EUROGRAPHICS 2016 / E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Efficient Multi-image Correspondences for On-line Light Field Video Processing

Ł. Dąbała[1,2] M. Ziegler[3] P. Didyk[1,4] F. Zilly[3] J. Keinert[3] K. Myszkowski[1] H.-P. Seidel[4] P. Rokita[2] T. Ritschel[5]

[1]MPI Informatik [2]Warsaw University of Technology [3]Fraunhofer IIS [4]Saarland University [5]University College London

Figure 1: *Our real-time multi-view correspondence algorithm extracts multi-view depth maps from sparse, wide-baseline light field video (here 3×3 cameras), in order to produce high-quality novel views for applications such as virtual apertures or virtual camera positions.*

Abstract

Light field videos express the entire visual information of an animated scene, but their shear size typically makes capture, processing and display an off-line *process, i. e., time between initial capture and final display is far from real-time. In this paper we propose a solution for one of the key bottlenecks in such a processing pipeline, which is a reliable depth reconstruction possibly for many views. This is enabled by a novel correspondence algorithm converting the video streams from a sparse array of off-the-shelf cameras into an array of animated depth maps. The algorithm is based on a generalization of the classic multi-resolution Lucas-Kanade correspondence algorithm from a pair of images to an entire array. Special inter-image confidence consolidation allows recovery from unreliable matching in some locations and some views. It can be implemented efficiently in massively parallel hardware, allowing for interactive computations. The resulting depth quality as well as the computation performance compares favorably to other state-of-the art light field-to-depth approaches, as well as stereo matching techniques. Another outcome of this work is a data set of light field videos that are captured with multiple variants of sparse camera arrays.*

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image processing and Computer Vision]: Scene Analysis—Shape

1. Introduction

Traditional images and videos are limited to a single view point of the scene. In contrast, a light field [Gab08, LH96] includes multiple view points by describing individual light rays. This information conveys to the human observer all important visual cues, which are known from the real world observation, such as binocular disparity, eye accommodation, motion parallax, and view-dependent effects, such as highlights and reflections. Light field video adds on top the temporal component that can express the full visual information of an animated scene, and consequently would be the ultimate entity to capture and reproduce in film production, broadcasting, teleconferencing, or interactive applications such as computer games and virtual reality. Currently, the use of light field video is limited by the explosion in bandwidth: directly capturing a dense light field video requires huge effort, processing time and storage, although the captured data is highly redundant.

Due to those challenges, light field videos are typically dealt within an *off-line* process, i. e., many images are captured, often by many cameras, next they are stored and processed, and, under some conditions, can be manipulated or displayed. One of the key bottlenecks that affects the proliferation of real-time light field video pipelines is the depth map computation for multiple-views, where

apart from the performance issues, also the inter-view consistency and robustness is a major challenge. This problem is even more difficult for sparse light fields as captured with wide baseline camera arrays, where disparities between corresponding image features may amount to hundreds of pixels. Such video light fields are in particular interesting as they enable for large ranges of virtual camera motion, and rely on a small number of video streams, which can ease the compression and transmission in practical applications.

In this work we address the problem of depth map reconstruction for sparse video light fields at interactive rates. This is possible due to a novel multi-video correspondence algorithm that finds a dense field of correspondences in all views of the light field at the same time. The algorithm makes use of specific inter-view and intra-view constraints to rely only on confident matches, and a novel consolidation step propagates such reliable information across different views. For images rectified onto a common plane, those correspondences can be interpreted as disparities which are proportional to the Euclidean distance between the respective camera centers. Conversion into multi-view depth maps can be achieved by simply calculating the reciprocal value of the disparity and multiplying by a proportionality constant. The resulting depth maps have edges aligning with the RGB image and can be used to provide many light-field enabled effects. The algorithm exploits redundancies in light field images, which enable a confidence-driven, robust multi-resolution approach that can easily be implemented on massively parallel graphics hardware. To test our algorithm we develop a representative dataset of video light fields for multiple configurations of synchronized and calibrated camera arrays.

2. Previous work

A range of different systems have been devised to capture, process and reproduce light fields. In this section, we will recall capture modalities and their relation to the required processing and resulting image quality.

Light field capture An ideal light field capture device is characterized by a high angular resolution (dense sampling of the different view points) over a large baseline. Unfortunately, this combination is extremely challenging due to high data volumes and capture costs. Consequently, commercial products from Lytro [NLB*05], Raytrix [PW12], Pelican [VLD*13] or Fraunhofer [BOD*16] and research prototypes such as programmable apertures [LLW*08] or kaleidoscopic imaging [MRK*13] are often limited to small baselines. Scaling the principle of dense angular sampling to larger baseline leads to voluminous capture devices like the recent Lytro Cinema camera, while the maximum distance between captured views is still limited. Due to these reasons, there is an increased interest in array-based light field capture. Gantry systems [LH96, WG14, KZP*13] permit a dense sampling of static objects. Video camera arrays are often restricted to rectangular capture [ZKU*04, TNM09]. Only a few camera arrays are known that can capture the complete light field information. Unfortunately, for many of them [YWB03, WJV*05, FMT*06, MGBP11], resolution is lower than full-HD. Ziegler et al. [ZEM*15] present a system that can capture full-HD light fields. A sparse sampling followed by a depth-based image warping permits to limit the number of required camera devices and hence enables video capture. Unfortunately, computation of the required disparity maps is computationally still quite expensive, preventing a real-time work flow.

Light field depth estimation Most Light field depth estimation algorithms rely on densely sampled input [LLW*08, MRK*13, WER16, THMR13, CLY*14, YGL*13, ZLD15] as produced from commercial products such as Lytro but require substantial computational effort. While our approach achieves interactive rates, 30 CPU minutes for one depth map reported by Liang et al. [LLW*08] or 240 GPU second achieved by a fast variant of Wanner and Goldlücke's approach [WG14] seem typical.

Dense sampling allows for the use of structure tensors on epipolar images [WG14] or the use of phase space [ZLD15]. We demonstrate our approach to work for sparse light fields, where the low number of slices would not allow fitting any slope to infer depth.

For sparse light fields, depth can be estimated by sweeping [YWB03, ZEM*15]. Regrettably, this does not scale well with input view count, pixel count and depth resolution or baseline: for each pixel, each depth hypothesis must be tested against each camera. If there are many views, many pixels and fine variations of depth together with a large baseline, computational complexity prevents interactive application. Thanks to our multi-resolution matching, our results show disparities of up to 256 px from HD resolution input at 3×5 views.

Denker and Umlauf [DU11] also have investigated multi-resolution matching for multi-view content. Not considering any explicit handling of confidence and without accounting for luminance guides, depth edges can become blurred. Our solution resolves them at quality similar to methods performing a full sweep as seen from our comparison.

Other methods have explicitly reasoned about occlusion [LLW*08] at a single resolution. Our approach does not handle occlusion explicitly and rather considers it as a special case of a general failure-to-match, which can also be due to specularity, to image noise or due to the inability to match on a certain resolution in our pyramid. Woetzel and Koch [WK04] have previously accounted explicitly for occlusions, but when using a plane sweep approach, at a single resolution, limiting speed, respectively, resolution.

