1: *Our method reconstructs accurate depth from light fields of complex scenes. The images on the left show a 2D slice of a 3D input light field, a so called epipolar-plane image (EPI), and two out of one hundred 21 megapixel images that were used to construct the light field. Our method computes 3D depth information for all visible scene points, illustrated by the depth EPI on the right. From this representation, individual depth maps or segmentation masks for any of the input views can be extracted as well as other representations like 3D point clouds. The horizontal red lines connect corresponding scanlines in the images with their respective position in the EPI.*

Abstract

This paper describes a method for scene reconstruction of complex, detailed environments from 3D light fields. Densely sampled light fields in the order of 10^9 light rays allow us to capture the real world in unparalleled detail, but efficiently processing this amount of data to generate an equally detailed reconstruction represents a significant challenge to existing algorithms. We propose an algorithm that leverages coherence in massive light fields by breaking with a number of established practices in image-based reconstruction. Our algorithm first computes reliable depth estimates specifically around object boundaries instead of interior regions, by operating on *individual light rays* instead of image patches. More homogeneous interior regions are then processed in a *fine-to-coarse* procedure rather than the standard coarse-to-fine approaches. At no point in our method is any form of global optimization performed. This allows our algorithm to retain precise object contours while still ensuring smooth reconstructions in less detailed areas. While the core reconstruction method handles general unstructured input, we also introduce a *sparse representation* and a *propagation scheme* for reliable depth estimates which make our algorithm particularly effective for 3D input, enabling fast and memory efficient processing of "Gigaray light fields" on a standard GPU. We show dense 3D reconstructions of highly detailed scenes, enabling applications such as automatic segmentation and image-based rendering, and provide an extensive evaluation and comparison to existing image-based reconstruction techniques.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth Cues; I.4.10 [Image Processing and Computer Vision]: Image Representation—Multidimensional;

Keywords: light fields, image-based scene reconstruction

Links: DL PDF WEB DATA

1 Introduction

Scene reconstruction in the form of depth maps, 3D point clouds or meshes has become increasingly important for digitizing, visualizing, and archiving the real world, in the movie and game industry as well as in architecture, archaeology, arts, and many other areas. For example, in movie production considerable efforts are invested to create accurate models of the movie sets for post-production tasks such as segmentation, or integrating computer-generated and real-world content. Often, 3D models are obtained using laser scanning. However, because the sets are generally highly detailed, meticulously designed, and cluttered environments, a single laser scan suffers from a considerable amount of missing data at occlusions [Yu et al. 2001]. It is not uncommon that the manual clean-up of hundreds of merged laser scans by artists takes several days before the model can be used in production.

Compared to laser scanning, an attractive property of passive, image-based stereo techniques is their ability to create a 3D representation solely from photographs and to easily capture the scene from different viewing positions to alleviate occlusion issues. Unfortunately, despite decades of continuous research efforts, the majority of stereo algorithms seem not well suited for today's challenging applications, e.g., in movie production [Sylwan 2010], to efficiently cope with higher and higher resolution images[1] while at the same time producing sufficiently accurate and reliable reconstructions. For specific objects like human faces stereo-based techniques have matured and achieve very high reconstruction quality (e.g., [Beeler et al. 2010]),

[1]Digital cinema and broadcasting are in the process of transitioning from 2k to 4k resolution (∼2 megapixels to ∼9 megapixels)

but more general environments such as the detailed outdoor scene shown in Figure 1 remain challenging for *any* existing scanning approach.

In this paper we follow a different strategy and revisit the concept of 3D light fields, i.e., a dense set of photographs captured along a linear path. In contrast to sparser and less structured input images, a perfectly regular, densely sampled 3D light field exhibits a very specific internal structure: every captured scene point corresponds to a linear trace in a so called epipolar-plane image (EPI), where the slope of the trace reflects the scene point's distance to the cameras (see Figure 1). The basic insight to leverage these structures for scene reconstruction was proposed as early as 1987 [Bolles et al. 1987], and has been revisited repeatedly since then (see, e.g., [Criminisi et al. 2005]). However, these methods do not achieve the reconstruction quality of today's highly optimized two or multi-view stereo reconstruction techniques.

With today's camera hardware it has become possible to capture truly dense 3D light fields. For example, for the results shown in Figure 1 we captured one hundred 21 megapixel (MP) images with a standard DSLR camera, effectively resulting in a two "Gigaray" light field. While such data can capture an unparalleled amount of detail of a scene, it also poses a new challenge. Over many years the basic building blocks in stereo reconstruction such as patch-based correlation, edge detection and feature matching have been tailored towards optimal performance at about 1–2 MP resolution. In addition, most algorithms involve some form of global optimization in order to obtain sufficiently smooth results. As a consequence, it is often challenging to scale such approaches to significantly higher image resolution.

In this paper we propose an algorithm that specifically leverages the properties of densely sampled, high resolution 3D light fields for reconstruction of static scenes. Unlike approaches based on patch-correlation our algorithm operates at the single pixel level, resulting in precise contours at depth discontinuities. Smooth, homogeneous image regions are handled by a hierarchical approach. However, instead of a standard coarse-to-fine estimation, we reverse this process and propose a *fine-to-coarse* algorithm that reconstructs reliable depth estimates at the highest resolution level first, and then proceeds to lower resolutions, avoiding the need for any kind of explicit global regularization. At any time the algorithm operates only on a small set of adjacent EPIs, enabling efficient GPU implementation even on light fields in the order of 10^9 rays. We further increase efficiency by propagating reliable depth estimates throughout the whole light field using a novel sparse data structure, such that the algorithm effectively computes depth maps for all input images concurrently. We demonstrate dense reconstructions of challenging, highly detailed scenes and compare to a variety of related stereo-based approaches. We also present direct applications to segmentation and novel-view synthesis, discuss practical issues when capturing high resolution 3D light fields, and discuss how our reconstruction algorithm generalizes to 4D light fields and unstructured input.

2 Related Work

Light field capture, representation, and depth estimation are closely connected and related to areas such as (multi-view) stereo. In this section we give an overview of the most related previous work.

Light field acquisition and representation. Light fields can be captured in various ways. Most setups rely on a controlled acquisition, e.g., using camera gantries [Levoy and Hanrahan 1996], camera arrays [Wilburn et al. 2005], lenslet arrays [Ng et al. 2005], or coded aperture techniques [Veeraraghavan et al. 2007] but unstructured acquisition like hand-held capture have also been considered [Gortler et al. 1996; Davis et al. 2012].

A significant challenge is that the captured set of images is very data-intensive and also redundant. Thus, already the seminal papers discussed compact representations and compression schemes. Levoy and Hanrahan [1996] propose several representations for 4D light fields and apply a lossy vector quantization followed by entropy coding. Gortler et al. [1996] applied standard image compression like JPEG to some of the views, and also point out the importance of depth information for more accurate view prediction and rendering. Isaksen et al. [2000] describe how an approximate depth proxy may compensate sparse angular sampling, with a focus on rendering photographic effects like varying depth-of-field. Similarly, Wanner et al. [2011] use a rough depth map to render light fields from a lenslet array camera. Chai et al. [2000] investigated the plenoptic sampling problem to determine the minimal number of views needed to perfectly reconstruct a light field. Solutions for efficient capture and rendering of unstructured light fields have been presented in [Zhu et al. 1999; Buehler et al. 2001; Rav-Acha et al. 2004; Davis et al. 2012]. Criminisi et al. [2005] investigated the segmentation of epipolar-plane images (EPIs) in 3D light fields into tubes representing layers of different objects. Storing colors and depth for each tube then gives a more compact representation of the light field. They also propose a method for detecting and removing specular highlights, but no solution for compactly storing this view-dependent information. Surface light fields [Wood et al. 2000; Chen et al. 2002] are an attractive solution to capture view-dependent effects, but they require accurate 3D geometry obtained by active scanning techniques. One component of our contribution is a sparse light field representation (Section 3) that differs from those previous approaches, fully reproduces the input light field including view dependent surface reflectance, and tightly integrates with our algorithm for depth estimation.

