SteReFo: Efficient Image Refocusing with Stereo Vision

Benjamin Busam∗,1,2 Matthieu Hog∗,1 Steven McDonagh1 *equal contribution
b.busam@tum.de matthieu.hog@huawei.com steven.mcdonagh@huawei.com
Gregory Slabaugh1 1Huawei Noah’s Ark Lab 2Technical University of Munich
gregory.slabaugh@huawei.com

Abstract

Whether to attract viewer attention to a particular object, give the impression of depth or simply reproduce human-like scene perception, shallow depth of field images are used extensively by professional and amateur photographers alike. To this end, high quality optical systems are used in DSLR cameras to focus on a specific depth plane while producing visually pleasing bokeh.

We propose a physically motivated pipeline to mimic this effect from all-in-focus stereo images, typically retrieved by mobile cameras. It is capable to change the focal plane a posteriori at 76 FPS on KITTI [13] images to enable real-time applications. As our portmanteau suggests, SteReFo interrelates stereo-based depth estimation and refocusing efficiently. In contrast to other approaches, our pipeline is simultaneously fully differentiable, physically motivated, and agnostic to scene content. It also enables computational video focus tracking for moving objects in addition to refocusing of static images. We evaluate our approach on publicly available datasets [13, 33, 9] and quantify the quality of architectural changes.

1. Introduction

Motivation. Around the turn of the millennium, Japanese photographers coined the term bokeh for the soft, circular out-of-focus highlights produced by near circular apertures [41]. To this day, bokeh is a sign of high quality photographs acquired using professional equipment, closely linked to the depth of field of the optical system in use [36]. Historically, producing such photos has been exclusively possible with high-end DSLRs. Synthesizing the effect of such high-end hardware finds application in particular in consumer mobile devices where the goal is to mimic the physical effects of high-quality lenses in silico [19]. Due to the inherent narrow aperture of cost-efficient optical systems commonly used in mobile phones, the acquired image is all-in-focus. This property hampers the natural image background defocus often desired in many types of scene capture, such as portrait images.

To address this problem a trend has emerged, where shallow depth of field images are computationally synthesized from all-in-focus images [50], usually by leveraging a depth estimation. In the rest of the paper we refer to this task as refocusing.

Drawbacks of recent approaches. The portrait mode of recent smartphones uses depth estimation from monocular [51] or dual-pixel [50] cameras. To circumvent depth estimation errors, previous approaches rely heavily on segmentation of a single salient object, making them limited to scenes with a unique, predominant region of interest. Moreover, this restriction limits the applicability of the underlying refocusing pipeline in other use cases such as object-agnostic image and video refocusing.

Contributions and Outline. We present a general approach that utilizes stereo vision to refocus images and videos (cf. Fig. 1). Our pipeline, entitled SteReFo, leverages the state-of-the-art in efficient stereo depth estimation to obtain a high-quality disparity map and uses a fast, dif-
Figure 2: Computational refocusing of an image from [9]. On the left, one input image (from the stereo pair) together with the intermediate disparity map is illustrated while the right part depicts a continuous sequential refocusing on depth planes from far (1) to close (10). Note the smooth transition of the refocus plane, not feasible with segmentation approaches. Also, note the physically motivated radial bokeh effect on the right traffic light in (7), (8) similar to the effects produced by high-end DSLR equipment. Stereo imagery enables in particular high depth precision for sharp boundaries which can be observed e.g. on the right side of the red car.

A differentiable layered refocusing algorithm to perform the refocusing (Fig. 3 shows the overall pipeline). A total runtime of 0.14 sec (0.11 sec for depth and 0.03 sec for refocusing) makes it computationally tractable for portable devices. Moreover, our method is agnostic to objects present in the scene and the user retains full control of both blur intensity and focal plane (cf. Fig. 2). We also conduct a study to assess the optimal way to combine depth information with the proposed layered refocusing algorithm. Unlike previous work, we quantify the refocus quality of our methods by means of a perceptual metric. More specifically, our contributions are:

2. The proposal and study of novel architectures to combine stereo vision and refocusing for physically motivated bokeh.
3. Both qualitative and quantitative analysis of our approach on synthetic and real images from SceneFlow [33], KITTI [13] and CityScapes [9].
4. A combination of 2D tracking and depth-based refocusing to enable computational focus tracking in videos with tractable computational complexity.

