# Video Depth-From-Defocus — Supplemental Material

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

## 1. Initialization and Implementation

The input to our video depth-from-defocus approach is a radiometrically linearized video with temporally changing focus distances containing one or more focus ramps, but with otherwise constant camera settings. We assume that we know camera properties such as the focal length, aperture $f$-number, sensor size, as well as temporally sparse readings of the camera's focus distances for some video frames.

**Focus Distance Initialization** Before the start of our algorithm in Section 4 of the main paper, we compute an initial set of temporally dense focus distance values. We use the sparse timestamped focus distance readings from the Magic Lantern firmware as starting point, see Section 3 in the main paper. We then solve for the per-frame focus distances $F$ using an energy minimization with the recorded focus data as data term, and additional smoothness and focus-consistency regularization terms:

$$\arg\min_{F} E^{\text{focus}}_{\text{data}} + \lambda_{\text{fs}} E^{\text{focus}}_{\text{smoothness}} + \lambda_{\text{focus}} E_{\text{focus}} . \quad (1)$$

The recorded focus distances $F^{\text{rec}}_t$ are available only for some frames $t \in T_{\text{rec}}$, so we constrain the unknown focus distances $F_t$ at those frames to lie close to them:

$$E^{\text{focus}}_{\text{data}} = \sum_{t \in T_{\text{rec}}} \left\| F_t - F^{\text{rec}}_t \right\|^2 . \quad (2)$$

As the focus is assumed to change smoothly over time, we enforce this by penalizing the second derivative of the focus distances:

$$E^{\text{focus}}_{\text{smoothness}} = \sum_{t} \left\| F_{t-1} - 2F_t + F_{t+1} \right\|^2 . \quad (3)$$

The focus-consistency term exploits the observation that similar focus distances result in similar depth-of-field and hence similar images, so if video frames appear very similar, then their focus distances should also be similar (see Figure 1):

$$E_{\text{focus}} = \sum_{t} \sum_{s \neq t} s_{t,s} \left\| F_t - F_s \right\|^2 , \quad (4)$$

where $s_{t,s}$ measures the (symmetric) similarity of the input video frames $V_t$ and $V_s$, so that more similar frames enforce consistency constraints more strongly. We compute the similarity using

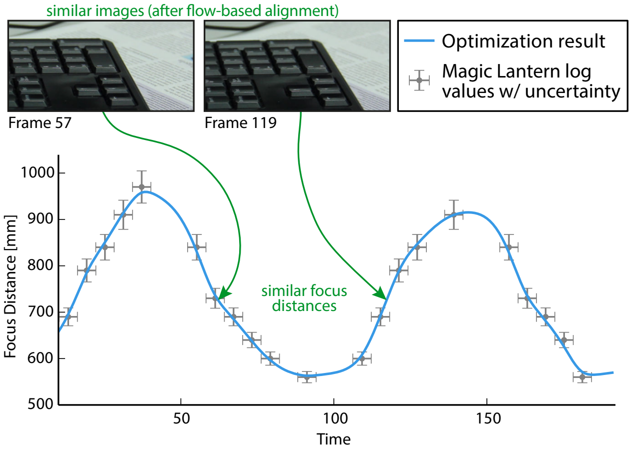

Figure 1. Focus distance initialization yields a smooth initial focus curve from sparse Magic Lantern data. Similar images according to $s_{t,s}$ (Equation 5) enforce consistent focus distances.

$$s_{t,s} = \min(0, 1 - \min(d_{t,s}, d_{s,t}) \,/ \tau_{\text{sim}}) , \quad (5)$$

based on the image dissimilarity $d_{t,s}$ which we compute using the RMSE between input frames $V_t$ and $V_s$ warped to $V_t$ using low-resolution (160×90) optical flow to compensate for camera and scene motion. The similarity threshold $\tau_{\text{sim}}$ determines which pairs of input frames result in consistency constraints and how strongly they are enforced. Typical parameter values are $\lambda_{\text{fs}} = \sqrt{10}$, $\lambda_{\text{focus}} = 0.1$ and $\tau_{\text{sim}} \in [0.01, 0.05]$.

### 1.1. Implementation of Video Depth-From-Defocus Algorithm

To implement our method from Section 4 (main paper), we use a multi-resolution approach with three levels to improve the convergence and visual quality of our results, as image defocus is more similar at coarser image resolutions. At each pyramid level, we perform three iterations of the stages described in Sections 4.1 to 4.4 of the main paper. At the coarsest level, we start the first iteration assuming that the all-in-focus image $I_t$ is the input video frame $V_t$, and also initialize our PatchMatch correspondences using optical flow [6] to provide a good starting point for our alignment computations. Between pyramid levels, we bilinearly upsample the all-in-focus images $I_t$, depth maps $D_t$ and all computed flow fields. We use scale-adjusted patch sizes for PatchMatch, using 25×25 pixels at the finest level and 7×7 at the coarsest.