Depth reconstruction from light fields. One of the first approaches to extract depth from a dense sequence of images is the seminal work of Bolles et al. [1987]. To our knowledge their technique is the first attempt to utilize the specific linear structures emerging in a densely sampled 3D light field for depth computation. However, the employed basic line fitting is not robust enough for a dense reconstruction of real world scenarios with occlusions, varying illumination, etc. and the reconstructions shown are sparse and noisy. The majority of methods adopt techniques from classical stereo reconstruction, i.e., matching corresponding pixels in all images of the light field using essentially robust patch-based block matching [Zhang and Chen 2004; Vaish et al. 2006; Bishop et al. 2009; Georgiev and Lumsdaine 2010]. Along similar lines, Fitzgibbon et al. [2005] and Basha et al. [2012] describe robust clustering techniques to identify matching pixels. Ziegler et al. [2007] propose to analyze the Fourier spectra of EPIs sheared according to a hypothesized depth. As we demonstrate in our comparisons, such approaches often do not scale well to high resolution light fields in terms of reconstruction quality and computational efficiency.

In order to achieve higher overall coherence, various methods estimate depth as the minimizer of a global energy functional where smoothness assumptions can be enforced; see for example [Adelson and Wang 1992; Stich et al. 2006; Liang et al. 2008; Bishop and Favaro 2010]. Notably, the recent energy-based approach of Wanner and Goldluecke [2012] gives high quality depth maps from 4D light fields. But as for any global optimization method this comes at a very high computational cost. For example, the authors of the latter work report 10 minutes per single view depth map at 1 MP resolution. The direct application of such approaches to higher spatio-angular resolutions seems impractical. A second difficulty with approaches based on global optimization is to tune the underlying smoothness assumptions to preserve precise depth discontinuities at object contours, which are of highest importance in practice [Sylwan 2010]. Fine details are often lost due to the involved coarse-to-fine multi-

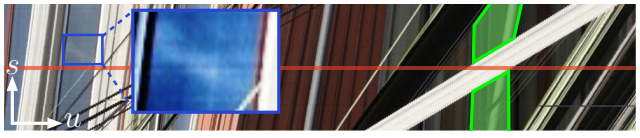
scale algorithms. Our fine-to-coarse approach is particularly suited for such applications as it reconstructs precise depth estimates at the single pixel level, without the need for explicit global regularization. Along similar lines one can extract depth from a light field using depth-from-focus techniques. However, those methods face challenges similar to standard stereo approaches such as inaccuracies at silhouettes, but also have limitations due to the aperture size [Schechner and Kiryati 2000].

To illustrate the novel challenges arising from high resolution, densely captured light fields, we compare our results to some of currently best performing two and multi-view stereo algorithms (for an overview please refer to the evaluations of Scharstein et al. [2002] and Seitz et al. [2006]). Despite considerable progress in this area [Kolmogorov and Zabih 2001; Hirschmüller 2005; Rhemann et al. 2011] with only two input views available one has to rely on some form of global smoothness. To alleviate over-smoothing of discontinuities, one can operate on larger image segments [Zitnick et al. 2004; Zitnick and Kang 2007], but this may lead to over-segmentation artifacts in the depth maps at textured image regions. Also, with only a few views available, explicit detection and handling of occlusions is often required [Humayun et al. 2011; Ayvaci et al. 2012], which further increases the computational load. An alternative is to only match a few reliable pixels [Čech and Šára 2007], and to densify the result later by spreading the sparse estimates [Sun et al. 2011]. However, existing approaches for sparse sample propagation generally require a global energy minimization [Geiger et al. 2010], or are prone to artifacts as shown in [Szeliski and Scharstein 2002]. Multi-view stereo techniques consider a larger number of images, spanning from tens [Seitz and Dyer 1999; Kang and Szeliski 2004; Zitnick et al. 2004; Vu et al. 2009; Beeler et al. 2010; Furukawa and Ponce 2010] to several thousands [Snavely et al. 2008; Furukawa et al. 2010] to compute a more complete scene representation rather than single depth maps. However, these methods often provide either accurate but still sparse, or dense but comparably smooth geometry and often do not scale well to very high resolution images. The coverage of the reconstructed scene with our method is higher than that of two-view stereo techniques, but lower than full 3D models generated with multi-view stereo. However, in contrast to the previously discussed techniques our algorithm produces a dense scene reconstruction with precise contours that is readily available for various applications such as novel view synthesis, depth-based segmentation, and other image-based applications. Some methods [Goldlücke and Magnor 2003; Bleyer et al. 2011] jointly estimate depth and segmentation, but these again rely on costly global optimization.

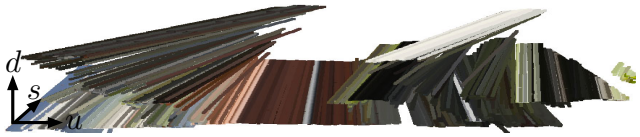
3 Sparse Representation

Light fields are typically constructed from a large set of images of a scene, captured at different viewing positions. A suitable representation of such data depends on a plethora of factors, including for example structured vs. unstructured capture of light fields, the targeted processing algorithms and applications, or just the sheer amount of data. Accordingly various representations have been proposed in the past [Levoy and Hanrahan 1996; Gortler et al. 1996; Isaksen et al. 2000; Buehler et al. 2001; Davis et al. 2012]. Our main focus in this paper is on 3D light fields of very high spatio-angular resolution, i.e., light fields constructed from hundreds of high resolution 2D images with their respective optical centers distributed along a 1D line. We introduce a novel compact representation that enables efficient parallel processing without the need to keep the full input light field in memory, and that can be efficiently constructed during our depth estimation described in Section 4.

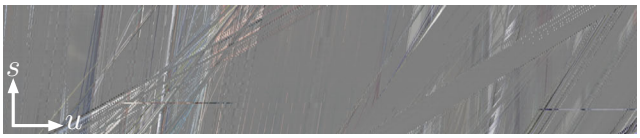
A 3D light field with radiance values captured in RGB color space can be denoted as a map $L : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The radiance $\mathbf{r} \in \mathbb{R}^3$ of a light ray is given as $\mathbf{r} = L(u, v, s)$, where s describes the 1D ray origin and (u, v) represent the 2D ray direction. In terms of the above mentioned capture setup, s can be interpreted as the different camera positions distributed along a 1D line, and (u, v) are the pixel coordinates in a corresponding image $I_s(u, v)$. For a concise exposition in the paper we assume regular, uniform sampling of u, v, and s, i.e., the optical centers are uniformly spaced and all captured images are rectified, so that the epipolar lines of a scene point coincide with the same horizontal scanline in all images. We provide details how to practically achieve this in Section 5.1.



(a) EPI E



(b) 3D visualization of



(c) Delta EPI ΔE

Figure 2: *Illustration of our sparse representation, using a cropped section from the EPI in Figure 1. The red line marks the central input view. (a) Concerning* completeness*, consider the region shaded in green on the right. It is occluded by the white structure and thus propagating color values from the central view only would not reconstruct the highlighted region. View-dependent* variation*, e.g., due to reflections in the building windows, is highlighted in the blue framed region. We increased color contrast in the inset for improved visibility of the color changes. Again, a reconstruction solely from the central view would not capture these effects. (b) 3D visualization of EPI $\widehat{E}$ reconstructed from our sparse representation . (c) Visualization of the difference between the input EPI and our reconstructed EPI $\widehat{E}$.*

While for given s a u-v-slice of this light field corresponds to input image I_s, a u-s-slice for a fixed v coordinate corresponds to a so called epipolar-plane image, or EPI [Bolles et al. 1987], which intuitively is simply a stack of the same row v taken from all input images. The left half of Figure 1 shows two out of 100 input images and an exemplary EPI. The horizontal red lines visualize both the respective s-parameters of the two input images in the EPI as well as the v-parameter in the input images from which the EPI has been constructed. Similar to above we denote an EPI as $E_v : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, with radiance $\mathbf{r} = E_v(u, s)$ of a ray at position (u, s). In analogy to image-pixels, we will use the term *EPI-pixel* (u, s) instead of the term ray at (u, s) for disambiguation. Most of our following discussion considers individual EPIs with parameter v fixed as our algorithm operates mostly on individual EPIs. Hence we will omit the subscript v for notational simplicity.