To the best of our knowledge, SteReFo is the first method that is jointly trainable for stereo depth and refocusing, made possible by the efficient design of our differentiable refocusing. Our model makes effective refocusing attainable, yet the approach does not require semantic priors and is not limited in blur intensity. We show that it is possible to mimic the manual refocusing effects found in video acquisition systems by autonomous parameter adjustment.

2. Related Literature

Large bodies of work exist in the domains of both vision-based depth estimation and computational refocusing. We briefly review work most relevant to ours, putting our contributions into context.

2.1. Depth Estimation

Depth estimation from imagery is a well studied problem with a long history to perform estimation from image pairs [43, 30, 47], from temporal image sequences in classical structure from motion (SfM) [12, 22] and simultaneous localization and mapping (SLAM) [18, 35, 11] and reasoning about overlapping images with varying viewpoint [2, 26]. In addition, the task of single image depth estimation has shown recent progress using contemporary learning based methods [29, 14, 16, 32].

Monocular vision. Deep learning based monocular depth estimation employ CNNs to directly infer depth from a static monocular image. They are either trained fully supervised (with a synthetic dataset or ground truth from different sensors) [10, 29] or leverage multiple cameras at training time to use photo-consistency for supervision [14, 40].
However, these approaches are in general tailored for a specific use case and suffer from domain shift errors. To address this drawback, stereo matching [16] or multi-view stereo [32] can be used as a proxy. While these recent approaches estimate reliable depth values, the depth often suffers from over-smoothing [15] which manifests as “flying pixels” in the free space found across depth discontinuities. Accurately and faithfully reproducing such boundaries is, however, critically important for subsequent refocusing quality. Therefore we focus our approach on a binocular stereo cue.

**Multi-view prediction.** For high-accuracy depth maps that preserve precise object boundaries, multiple views are still necessary [44] and binocular stereo is supported by large synthetic [33] as well as real datasets (cf. KITTI [13], CityScapes [9]). Leveraging this data, StereoNet [25] uses a hierarchical disparity prediction network with a deep visual feature backbone which is capable of running at 60 FPS on a consumer GPU. Its successor [56] extends the work with self-supervision to the domain of active sensing while maintaining the core efficiency. We build upon their work to leverage this computational advantage. More recently, Tonioni et al. [48] have proposed a way to perform continuous online domain adaptation for disparity estimation with real-time applicability.

**Other modalities and applications.** Fusion of different visual cues can boost accuracy of individual tasks. Leveraging temporal stereo sequences for unsupervised monocular depth and pose estimation, e.g. by warping deep features, improves the accuracy of both tasks [55]. With the same result, Zou et al. [60] jointly train for optical flow, pose and depth estimation simultaneously while Jiao et al. [23] mutually improve semantics and depth and GeoNet [53] jointly estimates depth, optical flow and camera pose from video. Fully unsupervised monocular depth and visual odometry can also be entangled [58] and 3D mapping applications [57] are realized by heavily relying on dense optical flow in 2D and 3D. Despite the superiority of these approaches, they suffer from larger computational burden or come at the cost of additional training data.

### 2.2. Refocusing

**Refocusing algorithms** are heavily utilized in video-games and animated movie production. A plethora of approaches has been proposed for shallow depth of field rendering in the computer graphics community. We follow the taxonomy in [38] and refer the reader to [5] for a complete survey. The first style of approach uses ray tracing to accurately reproduce the ray integration performed in a camera body. While some approaches focus on physical accuracy [39] and others on (relative) speed [52], these methods are very computationally expensive (up to hours per frame). It is also possible to render a set of views, at different viewpoints, with fast classical rasterization techniques (i.e. creating a light-field [31]). Views are then accumulated to produce a refocused image [17]. However, this requires the scene to be rendered using an amount of time quadratic in the size of the maximum equivalent blur kernel, which is computationally intractable.

This point motivated approaches that seek to reproduce the blur in the image domain directly. Applying depth-adaptive blur kernels can be formulated as scatter [28] or gather [42] operations. While the first is hard to parallelize, the latter suffers from sharply blurred object edges and intensity leakage. Moreover, because the blur kernel is different for each pixel, these approaches are hard to optimize for GPUs [51]. Finally, the last class of algorithms represents the scenes as depth layers in order to apply blur with fixed kernels separately [27]. We give special attention to this type of algorithm in Sec. 3.1.

