Video Depth-From-Defocus

Hyeongwoo Kim [1] Christian Richardt [1, 2, 3] Christian Theobalt [1]

[1] Max Planck Institute for Informatics [2] Intel Visual Computing Institute [3] University of Bath

Abstract

Many compelling video post-processing effects, in particular aesthetic focus editing and refocusing effects, are feasible if per-frame depth information is available. Existing computational methods to capture RGB and depth either purposefully modify the optics (coded aperture, light-field imaging), or employ active RGB-D cameras. Since these methods are less practical for users with normal cameras, we present an algorithm to capture all-in-focus RGB-D video of dynamic scenes with an unmodified commodity video camera. Our algorithm turns the often unwanted defocus blur into a valuable signal. The input to our method is a video in which the focus plane is continuously moving back and forth during capture, and thus defocus blur is provoked and strongly visible. This can be achieved by manually turning the focus ring of the lens during recording. The core algorithmic ingredient is a new video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video, and the focus distance for each frame. We extensively evaluate our approach, and show that it enables compelling video post-processing effects, such as different types of refocusing.

1. Introduction

Many cinematographically important video effects that normally require specific camera control during capture, such as focus and depth-of-field control, can be computationally achieved in post-processing if depth and focus distance are available for each video frame. This post-capture control is on high demand by video professionals and amateurs alike.

In order to capture video and depth in general, and specifically to enable post-capture focus effects, several methods were proposed that use specialized camera hardware, such as active depth cameras [31], light-field cameras [26], or cameras with coded optics [18].

We follow a different path and propose one of the first end-to-end approaches for depth estimation from – and focus manipulation in – videos captured with an unmodified commodity consumer camera. Our approach turns the often unwanted *defocus blur*, which can be controlled by lens aperture, into a valuable signal. In theory, smaller apertures produce sharp images for scenes covering a large depth range. When using a larger aperture, only scene points close to a certain focus distance project to a single point on the image sensor and appear in focus (see Section 3). Scene points at other distances are imaged as a *circle of confusion* [30]. The limited region of sharp focus around the focus distance is known as *depth of field*; outside of it the increasing *defocus blur* is an important depth cue [21].

Unfortunately, depth of field in normal videos cannot be changed after recording, unless a method like ours is applied. Our approach (Figure 1) takes as input a video in which focus sweeps across the scene, e.g. by manual change of lens focus. This means temporally changing defocus blur is purposefully provoked (see Section 3). Each frame thus has a different focus setting, and no frame is entirely in focus. Our core algorithmic contribution uses this provoked blur and performs space-time coherent *depth*, *all-in-focus color*, and *focus distance* estimation at each frame. We first segment the input video into multiple *focus ramps*, where the focus plane sweeps across the scene in one direction. The first stage of our approach (Section 4.1) constructs a *focus stack video* for each of them. Focus stack videos consist of a focus stack at each video frame, by aligning adjacent video frames to the current frame using a new defocus-preserving warping technique. At each frame, the focus stack video comprises multiple images with a range of approximately known focus distances (see Supplemental Section 1), which are used to estimate a depth map in the second stage (Section 4.2) using depth-from-defocus with filtering-based regularization. The third stage (Section 4.3) performs spatially varying deconvolution to remove the defocus blur and produce all-in-focus images. And the fourth stage of our approach (Section 4.4) further minimizes the remaining error by refining the focus distances for each frame, which significantly improves the depth maps and all-in-focus images in the next iteration of our algorithm. Our end-to-end method requires no sophisticated calibration process for focus distances, which allows it to work robustly in practical scenarios.

In a nutshell, the main algorithmic contributions of our paper are: (1) a new hierarchical alignment scheme between

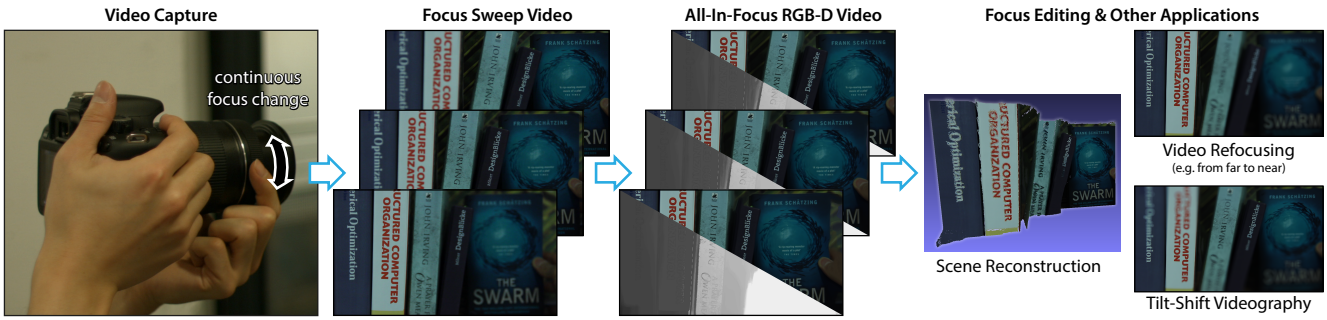

Figure 1. We capture focus sweep videos by continuously moving the focus plane across a scene, and then estimate per-frame *all-in-focus RGB-D videos*. This enables a wide range of video editing applications, in particular video refocusing. Please refer to the paper's electronic version and supplemental video to more clearly see the defocus effects we show in our paper.

video frames of different focus settings and dynamic scene content; (2) a new approach to estimate per-frame depth maps and deblurred all-in-focus color images in a space-time coherent way; (3) a new image-guided algorithm for focus distance initialization; (4) and a new optimization method for refining focus distances. We extensively validate our method, compare against related work, and show high-quality refocusing, dolly-zoom and tilt-shift editing results on videos captured with different cameras.