When the ray space of L is sampled densely enough, each scene point appears as a line segment in such an EPI with the slope of the line segment depending on the scene point's depth. Correspondingly, the EPIs of 3D light fields exhibit high coherence and contain

very redundant information that can be utilized for a more efficient representation. Rather than storing the full EPI, we can in principle reconstruct it by knowing the parameters of those line segments. As discussed in the related work section, this basic idea is well known. However, we propose a new representation that specifically considers two new aspects, namely *completeness* and *variation* of the represented light field.

Assume we can accurately estimate the slope of line segments or, equivalently, the depth of scene points. A first idea could be to simply collect and store the line segments and their color along a single horizontal line of an EPI. In principle this corresponds to storing a single input image and a depth map. A large number of captured light rays may be occluded in this particular part of the EPI, hence *completeness* of the representation would be compromised. In addition, scene points may change their color along their corresponding line segment due to specularities or other view dependent effects. Hence the above representation would not capture *variation* in the light field. See Figure 2 (a) for a visualization of both effects.

Our strategy for representing 3D light field data addresses these two issues. Firstly, we sample and store a set Γ of line segments originating at various locations in the input EPI E, until the whole EPI is completely represented and redundancy is eliminated to the extent possible. Secondly, we store a difference EPI ΔE that accounts for variations in the light field. More specifically, the slope m of a line segment associated with a scene point at distance z is given by

$$m = \frac{1}{d} = \frac{z}{f\,b}, \tag{1}$$

where d is the image space disparity defined for a pair of images captured at adjacent positions or, equivalently, the displacement between two adjacent horizontal lines in an EPI, f is the camera focal length in pixels and b is the metric distance between each adjacent pair of imaging positions. Correspondingly an EPI line segment can be compactly described by a tuple $\mathbf{l} = (m, u, s, \mathbf{r}^\top)$, where $\mathbf{r}$ is the average color of the scene point in the EPI. Γ is simply the set of all tuples $\mathbf{l}$. The actual scheme of how we select line segments $\mathbf{l}$ is part of the depth computation described in the following section. In Figure 2 (a), the red line represents the first and largest set of tuples that we will reconstruct. To ensure completeness, our representation will also store additional tuples inside the occluded regions highlighted in green.

From Γ, a reconstructed EPI $\widehat{E}$ can be generated by rendering the lines segments in the order of decreasing slopes, i.e., render the scene points from back to front. See Figure 2 (b) for a 3D visualization of the full representation Γ. Hence, for efficient EPI reconstruction, Γ is stored as ordered list of tuples in the order of decreasing slopes. The difference $\Delta E = E - \widehat{E}$ of the input E and the reconstruction $\widehat{E}$ captures the remaining variation and detail information in the light field, such as view dependent effects. This is illustrated in Figure 2 (c), where a grey color corresponds to zero reconstruction error. Note a high value of ΔE for the specularities and at inaccurate slope estimates.

Both Γ and ΔE compactly store all relevant information that is necessary to reconstruct the full 3D light field as well as extract an arbitrary input image with a corresponding depth map, or a full 3D point cloud. As an example, for the EPI in Figure 2, $\sim$277K EPI-pixels are reduced to $\sim$15K tuples (about 5.7%). Plain storage of the full tuple information without any further compression already results in a reduction to 21% compared to the RGB EPI. As discussed above various alternatives exist to store a coherent light field. A main benefit of our representation is its consistency with our algorithm for depth computation, enabling compact representation and efficient parallel computation as described in the next section.

4 Depth Estimation

Constructing Γ amounts to computing the line slopes at the EPI-pixels, i.e., estimating the depth of scene points. As mentioned before the ray coherence of a dense 3D light field allows our algorithm to operate on individual EPI-pixels instead of having to consider larger pixel-neighborhoods like most stereo approaches. As a consequence it performs especially well at depth discontinuities and reproduces precise object silhouettes due to the color contrast in these regions. This property is key to our *fine-to-coarse* depth estimation strategy: we estimate depth first at edges in the EPI at the highest resolution, propagate this information throughout the EPI, and then proceed to successively coarser EPI resolutions. In contrast to classic coarse-to-fine schemes, this allows us to preserve sharp depth discontinuities at object silhouettes, while also estimating accurate depth in homogeneous regions. Additionally, our strategy increases computational efficiency by restricting computations to small fractions of the high resolution input.

4.1 Overview

Starting at the full resolution of an EPI E, the first step consists of efficiently identifying regions where the depth estimation is expected to perform well. To this end we introduce a fast *edge confidence* measure C_e that is computed on the EPI. The algorithm then generates depth estimates for EPI-pixels with a high edge confidence. This is done by testing various discrete depth hypotheses d and picking the one that leads to the highest color density of sampled EPI-pixels. The density estimation is further leveraged to improve the initial confidence towards a refined *depth confidence* C_d, which provides a good indicator for the reliability of a particular depth estimate. All EPI-pixels with a high reliability are stored as tuples in Γ and propagated throughout the EPI. This process of depth estimation and propagation is iterated until all EPI-pixels with a high edge confidence C_e have been processed.

At this point all confident, i.e., sufficiently detailed regions at the current resolution level of the EPI E have a reliable depth value assigned, while the depth in more homogeneous regions is yet unknown. Our fine-to-coarse approach then downsamples E to a coarser resolution and starts over with the above procedure, computing edge confidence for yet unprocessed parts of the EPI and so forth. This procedure is continued until a depth value is assigned to every EPI-pixel, i.e., the line segment tuples in Γ reconstruct the complete light field.

4.2 Edge Confidence

As the edge confidence measure C_e is intended to be a fast test for which parts of the EPI a depth estimate seems promising, we define it as a simple difference measure

$$C_e(u,s) = \sum_{u' \in \mathcal{N}(u,s)} \|E(u,s) - E(u',s)\|^2, \tag{2}$$

where $\mathcal{N}(u,s)$ is a 1D window in the EPI E around the pixel (u,s). The size of this neighborhood can be small (9 pixels in our experiments) as it is supposed to measure only the local color variation.

C_e is then thresholded (with a value of 0.02), resulting in a binary confidence mask M_e, visualized as red pixels in Figure 5 (c)–(e). In order to remove spurious isolated regions, we apply a morphological opening operator to the mask. During the following depth computation this binary mask will be used to prevent the computation of depth estimates at ambiguous EPI-pixels and hence speed up the computation without sacrificing accuracy.

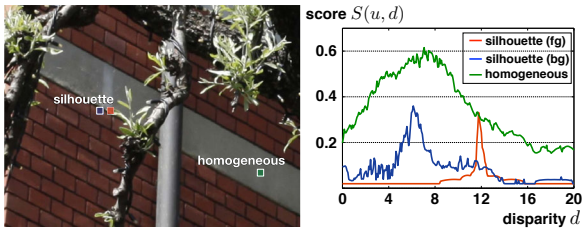


Figure 3: *At high image resolutions silhouette pixels result in a clear peak with a distinctive score profile whereas homogeneous regions lead to more flat and ambiguous scores. On coarser resolutions the scores in homogeneous regions become more distinct, which motivates our fine-to-coarse estimation.*

4.3 Depth Computation

Next our algorithm computes depth estimates for EPI-pixels in E marked as confident in M_e. For simpler parallelization on a GPU we perform this computation per scanline in the EPI, i.e., we select a fixed parameter $\hat{s}$ and compute a depth estimate for all $E(u, \hat{s})$ with $M_e(u, \hat{s}) = 1$. As discussed in Section 3, initially we select $\hat{s}$ as the horizontal centerline of E, as this generally allows us to compute a large fraction of the line segments visible in the EPI.

Following Equation (1) we try to assign a depth z, or equivalently a disparity d, to each EPI-pixel $(u, \hat{s})$. For a hypothetical disparity d the set $\mathcal{R}$ of radiances or colors of these EPI-pixels is sampled as

$$\mathcal{R}(u,d) = \{E(u+(\hat{s} \quad s)d,\ s) \mid s = 1, ..., n\}, \tag{3}$$

where n corresponds to the number of views in the light field. From the density of radiance values in $\mathcal{R}(u,d)$ a depth score $S(u,d)$ is computed in linearized RGB color space. The assumption here is that the scene is essentially Lambertian, i.e., a set $\mathcal{R}$ is likely to represent an actual scene point if the radiance samples are densely positioned in the underlying color space. Due to the high number of available samples in a dense light field our measure is very robust to outliers and hence implicitly handles occlusions. As we show in our results it is even robust to inconsistencies such as moving elements.

