Supplemental Materials for
Depth from Defocus in the Wild

Huixuan Tang[1] Scott Cohen[2] Brian Price[2] Stephen Schiller[2] Kiriakos N. Kutulakos[1]

[1] University of Toronto [2] Adobe Research

A1. Additional results

Additional results on real scenes We show four additional scenes captured by the Nexus5 cellphone camera in Figure A1 and seven scenes captured by the Samsung S3 cellphone camera in Figure A2. Among the seven Samsung scenes, three of them already appeared in [9]. Here we also show flow field estimations in addition to the depth map estimations. The scene, camera and imaging conditions of all our data are summarized in Table A1.

Comparison to the DFF method [8] On the Samsung dataset, we compare our LDFD and GDFD results to the results of [8]. Observe that our *two-image* method outperforms the DFF method even through the DFF technique takes a full focal stack of 25 to 41 images as input.

Comparison of GDFD to the HCF semi-dense method [4] In Figure A1 and Figure A2, we qualitatively compare our results to that of the recent hierarchical consensus framework [4] (HCF) applied to the sparse results of LDFD. Our method produces superior results with more accurate discontinuities. This is also shown in Figure A3 which contains comparisons on the dataset synthesized from the Middlebury dataset. We have already shown quantitative comparisons on this dataset in [9].

We tune the two main HCF parameter using grid search and choose the best setting by visual inspection. The depth map is scaled to be in unit of pixels before processing. We use an HCF smoothness weight of 0.4 for both depth and flow, and use an outlier cost of 0.03 for depth and 10 for flow.

Comparison of GDFD to the SPS semi-dense method [11] In Figure A1, A2 and A3 we also show comparison to the slanted-plane-smoothing method [11] (SPS) applied to the sparse results of LDFD. SPS tends to produce outliers where the

	scene	camera	ISO	resolution	aperture	focusing distances	depth range	camera motion	scene motion	flow magnitude
results not in paper	bottles	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -2m	yes	none	$<$ 5 pixels
	telephone	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -2m	yes	none	$<$ 5 pixels
	plants	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -3m	yes	none	$<$ 5 pixels
	window	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm-∞	yes	none	$<$ 5 pixels
	flower2	Nexus5	100	3280 × 2464	F2.4	13cm, 45cm	10cm-30cm	yes	none	$<$ 110 pixels
	christmas	Nexus5	152	3280 × 2464	F2.4	13cm, 45cm	15cm-80cm	yes	none	$<$ 70 pixels
	sushi	Nexus5	100	3280 × 2464	F2.4	13cm, 45cm	15cm-80cm	yes	none	$<$ 40 pixels
	potrait2	Nexus5	100	3280 × 2464	F2.4	13cm, 45cm	20cm-10m	yes	non-rigid	$<$ 140 pixels
results in paper	keyboard	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -2m	yes	none	$<$ 5 pixels
	balls	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -2m	yes	none	$<$ 5 pixels
	fruit	Samsung S3	N/A	640 × 360	F2.6	1.2cm, 2.5cm	1cm -2m	yes	none	$<$ 5 pixels
	bagels	Nexus5	222	3280 × 2464	F2.4	12cm, 30cm	10cm-30cm	yes	rigid	$<$ 80pixels
	flower	Nexus5	100	3280 × 2464	F2.4	8cm, 16cm	10cm-30cm	yes	piecewise-rigid	$<$ 80 pixels
	bell	Nexus5	143	3280 × 2464	F2.4	16cm, 90cm	15cm-80cm	yes	piecewise-rigid	$<$ 60pixels
	potrait	Nexus5	180	3280 × 2464	F2.4	16cm, 90m	20cm-∞	yes	non-rigid	$<$ 150 pixels
	patio	Canon7D	100	5184 × 3456	F4	10m, 20m	6m -∞	no	piecewise-rigid	$<$ 50 pixels
	stairs	Canon7D	100	5184 × 3456	F16	3m, 10m	2m -∞	no	non-rigid	$<$ 70 pixels

Table A1: Scenes, cameras and imaging conditions (Figures 1, 9, 10, A1, and A2).