2. Related Work

RGB-D Video Acquisition Many existing approaches for RGB-D video capture use special hardware, such as projectors or time-of-flight depth cameras [31], or use multiple cameras at different viewpoints. Moreno-Noguer et al. [23] use defocus blur and attenuation of the a projected dot pattern to estimate depth. Coded aperture optics enable single-shot RGB-D image estimation [1, 5, 18, 19, 20], but require more elaborate hardware modification. Stereo correspondence [3] or multi-view stereo approaches [43] require multiple views, for instance by exploiting shaky camera motions. Shroff et al. [36] shift the camera's sensor along the optical axis to change the focus within a video. They align consecutive video frames using optical flow to form a focus stack, and then apply depth from defocus to the stack. Unlike all mentioned approaches, ours works with a single unmodified video camera without custom hardware. There are also single-view methods based on non-rigid structure-from-motion [32], which interpret clear motion cues (in particular out-of-plane) under strong scene priors, and learning-based depth estimation methods [7, 10, 33, 37].

Depth from Focus/Defocus Focus stacking combines multiple differently focused images into a single *all-in-focus* (or *extended depth of field*) image [29]. Depth-from-(de)focus techniques exploit (de)focus cues within a focus stack to compute depth. Focus stacking is popular in macro photography, where the large lens magnification results in a very small depth of field. By sweeping the focus plane across a scene or an object, each part of it will be sharpest in one photo, and these sharp regions are then combined into the all-in-focus image. Depth from focus additionally determines the depth of a pixel from the focus setting that produced the sharpest focus [8, 24]; however, this requires a densely sampled focus stack and a highly textured scene. Depth from defocus, on the other hand, exploits the varying degree of defocus blur of a scene point for computing depth from just a few defocused images [28, 38]. The all-in-focus image is then recovered by deconvolving the input images with the spatially-varying point spread function of the defocus blur. Obviously, techniques relying on focus stacks only work well for scenes without camera or scene motion.

Suwajanakorn et al. [40] proposed a hybrid approach that stitches an all-in-focus image using motion-compensated focus stacking, and then optimizes for the depth map using depth-from-defocus. Their approach is completely automatic and even estimates camera parameters (up to an inherent affine ambiguity) from the input images. However, their approach is limited to reconstructing a single frame from a focus stack, and cannot easily be extended to videos, as this requires stitching per-frame all-in-focus images. Our approach is tailored for videos, not just single images.

Refocusing Images and Videos Defocus blur is an important perceptual depth cue [9, 21], thus refocusing is a powerful, appearance-altering effect. However, just like RGB-D video capture, all approaches suitable for refocusing videos require some sort of custom hardware, such as special lenses [22, 26], coded apertures [18] or active lighting [23]. Special image refocusing methods are difficult to extend to videos as they rely on multiple captures from the same view, for example, for depth from (de)focus [40].

A single image captured with a light-field camera with special optics [26, 41] can be virtually refocused [13, 25], but the light-field image often has a drastically reduced spatial resolution. Miau et al. [22] use a deformable lens to quickly and repeatedly sweep the focus plane across the scene while recording video at 120 Hz. They refocus video by selecting appropriately focused frames. Some approaches exploit residual micro-blur that is hard to avoid even in a

photograph set to be in focus everywhere. It can be used for coarse depth estimation [34], or removed entirely [35].

Defocus deblurring is also related to motion deblurring [6, 12, 42]. Their characteristics differ; motion blur is, for instance, mostly depth-independent.

3. Preliminaries

Both our lens model and aspects of video recording influence algorithm design.

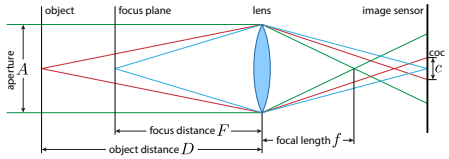

Figure 2. Thin-lens model and circle of confusion.

Defocus Model We assume a standard video camera with a finite aperture lens that produces a limited depth of field. According to the thin-lens model (Figure 2), the amount of defocus blur is quantified by the diameter c of the circle of confusion [30]:

$$c = \frac{Af\,|D-F|}{D(F-f)} = \frac{f^2|D-F|}{ND(F-f)}, \tag{1}$$

where $A = f/N$ is the diameter of the aperture, f is the focal length of the lens, N is the f-number of the aperture, D is the depth of a scene point and F is the focus distance. We assume fixed aperture and focal length, and a focus distance F changing over time. Therefore, the defocus blur of a 3D point only depends on its depth D and the focus distance F, which we express as the point-spread function $K(D,F)$ corresponding to the circle of confusion in Equation 1. We model the color of a defocused image V at a pixel $\mathbf{x}$ using

$$V(\mathbf{x}) = (K(D(\mathbf{x}),F) * I)(\mathbf{x}), \tag{2}$$

where I denotes the all-in-focus image. Note that K is spatially varying, because each pixel $\mathbf{x}$ may have a different depth $D(\mathbf{x})$. For brevity, we hence omit the pixel index $\mathbf{x}$.

Video Recording We record our focus sweep input video simply by manually adjusting the focus on the lens, roughly following a sinusoidal focus distance curve – leading to several focus ramps (see Figure 3). The exact focus distance for each video frame is unknown; reading it out from the camera is difficult in practice and may require low level modification of the firmware. We use the Magic Lantern[1] software for Canon EOS digital DSLR cameras to record timestamped lens information at about 4 Hz. In practice, focus distance values are not measured for most frames, timestamps may not be exactly aligned with the time of frame capture, and the recorded focus distances are quantized and not fully accurate. There is also natural variation in the focus distance curves as people cannot exactly reproduce a curve. Therefore, our algorithm uses the sparsely recorded lens information only as a guide and explicitly optimizes for the dense correct focus distances at every frame (Section 4.4).