**Refocusing pipelines.** In contrast to computer graphics, where scene depth and occluded parts can be retrieved easily, in computer vision, the estimated depth is often noisy and background information is not necessarily available due to the projective nature of cameras. This issue can be addressed through a hole filling task for missing pixel depth [8] or by leveraging an efficient bilateral solver for stereo-based depth [4]. Yu et al. [54] directly reconstruct such a light field from stereo images, similar to the techniques discussed previously. It leverages depth estimation, forward warping plus inpainting to reconstruct...
a reasonable number of views that can be interpolated and summed to render the final image. The same idea is presented in [46], but the approach is fully learned with single image input and light field supervision. Zhu et al. [59] consider a refocus task using smartphone-to-DSLR image translation. However, authors concede that average performance is considerably worse than highlighted results. More recently, in [50] is presented a complete pipeline that computes a person segmentation, a depth map from dual pixel and finally the refocused image. While the results are visually compelling, the method is limited to focusing on a person in the image foreground. In [51], the authors decompose the problem into three modules: monocular depth estimation, blurring and upsampling. While the approach provides visually pleasing images it is unclear how it generalizes, given its fully synthetic training set. Due to the blur step being completely learned, we observed that the images lack the distinctive circular bokeh that professional DSLR cameras produce (see Fig. 4). Finally, Srinivasan et al. [45] proposes light field synthesis and supervision with refocused images (i.e. aperture supervision) to learn refocusing. Due to the synthesis of a multitude of views in the first approach, it does not scale well with large kernels. The latter uses the all-in-focus image with a variety of radial kernels and the network is trained to select, for each pixel, which blur value is most likely. The final image is a composition of these blurred images. This approach has the limitation that both the focal plane and the aperture are fixed and cannot be manipulated by a user. In contrast, we want to keep the system parameterizable.

3. Methodology

As illustrated in Fig. 3, our pipeline is split into two modules: depth estimation and refocusing. The inputs to the pipeline are a pair of rectified left and right stereo images. A focus plane and an aperture are two user-controllable parameters. In the following, we explain our proposal for the SteReFo module of Fig. 3 in four different variants which we compare subsequently.

3.1. Disparity Estimation

To produce high quality depth maps, our architecture takes inspiration from two state-of-the-art pipelines for real-time disparity prediction, namely StereoNet [25] and ActiveStereoNet [56] which estimate a subpixel precise low-resolution disparity map that is consecutively upsampled and refined with RGB-guidance from the reference image. Our depth estimation network consists of two Siamese towers with shared weights that extract deep image features at 1/8 of the stereo pair resolution following the architecture described in [56]. We construct a cost volume (CV) by concatenation of the displaced features along the epipolar lines of the rectified input images. The discretization is chosen to include 18 bins. A shifted version of the differentiable ArgMin operator [24] recovers disparities from $i = 0$ to $D_{max} = 17$ where the disparities are given by

$$d_i = \sum_{d=1}^{D_{max}+1} d \cdot \sigma (-C_i (d)) - 1 \quad (1)$$

with the softmax operator $\sigma$ and the cost $C_i$. The low resolution disparity map defined in Eq. 1 is then hierarchically upsampled ($\frac{1}{8} \rightarrow \frac{1}{4} \rightarrow \frac{1}{2} \rightarrow$ full resolution) using bilinear interpolation. Following the idea of Khamis et al. [25], we use residual refinement to recover high-frequency details. Prior to stacking the resized image and low-resolution disparity map, we pass both individually through a small network with 1 convolution and 3 ResNet [20] blocks, as we observed this robustifies our depth prediction quality. The module is trained using a Barron loss [3] with parameters $\alpha = 1$, $c = 2$ and RMSProp [21] optimization with an exponentially decaying learning rate.