Large-baseline correspondences Corresponding image features that are hundreds of pixels apart from each other can be hard to find using sweeping or epipolar analysis (e. g., structure tensors or phase space). Solving this issue is possible by recurring to modern correspondence algorithms that preserve luminance edges [MZK10, YWB03]. A large variety of flavors have been proposed in literature [SCD*06, HI16] trading computation effort against achievable quality. Some of them have been particularly designed for real-time operation [Hir08, RHB*11]. Innovative concepts have been proposed to perform fast multi-camera disparity estimation [FWAS10, ZRMK12], but typically the number of cameras is limited to three or four when achieving up to 24 Hz for full-HD resolution, while our light fields consist of nine to sixteen views.

In order to design a disparity estimation for large and many-camera light fields, we seek inspiration in traditional stereo multi-resolution correspondence algorithms like Lucas-Kanade [LK81]

and Horn-Schunck [HS81] to extend it to the multi-view case. Modern image-pair (stereo-) correspondence algorithms based on multi-resolution optical flow [BWS05, YWA10, LYT11] combined multiple resolution with warping, variational optimization or Belief Propagation with great success. However, to our knowledge, no combination of multi-resolution processing to multi-view data has been proposed such that edges are preserved. Our approach achieves this by tracking confidence and avoiding up-sampling across luminance edges.

Our multi-view case is both more challenging, as more data has to be produced, but at the same time also more simple, as the multiple views impose mutual constraints on the depths. Our approach identifies correspondences that contradict those constraints and eliminate them during the multi-resolution procedure by tracking confidence.

3. Overview

Our main contribution is a method to establish dense correspondences between all pairs of neighboring cameras, by assigning a per-pixel disparity label in all views. Input is the rectified multi-view video set. Output is a depth map for every view in every frame. Note, that this is different from other solutions that estimate a single depth map from multiple images.

In Sec. 4 we present our multi-resolution and multi-view correspondence algorithm. Fig. 2 summarizes major components of this algorithm, which we briefly overview next. The initial disparity values are acquired based on the computation outcome for the lower resolution level (Sec. 4.1). Next, the correspondence matching is done between four nearest neighbors, which results in the updated disparity (Sec. 4.2) and confidence (Sec. 4.3) maps. The maps are then used in the consolidation step (Sec. 4.4), which produces combined disparity and confidence for all views. Finally, the confidence is used again, but this time in the spatial domain along with RGB-guided and edge- preserving filtering, in the disparity map up-sampling step to the next resolution level (Sec. 4.5). In Sec. 4.6 we provide details of our massively parallel implementation.

4. Interactive multi-view correspondence

A simple solution to assign depth labels would be to sweep all depth values and choose the one for each pixel that produces the best agreement when re-projecting the patch around the pixel into all other views. In a second step, the resulting depth could be filtered for outliers and to enforce smoothness. Bilateral filtering as in [MZK10] is very common for this purpose. This approach is simple to implement, yet effective, but regrettably requires substantial computational effort as all depth hypothesis for all pixels need to be swept. With a baseline of hundreds of pixels, this is no option for the desired on-line capture.

Instead, we draw inspiration from Lukas-Kanade correspondence [LK81]. Here, the matching of two images was accelerated, by working in multiple resolutions. In a coarser resolution, the neighborhood to search becomes smaller: What might have been patches of 200 pixels spatial distance, on a higher level become neighboring pixels. We apply the same idea to multiple images: All images are first matched in a reduced resolution, then the solution is up-sampled and refined, where the refinement uses the previous solution as an initial guess. Typically, we start at level 6, what reduces the input resolution by 2^6.

One difficulty of multi-resolution approaches is that, if matching fails at one level, it will not recover from it at higher levels. Fortunately, light-fields are highly redundant, allowing us to devise an intra- and inter-view consolidation step which identifies which matches are reliable and which are not.

Matching is done independently between different frames of the videos, so the following will describe matching images from a single, synchronized point in time.

4.1. Basic terms

Definitions Let $\mathcal{L} = \{L_i \in \mathbb{R}^2 \to \mathbb{R}^3\}$ be the input, a set of n_v images, where L_i associates each 2-dimensional pixel coordinate with an RGB color triple. Typically, n_v is 3×3 or 4×4, and all images are rectified onto a common plane, but the spacing in camera placement can be in general different between particular camera rows and columns. Therefore, disparities need to be normalized according to the multi-baseline-matching approach described in [Hir08]. To account for such possibly variable camera spacing we introduce a distance vector $\mathbf{r}_{i,i_2} \in \mathbb{R}^2$ for every camera pair i and i_2. This way for any point $\mathbf{x} \in \mathbb{R}^2$ at view i the corresponding visible point $\mathbf{x}_2$ in view i_2 can be found as $\mathbf{x}_2 = \mathbf{x} + \mathbf{r}_{i,i_2} \cdot D_i(\mathbf{x})$, or conversely $\mathbf{x} = \mathbf{x}_2 + \mathbf{r}_{i_2,i} \cdot D_{i_2}(\mathbf{x}_2)$, where $D_i(\mathbf{x})$ and $D_{i_2}(\mathbf{x}_2)$ denote a form of disparity magnitude akin to the scene depth, which should be the same for non-occluded scene regions. This way the distance vector $\mathbf{r}_{i,i_2}$ encapsulates information on the geometrical configuration between each camera pair, and the disparity magnitude $D_i(\mathbf{x})$ is just a scalar multiplier which enables to derive the actual pixel disparity between view i and any other view i_2. The output of our method is a set $D = \{D_i \in \mathbb{R}^2 \to \mathbb{R}^+ \cup \{0\}\}$ of such disparity magnitude maps for all views as specified by $\mathcal{L}$. For simplicity in the following text we refer to D_i as disparity.

Pyramid First, Gaussian pyramid [BA83] representation of L is built, using a 3×3 stencil. The j-th level in this pyramid is denoted as $L^{(j)}$. The algorithm will estimate disparity magnitude for all levels j and views i, in parallel over the view image pixels and sequentially over the levels.

Descriptors Instead of matching directly over the RGB values in $\mathcal{L}$, simple 5×5 census [ZW94] binary descriptors are used. Each descriptor is stored into a single 32-bit integer value. This both allows to perform faster matching by comparing only small integer descriptors as well as it makes the matching more robust [LYT11]. The pyramid in this descriptor space is denoted as $\mathcal{B} = \{B^{(j)}\}$, and $\ominus$ indicates the scalar differences operator between two descriptors.