[1] http://www.magiclantern.fm

4. All-In-Focus RGB-D Video Recovery

Given a video $\mathcal{V}$ with frames $\{V_t\}_{t\in T}$ containing one or more focus sweeps, we formulate our algorithm as a joint optimization framework that seeks the optimal depth maps D_t, all-in-focus images I_t, and focus distances F_t for all frames $t \in T$. Let us assume that $W_{s\to t}(\cdot)$ is a warping function that aligns an image at time s with time t, while preserving the original defocus blur (explained in Section 4.1). Then, we can construct a focus stack at each frame t by warping all input video frames to it using $\{W_{s\to t}(V_s)\}_{s\in T}$ (in practice, we only warp keyframes, as explained later). We seek the optimal depth map D_t and all-in-focus image I_t, and focus distances $\{F_t\}_{t\in T}$ which best reproduce the focus stack at frame t with the defocus model in Equation 2. D_t, I_t and F_t, for $t \in T$, are the unknowns we solve for. The core ingredient of our joint optimization is a data term that penalizes the defocus model error of the focus stack at all frames:

$$E_{\text{data}} = \sum_{t\in T}\sum_{s\in T} w_{t,s}\,\|K(D_t,F_s) * I_t - W_{s\to t}(V_s)\|^2\,. \tag{3}$$

We introduce the weighting term $w_{t,s}$ to give lower weights to pairs of frames that are further apart, and which hence need warping over longer temporal distances. In our implementation, we use a Gaussian function $w_{t,s} = \exp(-|t-s|^2/2\sigma_{\text{w}}^2)$ with σ_{w} set to 85 percent of the length of each focus ramp.

Simultaneously estimating depth, deblurring the input video and optimizing focus distances from purposefully defocused and temporally misaligned images is highly challenging; many invariance assumptions used by correspondence finding approaches break down in this case. To solve this joint optimization problem efficiently, we decompose it into four subproblems, or *stages*, that we solve iteratively: defocus-preserving alignment (Section 4.1), depth estimation (4.2), defocus deblurring (4.3), and focus refinement (4.4). Initialization and implementation details can be found in Supplemental Section 1. Each subproblem requires solving for a subset of the unknowns by minimizing a cost functional like Equation 3, with additional regularization terms explained later. We adopt a multi-scale, coarse-to-fine approach. At each resolution level, we perform three iterations of the four stages, each of which is solved for the entire length of the input video. The multi-resolution

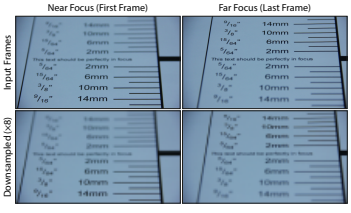

Figure 4. Downsampling (bottom) effectively reduces defocus difference, helping correspondence finding.

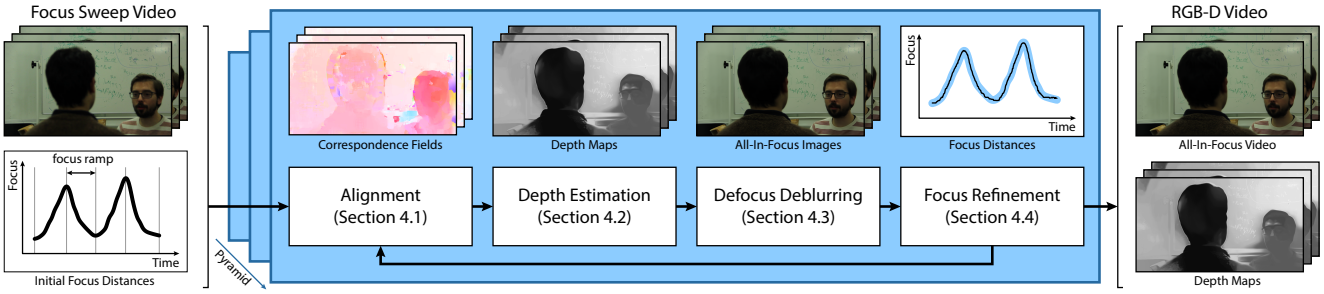

Figure 3. Overview of our approach. For each video frame, we first align neighboring frames to it to construct a focus stack. We then estimate spatially and temporally consistent depth maps from the focus stacks, and compute all-in-focus images using non-blind deconvolution. Finally, we refine the focus distances for all frames. We perform these steps in a coarse-to-fine manner and iterate until convergence.

approach improves convergence, but more importantly, any focus difference between two video frames is reduced when the images are downsampled in the pyramid (see Figure 4). This insight enables us to compute reliable initial correspondence fields with less influence from different defocus blurs. Once all parameters are estimated at a coarse level, the higher level of the pyramid uses them as initialization for its iterations.

4.1. Patch-Based Defocus-Preserving Alignment

Here, we construct a focus stack for each frame of the input video using patch-based, defocus-preserving image alignment. The result are the warping functions $W_{s\to t}$ for pairs (s, t) of frames, while all other unknowns (I, D and F) remain constant. Two frames in the focus sweep video, V_s and V_t, generally differ in defocus blur and maybe scene or camera motion. The main challenge of the defocus-preserving alignment is to compute a reliable correspondence field that is robust to both complex motion and defocus changes between the frames.

Using standard correspondence techniques, such as optical flow, to directly warp the input video frame V_s to V_t is prone to failure, because the different defocus blurs in the two images are not modeled by standard matching costs. Optical flow will try to explain differences in defocus blur using flow displacements, which produces erroneous correspondences (see Figure 5).

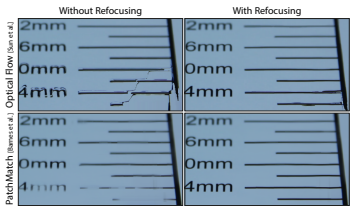

Figure 5. Comparison of focus stack alignment from near to far focus (Figure 4). Without refocusing to match blur levels, both optical flow and PatchMatch fail. With refocusing, PatchMatch improves on flow (used by Shroff et al. [36]).