3.2. Efficient Layered Depth of Field

Our refocusing module utilizes layered depth of field rendering to enable efficient refocusing. The core idea of layered depth of field rendering [27] is to first decompose the scene into depth layers in order to separately blur each layer before compositing them back together. In contrast to [51], which learns kernel weights, this physically motivated choice directly reflects the effects obtained by DSLR lenses while providing an appropriate balance between efficiency and accuracy for our runtime requirements. Using this approach, the blur operation is applied by combining fixed-kernel convolutions, that make it very efficient in practice.

---

**Algorithm 1:** The layered depth of field base algorithm used in our approach.

**Input:** All-in-focus Image $I$ with associated disparity map $D$, focus plane $d_f$, aperture $a$, and disparity range $[d_{min}, d_{max}]$.

**Output:** Image $I_b$ refocused on the depth plane $d_f$.

```plaintext
1 $I_s = [0]
2 M_s = [0]
3 for d ← d_{min} to d_{max} by $\frac{1}{a}$ do
4 $M^d = \lvert D - d \rvert < \frac{1}{a}$
5 $I^d = M^d \otimes I$
6 $r = a \cdot (d - d_f)$
7 $M_b^d = M^d \ast K(r)$
8 $I_b^d = I^d \ast K(r)$
9 $M_s = M_s \ast (1 - M_b^d) + M_b^d$
10 $I_s = I_s \ast (1 - M_b^d) + I_b^d$
11 end
12 $I_b = I_s \otimes M_s$
```
due to contemporary GPU convolutional implementations. We describe the algorithm in Alg. 1, where the \( \circ \) and \( \otimes \) notation are used for the entrywise Hadamard product and division respectively, and \( * \) denotes convolution. We start from an all-in-focus image \( I \) with its associated disparity map \( D \), a user-set focus plane \( d' \), an aperture \( a \) and a disparity range \( [d_{\text{min}}, d_{\text{max}}] \) defined by the stereo setup capabilities. \( I_s \) and \( M_s \) are two accumulation buffers. We sweep the scene from back to front within a given disparity range using a step size of (optimally) \( 1 \). A mask \( M^d \), defining the zones within a disparity window around the disparity plane \( d \), is used to extract the corresponding texture of objects within depth plane \( I^d \). The corresponding blur radius \( r \) is computed from the distance of the focal plane \( d' \) to the current depth plane \( d \) and the given aperture \( a \). The extracted mask and texture are blurred with a radial kernel \( K(r) \) of diameter \( r \). The blurred mask and texture are accumulated in the buffers \( I_s \) and \( M_s \), overwriting the previous values where the mask is not 0, in order to handle clipping in the blur (i.e. prevent out of focus regions to bleed into in-focus regions). The final blurred image \( I_b \) is rendered by normalizing the accumulated blur texture with the accumulated masks.

Adaptive downsampling. We alter the base algorithm described in Section 3.1 in two ways. To further increase the efficiency of the pipeline, we set a maximum kernel size \( k_{\text{max}} \) that may be applied. For a given disparity plane \( d \), we resize the input image by a factor of \( \gamma = \left\lceil \frac{2r+1}{k_{\text{max}}} \right\rceil \) and apply convolution with a kernel size \( K(\gamma \cdot r) \). The blur result is then upsampled to full resolution using bilinear interpolation. While this is an approximation, the visual difference is marginal due to its application to out of focus regions. However, the computational efficiency is improved by several orders of magnitude (cf. Sec. 4.2).

Differentiability. The second modification we carry out is making this algorithm differentiable in order to use it in an end-to-end trainable pipeline. In Alg. 1, it can be observed that all operations carried out are differentiable, except for computation of the mask which relies on the non-smooth less than operator in line 4. By expressing this operator using the Heaviside step function, the mask computation can be written as:

\[
M^d = H \left( \frac{1}{a} - |D - d| \right) \quad \text{where} \quad H(x) = \begin{cases} 
0 & x < 0 \\
1 & x \geq 0 
\end{cases}
\] (2)

While the Heaviside step function itself is non-differentiable, a smooth approximation is given by \( H(x) = \frac{1}{2} + \frac{1}{2} \tanh(x) \). Hence we can replace line 4 in Alg. 1 with

\[
M^d = \frac{1}{2} + \frac{1}{2} \tanh \left( \alpha \cdot \left( \frac{1}{a} - |D - d| \right) \right) \quad \text{(3)}
\]

where \( \alpha \) controls the transition sharpness of the Heaviside step function approximation. We empirically set \( \alpha = 10^3 \).