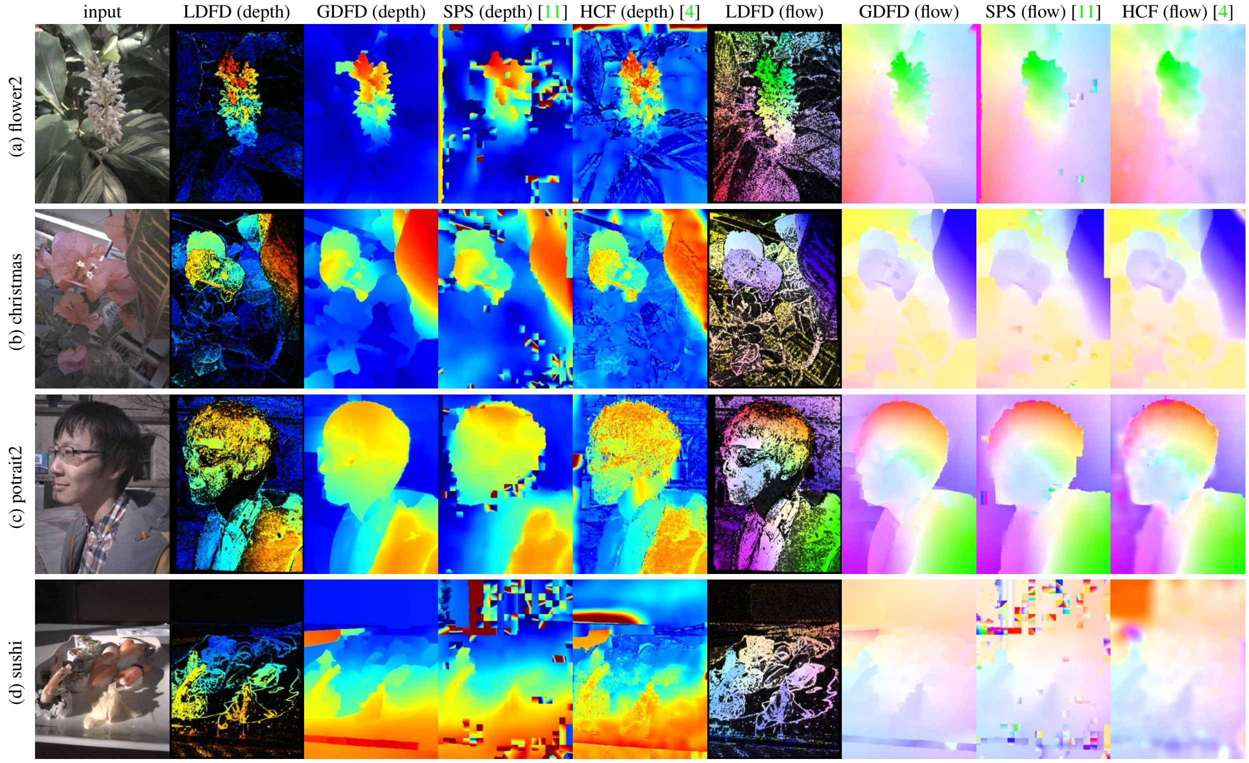

Figure A1: Additional results on real scenes we captured with the Nexus5 phone. Note that SPS, HCF and GDFD are all applied to the same input, generated by LDFD.

LDFD data is sparse, and produces blocky boundaries around outliers. Our GDFD optimization is much more robust to these problems.

The SPS method has eleven parameters. We set the number of local planes to 500, as our model uses $N = 500$ control points. Starting from the recommended setting for the KITTI dataset by the authors [11], we tuned the remaining parameters by trial and error. We used an SPS appearance weight of 1000 and regularization weight of 10. We set length and size weight both to 1000 and the disparity weight to 1000. The hinge and coplanar weights were set to 0.2. The thresholds for coplanar label, hinge label and inier are set to $15, 2$ and 3 respectively. Before running SPS, we scale the LDFD depth map to the range $[100, 150]$ and the DFD flow maps to the range $[0, 255]$. In this way, the depth and flow values of our problem are have similar magnitude to the disparity range that SPS was designed to process.

Discussion It may appear surprising that the two semi-dense methods [4, 11] perform so poorly on our DFD problem despite our efforts to tune their parameters. We see two main reasons. First, defocus is usually much sparser and noisier than stereo data which the two methods were designed to handle. Second, both SPS and HCF initializes the pixel-based depth map and the flow field by inpainting the LDFD results with a naive scanline-based method. Although this works reasonably well on clean stereo data, it performs poorly on our noisy defocus data.

See [1] for more results.