The solution is to compensate any focus differences before computing correspondences [36]. We therefore refocus the target frame V_t to match the focus distance F_s of the source frame s using the refocusing operator $R(V_t, F_s) = K(D_t, F_s) * I_t$. In the first iteration at the coarsest resolution level, the refocusing operator returns the input frame V_t unchanged, as the downsampling in the pyramid has already removed most of the defocus blur. In subsequent iterations, the refocusing operator uses the current estimates of frame t's depth map D_t and all-in-focus image I_t to perform the refocusing. The embedding of this focus difference compensation and alignment process into the overarching coarse-to-fine scheme enables reliable focus stack alignment even for large scene motions and notable defocus blur differences.

We use PatchMatch [2] to robustly compute the warping function $W_{s\to t}$ between the source frame V_s and the refocused target frame $R(V_t, F_s)$. PatchMatch handles complex motions and is fairly robust to the remaining focus differences, while traditional optical flow techniques tend to fail in such cases. PatchMatch correspondences are not always geometrically correct (see right), as they exploit visually similar patches from other regions of the image which have similar defocus blur (note the purple and yellow regions on the left, which indicate vertical motion along the edge of the books). In our case, this is an advantage, as it improves the warping quality while preserving defocus blur. At the coarsest level of the pyramid, we initialize the PatchMatch search using optical flow [39]; at this level, focus-induced appearance differences are minimal. We also constrain the size of the search window to find the best matches around the initial correspondences. This encourages the estimated correspondence field to be more spatially consistent. Since the warping is computed by refocusing the target frame, the defocus blur in the source frame is preserved, which is crucial for constructing valid focus stacks from a dynamic focus sweep video.

Correspondences for the BOOK dataset (from the first frame to the last frame).

We apply the estimated defocus-preserving warping operators $W_{s\to t}$ to create a *focus stack video* with per-frame focus stacks, as shown in Figure 6. However, we do not warp all frames to all others, to prevent artifacts introduced by

aligning temporally distant videos frames in which the scene may have changed drastically. Instead, we first segment the input video into contiguous focus ramps (see Figure 3), $T_i \subset T$ for $i \in R$, which contain only temporally close frames. For each input video frame t, we then create a focus stack by warping the other frames in its ramp to it using our defocus-preserving alignment. This reduces the computational complexity of alignment from $\mathcal{O}(|T|^2)$ for all-pairs warping, to $\mathcal{O}(|T|^2/|R|)$ for all-pairs warping within each focus ramp.

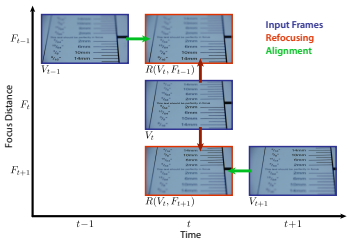

Figure 6. Defocus-preserving alignment. We refocus the input frame V_t (center) to match the focus distances of neighboring frames $F_{t\pm1}$ (red arrows), and compute correspondences (green arrows) between neighboring frames, $V_{t\pm1}$, and the corresponding refocused image $R(V_t, F_{t\pm1})$.

4.2. Filtering-Based Depth Estimation

The second stage of our approach estimates spatially and temporally consistent depth maps D_t for all focus stacks, while keeping all other variables constant. In this case, our data term E_{data} from Equation 3 measures how well the estimated depth maps fit to the defocus observations in all focus stacks, which is equivalent to depth from defocus [28] applied per video frame. This step requires the pixel-wise alignment across each focus stack, computed in the previous stage, to measure the fitting error. Since this error is individually penalized at each pixel, it can lead to spatial inconsistencies in the depth map. To avoid this issue, we introduce a long-range linear Potts model. In contrast to the pairwise Potts model which compares depth values only between immediately adjacent pixels, our version performs long-range comparisons which benefit globally consistent depth estimation, yet prevents erroneous smoothing of actual features in the depth map:

$$E_{\text{smoothness}}^{\text{spatial}} = \sum_{t\in T} \sum_{\mathbf{x}} \sum_{\mathbf{y}\neq\mathbf{x}} \min(\alpha(\mathbf{x},\mathbf{y})\,|D_t(\mathbf{x}) - D_t(\mathbf{y})|, \tau_d), \quad (4)$$

where τ_d is the truncation value of the depth difference. We use the bilateral weight α between two pixels $\mathbf{x}$ and $\mathbf{y}$, $\alpha(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_s^2} - \frac{\|I(\mathbf{x})-I(\mathbf{y})\|^2}{2\sigma_r^2}\right)$, to encourage consistent depth estimation between nearby pixels with similar colors, where σ_s and σ_r denote the standard deviation for the spatial and range terms, respectively. We use $\sigma_s = 0.075\times$ the image width, and $\sigma_r = 0.05$.

In addition, we want the depth maps to be temporally coherent across all frames. We minimize the discrepancy between depth maps using

$$E_{\text{smoothness}}^{\text{temporal}} = \sum_{t\in T} \sum_{s\in T\setminus t} \|D_t - W_{s\to t}(D_s)\|^2, \quad (5)$$

which encourages temporal consistency over extended depth map sequences. The total cost function for depth map estimation is defined by combining the data term from Equation 3 and the smoothness terms from Equations 4 and 5:

$$\arg\min_D E_{\text{data}} + \lambda_{\text{ss}} E_{\text{smoothness}}^{\text{spatial}} + \lambda_{\text{ts}} E_{\text{smoothness}}^{\text{temporal}}, \quad (6)$$

where $\lambda_{\text{ss}} = 1$ and $\lambda_{\text{ts}} = 0.2$ are balancing weights.