4.2. Matching

Matching is done sequentially for all levels, but independently for all views and pixels. Outcome is a multi-view disparity map $D_i^{(j)}$, as well as a multi-view confidence map $C_i^{(j)}$. Different from other approaches, the output are multiple maps of depth and confidence, one

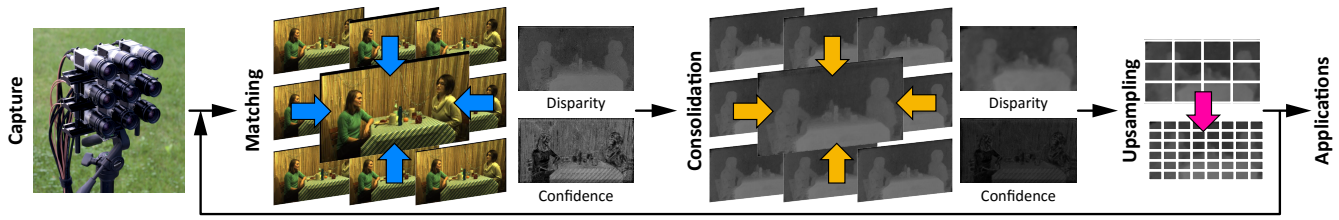

Figure 2: *The computation flow in our multi-resolution and multi-view correspondence algorithm (refer to the text for details).*

for each view, instead of a single depth map. The disparity $D_i^{(j)}$ and confidence $C_i^{(j)}$ maps result from consolidating of per-view disparity $\bar{D}_i^{(j)}$ and confidence $\bar{C}_i^{(j)}$ maps for all views in $\mathcal{L}$. Additionally, we assume the disparity map $D_i^{(j-1)}$ of the start level to be already known. This level is at a coarse resolution and can be initialized with zero or with an exhaustive search for a low number of pixels. $\uparrow D_i^{(j-1)}$ denotes an upscaled version of $D_i^{(j-1)}$ to the resolution of j-th level and is used as an initial disparity approximation in the $D_i^{(j)}$ derivation.

Since an initial disparity guess is given at every level j, only a small discrete number n_d of disparity correction hypotheses need to be swept. Typically, $n_\mathrm{d} = 3$. This is possible, as at every level j, the correction displacement is small, while in the full resolution and at large baselines, the number of candidates to match is prohibitively large, in the order of hundreds. For our image quality, we did not manage to reduce n_d to 1, the theoretical minimum, without loosing too much depth quality. When employing higher-quality sensors or more advanced descriptors this might become possible, resulting in a substantial speed-up.

In particular, for pixel $\mathbf{x}$ in view i, all disparities $-n_\mathrm{d}$ to n_d are swept, and a per-view cost f_i of disparity d at location $\mathbf{x}$ is minimized:

$$\bar{D}_i^{(j)}(\mathbf{x}) = \underset{d \in \mathcal{S}_i^{(j)}(\mathbf{x})}{\arg\min}\, f_i^{(j)}(\mathbf{x}, d), \tag{1}$$

where $\mathcal{S}_i^{(j)}(\mathbf{x}) = \{\uparrow D_i^{(j-1)}(\mathbf{x}) - n_\mathrm{d}, \ldots, \uparrow D_i^{(j-1)}(\mathbf{x}) + n_\mathrm{d}\}$ is the *sweep range* set of pixel $\mathbf{x}$ in view i at level j. Sweeping is done *relative* to the disparity $\uparrow D_i^{(j-1)}$, starting at $-n_\mathrm{d}$ and ranging to $+n_\mathrm{d}$.

The cost f_i comprises the sum of all view-pair costs g_{i,i_2} between the view i and its four immediate neighbors N_i in the horizontal and vertical directions. It combines matching one image to all others, and consequently, there is one cost per position per view, even if it combines matching N_i neighboring view pairs.

$$f_i^{(j)}(\mathbf{x}, d) = \frac{1}{|N_i|} \sum_{i_2 \in N_i} g_{i,i_2}^{(j)}(\mathbf{x}, d). \tag{2}$$

The view-pair cost g_{i,i_2} compares the descriptor $B_i^{(j)}$ at $\mathbf{x}$ in view i to the corresponding descriptor $B_{i_2}^{(j)}$ in the neighboring view i_2 at a location, offset by the disparity d in the direction vector $\mathbf{r}_{i,i_2}$; more formally,

$$g_{i,i_2}^{(j)}(\mathbf{x}, d) = B_i^{(j)}(\mathbf{x}) \ominus B_{i_2}^{(j)}(\mathbf{x} + d \cdot \mathbf{r}_{i,i_2}). \tag{3}$$

The ability to restrict the sweep to a compact neighborhood is a key result that allows our algorithm to perform at interactive rates.

4.3. Confidence

Besides storing the disparity that is locally optimal, we also compute the confidence $\bar{C}_i^{(j)}$ of pixel $\mathbf{x}$ in view i at level j as:

$$\bar{C}_i^{(j)}(\mathbf{x}) = 1 - \frac{1}{1 + \sigma_m h_i^{(j)}(\mathbf{x})}, \tag{4}$$

where

$$h_i^{(j)}(\mathbf{x}) = \Big|\underbrace{\Big(\frac{1}{|\mathcal{S}_i^{(j)}(\mathbf{x})|} \sum_{d \in \mathcal{S}_i^{(j)}(\mathbf{x})} f_i^{(j)}(\mathbf{x}, d)\Big)}_{\text{Average cost}} - \underbrace{\big(f_i^{(j)}(\mathbf{x}, \bar{D}_i^{(j)}(\mathbf{x}))\big)}_{\text{Best cost}}\Big|^{\frac{1}{2}}.$$

Again, $\mathcal{S}_i^{(j)}(\mathbf{x})$ is the sweep range of pixel $\mathbf{x}$ in view i on level j, as defined in Eq. 1. The difference of the average cost and the cost around the optimal disparity is used as a measure of confidence: If one disparity was much better than its alternatives, it is confident. If it is similar to other alternatives, it will be down-weighted further down in the consolidation and up-sampling steps to follow. The square root function in the nominator has been added to relax the confidence measure penalization as soon as the differences between the best and average costs can be observed. The strength of such confidence measure is additionally controlled by σ_m, whose typical value used for all results is 10.

Discussion While the above is often correlated with the true confidence, the confidence itself can be in error. In low-intensity areas for example, the cost landscape becomes random and can result in false confidence in the wrong solution. Identifying this condition is a typical challenge to all matching-based approaches that make use of confidence which is to be resolved in future work.

Our notion of confidence jointly accounts for occlusions, specularity and inability-to-match at a specific resolution. Previously, they were modeled explicitly [LLW*08]. As seen from typical results, our results show some robustness to not only occlusions, but also specularities as well as the ability to recover from inability-to-match at coarse resolutions, which is critical to the speed we achieve.

4.4. Consolidation

The matching procedure above was done independently for all views. In a diffuse world without occlusions, disparities would need to *agree* between different views: When re-projecting using disparity d from view i to view i_2 the disparity magnitude should match the disparity d_2 when going from i_2 to i. Similar constraints exist between all views $i_2 \in \mathcal{L}$. In the presence of capture noise, occlusion or specularities, no good estimations are possible or disparity is not even defined properly.

Thanks to our multi-resolution procedure, dealing with unreliable data can be surprisingly simple: We will just define disparity that was unreliable or contradicting other views to have a low confidence. As the confidence will be used in the next step to provide up-sampling, unreliable data will not contribute to the next step of estimation.