3.3. Refocusing Architectures

To intertwine stereo depth estimation and refocusing in SteReFo, we investigate the four architectures illustrated in Fig. 5. The first (\( A_1 \)), dubbed Sequential depth, takes the disparity, estimated from the stereo network, at full resolution and uses it in the layered depth of field technique described in Section 3.1. While the first part is supervised with the ground truth depth, the second stage is not learned. Sequential aperture (\( A_2 \)) is a variation of the first architecture where aperture supervision is used to train the network end-to-end from the blur module. A ground truth blurred image is used instead of the depth, and the loss is defined in the final image domain by applying a pixel-wise Euclidean loss. This is possible thanks to the differentiability of the refocusing algorithm. We use an image refocused with the ground truth disparity for supervision.

The third technique, \( B \), leverages the fact that the cost volume of the stereo network provides a scene representation very similar to the layer decomposition used in the blurring algorithm. We use each slice of the cost volume (after the StereoNet ArgMin step) directly as a mask \( M^d \). We note these slices are of low resolution (1/8 of the input) and therefore bilinearly upsample and refine them using a network with shared weights, up to the resolution required to apply the blurring convolution with a kernel of maximum size \( k_{\text{max}} \). To train this network, we again use images blurred with the ground truth depth map and supervise with an \( L_2 \) loss. A pretrained StereoNet is utilized for which the weights are frozen before the cost volume computation step. The refinement network uses the same blocks as in a refinement scale of ActiveStereoNet [56]. We call this method cost volume refinement.

Branch \( C \) depicts a blur refinement for which we propose to start from the \( 1/2 \) resolution depth map provided by a StereoNet intermediate step, blur the image at half resolution, and then upsample the blurred images back to full resolution using an upsampling akin to [56]. Once again, the network is trained with an \( L_2 \) loss on a ground truth blurred image and the weights of the StereoNet part, up to the second refinement scale are frozen.

4. Experimental Evaluation

In the following section we provide qualitative and quantitative analysis of our approach on synthetic and real imagery using the public datasets SceneFlow [33], KITTI [13] and CityScapes [9]. All experiments are conducted on an Intel(R) Core(TM) i7-8700 CPU machine at 3.20 GHz and we trained all neural networks until convergence using an NVidia GeForce GTX 1080 Ti GPU with Tensorflow [1].
Figure 5: SteReFo architectures. A Siamese tower extracts deep features from a stereo pair which form a cost volume. Four instantiations of our pipeline are depicted. Branch A: the cost volume is then sequentially processed by a depth estimation and differentiable refocusing module and trained with disparity ($A_1$) or aperture ($A_2$) supervision. Branch B: cost volume refinement. Cost volume slices are adaptively upsampled and fed into the layered depth of field pipeline. Branch C: a low resolution depth map is used to predict a downsized refocus image which is consecutively upsampled.

**Qualitative Evaluation Metrics.** Comparing the quality of blurred images is a very challenging task. Barron et al. [4] propose to utilize structural metrics to quantify image quality with a light field ground truth. This modality is difficult to acquire and is therefore usually not present in real datasets of trainable size. Classical image quality metrics, like PSNR and SSIM, do not fully frame the perceptual quality of refocused images [6]. Because there is no consensus on what, quantitatively, makes for a good refocused images (bokeh-wise but also in terms of object boundaries and physical blur accuracy), subjective assessments are often used [19] and some papers exclusively focus on qualitative assessments [54, 52, 46, 51, 45]. In order to provide quantitative evaluation of our results, in addition to classical metrics, we propose to utilize a perceptual metric commonly used by the super resolution community [7, 6], the NIQE score [34].

In a first experiment, we use a synthetic dataset to train and test the four different approaches described in Section 3.3. In a second experiment we assess how our pipeline performs on real data.

### 4.1. Architecture Comparison on Synthetic Data

We train the introduced approaches on the full 35mm driving set of SceneFlow [33] and exclude 11 frames for testing. The virtual aperture and focal plane is fixed to $a = 0.1$ and $d_f = 100$, while the disparity range is set to $d \in [0, 300]$. The maximum blur kernel size is $k_{max} = 11$.