The direct minimization of Equation 6 requires global optimization with respect to all depth images, which is computationally expensive. Instead, we solve an efficient approximation of the global optimization problem. We pose the minimization task as a labeling problem, and first estimate spatially consistent depth maps for all frames by applying a variant of cost-volume filtering [11], and then refine the per-frame depth maps to enforce temporal consistency [16].

We start by computing per-frame depth maps D_t in three steps. (1) We evaluate the data term (Equation 3) for n predefined, uniformly spaced depth layers, and store the error for each pixel $\mathbf{x}$ and depth label d in the cost volume $C(\mathbf{x}, d)$. As in previous depth-from-defocus techniques [28], we perform this evaluation in the frequency domain, where convolution can be efficiently computed using element-wise multiplication. (2) We apply fast joint-bilateral filtering [27] on each depth-cost slice, to minimize the long-range spatial smoothness term in Equation 4. For this, we use the all-in-focus image I_t as the guide image in computing the bilateral weight α. As in the previous section, we take the estimated all-in-focus image I_t from the previous iteration, and assume $I_t = V_t$ in the first iteration of the coarsest resolution level. (3) We select the spatially optimal depth for each pixel using $D_t(\mathbf{x}) = \arg\min_d C(\mathbf{x}, d)$.

After computing depth maps independently from each focus stack, we apply temporal smoothing to make the depth maps consistent over time. We use a keyframe-based approach with a sliding temporal window. For each frame t, we align the depth maps of the previous and following two keyframes to the current depth map D_t using our warping operator $W_{s\to t}$ computed on the all-in-focus images. The updated depth map D_t is the Gaussian-weighted mean of aligned depth maps. The used keyframes are not restricted to be from the same focus ramp as the frame t; this enforces temporal consistency also

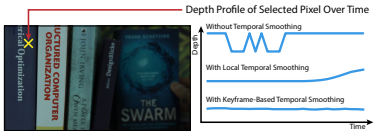

Figure 7. Our keyframe-based smoothing produces more consistent depth than local smoothing of adjacent frames. (The pixel should have constant depth in this scene.)

across focus ramp boundaries. In Figure 7, we show that our approach successfully produces temporally coherent depth maps, compared to the unfiltered input depth maps and also the simpler local filtering of adjacent frames, as some bias remains due to the short temporal range of the filtering.

4.3. Defocus Deblurring

Now that we have computed the depth maps D_t, we estimate all-in-focus images I_t using non-blind deconvolution with the spatially varying point-spread function (PSF) corresponding to the depth-dependent circle of confusion (Equation 1). While the disc shape of the PSF is a good approximation of the actual shape of the camera aperture, in practice, its sharp boundary causes ringing artifacts in the deconvolution process due to zero-crossings in the frequency domain [18], see Figure 8. We therefore adopt the smoothness term introduced by Zhou et al. [44] to prevent ringing artifacts:

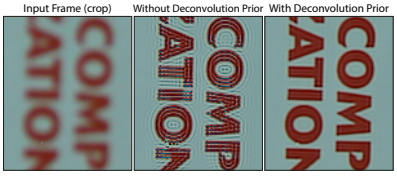

Figure 8. Deconvolution can result in ringing artifacts, which are suppressed by our deconvolution smoothness prior.

$$E_{\text{smoothness}}^{\text{all-in-focus}} = \sum_{t \in T} \|H * I_t\|^2, \tag{7}$$

where H is an image statistics prior.

The smoothness term uses a learning approach to capture natural image statistics. We first take sample images at the same resolution as the input image from a database of natural images, and then apply the Fourier transform to the samples to compute the frequency distribution of image statistics. The final image statistics of the natural image dataset H is obtained by averaging the squared per-frequency modulus of all sample distributions. The smoothness term Equation 7 enforces the all-in-focus image I_t to follow a similar frequency distribution as the learned one in H. The image statistics prior H only needs to be computed once for each video resolution to be processed, and can then be reused for new videos. For the details of the smoothness term, we refer the reader to Zhou et al. [44].

The total cost function of the defocus deblurring is a combination of the data term in Equation 3 and the learned smoothness term in Equation 7:

$$\arg\min_{I} E_{\text{data}} + \lambda_{\text{as}} E_{\text{smoothness}}^{\text{all-in-focus}}, \tag{8}$$

where $\lambda_{\text{as}} = 10^{-3}$ balances the two cost terms. We compute the optimal all-in-focus image I_t by performing Wiener deconvolution independently on a range of n depth layers, each with a fixed, depth-dependent point spread function, and then composite the sub-images to obtain the all-in-focus image I_t.

4.4. Focus Distance Refinement

This step refines the focus distances to reduce the data term E_{data} (Equation 3). As explained in Section 3, we can at best read out temporally sparse focus distance values from the camera, which are moreover subject to inaccuracies. To overcome this difficulty, we refine the focus distances for all frames in the final stage of our approach. By rearranging the terms associated with the focus distance F_t in Equation 3, we define the focus refinement subproblem as

$$\arg\min_{F_t} \sum_{s \in T} w_{s,t} \left\| R(V_s, F_t) - W_{t \to s}(V_t) \right\|^2. \tag{9}$$

We optimize this equation by gradient descent. Since the cost function is highly nonlinear in F_t, we compute the gradient numerically by examining the costs for focus distances $F_t \pm \delta$ with $\delta = 5\,\text{mm}$. In practice, we refocus each source frame V_s to focus distances $F_t \pm \delta$, and compare it to the aligned target frame $W_{t \to s}(V_t)$ (see Supplementary Figure 3). We then set the focus distance F_t to the new minimum.

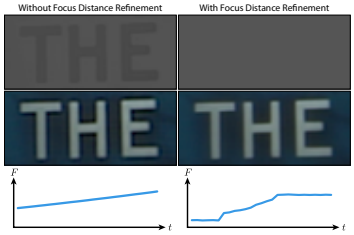

Figure 9. Focus distance refinement improves depth maps (top) by reducing texture copy artifacts, and also removes halos in all-in-focus images (middle). The refined focus distances (bottom) correctly reflect the constant focus at the beginning and end of the video.