In practice, we will consolidate the new confidence $C_i^{(j)}(\mathbf{x})$ for view i and pixel $\mathbf{x}$ at level j by summing up per-view confidences for all views $i_2 \in \mathcal{L}$ that are weighted by their respective inverse re-projection errors. The reprojection error is computed by moving the current location $\mathbf{x}$ from the current view i along the direction vector $\mathbf{r}_{i,i_2}$ by the disparity $\bar{D}_i^{(j)}(\mathbf{x})$ and testing if the disparity found there maps back to the start position, but only if that other point was reliable. This way the per-view confidence will propagate across other views in the camera array, and the consolidated confidence $C_i^{(j)}(\mathbf{x})$ will be reduced when its value is different then the corresponding disparity values, which have been found per-view confident.

$$C_i^{(j)}(\mathbf{x}) = \sum_{i_2 \in \mathcal{L}} \hat{C}_{i,i_2}^{(j)}(\mathbf{x}, \mathbf{x}') \tag{5}$$

$$\hat{C}_{i,i_2}^{(j)}(\mathbf{x}, \mathbf{x}') = \bar{C}_{i_2}^{(j)}(\mathbf{x}') / (1 + \sigma_r \underbrace{\left| \bar{D}_i^{(j)}(\mathbf{x}) - \bar{D}_{i_2}^{(j)}(\mathbf{x}') \right|}_{\text{Reprojection}}), \tag{6}$$

where $\mathbf{x}' = \mathbf{x} + \mathbf{r}_{i,i_2} \cdot \bar{D}_i^{(j)}(\mathbf{x})$ i. e., the position of $\mathbf{x}$ from view i in view i_2. The re-projection weight σ_r controls how much an error in reprojection implies a decrease in confidence; a typical value for it used in all results is 10. Similarly, the consolidated disparity map can be derived:

$$D_i^{(j)}(\mathbf{x}) = \sum_{i_2 \in \mathcal{L}} \left[\bar{D}_{i_2}^{(j)}(\mathbf{x}') \hat{C}_{i,i_2}^{(j)}(\mathbf{x}, \mathbf{x}') / C_i^{(j)}(\mathbf{x}) \right]. \tag{7}$$

Effectively, the reprojection error is weighted by the confidence of the disparity in the other view $\bar{C}_{i_2}^{(j)}(\mathbf{x}')$ to not conclude failure from disparity $\bar{D}_{i_2}^{(j)}(\mathbf{x}')$ that is already known to be wrong, and should not contribute in the disparity estimation that will be conveyed to the next level $j+1$.

Performing multi-view matching with reliable confidence is hard to achieve in real-time. Others have used recursive matching for multi-view correspondence [RZMK12], but this does not allow for fine-grained parallelization. Since we compute disparity and confidence independently for every pixel for every view in the camera array, we can take advantage of data processing on GPU, where every thread does not need to wait for others in producing outputs.

4.5. Up-sampling

After matching and consolidating level j, up-sampling is used to produce the initial value for level $j+1$. This is done as convex combination of spatially close pixels, using the consolidated confidence, spatial proximity and the color of the next-higher level as a guide. The consolidated confidence makes sure, that unreliable disparity values contribute less and are not propagated to the next level. Note that in the consolidation step the influence of unreliable disparity propagation between neighboring views was suppressed (Eq. 7), while in the upsampling step spatial pixel neighborhood within the same view is considered (Eq. 8). The guide by the color ensures that sharp edges in appearance remain sharp edges in the estimated disparity, similar as in joint bilateral filtering [KCLU07]. Using the appearance edges of the next level assures no blurring occurs at edges. We are inspired here by previous work that has identified the importance of preserving luminance edges when reconstructing depth [KCLU07, MZK10].

The upsampled disparity $\uparrow D_i^{(j)}(\mathbf{x})$ (refer to Fig. 2), which is used as an initial guess at the next level $j+1$, is computed as a weighted average of the spatially neighboring disparities:

$$\uparrow D_i^{(j)}(\mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{y} \in N_i(\mathbf{x})} w(\mathbf{x}, \mathbf{y}) D_i^{(j)}(\mathbf{y}), \tag{8}$$

where $N_i(\mathbf{x})$ is the neighborhood of pixel $\mathbf{x}$ in view i, typically, a 5×5 stencil, and Z is the sum over all kernel weights $w(\mathbf{x}, \mathbf{y})$ for all pixels $\mathbf{y} \in N_i(\mathbf{x})$. The kernel has a high value, where a pixel $\mathbf{x}$ should mix with a pixel $\mathbf{y}$, i. e., pixels that are confident, proximal and have a similar appearance, which is a product of three other kernels, as in

$$w(\mathbf{x}, \mathbf{y}) = \underbrace{C_i^{(j)}(\mathbf{y})}_{\text{Confidence}} \underbrace{w_s(\mathbf{x}, \mathbf{y})}_{\text{Proximity}} \underbrace{w_a(\mathbf{x}, \mathbf{y})}_{\text{Appearance}} \tag{9}$$

where

$$w_s(\mathbf{x}, \mathbf{y}) = \frac{1}{0.1 + (\sigma_s ||\mathbf{x} - \mathbf{y}||_2^2)}$$

is the spatial weight, and

$$w_a(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + (\sigma_a \left|\left| L^{(j+1)}(\mathbf{x}) - L^{(j)}(\mathbf{y}) \right|\right|_2^2)}$$

is the appearance weight. The constant σ_s is the spatial bandwidth (typically, $\sigma_s = 0.1$) and σ_a is the appearance bandwidth (typically, $\sigma_a = 30$). CIELAB differences are used as distances between colors in L.

As disparity is in units of pixels, it is scaled by a factor of two in a practical implementation when increasing the level, where the resolution is doubled every step.

4.6. GPU implementation

All the above steps are designed to be executed in parallel. Pyramid construction is sequential over levels and parallel over views and pixels. Feature computation is parallel over views, levels and pixels. Matching, consolidation and up-sampling happen in consecution and sequential over levels but can be parallel over views and pixels. Our implementation uses OpenGL pixel shaders and reads and writes multi-view images to array textures.

5. Results

In this section, qualitative and quantitative results of our approach are shown. We start by describing our capture setups (Sec. 5.1) and captured data sets (Sec. 5.2). We will show the quality of light-field enabled effects (Sec. 5.3), before comparing to other depth-from-light field approaches (Sec. 5.4). The section concludes with possible novel applications of our system (Sec. 5.5).

5.1. Capture

In contrast to classical video capture, capturing light field video requires not only a plurality of cameras: Cameras also need to be calibrated and synchronized and the much higher data rate needs to be handled. For this work we had two different systems available that will next be described briefly.

The first camera system is shown in Fig. 2, left. It consists of nine industrial video cameras (Basler acA2000-50gc, 2046×1086 px) arranged on a 3×3 grid, equipped with 12 mm lenses and connected to a PC via Ethernet. The PC controls the cameras and stores the data allowing to capture 1080 p video at 25 Hz.

Our second system consists of up to 16 video cameras equipped with local storage and able to capture video at 30 Hz (GoPro Hero3+ Black Edition) arranged in a 4×4 or 3×5 grid. In order to increase angular resolution and minimize lens distortion the default lenses have been replaced resulting in an opening angle of about 70°.

In both cases, cameras within one array have identical focal length and run on identical settings. The baseline between two cameras is variable and amounts from 6 to 10 cm depending on the scene setup. Cameras within an array are hardware-synchronized. Images captured have a spatial resolution of 1920×1080 pixels. De-bayering and calibration [XMMT15] are assumed to be part of the capture.