In Fig. 6 we display the result of the forward pass on our test images and display a representative crop example in Fig. 7. Qualitatively we notice that, overall, the result of the sequential approach $A_1$ outperforms the other three in terms of boundaries, bokeh appearance, and blur accuracy. The blur upsampling method $C$ produces blurry output, even in areas that are intended to be sharp, and we observed a loss in the bokeh circularity. The cost volume refinement approach $B$, although a conceptually interesting idea, was found to introduce some high frequency artifacts in the blurred zones and also has generally lower quality boundaries. Finally, aperture supervision ($A_2$) is by far the worst of the approaches, qualitatively, as we find high sensitivity to uniform areas in the image in addition to poor performance at object boundaries.

We further investigate the source of the quality drop in Fig. 8, where we compare the output of the depth using ground truth depth supervision and ground truth blurred images (i.e. aperture supervision). While the depth supervision retrieves disparity precision in particular along depth discontinuities, the supervision with aperture fails to recover small details and depth boundaries, ultimately destroying the depth map gradients.

**Quantifying the result.** The NIQE score [34] unifies a collection of statistical measures to judge the visual appearance of an image. We initially evaluate our introduced approaches for the test images numerically in Tab. 1 and analyze the absolute difference from the ground truth retrievals.
Ground Truth | Sequential Depth ($A_1$) | Sequential Aperture ($A_2$) | CV Refinement ($B$) | Blur Refinement ($C$)
---|---|---|---|---
(a) G.T. | (b) Seq. D. | (c) Seq. A. | (d) CV Ref. | (e) Blur Up.

Figure 6: Comparison of the tested approaches. Each column correspond to one method and each row to one test image. We display on the very left column a ground truth image refocused using the provided ground truth disparity. While we invite the reader to zoom-in to see the details, more example images are included in the supplementary material.

(a) G.T. | (b) Seq. D. | (c) Seq. A. | (d) CV Ref. | (e) Blur Up.

Figure 7: Crop on a representative artifact for the proposed methods. (a) is the ground truth, (b) the output of the sequential approach with depth supervision, (c) the sequential approach trained with aperture supervision, (d) refocusing from the cost volume, (e) the blur upsampling.

---|---|---

Figure 8: Depth map comparison. Left to right columns correspond to ground truth disparity, the disparity from depth supervision and from aperture supervision, respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NIQE ↓</td>
<td>6.5 ± 1.4</td>
<td>7.3 ± 2.4</td>
<td>8.2 ± 3.4</td>
</tr>
<tr>
<td>Rel. ↓</td>
<td>0.1 ± 0.04</td>
<td>1.2 ± 0.8</td>
<td>1.7 ± 1.1</td>
</tr>
<tr>
<td>SSIM ↑</td>
<td>0.98 ± 0.01</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.03</td>
</tr>
<tr>
<td>PSNR ↑</td>
<td>39.16 ± 1.1</td>
<td>36.25 ± 1.2</td>
<td>36.60 ± 1.8</td>
</tr>
</tbody>
</table>

Table 1: Evaluation results for perceptual [34] and structural metrics. ↓ indicates that lower, ↑ that higher is better.

Discussion. We believe our experimental work gives valuable insight into how the tasks of depth and refocusing can be entangled and the resulting benefits of doing so. Firstly, it suggest that depth supervision, and therefore high quality depth data, is essential for refocusing, even more so than retrieving images that are numerically close to the ground truth. This is quantitatively supported by the retrieved NIQE scores. Secondly, upsampling and refining the depth gives better quantitative and qualitative results than upsampling the blurred image. This suggests that the task of correcting the depth is superior to adjusting a blurred image with residual refinement, especially in the boundaries of in-focus objects.

Counter-intuitively, upsampling and filtering the cost volume reveals to be a difficult task, and while the results are still visually appealing, the high computational complexity makes this approach less tractable.
4.2. Results on Real Data

The lack of a publicly available dataset for stereo-based refocusing approaches and the requirements of recent methods for additional information such as segmentation masks [51], varying aperture [45] or co-modalities given by dual-pixel sensing [30] and light fields [4] impede a standardized evaluation protocol. In order to assess how our approach performs on real data, we utilize that our pipeline does not require these additional cues and use the datasets proposed in [9] and [13]. We pick our best-performing approach, i.e. the sequential pipeline using depth supervision (A1) and pretrain on CityScapes [9] to perform static image refocusing (cf. Fig. 2) and refine with the coarse KITTI [13] ground truth for temporal evaluation.