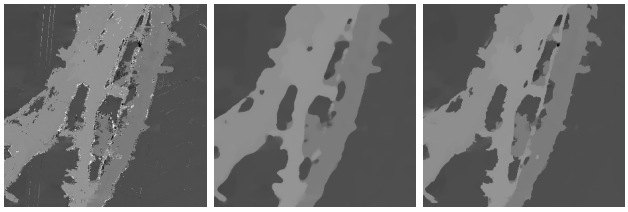
We compute the density efficiently using iterations of a modified Parzen window estimation [Duda et al. 1995] with an Epanechnikov kernel, and define the initial depth score as

$$S(u,d) = \frac{1}{|\mathcal{R}(u,d)|} \sum_{\mathbf{r} \in \mathcal{R}(u,d)} K(\mathbf{r} \quad \bar{\mathbf{r}}), \tag{4}$$

where $\bar{\mathbf{r}} = E(u, \hat{s})$ is the radiance value at the currently processed EPI-pixel, and the kernel $K(\mathbf{x}) = 1 \quad \|\mathbf{x}/h\|^2$ if $\|\mathbf{x}/h\| \leq 1$ and 0 otherwise. The bandwidth parameter was set to $h = 0.02$ in our experiments. Gaussian or other bell-shaped kernels also work well, but the chosen kernel is cheaper to compute. For a rather noise-free EPI this initial depth score is sufficient. To reduce the influence of noisy radiance measurements we borrow ideas from the mean-shift algorithm [Comaniciu and Meer 2002] by computing an iteratively updated radiance mean

$$\bar{\mathbf{r}} \leftarrow \frac{\sum_{\mathbf{r} \in \mathcal{R}} K(\mathbf{r} \quad \bar{\mathbf{r}})\mathbf{r}}{\sum_{\mathbf{r} \in \mathcal{R}} K(\mathbf{r} \quad \bar{\mathbf{r}})} \tag{5}$$

before computing Equation (4). Regarding the efficiency of this approach it is important to note that a full mean-shift clustering process or even just running the above mean-shift steps to convergence is counter-productive, as it significantly increases the computational complexity, in particular on a GPU due to the required branching and possibly different control flow. The main purpose, i.e., robustness to noise, is achieved already after a few iterations, hence the algorithm performs a constant number of 10 iterations for all results shown in the paper.



(a) No median (b) Standard median (c) Bilateral median

Figure 4: *Our proposed bilateral median filter removes speckles, while preserving fine details like the thin vertical string in the middle.*

For each EPI-pixel $(u, \hat{s})$ we compute scores $S(u,d)$ for the whole range of admissible disparities d, and assign the disparity with the highest score as the pixel's depth estimate

$$D(u, \hat{s}) = \arg\max_d S(u,d). \tag{6}$$

In addition we also compute the refined confidence C_d as a measure for the reliability of a depth estimate. C_d combines the edge confidence C_e with the difference between the maximum score $S_{\max} = \max_d S(u,d)$ and the average score $\bar{S} = \sum_d S(u,d)$

$$C_d(u, \hat{s}) = C_e(u, \hat{s}) \|S_{\max} \quad \bar{S}\| \tag{7}$$

The refined confidence measure C_d is meaningful as it combines two complementary measures. For instance, noisy regions of an EPI would result in a high edge-confidence C_e, while a clear maximum $S_{\max}$ is not available. Similarly, ambiguous homogenous regions in an EPI, where C_e is low, can produce a strong, but insufficiently unique $S_{\max}$; see Figure 3.

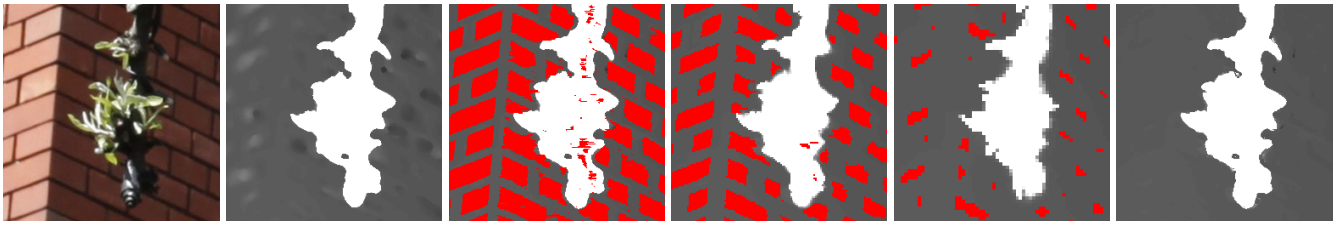
In order to eliminate the influence of outliers that might have survived the density estimation process, we apply a median filter on the computed depths. However, we observed that a straightforward median filter compromises the precise localization of silhouettes. We therefore use a bilateral median filter that preserves the localization of depth discontinuities by leveraging information from the radiance values of nearby EPIs. This is implemented by replacing the depth estimate $D_v(u, \hat{s})$ by the median value of the set

$$\begin{aligned} \{D_{v'}(u', \hat{s}) \quad \mid \quad & (u', v', \hat{s}) \in \mathcal{N}(u, v, \hat{s}), \\ & \|E_v(u, \hat{s}) \quad E_{v'}(u', \hat{s})\| < \varepsilon, \\ & M_e(u', v', \hat{s}) = 1\}, \end{aligned} \tag{8}$$

where $(u', v', \hat{s}) \in \mathcal{N}(u, v, \hat{s})$ denotes a small window over $I_{\hat{s}}$. The second condition assures that we only consider EPI-pixels of similar radiance and the last condition masks out unconfident EPI-pixels for which no depth estimation is available. In all our experiments we use a window size of 11×11 and a threshold value $\varepsilon = 0.1$. Correspondingly, we always store at most 11 EPIs during computation. The effect of this filtering step is illustrated in Figure 4.

4.4 Depth Propagation

Each confident depth estimate $D(u, \hat{s})$ with $C_d(u, \hat{s}) > \varepsilon$ is now stored as a line segment tuple $\mathbf{l} = (m, u, \hat{s}, \bar{\mathbf{r}}^\top)$ in (see Equation (1)), where $\bar{\mathbf{r}}$ represents the mean radiance of $(u, \hat{s})$ computed in Equation (5). Then the depth estimate is propagated along the slope of its corresponding EPI line segment to all EPI-pixels (u', s') that have a radiance similar to the mean radiance, i.e.,



(a) Image (b) No fine-to-coarse (c) Level 0 (d) Level 1 (e) Level 2 (f) With fine-to-coarse

Figure 5: *Our fine-to-coarse refinement yields reliable depth estimates also in homogeneous image regions, like the bricks. This is achieved by applying our confidence measure to detect unreliable pixels (marked in red) and estimate their depth at coarser image resolutions with the depth range bounded by estimates on the higher resolutions.*

$\|E(u', s') - \bar{\mathbf{r}}\| < \varepsilon$ with ε having the same value as in Equation (8). This step is a conservative visibility estimate and ensures that foreground objects in the EPI are not overwritten by background objects during the propagation.

As an alternative to the above test of radiance similarities, we experimented with running the full mean shift clustering on the set $\mathcal{R}(u, d)$ and propagating the depth estimate directly to the cluster elements, but we found that our simplified density estimation and the above procedure provide similar results at a fraction of the time.

Finally, low confidence depth estimates are discarded and marked for re-computation, and all EPI-pixels with a depth estimate assigned during the propagation are masked from further computations. A new part of the EPI is selected for depth computation by setting $\hat{s}$ to the nearest s with respect to the center of the EPI that still has unprocessed pixels. The method then starts over with the radiance sampling and depth computation as described in Section 4.3, until all edge confident EPI-pixels at the current EPI resolution have been either processed or masked during by the propagation.

4.5 Fine-to-Coarse Refinement

Parts of the EPI without assigned depth values are either ambiguous due to homogeneous colors (insufficient edge confidence), or have a strongly view dependent appearance (insufficient depth-confidence). However, since our method starts processing at the highest available resolution, the set Γ provides reliable reconstructions of all detailed features in the EPI and, in particular, of object silhouettes. The core idea of our fine-to-coarse strategy is now to compute depth in less detailed and less reliable regions by exploiting the regularizing effect of an iterative downsampling of the EPI. Furthermore, we enhance robustness and speed up the computation by using the previously computed confident depth estimates as depth interval bounds for the depth estimation at coarser resolutions. See Figure 5 for an example of our refinement strategy and note the improvement from subfigure (b) to (f) at the bricks.