5.2. Video light field dataset

Besides dense light fields captured with a gantry we evaluate our method on two new data sets generated with the capture setups described in Sec. 5.1. In contrast to existing data sets [WG14] our data sets are not static but show live action at video frame rates. The scenes show visually rich appearance: static objects and dynamic human characters at different depths, combined with detailed objects having complex occlusions (e. g., plants) as well as specularity and transparency. The camera setup and position was typically selected such that the total amount of parallax in each scene amounts to 3-4 %. This allows for wide baselines that have significant changes in perspective and is common in stereoscopic productions. Currently the data set includes the following scenes, available at `resources.mpi-inf.mpg.de/LightFieldVideo/`:

1. BAR (Fig. 4 and the supplemental video): It features a human character, transparent objects like bottles and glasses and plants with large occlusions and fine details.
2. BEER GARDEN (Fig. 1 and the supplemental video): Two young women sitting opposite to each other on a table. The scene features highly structured, partially repetitive patterns, and only little transparent or semi-transparent objects.
3. CHESS PLAYERS (Fig. 3): Two persons playing chess. A natural scene with slow movements.
4. TABLETOP SOCCER (Fig. 3): Two people are playing on a tabletop soccer. The scene features highly specular objects in combination with fast movements causing strong motion blur.
5. CGI-STUDIO (Fig. 3): A character standing in front of a green screen. This setup is comparable to a typical anchorman scene in a virtual news studio.

Figure 3: *Sample images from* CHESS PLAYERS, CGI-STUDIO *and* TABLETOP SOCCER.

5.3. Light field-enabled effects

The multi-view depth maps produced by our algorithm in Sec. 4 enable several standard visual effects such as virtual camera movements or synthetic aperture renderings [ZEM*15]. Typically the visual quality is enhanced in our approach as most occlusions in one view can be resolved in another one, avoiding missing information when resolving the novel view L. Fig. 4 shows such novel views examples. Synthetic aperture rendering can be used to control the depth-of-field in the scene. A typical example of such a refocus rendering is shown in Fig. 1. View-dependent shading as sampled in the original views can also be supported, as no single color is stored per position, but every view can store a different color information, which can then be blended into the novel view L.

5.4. Comparison to other approaches

Quality Here, we compare to several previous approaches to derive depth from a light field. These approaches typically compute a single depth map, while we compute multiple depth maps. For our data set we do not have ground truth depth maps, and in general it is difficult to acquire reliable depth maps for multi-camera systems [WG14]. In fact, as observed in [MMS*09] a quantitative evaluation of depth map quality might be a poor predictor of the actual quality of images derived using such maps. For all these reasons, to quantitatively evaluate our method we compare predicted novel views to ground-truth reference images.

The Stanford light field archive [Sta08] provides several dense light fields captured with a gantry built from Lego bricks. Those data sets consist of 17×17 views. We can sub-sample this data set by skipping rows and columns regularly obtaining a variety of different camera setups. From this subsampled setup depth maps with wider baseline can be computed. As the position of intermediate but unused views is known one can predict a novel view at this point and compare it to the ground-truth image at this position.

In our experimental setup we sub-sampled the 17×17 views to obtain a 5×5 setup skipping three views in horizontal as well as in vertical direction. With our view synthesis algorithm we computed a novel view in the 7^{th} row and column of the original dense light field. This algorithm is similar to the one presented

Figure 4: *Novel views generated from a fixed time frame. The left images shows the leftmost camera, the central image the top-center while the right image is at the rightmost location. This quality was previously only achieved at much higher computational cost.*

Table 1: *SSIM image similarity (higher is better) for different scenes and different approaches. We achieve quality similar to other approaches at a fraction of the computational time as seen in Table 2.*

		TRUCK		BRACELET		CHEST	
Views		1	5	1	5	1	5
SGM	[Hir08]	0.91	0.95	0.93	0.95	0.94	0.96
CVF	[RHB*11]	0.94	0.96	0.88	0.96	0.93	0.96
Wang	[WER16]	0.93	0.95	0.93	0.98	0.94	0.96
Ours		0.94	0.95	0.94	0.98	0.94	0.95

Table 2: *Timings of our algorithm measured on NVidia GTX 670 for input images of resolution* 960×540.

Camera array size	3×3	5×3
Matching	50 ms	70 ms
Confidence	31 ms	60 ms
Consolidation	32 ms	90 ms
Upsampling	34 ms	57 ms
Total	147 ms	277 ms

by Zilly et al. [ZRM*14]. We basically extended their approach to allow renderings also in vertical direction.

We compare our results to a very recent method reported from Wang et al. [WER16] for dense light fields as well as to two algorithms known for highly efficient stereo matching: Semi-global Matching (SGM) [Hir08] and Cost-Volume-Filtering (CVF) [RHB*11]. In order to get depth maps for every view from the stereo matching algorithms we run these algorithms on horizontal adjacent stereo pairs. In case of a 3×3 setup this results in six independent stereo pairs. Vertical stereo pairs are ignored as this would significantly decrease efficiency and would also require an additional merging procedure. Table 1 shows an SSIM score between a known view and a re-synthesized version of this same view using our depth reconstruction. We compare using both one and five depth maps. SSIM explicitly accounts for image contrast and structure distortions, and gives a more reliable judgment of actually perceived distortions by the human observer [WB06] than numerical evaluation of depth would do. The SSIM score increases with increasing number of views and saturates for five views. Our proposed method performs best on the BRACELET data set and has only slightly lower quality on TRUCK and TREASURE CHEST data set.

Fig. 5 shows synthesized views for TRUCK, BRACELET, and TREASURE CHEST. Small details have been exposed and magnified to see differences between the novel view and the ground truth reference. Noticeable changes are visible on fine details. None of the shown results visually over- or under-performs which is in good coherency with the numeric results.

In addition to the obtained view synthesis results Fig. 6 shows false color sample disparity maps for each of the selected methods. Blue areas denote high distance, green areas denote medium distance while red areas are closely to the camera. Fig. 7 shows sample disparity maps for two of our light field video datasets (Sec. 5.2). In this case we limit evaluation to our method, SGM and CVF. In contrast to the Stanford dataset, these dataset have a much higher baseline and we did not manage to get acceptable results for Wang's method.

Performance We measured the performance of subsequent stages in our multi-view depth reconstruction algorithm on an NVidia GeForce GTX 670 graphics card. The timings can be seen in Table 2. We provide also the performance information for [WER16, Hir08, RHB*11] as reported by the respective authors. To process a single stereo image pair the SGM algorithm takes around 119 ms for the image size of 640×480 or around 447 ms for the resolution of 1282×1110. The timings for this method were computed on a comparable graphics card: NVidia GTX 580. The CVF algorithm tested on an NVidia GTX 480, which has less power than two other graphics cards, was able to achieve the performance ranging from 19 to 98 ms for the image resolution from 384×288 to 450×375 pixels, and the range of disparities from 16 to 60 pixels. The output of the method is also a single disparity map. The Wang algorithm was not tested on GPU, but the authors report that it takes 3 minutes to compute a single depth map for Lytro Illum Image on an Intel i7 machine with 8 GB of RAM.