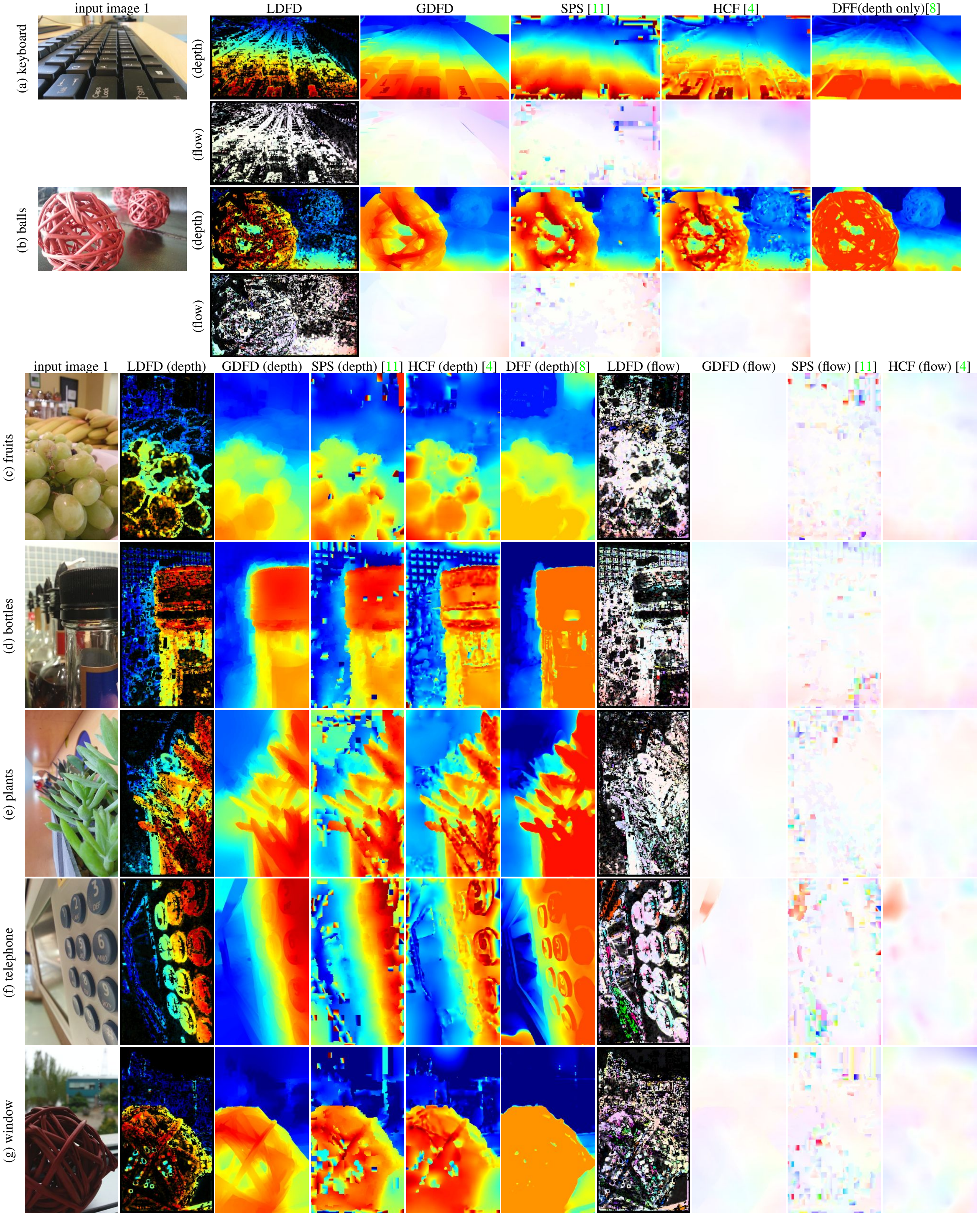

Figure A2: Results on Samsung data [8]. We take *two* input images out of the full focal stack of 20-40 images, and compute LDFD results from the image pair. The GDFD, SPS and HCF results are computed from the LDFD results. The DFF results are computed from the *entire focal stack*. Despite this the DFF depth maps are significantly less detailed than both LDFD and GDFD.