First the depth bounds are set for all EPI-pixels without a depth estimate. As depth bounds, the algorithm uses the upper and lower bounds of the closest reliable depth estimates in each horizontal row of the EPI. Then the EPIs are downsampled by a factor of 0.5 along the spatial u and v-dimensions, while the resolution along the angular s-dimension is preserved. We presmooth the EPIs along the spatial dimensions using a 7×7 Gaussian filter with standard deviation $\sigma = \sqrt{0.5}$ to avoid aliasing. The required 7 EPIs are already in memory from the bilateral median filtering step (Equation (8)).

The algorithm then starts over at the new, coarser resolution with the previously described steps, i.e., edge confidence estimation, depth estimation and propagation. EPI-pixels with reliable depth estimates computed at higher resolutions are not considered anymore but only used for deriving the above described depth bounds. This fine-to-coarse procedure is iterated through all levels of the EPI pyramid until any of the image dimensions becomes less then 10 pixels. At the coarsest level, depth estimates are assigned to all pixels regardless of the confidence measurements. The depth estimates at coarser resolution levels are then successively upsampled to the respective higher resolution levels and assigned to the corresponding higher resolution EPI-pixels without a depth estimate, until all EPI-pixels at the finest resolution level have a corresponding depth estimate. As a final step we apply a 3×3 median to remove spurious speckles.

Note that unlike other algorithms based on multi-resolution processing and global regularization, our fine-to-coarse procedure (similar in spirit to the push-pull algorithm [Gortler et al. 1996]) starts at the highest resolution level and hence preserves all details, which is generally very challenging in classical, coarse-to-fine multi-resolution approaches. Our downsampling achieves an implicit regularization for less reliable depth estimates so that all processing steps are purely local at the EPI-level. Hence, even massive light fields can be processed efficiently.

5 Experimental Evaluation

This section briefly presents our setup for capturing 3D light fields and its calibration. We then show results and evaluations of our method, including comparisons to various state-of-the-art techniques in (multi-view) stereo. We also demonstrate exemplary applications such as segmentation and image-based rendering. Finally, we discuss how to generalize the algorithm for handling 4D light fields and unstructured input. The input light fields, our reconstructions, and additional results are available on our project webpage.

5.1 Capture Setup and Calibration

Setup. We captured 3D light fields by mounting a consumer DSLR camera on a motorized linear stage. The camera was a Canon EOS 5D Mark II with a 50 mm lens with which we captured images at various resolutions up to 21 MP. The linear stage was a Zaber T-LST1500D that is 1.5 meter long and can be controlled from a computer to obtain an accurate spacing of camera positions. We captured 100 images of each scene with uniform spacing between the camera positions and used them for reconstruction. The spacing between camera positions ranges from 2 mm to 15 mm.

The described setup worked well in practice for capturing high spatio-angular resolution light fields: it is cheaper and easier to handle than a full array of cameras, while yielding much higher spatial and angular resolutions than single light field cameras based on lenslet arrays or coded aperture. A typical capture session takes about 2 minutes, because for every picture we first move the camera, stop, take the picture, and move again to avoid motion blur during capture and to achieve higher image resolution. With a continuously moving setup the time could be reduced to a few seconds.

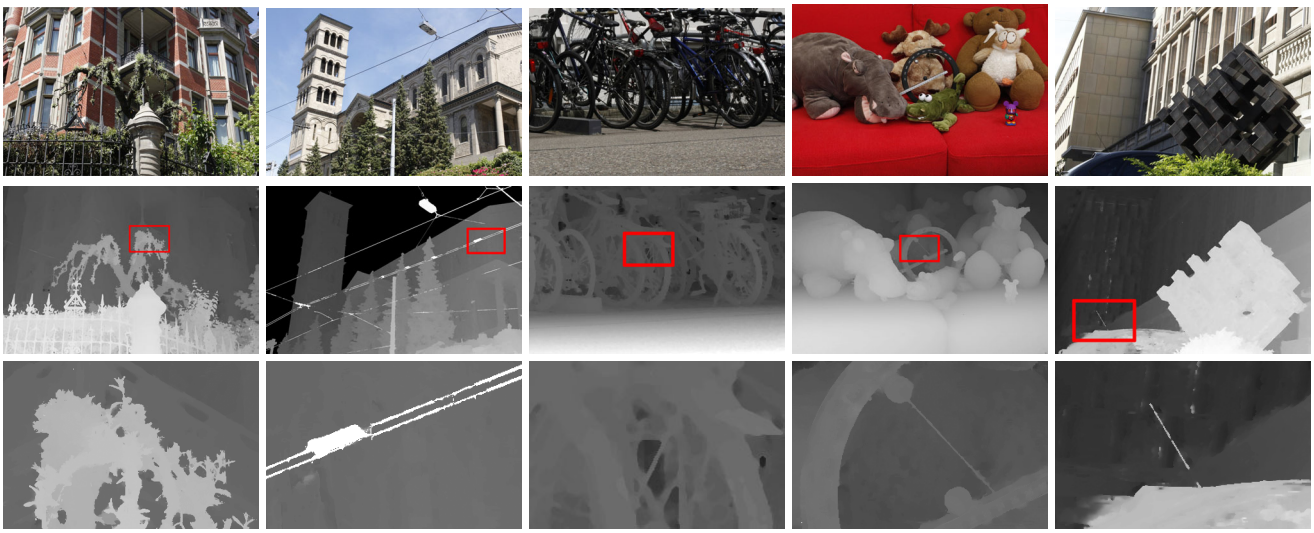


Figure 6: *Results on various 3D light fields.* ***Top to bottom:*** *One input image, corresponding depth map, and close-up of the highlighted region. For the Church we used color-based segmentation to exclude the homogeneous sky as no meaningful depth can be computed there.*

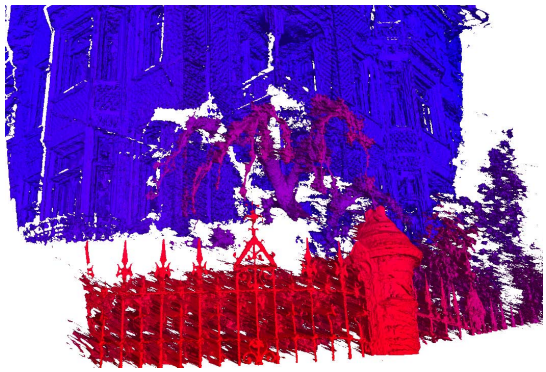
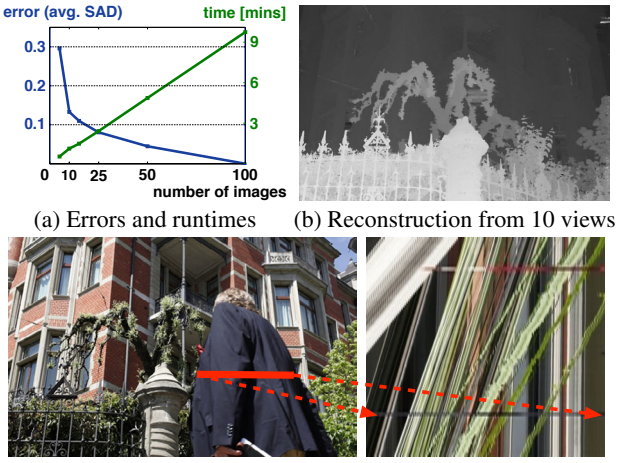


Figure 7: *Shaded 3D mesh, generated by triangulating individual depth maps and merging them into a single model. Color encodes depth. More 3D meshes are shown in our supplementary material.*



(a) Errors and runtimes (b) Reconstruction from 10 views

(c) Robustness to inconsistencies and outliers

Figure 8: *Robustness of our method. (a) Reconstruction error and runtimes for varying numbers of input views. (b) Reconstruction from only 10 views. (c) Our method is also robust to inconsistencies and outliers in the data, e.g., people walking by (horizontal lines) or moving plants (jagged green lines, see also plants in Figure 1).*

Calibration. To closely approximate a regularly sampled 3D light field we first correct the captured images for lens distortion using PTLens[2], and then compensate for mechanical inaccuracies of the motorized linear stage. To this end we estimate the camera poses using Voodoo camera tracker[3], compute the least orthogonal distance line from all camera centers as a baseline, and then rectify all images with respect to this baseline [Fusiello et al. 2000].