5.5. Applications

We propose two novel applications enabled by our system: A light field view finder supporting capture of a scene using a light field rig with interactive feedback as illustrated in Fig. 8. Our proposed algorithm potentially also enables applications like light-field based tele-presence system that can transmit immersive visual information. Such data can be inspected in real-time on novel devices such as

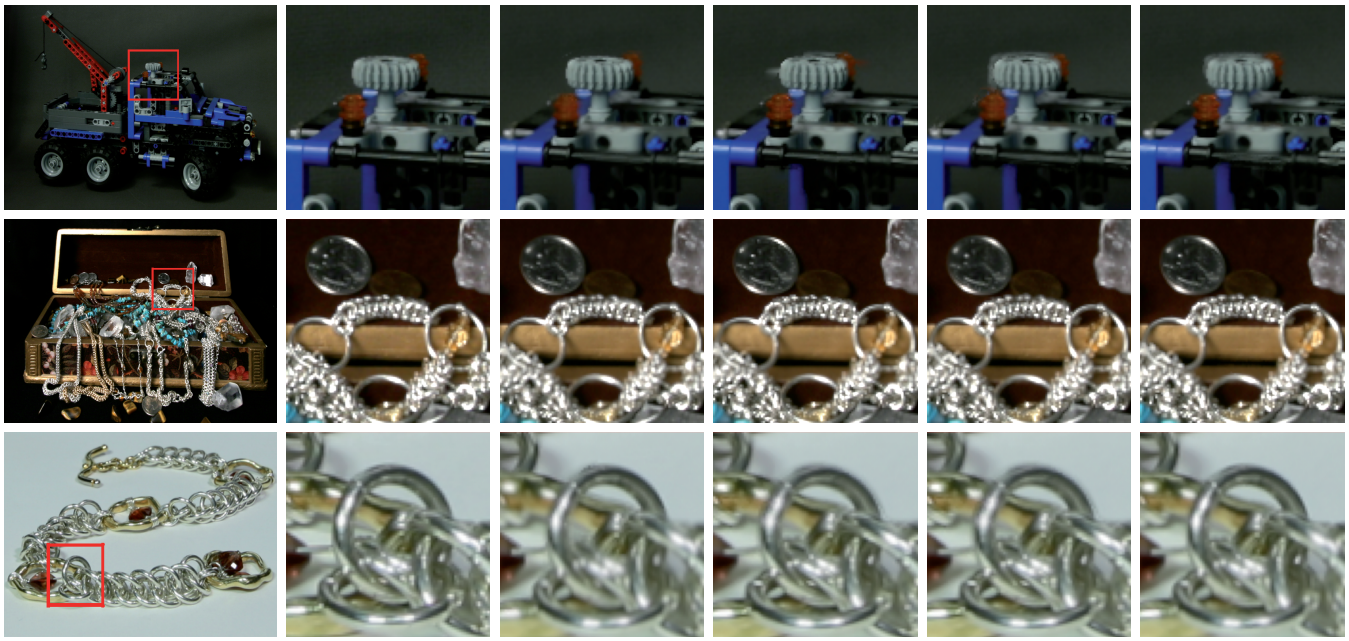

Figure 5: *Qualitative evaluation: We compare to state-of-the art approaches. The first and second column shows the reference image and selected magnified detail. The next four images show view synthesis results with underlying depth maps obtained using SGM, CVF, Wang and our proposed method. Though complexity of our proposed method is low, the visual quality is comparable to competing approaches.*

a head-mounted display (HMD) or might be used to enable eye-contact during tele-conferences.

6. Conclusion

In this work, we presented a novel multi-view depth reconstruction method for video light fields. Our approach achieves quality similar to previous work, but at a fraction of the computational cost. While many of the concepts we employ (multi-resolution, confidence, inter-view constraints) have been used before in correspondence methods, our system is first to show a practical GPU implementation with interactive performance combined with a systematic evaluation on a data set of light field videos we make available.

Our depth maps are simultaneously derived for all views in order to ensure inter-view consistency. Moreover, preservation of sharp depth discontinuities enables spatially consistent view rendering. Core of our approach is a multi-view matching procedure that turns the redundancy of light fields into an advantage by filtering unreliable matches in angular and spatial domains. This way, potential errors in the multi-resolution correspondence search are mostly corrected, and thus not propagated into higher resolutions, which makes our depth reconstruction both fast and reliable. The method maps well to modern GPU architectures and achieves interactive performance. Overall, the performance of our method is comparable to highly efficient stereo pair matching methods, but at the same time we are able to generate multiple, consistent depth maps, while competing algorithms derive a single stereo map.

Our approach is limited to opaque surfaces. While we can tolerate a moderate amount of specularities and sensor noise, larger mirrors, transparent surfaces or low-fidelity sensors are currently not handled well. Since those are relevant in practical light field capture, we would like to generalize to such scenes in future work.

Key to our approach is accounting for uncertainty due to occlusion or difficulties to match at lower resolution at edges. This finally leads to a high-resolution confidence value for every pixel. The novel-view synthesis we currently employ ignores this information. Accounting for this information and only using reliable depths is an interesting avenue of further research specifically enabled by our approach.

Finally, while our method imposes constraints across multiple views and our computational effectiveness allows dealing with light field video, we do not impose any inter-view/time constraints. As the scenes are not limited to rigid camera motions, imposing temporal coherence poses a challenging avenue of future work ready to explore given the data set we have provided.

Acknowledgment: Supported by the Fraunhofer and Max Planck cooperation program within the German pact for research and innovation (PFI).

References

[BA83] BURT P. J., ADELSON E. H.: The Laplacian pyramid as a compact image code. *IEEE Trans. Comm. 31*, 4 (1983), 532–40. 3

[BOD*16] BRRUCKNER A., OBERDOERSTER A., DUNKEL J., REIMANN A., WIPPERMANN F.: Ultra-slim 2D- and depth-imaging camera modules for mobile imaging. In *Proc. SPIE MOEMS and Miniaturized Systems* (2016), vol. 9760. 2

[BWS05] BRUHN A., WEICKERT J., SCHNÖRR C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J Computer Vision 61*, 3 (2005), 211–31. 3