We demonstrate the performance of our focus distance refinement in Figure 9. It improves the depth estimation as well as visual quality of the all-in-focus images by suppressing excessive edge contrasts. Because this strategy frees us from requiring artificial patterns or special hardware for the accurate calibration of focus distances, it allows for our flexible and simple acquisition of the focus sweep video.

5. Results and Evaluation

We thoroughly evaluate our proposed video depth-from-defocus approach for reconstructing all-in-focus RGB-D videos. We first show qualitative results on natural, dynamic scenes with non-trivial motion, captured with static and moving video cameras. We then compare our approach against the two closest approaches, by Shroff et al. [36] and Suwajanakorn et al. [40]. We further evaluate the design choices made in our approach with an ablation study on a ground-truth dataset. Lastly, we evaluate our focus refinement optimization in Supplemental Section 3.

We show all-in-focus images and depth map results on a range of datasets in Figure 10, in Supplemental Figure 2 and in our video. Our depth maps capture the gist of each scene, including the main depth layers and their silhouettes, and the depth gradients of slanted planes with sufficient texture. As demonstrated by the results, our approach works for dynamic scenes, and handles a fair degree of occlusions,

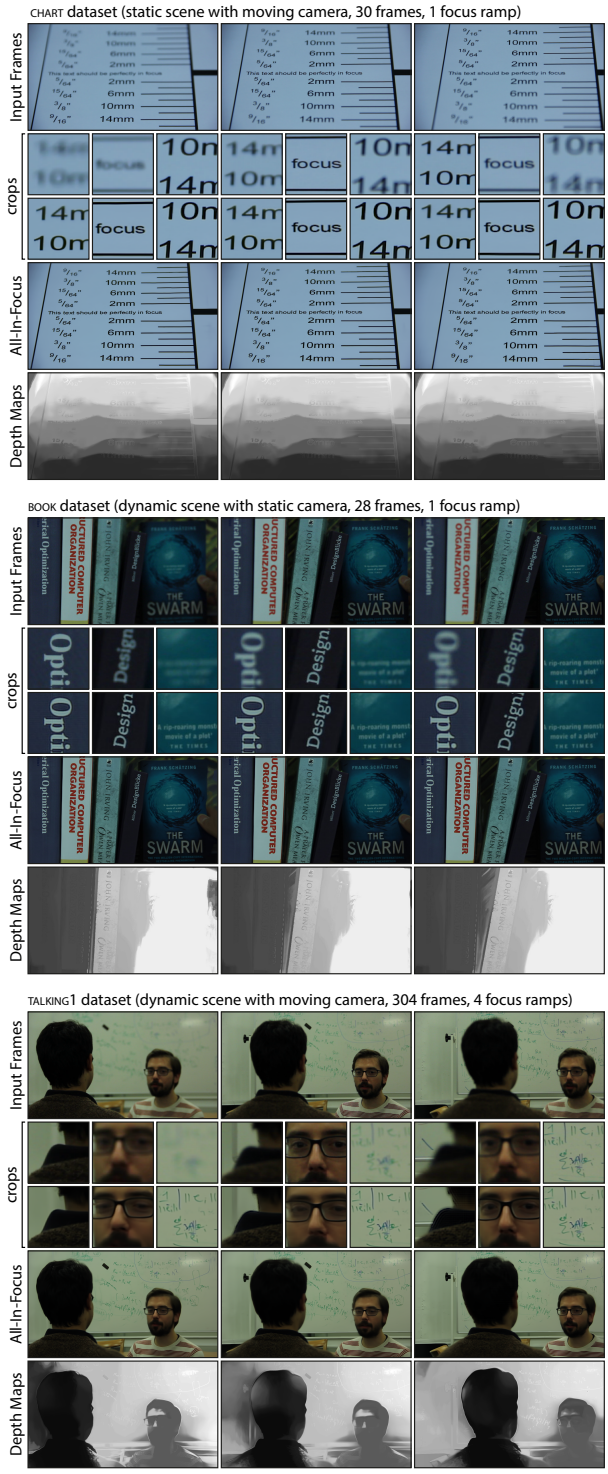

Figure 10. RGB-D video results. We show reconstructed all-in-focus images and depth maps for three focus sweep videos with various combinations of scene and camera motion. The image crops (top: input frame cropped, bottom: all-in-focus images cropped) focus on regions at the near, middle and far end (from left to right) of the scene's depth range. Note that each input frame is in focus in only one of the three crops, while our all-in-focus images are in focus everywhere. Please zoom in to see more details.

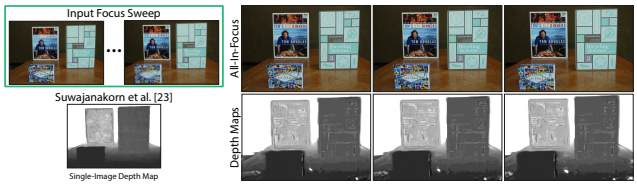

Figure 11. Comparison of our approach to Suwajanakorn et al. [40] on their dynamic dataset. Left: Input focus stack, focused near (left) to far (right), and their estimated depth map for the last frame only. Right: We reconstruct all-in-focus images and depth maps for all frames of this dynamic sequence (also see our video).

dis-occlusions and out-of-plane motions. It also properly reconstructs the depth and all-in-focus appearance of small objects, like the earrings in sequence TALKING2 (Supplemental Figure 2), which is highly challenging. Note that our approach also works if scene and camera are rather static, where approaches requiring notable disparity for depth estimation would fail – even on unblurred footage. Similar to previous depth-from-(de)focus techniques, our approach works best for textured scenes that are captured in a full focus stack. Although our depth maps are not perfect, they are temporally coherent and enable visually plausible video refocusing applications, as shown in Supplemental Section 4.