**Computation Times** Our all-in-focus RGB-D video estimation approach processes 30 video frames at 854×480 resolution in 4 hours on a 30-core 2.8 GHz processor with 256 GB memory. This runtime breaks down as follows, per video frame: 8.6 minutes for defocus-preserving alignment, 19 minutes for depth estimation, 2.4 minutes for defocus deblurring, and 5 seconds for focus distance refinement. Our MATLAB implementation is unoptimized, but parallelized over the input video frames. We believe an optimized, possibly GPU-assisted implementation would yield significant speed-ups.

## 2. Results

We show additional all-in-focus images and depth map results on a range of datasets in Figure 2.

## 3. Evaluation of Focus Distance Refinement

Here, we investigate the contribution of the focus distance refinement (Figure 3 and Section 4.4 in the main paper) to estimating better all-in-focus images and recovering from inaccurate initial focus distances. For this, we process the synthetically refocused 'alley_1' dataset from MPI-Sintel [2] with initial focus distances perturbed by varying degrees of additive Gaussian noise (but without imaging noise), with and without our focus distance refinement, and compare the all-in-focus images and estimated focus distances to the ground truth. Figure 4 shows that our focus distance refinement consistently reduces the errors in estimated focus distances. This in turn leads to better refocusing results for our defocus-preserving alignment (Section 4.1 in the main paper), which produces cleaner all-in-focus images and improves the overall performance of our approach.

## 4. Applications of Video Depth-From-Defocus

**Video Refocusing** Given the estimated all-in-focus images and depth maps, we can now freely refocus the original input video according to the user's wishes by simply rendering the appropriate defocus blur in a post-process. For this, we use the same thin-lens defocus model as in Section 3 of the main paper, and blur each pixel's neighborhood with the blur kernel $K(D(\mathbf{x}), F)$ corresponding to its depth $D(\mathbf{x})$ and the focus distance $F$ of the virtual lens [5]. This approach provides complete freedom, as the camera's aperture, focal length and focus distance can be changed independently

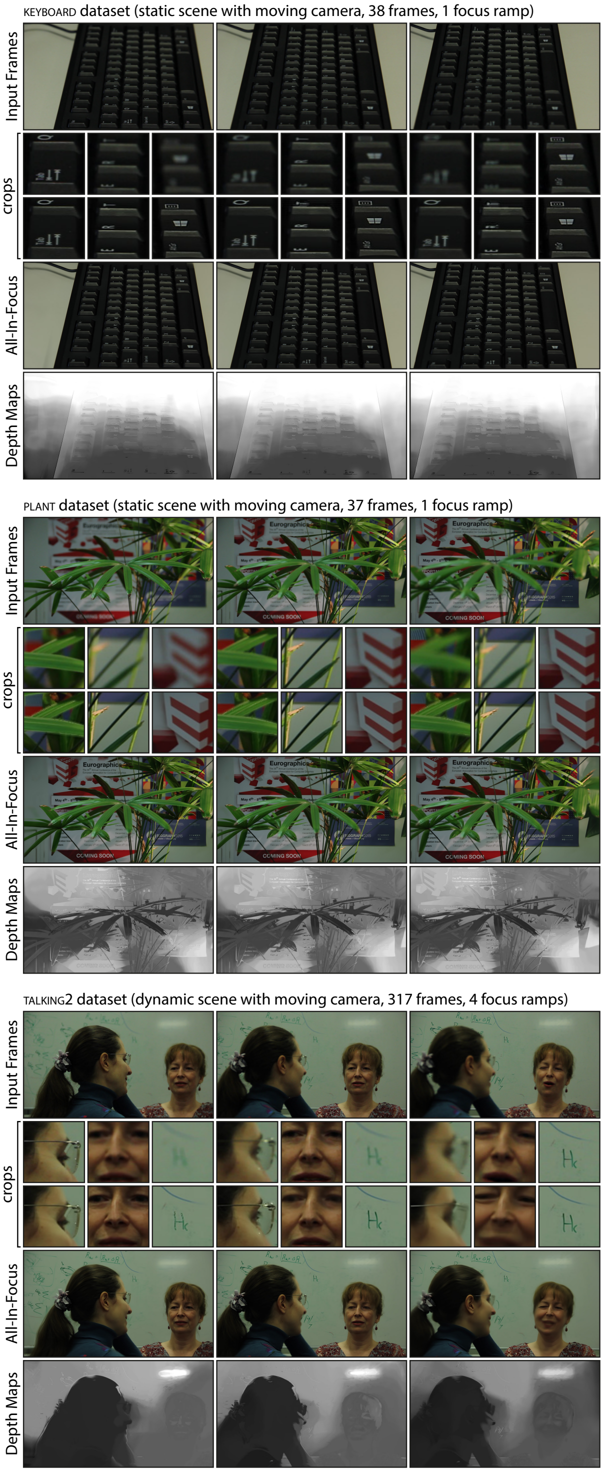
KEYBOARD dataset (static scene with moving camera, 38 frames, 1 focus ramp)
Input Frames
crops
All-In-Focus
Depth Maps
PLANT dataset (static scene with moving camera, 37 frames, 1 focus ramp)
Input Frames
crops
All-In-Focus
Depth Maps
TALKING2 dataset (dynamic scene with moving camera, 317 frames, 4 focus ramps)
Input Frames
crops
All-In-Focus
Depth Maps