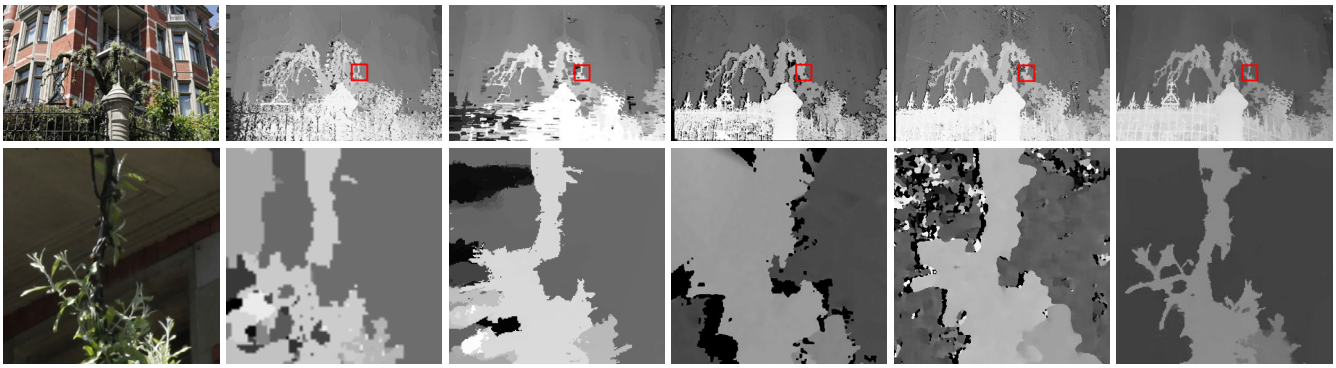
5.2 Results

Using above setup we captured a variety of 3D light fields of challenging outdoor and indoor scenes. In Figure 6 we show example input images and corresponding depth maps. However, our algorithm computes depth for every scene point that is visible in the input images. Hence, from our internal representation we can efficiently extract depth maps for each input view, as well as generate alternative scene representations like 3D point clouds. Figure 7 additionally shows a 3D mesh extracted from our reconstructions. Although we usually achieve a lower accuracy in terms of absolute distance compared to a laser scanner, our method faithfully reproduces fine details of complex, cluttered scenes, with precise reconstruction of object contours, performing well on homogeneous regions at the same time. These properties are highly desirable in applications such as segmentation (Figure 12) or novel view synthesis with only moderate viewpoint changes (Figure 13).

Robustness and performance. Figure 8 (a) and (b) demonstrate the robustness of our algorithm for different numbers of input views. We ran our experiments on a desktop PC with an Intel iCore 7 3.2 GHz CPU and an NVidia GTX 680 graphics card, and tested a set of 256 depth hypotheses for every EPI-pixel in all experiments. As a baseline solution, we computed a result from 100 input views at the full 21 MP resolution and evaluated the error using normalized sum-of-absolute differences (SAD). While our algorithm benefits from a large number of input views, reasonable results can still be achieved

[2] http://www.epaperpress.com/ptlens/

[3] http://www.digilab.uni-hannover.de/docs/manual.html



(a) Input (b) Graph Cut [∼1 day on 1 MP] (c) CostVolume Filter [∼1 day on 1 MP] (d) Dense seed&grow [10 mins] (e) Semi-global match [4 mins on 4 MP] (f) ours [50 mins at 21MP]

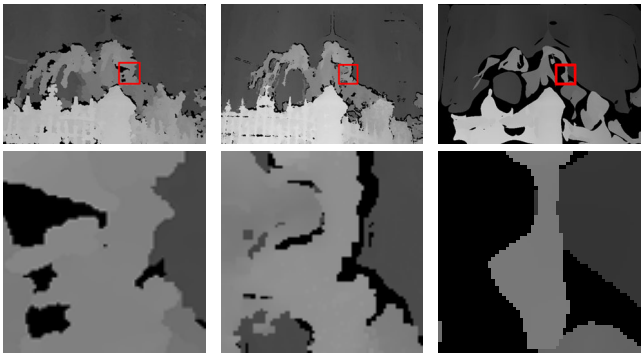
Figure 9: *Comparison to two-view stereo methods on the Mansion data set.* ***From left to right:*** *(a) One input image, (b) [Kolmogorov and Zabih 2001], (c) [Rhemann et al. 2011], (d) [Geiger et al. 2010], (e) [Hirschmüller 2005] . The numbers in brackets denote the running time for 50 views in the light field, but are measured with different implementations (C/Matlab) and processor types (CPU/GPU).*

with only 10 input views (see Figure 8 (b)). A typical runtime for a single depth map using 100 views at 21 MP resolution is about 9 minutes. With our current implementation, the full propagation to 50 views takes about 50 minutes. The linear dependence of the runtimes on the number of images is illustrated in Figure 8 (a). For example, for 10 views a single depth map requires about 1 minute.

Our method is robust against varying baseline and angular separations caused by different distances between the camera positions and the scene points. For the results shown in Figure 6 the angular separations range from 1.5° up to 13°. The example in Figure 15 captured with a hand-held camera features a considerable angular separation from 9° to 41° as well as a large baseline of about 300 meters. In addition our algorithm is robust to non-static scene elements like people moving in front of the camera or plants moving in the wind (Figure 8 (c)). For instance, the sparse horizontal color artifacts visible in the input EPI in Figure 1 are caused by people passing by during capture. The density estimation in Equation (4) simply regards those radiance values as outliers and still produces a consistent result from the remaining samples.

The influence of the two most relevant parameters in our method, the kernel bandwidth h and the color tolerance ε of the bilateral median, is conceptually similar to adjusting the window size in stereo methods comparing image patches. An increase of h and ε compared to our default values increases robustness to noise, whereas smaller values better preserve fine details.

Comparison to (multi-view) stereo. We processed the Mansion data set with a number of state-of-the-art techniques in two-view and multi-view stereo, and also ran our algorithm on a number of standard benchmark datasets. However, please note that most of these algorithms have been designed with different application scenarios in mind. Hence these comparisons are meant to illustrate the novel challenges for the field of image-based reconstruction arising from the ability to capture increasingly dense and higher resolution input images. For each method we hand-optimized parameters and the camera separation of the input images for best reconstruction quality. Comparing the results in Figure 9 and focusing on the closeups, issues of existing methods with such highly detailed scenes become obvious. The popular *graph cuts* [Kolmogorov and Zabih 2001] as well as the more recent *cost volume filtering* approach [Rhemann et al. 2011] are time and memory intensive and could not process resolutions higher than 1 MP. Both methods reconstruct sharp boundaries, but they are not well localized due to the low resolution. Homogeneous image regions are problematic as well. Good



(a) Furukawa and Ponce [2010] [6 mins on 10 MP] (b) Beeler et al. [2010] [20 mins on 10 MP] (c) 123D Catch [5 mins on 10 MP]

Figure 10: *Comparison to multi-view stereo methods.*

Figure 11: *Result on the* flower garden *sequence with 50 images.* ***Left:*** *One input image with 0.08 MP resolution.* ***Right:*** *Our depth map. The computation time was 3 seconds.*

performances in terms of memory and runtime are achieved by the *dense seed-and-grow* approach of Geiger et al. [2010] and by *semi-global matching* [Hirschmüller 2005] (as implemented in OpenCV). However, these methods show problems in homogeneous regions and around object contours as well (see black pixels). Leveraging the huge amount of data in a corresponding light field of the scene, our fine-to-coarse procedure reconstructs detailed, well-localized silhouettes and plausible depth estimates in homogeneous regions at reasonable run times.