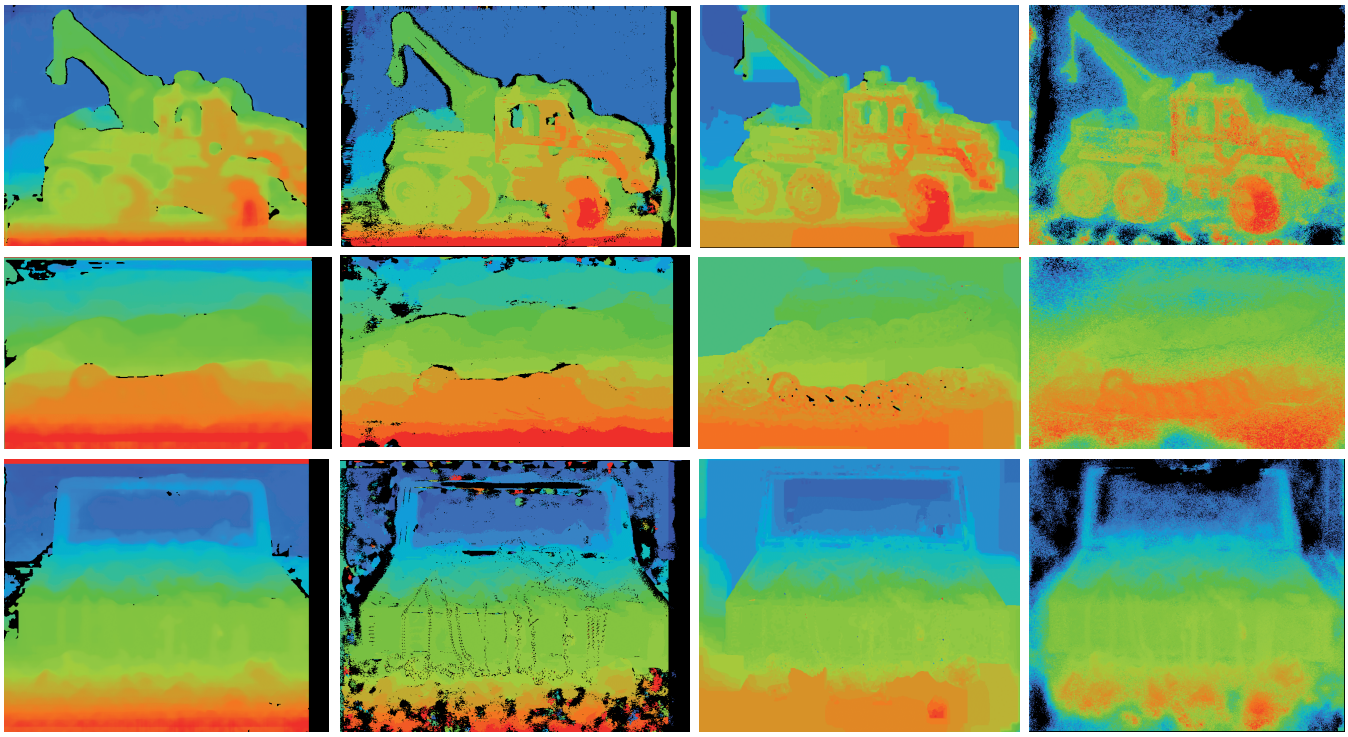

Figure 6: *False color disparity map for the central view. From left to right we show result for SGM, CVF, Wang and our proposed method.*

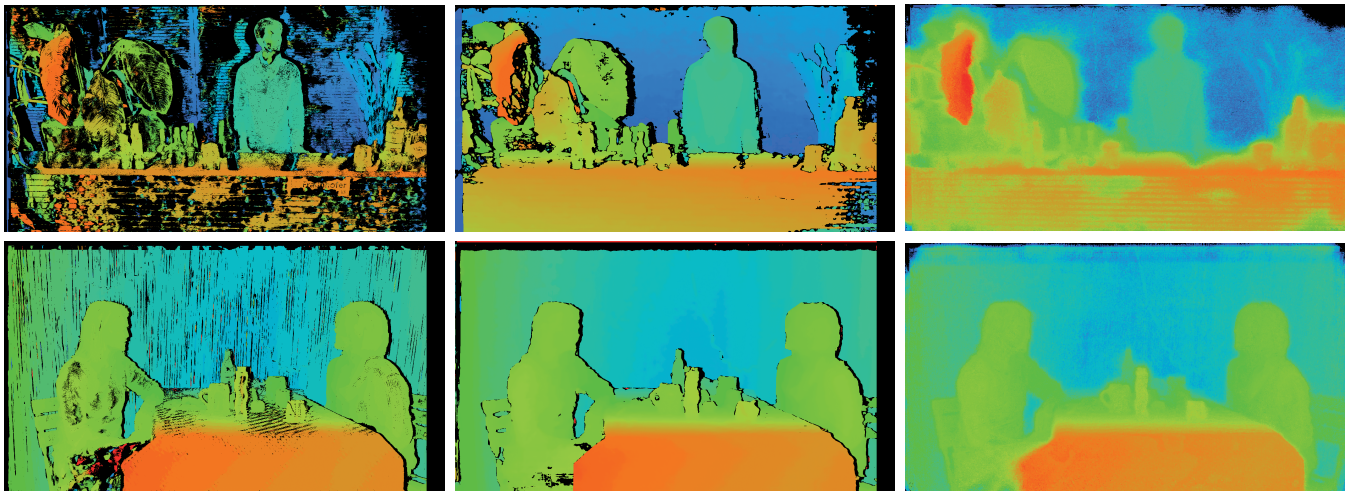

Figure 7: *Sample false color disparity maps for* BAR *and* BEER GARDEN*: From left to right: CVF, SGM and our proposed method.*

[CLY*14] CHEN C., LIN H., YU Z., KANG S., YU J.: Light field stereo matching using bilateral statistics of surface cameras. In *CVPR* (2014), pp. 1518–1525. 2

[DU11] DENKER K., UMLAUF G.: Accurate real-time multi-camera stereo-matching on the GPU for 3D reconstruction. *J WSCG 19*, 1-3 (2011), 9–16. 2

[FMT*06] FUJII T., MORI K., TAKEDA K., MASE K., TANIMOTO M., SUENAGA Y.: Multipoint measuring system for video and sound - 100-camera and microphone system. In *IEEE Multimedia and Expo* (2006), pp. 437–440. 2

[FWAS10] FELDMANN I., WAIZENEGGER W., ATZPADIN N., SCHREER O.: Real-time depth estimation for immersive 3d videoconferencing. In *Proc. 3DTV-CON* (2010), pp. 1–4. 2

[Gab08] GABRIEL L.: La photographie intégrale. *Comptes-Rendus, Académie des Sciences 146* (1908), 446–551. 1

[HI16] HAMZAH R. A., IBRAHIM H.: Literature survey on stereo vision disparity map algorithms. *J Sensors 2016*, 8742920 (2016). 2

[Hir08] HIRSCHMULLER H.: Stereo processing by semiglobal matching and mutual information. *IEEE PAMI 30*, 2 (2008), 328–341. 2, 3, 7

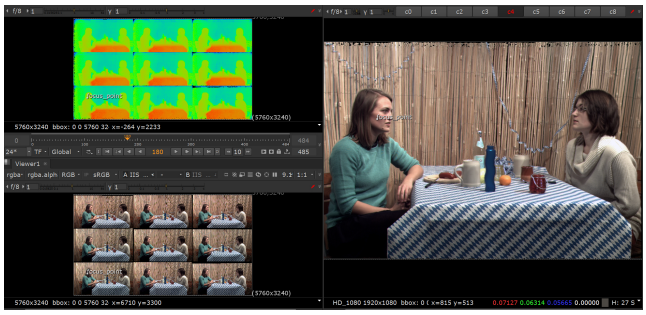

Figure 8: *The high efficiency of our proposed method in combination with real-time view synthesis methods potentially allows to build a system that can i.e. be used as a viewfinder known from classical cameras. Depth of scene elements, possible viewpoints as well as depth-of-field could be inspected and adjusted on set.*