Figure 2. 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.

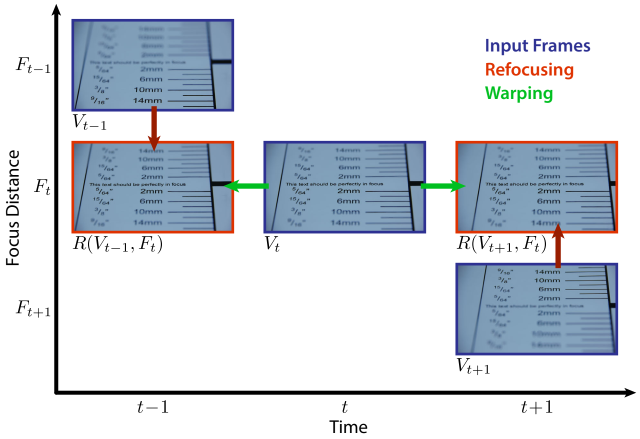

Figure 3. We refine focus distances by refocusing input frames to each frame $t$, and minimizing the difference to the frame $V_t$ warped to each of the refocused input images (Equation 9 in main paper).

and arbitrarily. The user can for example change the aperture, while keeping the original focus settings, to reduce or magnify the defocus blur (see Figure 5), similar to Bae and Durand [1], but for videos. The focus can also be fixed on an object of interest or follow it through the video using a 'focus pull', or the focus can be interactively controlled by the user using a 'focus-follow' function that keeps the region under the user's mouse pointer in focus. The reconstructed focus settings can also be smoothed to correct auto-focus failures and produce a more professional-looking result.

**Tilt-Shift Videography** The *tilt-shift effect* is created by tilting the camera's lens relative to its image plane which results in a slanted focus plane with a wedge-shaped depth of field that produces the iconic miniature look [3]. (The purpose of lens *shift* is to correct for perspective distortions like converging parallel lines; however, it does not affect the focus plane or depth of field.) While the lens in most *view cameras* can be tilted and shifted freely thanks to the flexible bellows between lens and film, most lenses in modern cameras are fixed to be parallel to the image sensor, which prevents this effect. There are some special-purpose tilt-shift lenses for modern cameras, e.g. from Canon, Nikon or Lensbaby, which can be expensive, but the tilt-shift look is baked into the recorded footage and cannot be modified after capture. We show virtual tilt-shift videography in Figure 5 and our video by refocusing with a tilted virtual lens [4]. This provides ultimate flexibility as the desired look can be modified and tweaked interactively.

**Dolly Zoom** Depth maps also enable other applications such as limited novel-view synthesis. When combined with the video refocusing presented earlier, this provides the two ingredients required for a dolly zoom (or 'Hitchcock Zoom'): a camera on a virtual dolly that moves towards or away from the scene, and a carefully controlled virtual camera zoom that keeps an object of interest at constant size (see supplemental video). Assuming thin-lens optics, this is achieved by varying the focal length $f$ and object-to-lens distance $u$ such that the magnification $M = f/(u-f)$ remains constant for the selected object.