In Figure 10 we show results of recent multi-view stereo methods. For comparison to our result in Figure 9 (f) we show a 3D rendering of the point clouds which is colored in accordance to depth and selected a similar closeup region as before. The method of Furukawa

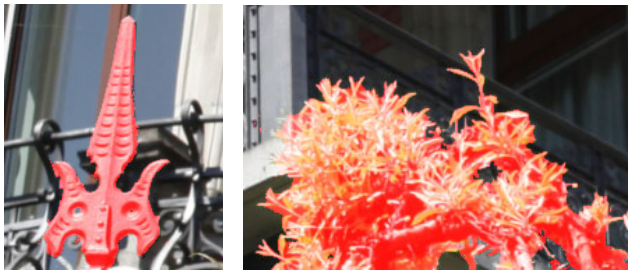


Figure 12: *Closeups of depth-based segmentations of the Mansion data set. Note the high level of detail and that foreground and background would be very difficult to distinguish solely based on color.*

and Ponce [2010] leverages information from 50 views of the light field. We also compare against the method of [Beeler et al. 2010] that was originally developed for high quality face reconstruction and that uses 8 input images. As it is optimized for faces, its core assumptions regarding smoothness and surface continuity are violated, hence the authors processed our dataset running only the initial multi-view matching part of their pipeline. Overall both approaches achieve good reconstructions, but lack details around contours and miss some homogeneous regions in comparison to our method. We also show a result produced using the commercial tool *Autodesk 123D Catch*[4] that to our knowledge is based on the work of Vu et al. [2009]. The application could process 10 images and produced a very smooth result that, however, lacks any detail.

We also ran our method on classic stereo data that has been used in the stereo community for benchmarking. These datasets differ significantly from the fundamental assumptions behind our algorithm as they encompass a relatively small number of low resolution input images. In Figure 11 we show our result on the *flower garden* sequence[5] (50 images, 0.08 MP). On this small spatial resolution, our method takes about 3 seconds to compute a depth map with quite accurate silhouettes. However, due to missing texture in the sky, artifacts in the top left corner arise. In our supplementary material we show additional comparisons on classic stereo data [Szeliski and Scharstein 2002; Zitnick et al. 2004]. For this low spatio-angular resolution data (5–8 images, $\leq$ 0.8 MP) the quality degrades tangibly as our method has been specifically designed to operate on the pixel level by leveraging highly coherent data. In such scenarios, methods employing comparisons of whole image patches and global regularization are advantageous.

5.3 Applications

Scene reconstruction finds a number of immediate uses in applications related to computer graphics besides generating a 3D model of a scene. In the following we illustrate how the output of our method can be directly used for applications such as automatic image segmentation as well as image-based rendering.

Segmentation. Despite being a common task in movie production, automatic segmentation like background removal is still a challenge in detailed scenes. Due to the precise object contours in our reconstructions we can use our method for automatically creating high quality segmentations. For the shown results we simply thresholded all pixels within a prescribed depth interval. Using our depth this approach is not only easy to implement, but also supports real-time updates to the segmentation even on the high resolution images. In Figure 12 we show results on the Mansion data set. We wish to stress that such results would be very difficult to obtain using classical color-based or manual segmentation due to the extreme detail in this scene and the partially similar colors between foreground and background.

[4]http://www.123dapp.com/catch

[5]http://persci.mit.edu/demos/jwang/garden-layer/orig-seq.html

Figure 13: *Examples for novel view-synthesis by rendering a colored point cloud. The leftmost image is from the set of input images.*

Image-based rendering. Another benefit of our method is that we get consistent depth estimates for any input view of the light field, i.e., we compute as complete a scene reconstruction as possible from the available input data. Thus, we can directly visualize our results as a colored 3D point cloud using splat-based rendering , with the ability to look around occluding objects (see Figure 13). Moreover, we can use the delta EPI representation to reproduce view dependent effects during rendering, e.g., using a weighting scheme as proposed in [Buehler et al. 2001].

5.4 Extension to 4D and Unstructured Light Fields

It is straightforward to generalize our reconstruction algorithm to inputs that do not correspond to a regularly sampled 3D light field.

4D light fields. In a regular 4D light field the camera centers are horizontally and vertically displaced, leading to a 4D parametrization of rays as $\mathbf{r} = L(u, v, s, t)$, where t denotes the vertical ray origin. The ray sampling from Equation (3) is then extended to

$$\mathcal{R}(u, v, s, t, d) = \{L(u+(\hat{s}-s)\,d, v+(\hat{t}-t)\,d, s, t) \mid s = 1, ..., n,\ t = 1, ..., m\}, \quad (9)$$

where $(\hat{s}, \hat{t})$ is the considered view and m denotes the number of vertical viewing positions. This leads to sampling a 2D plane in a 4D ray space instead of the 1D line in case of 3D light fields. The depth propagation also takes place along both the s and t-directions.

(a) Image (b) GCL [15 min] (c) Ours [1 min]

Figure 14: *Comparison of globally consistent labeling (GCL) [Wanner and Goldlücke 2012] (b) to our result (c) on a 4D light field.*

A result for a 4D light field from the Stanford database[6] is shown in Figure 14 where we also provide a visual comparison to the 4D light field depth estimation method by [Wanner and Goldlücke 2012]. While they achieve already appealing results, our method resolves additional details, e.g., on the wheels and the small holes in the Lego bricks. They report a timing of 15 minutes, whereas ours takes 64 seconds. More results on 4D light fields including quantitative ground truth comparisons are given in our supplementary material.

[6]http://lightfield.stanford.edu/lfs.html

Unstructured light fields. For arbitrary, unstructured input we loose the efficiency of the EPI-based processing, but the reconstruction quality remains. In this scenario we use the camera poses estimated in the calibration phase (Section 5.1) to determine the set of sampled rays for a depth hypothesis. More precisely, we back-project each considered pixel to 3D space in accordance to the hypothesized depth and then re-project the 3D position to the image coordinate systems of all other views to obtain the sampling positions. Formally, the set of sampled rays becomes

$$\mathcal{R}(u,v,s,d) = \{L(u',v',s) \mid s = 1,...,n, \\ P_s^{-1}[u'\ v'\ f\ d]^\top = P_{\hat{s}}^{-1}[u\ v\ f\ d]^\top\}, \quad (10)$$

where P_s denotes the camera rotation and translation of view s (estimated in the calibration phase) and f is the camera focal length in pixels.

In Figure 15 we show an example for a challenging hand-held capture scenario. The input images have been taken on a boat in front of the skyline of Shanghai, with considerable variation in orientation of the camera and of the colors within the scene. We segmented the sky and the water surface. To assess the quality of our reconstruction we also show a bird's-eye view overlaid on a satellite image of this area. Please see also the supplemental video for the input sequence and animated novel viewpoint renderings. Computing depth took 162s per view at 3 MP spatial resolution using 100 images. For such unstructured input we observed an increase in running time of about 50% compared to structured 3D input.

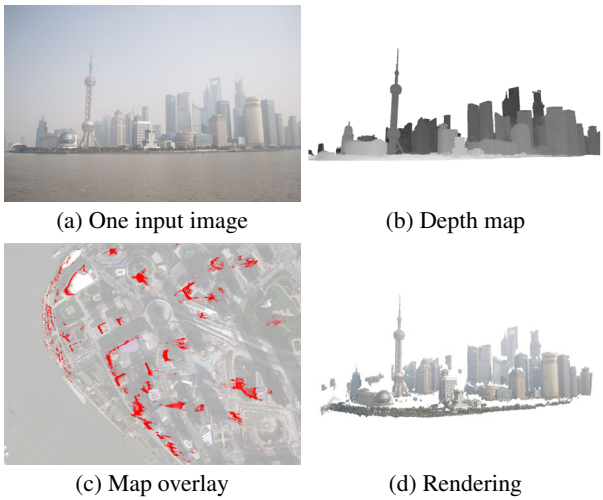
6 Limitations and Future Work

We presented a method for scene reconstruction from densely sampled 3D light fields. A limitation of our method are surfaces with spatially varying reflectance, as they violate the assumptions behind the radiance density estimation. This is for example apparent in the reconstruction of the metallic car surface on the bottom left in the Statue dataset, Figure 6. This dataset also contains comparably large homogeneous areas in the background, leading to slightly noisy depth estimates in these regions. In some cases, however, like for the windows in the Mansion dataset, the combination of our confidence measures and the fine-to-coarse approach succeeds in plausibly filling even such difficult regions. However, a more principled approach would of course be desirable, e.g., following Criminisi et al. [2005]. In future work we plan to combine our ray density estimation with more sophisticated reflectance models. Low contrast between foreground and background objects over the whole light field may also lead to problems, as witnessed on some parts of the cables in the Church sequence in Figure 6. Finally, while our reconstructions feature precise contours and are very complete as they produce a depth estimate for every input ray, we achieve lower accuracy in terms of absolute distance measurements than a laser scanner. To improve accuracy, investigating a continuous refinement of our discrete depth labels also seems promising.