[HS81] HORN B. K., SCHUNCK B. G.: Determining optical flow. In *1981 Technical symposium east* (1981), pp. 319–331. 3

[KCLU07] KOPF J., COHEN M. F., LISCHINSKI D., UYTTENDAELE M.: Joint bilateral upsampling. *ACM Trans. Graph. (Proc. of SIGGRAPH 2007) 26*, 3 (2007), to appear. 5

[KZP*13] KIM C., ZIMMER H., PRITCH Y., SORKINE-HORNUNG A., GROSS M. H.: Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans.Graph.(Proc.SIGGRAPH) 32*, 4 (2013), 73–1. 2

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. SIGGRAPH* (1996), pp. 31–42. 1, 2

[LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Int. J CAI* (1981), vol. 81, pp. 674–679. 2, 3

[LLW*08] LIANG C.-K., LIN T.-H., WONG B.-Y., LIU C., CHEN H. H.: Programmable aperture photography: multiplexed light field acquisition. In *ACM Trans. Graph. (Proc. SIGGRAPH)* (2008), vol. 27, ACM, p. 55. 2, 4

[LYT11] LIU C., YUEN J., TORRALBA A.: Sift flow:dense correspondence across scenes and its applications. *IEEE PAMI* (2011), 978–94. 3

[MGBP11] MARTON F., GOBBETTI E., BETTIO F., PINTUS R.: A real-time coarse-to-fine multiview capture system for all-in-focus rendering on a light-field display. In *Proc. 3DTV-CON* (2011), pp. 1–4. 2

[MMS*09] MERKLE P., MORVAN Y., SMOLIC A., FARIN D., MÜLLER K., DE WITH P. H. N., WIEGAND T.: The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communcation 24*, 1-2 (2009). 6

[MRK*13] MANAKOV A., RESTREPO J. F., KLEHM O., HEGEDÜS R., EISEMANN E., SEIDEL H.-P., IHRKE I.: A reconfigurable camera add-on for high dynamic range, multi-spectral, polarization, and light-field imaging. *ACM Trans. Graph. (Proc. SIGGRAPH 2013) 32*, 4 (2013). 2

[MZK10] MÜLLER M., ZILLY F., KAUFF P.: Adaptive cross-trilateral depth map filtering. In *Proc. 3DTV* (2010), pp. 1–4. 2, 3, 5

[NLB*05] NG R., LEVOY M., BRÉDIF M., DUVAL G., HOROWITZ M., HANRAHAN P.: Light field photography with a hand-held plenoptic camera. *Comp. Science TR 2*, 11 (2005), 1–11. 2

[PW12] PERWASS C., WIETZKE L.: Single lens 3D-camera with extended depth-of-field. *Proc. SPIE 8291* (2012). 2

[RHB*11] RHEMANN C., HOSNI A., BLEYER M., ROTHER C., GELAUTZ M.: Fast cost-volume filtering for visual correspondence and beyond. In *CVPR* (2011). 2, 7

[RZMK12] RIECHERT C., ZILLY F., MÜLLER M., KAUFF P.: Real-time disparity estimation using line-wise hybrid recursive matching and cross-bilateral median upsampling. In *Proc.ICPR* (2012), pp. 3168–71. 5

[SCD*06] SEITZ S., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR* (2006), pp. 519– 528. 2

[Sta08] STANFORD UNIVERSITY: Stanford lightfield archive, 2008. 6

[THMR13] TAO M., HADAP S., MALIK J., RAMAMOORTHI R.: Depth from combining defocus and correspondence using light-field cameras. In *ICCV* (2013), pp. 673–680. 2

[TNM09] TUNG T., NOBUHARA S., MATSUYAMA T.: Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *ICCV* (2009), pp. 1709–1716. 2

[VLD*13] VENKATARAMAN K., LELESCU D., DUPARRÉ J., MCMAHON A., MOLINA G., CHATTERJEE P., MULLIS R., NAYAR S.: Picam: An ultra-thin high performance monolithic camera array. *ACM Trans. Graph. 32*, 6 (2013), 166:1–166:13. 2

[WB06] WANG Z., BOVIK A. C.: *Modern Image Quality Assessment*. Morgan & Claypool Publishers, 2006. 7

[WER16] WANG T.-C., EFROS A., RAMAMOORTHI R.: Depth estimation with occlusion modeling using light-field cameras. *IEEE PAMI* (2016). 2, 7

[WG14] WANNER S., GOLDLUECKE B.: Variational light field analysis for disparity estimation and super-resolution. *IEEE PAMI 36*, 3 (2014), 606–19. 2, 6

[WJV*05] WILBURN B., JOSHI N., VAISH V., TALVALA E.-V., ANTUNEZ E., BARTH A., ADAMS A., HOROWITZ M., LEVOY M.: High performance imaging using large camera arrays. *ACM Trans. Graph 24*, 3 (2005), 765–76. 2

[WK04] WOETZEL J., KOCH R.: Real-time multi-stereo depth estimation on gpu with approximative discontinuity handling. In *1st European Conference on Visual Media Production* (2004), vol. 3, Citeseer. 2

[XMMT15] XU Y., MAENO K., MAGAHARA H., TANIGUCHI R.-I.: Camera array calibration for light field acquisition. *Frontiers of Computer Science 9*, 5 (2015), 691–702. 6

[YGL*13] YU Z., GUO X., LIN H., LUMSDAINE A., YU J.: Line assisted light field triangulation and stereo matching. In *ICCV* (2013), pp. 2792–2799. 2

[YWA10] YANG Q., WANG L., AHUJA N.: A constant-space belief propagation algorithm for stereo matching. In *CVPR* (2010), pp. 1458–65. 3

[YWB03] YANG R., WELCH G., BISHOP G.: Real-time consensus-based scene reconstruction using commodity graphics hardware. *Comp. Graph. Forum 22*, 2 (2003), 207–216. 2

[ZEM*15] ZIEGLER M., ENGELHARDT A., MÜLLER S., KEINERT J., ZILLY F., FOESSEL S., SCHMID K.: Multi-camera system for depth based visual effects and compositing. In *Proc. CVMP* (2015), ACM, p. 3. 2, 6

[ZKU*04] ZITNICK C. L., KANG S. B., UYTTENDAELE M., WINDER S., SZELISKI R.: High-quality video view interpolation using a layered representation. In *Proc. ACM SIGGRAPH* (2004), pp. 600–608. 2

[ZLD15] ZHANG Z., LIU Y., DAI Q.: Light field from micro-baseline image pair. In *CVPR* (2015), pp. 3800–3809. 2

[ZRM*14] ZILLY F., RIECHERT C., MÜLLER M., EISERT P., SIKORA T., KAUFF P.: Real-time generation of multi-view video plus depth content using mixed narrow and wide baseline. *J Vis. Comm. and Image Rep. 25*, 4 (2014). 7

[ZRMK12] ZILLY F., RIECHERT C., MÜLLER M., KAUFF P.: Generation of multi-view video plus depth content using mixed narrow and wide baseline setup. In *Proc. 3DTV-CON* (2012), pp. 1–4. 2

[ZW94] ZABIH R., WOODFILL J.: Non-parametric local transforms for computing visual correspondence. In *ECCV* (1994), pp. 151–8. 3