Comparison to Shroff et al. [36] This work moves a camera's sensor along the optical axis to compute all-in-focus RGB-D videos in an approach similar to ours. However, our approach improves on theirs in several important ways: (1) we use a commodity consumer video camera that does not require any hardware modifications like in their approach, (2) our defocus-preserving alignment finds more reliable correspondences than optical flow, (3) our depth maps are more detailed and temporally coherent, and (4) our all-in-focus images and hence refocusing results improve on theirs. We simulate their focus stack alignment approach by replacing PatchMatch in our implementation with optical flow [39]. Figure 5 shows that PatchMatch achieves visually better alignment results.

Comparison to Suwajanakorn et al. [40] This recent depth-from-focus technique computes a *single* depth map with all-in-focus image from quick focus sweeps of around 30 photos of static scenes with little camera motion. Their approach first reconstructs the all-in-focus image by aligning the input photos and stitching the sharpest regions. This will fail for videos, as dynamic scenes break their alignment strategy of concatenating the optical flows. Any estimated per-frame depth maps are also most likely not temporally coherent. Our approach, on the other hand, computes temporally coherent all-in-focus RGB-D videos of dynamic scenes. Our robust defocus-preserving alignment (Section 4.1) enables us to construct per-frame focus stacks for dynamic scenes (moving scene and camera), and hence to compute per-frame depth maps and all-in-focus images. On top, we implement keyframe-based temporal consistency filtering

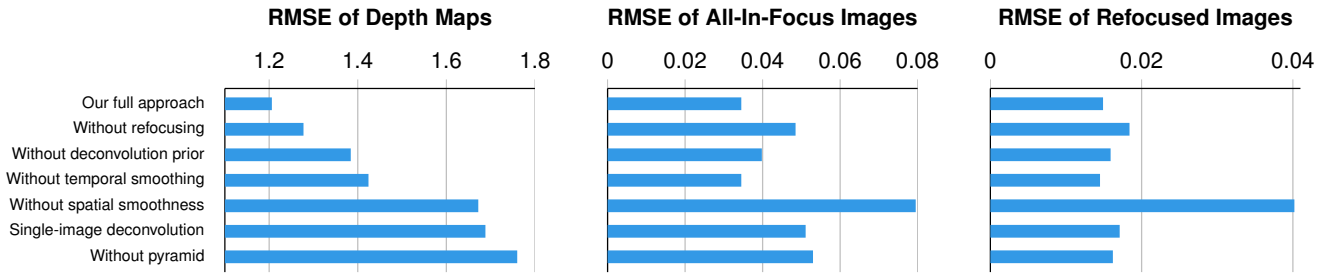

Figure 12. Validation of design choices using an ablation study (lower RMSE is better). Our approach is best overall, but each components is required for achieving best results.

(Section 4.2) to remove any flickering from the depth maps. We visually compare our results to theirs on one of their datasets in Figure 11. We use their provided camera parameters without further focus distance refinement (Section 4.4).

Validation of Design Choices We performed a quantitative ablation study to analyze the influence of the design choices in our algorithm. For this, we synthetically defocus 10 frames from the MPI-Sintel dataset 'alley_1' [4] using two focus ramps, and apply additive Gaussian noise with $\sigma = 3/255$ to simulate camera imaging noise. We then process the resulting video while disabling or replacing individual components of our approach. In Figure 12, we evaluate the accuracy of the estimated depth maps and all-in-focus images using the root-mean-squared error (RMSE) compared to the ground truth. Our full approach produces overall the best results. One can clearly see the importance of each component in our approach, as leaving them out significantly degrades the quality of the estimated depth maps or all-in-focus images, or both. We also evaluate how accurately each alternative explains the input defocus images when refocusing the all-in-focus image using the estimated depth map. Without temporal smoothing, refocused images have lower RMSE than our full approach, but the images lack temporal consistency (which is not measured by RMSE).

6. Discussion and Conclusion

Limitations Our approach relies on aligning all frames of a focus ramp to each other. This works well for focus ramps of up to around 30 frames, but becomes more difficult for longer ramps, as more motion needs to be compensated. This is significantly more difficult than for example the alignment required for HDR video reconstruction [14], which only needs to align three subsequent frames instead of 30–100. While our alignment approach produces good results within one ramp, even a long one, the consistency across long ramps becomes more difficult to enforce. This may lead to popping artifacts in the all-in-focus video.

As in previous depth-from-defocus methods, each aligned focus stack yields a single depth map. However, this limits objects in adjacent video frames to have similar depths, which restricts their motion in depth.

Large occlusions are also problematic as the focus stack alignment degrades in quality when part of the scene is not visible during a focus ramp, for example at the image boundaries. Like static depth-from-defocus methods, we assume the appearance of objects as well as lighting remain constant. Additionally, untextured regions are harder to reconstruct, and may show some temporal flickering, similar to previous depth-from-defocus methods but also passive, image-based depth reconstruction approaches in general.

We employ a simple blur model, which uses a spatially varying convolution with a point-spread function. This may cause blurs across depth boundaries, which can create halos in the depth maps (see discussion in Lee et al. [17]). A potential solution are more sophisticated, multi-layer defocus blur models [15], which are harder to integrate into our optimization. Our depth maps are plausible. They may not entirely match the quality of depth maps from RGB-D cameras or multi-camera systems, but they were recorded with a completely unaltered camera, along with focus distances and all-in-focus frames, and enable video focus post-processing at good quality. Our goal was to explore what is possible with unaltered hardware and what information may lie in typical artifacts, even auto-focus pulls.