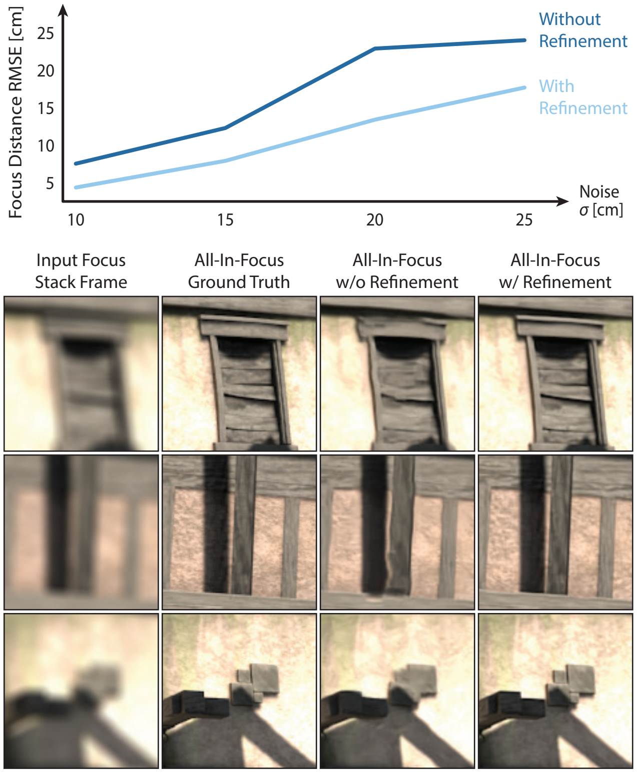

Figure 4. Focus distance refinement improves the focus estimates and all-in-focus images when the initial focus distances are inaccurate or noisy. **Top:** Plot of noise level versus RMSE of focus distances compared to the ground truth; note that refinement consistently reduces the error. **Bottom:** Crops of a single frame for noise level $\sigma = 10\,\text{cm}$. Without refinement, the all-in-focus images are distorted and lack details; with refinement, the image is close to the ground truth.

## References

[1] S. Bae and F. Durand. Defocus magnification. *Computer Graphics Forum (Eurographics)*, 26(3):571–579, 2007.

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

[3] 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.

[4] H. M. Merklinger. *Focusing the View Camera*. 1.6.1 edition, 2010.

[5] G. Riguer, N. Tatarchuk, and J. Isidoro. Real-time depth of field simulation. In *Shader$^2$*, chapter 4. Wordware, 2003.

[6] 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.

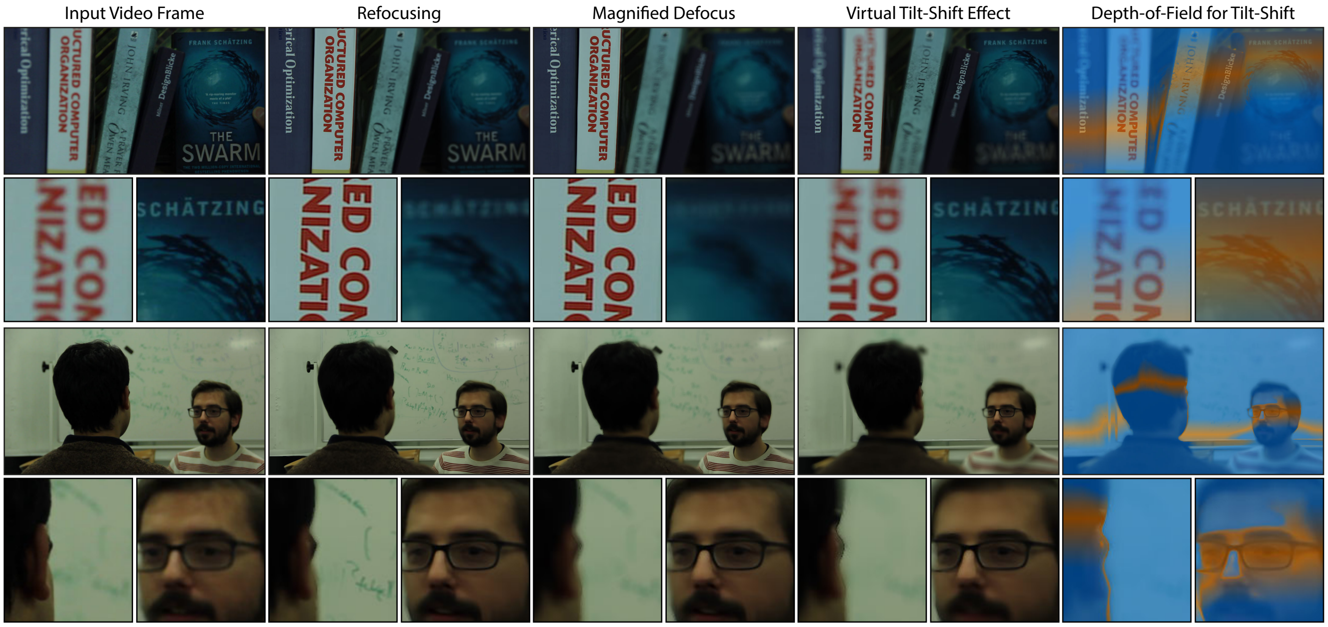

Figure 5. **Video refocusing results.** We first synthetically refocus the input video, then increase the defocus blur by increasing the aperture (smaller *f*-number), and finally apply a virtual tilt-shift effect, which results in a slanted focus plane. Please see our video for full results.