To examine the efficiency of our pipeline, we apply SteReFo individually on consecutive frames of [13]. We utilize correlation-filter based 2D object tracking [49] to reason about the spatial location of objects of interest, per frame, prior to applying image refocusing. The 2D tracker provides state-of-the-art performance at high frame rates, a choice that makes a lightweight overall pipeline feasible. For the sake of comparison, we also retrained the monocular depth estimation approach in [14], and refocus the video using the generated depth as input for our layered refocusing pipeline.

For both approaches, we use disparity values of $d \in [0, 80]$ and an aperture of $a = 0.25$, the focal plane is defined as the median value of the disparities inside the tracking bounding box, and $k_{max} = 11$ (cf. Fig. 1). We compare the results directly in Fig. 9 and notice that blurring is significantly less consistent with respect to the scene geometry in the case of [14] compared to our approach. Indeed, the background is defocused as if it was not at infinity, the cars appear to reside in the same depth plane and the lower section of the car in the middle of the image is blurred (where it should not). The disparity maps for each approach support these observations. The interested reader is referred to the full video sequence https://youtu.be/sX8N702uIag.

<table>
<thead>
<tr>
<th>FPS</th>
<th>$a = 0.1$</th>
<th>$a = 0.2$</th>
<th>$a = 0.5$</th>
<th>$a = 0.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{max} = \infty$</td>
<td>76</td>
<td>17</td>
<td>2</td>
<td>0.4</td>
</tr>
<tr>
<td>$k_{max} = 11$</td>
<td>76</td>
<td>38</td>
<td>23</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 2: Runtime evaluation of the layered refocusing pipeline without ($k_{max} = \infty$) and with ($k_{max} = 11$) adaptive downsampling on images of [13]. Frames per second are evaluated by average of 10 runs.

**Limitations.** The current approach has some limitations. The first one is inherent to all approaches relying on pyramidal depth estimation: small details that are lost at the lowest scale are difficult to recover at the upper scale which is why thin structures are problematic for our depth estimation (cf. the mirror of the truck in Fig. 9). The second observation we make is that StereoNet does not perform as well on real data as on synthetic data. Apart from the obvious difficulties (e.g., specularities, rectification errors, noise, optical aberrations) real data embeds, we also believe the sparse ground truth provided by the projected Lidar data used for supervision does not encourage the network to refine well at the object boundaries. Finally, the refocusing part suffers from the same problems as all image-based shallow DoF rendering techniques: it does not handle a defocus foreground very well. This is due to the fact that we miss occluded information when blurring from one view only.

5. Conclusion

The entanglement of stereo-based depth estimation and refocusing proves to be a promising solution for the task of efficient scene-aware image relurring with appealing bokeh. Future improvements can address a fusion with segmentation methods to enhance boundary precision similar to [37] who entangle the task of semantic segmentation and depth estimation in real-time. The lightweight differentiable architecture with the insight about the value given by a sequential approach with depth supervision can be used for a variety of image and video refocusing applications in other vision pipelines that utilize our refocusing module. For instance in mobile applications, an image pair is taken at one point and different refocusing results may be calculated to help selection by a user afterwards, and the efficiency of our pipeline paves the way to real-time video editing applications on the edge.
References


[30] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos. Review of stereo vision algorithms: from software to hardware. Inter-
6. Additional Results

In addition to Figure 6 found in the main paper, Figure 10 shows further results on the SceneFlow dataset [33] for the different pipelines, \( A_1 \), \( A_2 \), \( B \) and \( C \), which we propose.
<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Sequential Depth ($A_1$)</th>
<th>Sequential Aperture ($A_2$)</th>
<th>CV Refinement ($B$)</th>
<th>Blur Refinement ($C$)</th>
</tr>
</thead>
</table>

Figure 10: Comparison of the four different approaches for the rest of the test set. Each column corresponds to one method and each row to one test image. We display on the very left column the ground truth image refocused using the provided ground truth disparity. We invite the reader to zoom-in to see the details.