Conclusion We presented the first algorithm for space-time coherent depth-from-defocus from video. It reconstructs all-in-focus RGB-D video of dynamic scenes with an unmodified commodity video camera. We open a different view on RGB-D video capture by turning the often unwanted defocus blur artifact into a valuable signal. From an input video with purposefully provoked defocus blur, e.g. by simply turning the lens, we compute space-time-coherent depth maps, deblurred all-in-focus video and per-frame focus distance. Our end-to-end approach relies on several algorithmic contributions, including an alignment scheme robust to strongly varying focus settings, an image-based method for accurate focus distance estimation, and a space-time-coherent depth estimation and deblurring approach. We have extensively evaluated our method and its components, and show that it enables compelling video refocusing effects.

Acknowledgements We thank the authors of the used datasets. Funded by ERC Starting Grant 335545 CapReal.

References

[1] Y. Bando, B.-Y. Chen, and T. Nishita. Extracting depth and matte using a color-filtered aperture. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 27(5):134:1–9, 2008. 2

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3):24, 2009. 4

[3] J. T. Barron, A. Adams, Y. Shih, and C. Hernández. Fast bilateral-space stereo for synthetic defocus. In *CVPR*, 2015. 2

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 8

[5] A. Chakrabarti and T. Zickler. Depth and deblurring from a spectrally-varying depth-of-field. In *ECCV*, pages 648–661, 2012. 2

[6] S. Cho, J. Wang, and S. Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):64:1–9, 2012. 3

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[8] P. Grossmann. Depth from focus. *Pattern Recognition Letters*, 5(1):63–69, 1987. 2

[9] R. T. Held, E. A. Cooper, J. F. O'Brien, and M. S. Banks. Using blur to affect perceived distance and size. *ACM Transactions on Graphics*, 29(2):19:1–16, 2010. 2

[10] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3): 577–584, 2005. 2

[11] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2013. 5

[12] Z. Hu, L. Xu, and M.-H. Yang. Joint depth estimation and camera shake removal from single blurry image. In *CVPR*, pages 2893–2900, 2014. 3

[13] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *SIGGRAPH*, pages 297–306, 2000. 2

[14] N. K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D. B. Goldman, and P. Sen. Patch-based high dynamic range video. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 32(6): 202:1–8, 2013. 8

[15] M. Kraus and M. Strengert. Depth-of-field rendering by pyramidal image processing. *Computer Graphics Forum (Eurographics)*, 26(3):645–654, 2007. 8

[16] M. Lang, O. Wang, T. O. Aydın, A. Smolic, and M. Gross. Practical temporal consistency for image-based graphics applications. *ACM Transactions on Graphics (SIGGRAPH)*, 31 (4):34:1–8, 2012. 5

[17] S. Lee, E. Eisemann, and H.-P. Seidel. Real-time lens blur effects and focus control. *ACM Transactions on Graphics (SIGGRAPH)*, 29(4):65:1–7, 2010. 8

[18] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):70, 2007. 1, 2, 6

[19] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen. Programmable aperture photography: multiplexed light field acquisition. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):55:1–10, 2008. 2

[20] M. Martinello, A. Wajs, S. Quan, H. Lee, C. Lim, T. Woo, W. Lee, S.-S. Kim, and D. Lee. Dual aperture photography: Image and depth from a mobile camera. In *ICCP*, 2015. 2

[21] G. Mather. Image blur as a pictorial depth cue. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1367):169–172, 1996. 1, 2

[22] D. Miau, O. Cossairt, and S. K. Nayar. Focal sweep videography with deformable optics. In *ICCP*, 2013. 2

[23] F. Moreno-Noguer, P. N. Belhumeur, and S. K. Nayar. Active refocusing of images and videos. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):67, 2007. 2

[24] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, 1994. 2

[25] R. Ng. Fourier slice photography. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):735–744, 2005. 2

[26] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. CSTR 2005-02, Stanford University, 2005. 1, 2

[27] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81:24–52, 2009. 5

[28] A. P. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4): 523–531, 1987. 2, 5

[29] S. Pertuz, D. Puig, M. A. Garcia, and A. Fusiello. Generation of all-in-focus images by noise-robust selective fusion of limited depth-of-field images. *IEEE Transactions on Image Processing*, 22(3):1242–1251, 2013. 2

[30] M. Potmesil and I. Chakravarty. Synthetic image generation with a lens and aperture camera model. *ACM Transactions on Graphics*, 1(2):85–108, 1982. 1, 3

[31] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Eurographics)*, 31(2):247–256, 2012. 1, 2

[32] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *ECCV*, pages 583–598, 2014. 2

[33] A. Saxena, M. Sun, and A. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 2

[34] J. Shi, X. Tao, L. Xu, and J. Jia. Break Ames room illusion: Depth from general single images. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 34(6):225:1–11, 2015. 3

[35] J. Shi, L. Xu, and J. Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 3

[36] N. Shroff, A. Veeraraghavan, Y. Taguchi, O. Tuzel, A. Agrawal, and R. Chellappa. Variable focus video: Re-

constructing depth and video for dynamic scenes. In *ICCP*, 2012. 2, 4, 6, 7

[37] S. Srivastava, A. Saxena, C. Theobalt, S. Thrun, and A. Y. Ng. i23 - rapid interactive 3D reconstruction from a single image. In *Vision, Modeling, and Visualization Workshop (VMV)*, 2009. 2

[38] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994. 2

[39] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106 (2):115–137, 2014. 4, 7

[40] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *CVPR*, 2015. 2, 6, 7

[41] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):69, 2007. 2

[42] J. Wulff and M. J. Black. Modeling blurred video with layers. In *ECCV*, pages 236–252, 2014. 3

[43] F. Yu and D. Gallup. 3D reconstruction from accidental motion. In *CVPR*, 2014. 2

[44] C. Zhou, S. Lin, and S. K. Nayar. Coded aperture pairs for depth from defocus and defocus deblurring. *International Journal of Computer Vision*, 93(1):53–72, 2011. 6