While the reconstruction of static scenes already has a number of applications, extending our method to temporally varying light fields of dynamic scenes, e.g., using an array of high resolution cameras, provides many interesting new opportunities and challenges. We believe that such very high resolution data may require a rethinking of existing algorithm designs, e.g., using global optimization.

Acknowledgements

We would like to thank Simon Heinzle, Wojciech Matusik, Thabo Beeler and Oliver Wang for valuable feedback and help with comparisons. We are grateful to Paul Beardsley and Skye project team for the Shanghai dataset.



(a) One input image (b) Depth map

(c) Map overlay (d) Rendering

Figure 15: *Results on a challenging unstructured light field, obtained by hand-held capture (a) from a floating boat. (b) A resulting depth map. (c) Overlay of our reconstruction on a satellite image ©2013 DigitalGlobe, Google. (d) Rendering from a novel viewpoint.*

References

ADELSON, E. H., AND WANG, J. Y. A. 1992. Single lens stereo with a plenoptic camera. *IEEE PAMI 14*, 2.

AYVACI, A., RAPTIS, M., AND SOATTO, S. 2012. Sparse occlusion detection with optical flow. *IJCV 97*, 3.

BASHA, T., AVIDAN, S., HORNUNG, A., AND MATUSIK, W. 2012. Structure and motion from scene registration. In *CVPR*.

BEELER, T., BICKEL, B., BEARDSLEY, P. A., SUMNER, B., AND GROSS, M. H. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. 29*, 4.

BISHOP, T. E., AND FAVARO, P. 2010. Full-resolution depth map estimation from an aliased plenoptic light field. In *ACCV*.

BISHOP, T., ZANETTI, S., AND FAVARO, P. 2009. Light field superresolution. In *ICCP*.

BLEYER, M., ROTHER, C., KOHLI, P., SCHARSTEIN, D., AND SINHA, S. 2011. Object stereo — joint stereo matching and object segmentation. In *CVPR*.

BOLLES, R. C., BAKER, H. H., AND MARIMONT, D. H. 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV 1*, 1.

BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S. J., AND COHEN, M. F. 2001. Unstructured lumigraph rendering. In *SIGGRAPH*.

ČECH, J., AND ŠÁRA, R. 2007. Efficient sampling of disparity space for fast and accurate matching. In *CVPR*.

CHAI, J., CHAN, S.-C., SHUM, H.-Y., AND TONG, X. 2000. Plenoptic sampling. In *SIGGRAPH*.

CHEN, W.-C., BOUGUET, J.-Y., CHU, M. H., AND GRZESZCZUK, R. 2002. Light field mapping: Efficient representation and hardware rendering of surface light fields. In *SIGGRAPH*.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE PAMI 24*, 5.

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. *CVIU 97*, 1.

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

DUDA, R., HART, P., AND STORK, D. 1995. *Pattern Classification and Scene Analysis, 2nd ed.*

FITZGIBBON, A., WEXLER, Y., AND ZISSERMAN, A. 2005. Image-based rendering using image-based priors. *IJCV 63*, 2.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multi-view stereopsis. *IEEE PAMI 32*, 8.

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

FUSIELLO, A., TRUCCO, E., AND VERRI, A. 2000. A compact algorithm for rectification of stereo pairs. *Mach. Vis. Appl. 12*, 1.

GEIGER, A., ROSER, M., AND URTASUN, R. 2010. Efficient large-scale stereo matching. In *ACCV*.

GEORGIEV, T., AND LUMSDAINE, A. 2010. Reducing plenoptic camera artifacts. *Comp. Graph. Forum 29*, 6.

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

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The Lumigraph. In *SIGGRAPH*.

HIRSCHMÜLLER, H. 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*.

HUMAYUN, A., MAC AODHA, O., AND BROSTOW, G. 2011. Learning to find occlusion regions. In *CVPR*.

ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *SIGGRAPH*.

KANG, S. B., AND SZELISKI, R. 2004. Extracting view-dependent depth maps from a collection of images. *IJCV 58*, 2.

KOLMOGOROV, V., AND ZABIH, R. 2001. Computing visual correspondence with occlusions via graph cuts. In *ICCV*.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *SIGGRAPH*.

LIANG, C.-K., LIN, T.-H., WONG, B.-Y., LIU, C., AND CHEN, H. H. 2008. Programmable aperture photography: multiplexed light field acquisition. *ACM Trans. Graph. 27*, 3.

NG, R., LEVOY, M., BRÉDIF, M., DUVAL, G., HOROWITZ, M., AND HANRAHAN, P. 2005. Light field photography with a hand-held plenoptic camera. *Comp. Sci. Techn. Rep. CSTR 2*.

RAV-ACHA, A., SHOR, Y., AND PELEG, S. 2004. Mosaicing with parallax using time warping. In *IVR*.

RHEMANN, C., HOSNI, A., BLEYER, M., ROTHER, C., AND GELAUTZ, M. 2011. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV 47*, 1-3.

SCHECHNER, Y. Y., AND KIRYATI, N. 2000. Depth from defocus vs. stereo: How different really are they? *IJCV 39*, 2.

SEITZ, S. M., AND DYER, C. R. 1999. Photorealistic scene reconstruction by voxel coloring. *IJCV 35*, 2.

SEITZ, S., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2008. Modeling the world from Internet photo collections. *IJCV 80*, 2.

STICH, T., TEVS, A., AND MAGNOR, M. A. 2006. Global depth from epipolar volumes–a general framework for reconstructing non-lambertian surfaces. In *3DPVT*.

SUN, X., MEI, X., JIAO, S., ZHOU, M., AND WANG, H. 2011. Stereo matching with reliable disparity propagation. In *3DIMPVT*.

SYLWAN, S. 2010. The application of vision algorithms to visual effects production. In *ACCV*.

SZELISKI, R., AND SCHARSTEIN, D. 2002. Symmetric sub-pixel stereo matching. In *ECCV*.

VAISH, V., LEVOY, M., SZELISKI, R., ZITNICK, C., AND KANG, S. 2006. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *CVPR*.

VEERARAGHAVAN, A., RASKAR, R., AGRAWAL, A. K., MOHAN, A., AND TUMBLIN, J. 2007. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph. 26*, 3.

VU, H.-H., KERIVEN, R., LABATUT, P., AND PONS, J.-P. 2009. Towards high-resolution large-scale multi-view stereo. In *CVPR*.

WANNER, S., AND GOLDLÜCKE, B. 2012. Globally consistent depth labeling of 4D light fields. In *CVPR*.

WANNER, S., FEHR, J., AND JAEHNE, B. 2011. Generating EPI representations of 4D light fields with a single lens focused plenoptic camera. In *IISVC*.

WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E.-V., ANTÚNEZ, E. R., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High performance imaging using large camera arrays. *ACM Trans. Graph. 24*, 3.

WOOD, D. N., AZUMA, D. I., ALDINGER, K., CURLESS, B., DUCHAMP, T., SALESIN, D. H., AND STUETZLE, W. 2000. Surface light fields for 3D photography. In *SIGGRAPH*.

YU, Y., FERENCZ, A., AND MALIK, J. 2001. Extracting objects from range and radiance images. *IEEE TVCG 7*, 4.

ZHANG, C., AND CHEN, T. 2004. A self-reconfigurable camera array. In *EGSR*.

ZHU, Z., XU, G., AND LIN, X. 1999. Panoramic EPI generation and analysis of video from a moving platform with vibration. In *CVPR*.

ZIEGLER, R., BUCHELI, S., AHRENBERG, L., MAGNOR, M. A., AND GROSS, M. H. 2007. A bidirectional light field - hologram transform. *Comput. Graph. Forum 26*, 3.

ZITNICK, C. L., AND KANG, S. B. 2007. Stereo for image-based rendering using image over-segmentation. *IJCV 75*, 1.

ZITNICK, C. L., KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2004. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. 23*, 3.