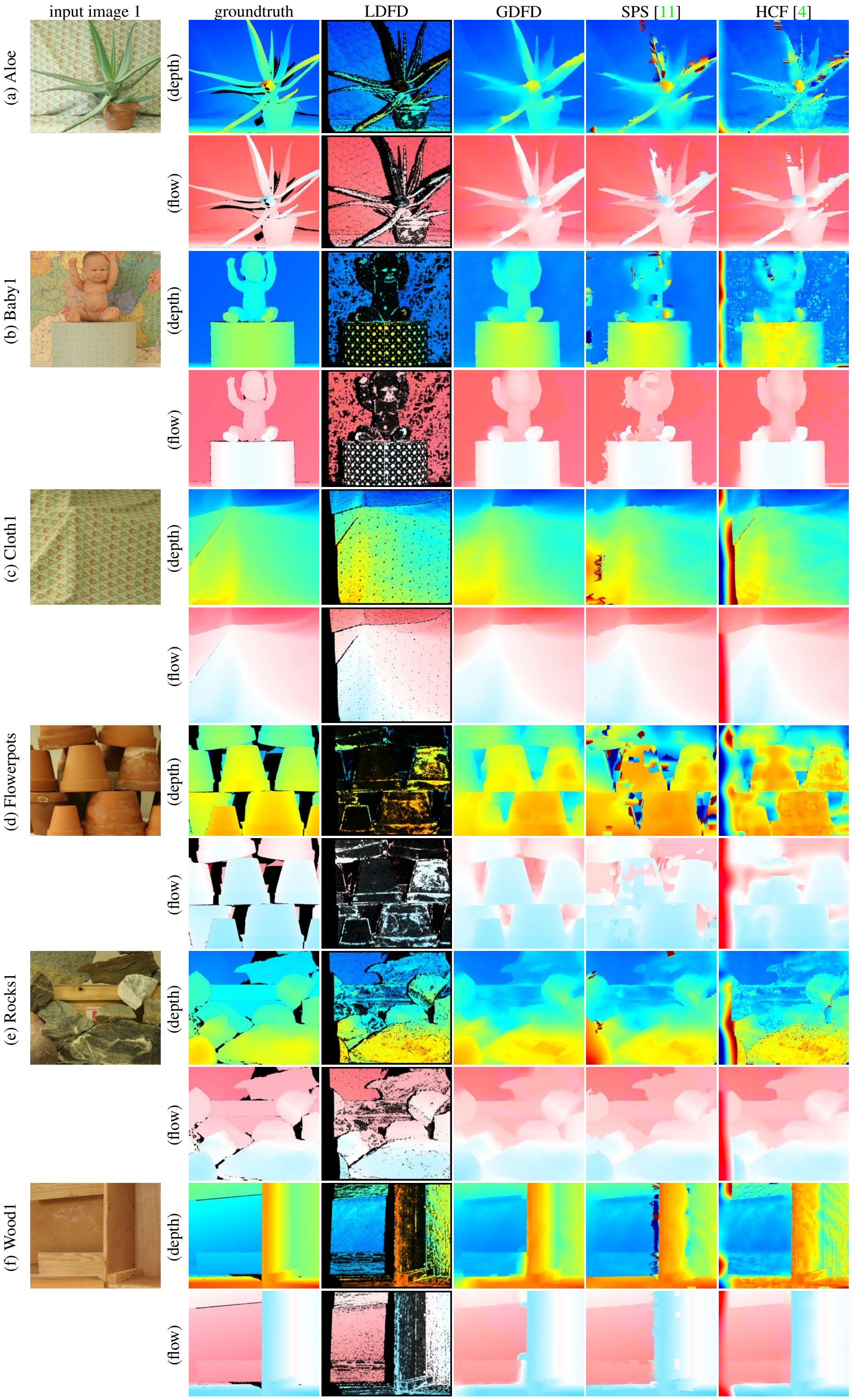

Figure A3: Qualitative results on a synthetic dataset with groundtruth. We generate input images from the Middlebury stereo dataset, where displacement between images is due to parallax and is along scanlines. The black pixels in the groundtruth correspond to pixels whose disparity is not available in the dataset. We ran our Local DFD method to estimate depth and flow and applied the Global DFD optimization and the two semi-dense methods to the LDFD results.

A2. Basic Expressions

Below we provide expressions and proofs supporting the technical discussions in [9].

A2.1. Analytical expression for blur kernels k_1^d and k_2^d

Consider a thin lens model with focal length F and aperture radius A. The defocus radii at focus settings f_1 and f_2 are

$$r_n(d) = A\left(1 - \frac{f_n}{F}\right) + Af_n d \qquad n = 1, 2 \ . \tag{A1}$$

A negative defocus radius indicates that the sensor plane is behind the point's in-focus plane.

The magnification difference between images i_1 and i_2 is

$$m_{12} = f_1/f_2 \ . \tag{A2}$$

After correcting for magnification, the blur kernels $\mathrm{k}_1^d, \mathrm{k}_2^d$ for a point of depth d are pillbox kernels with radius $r_1(d)$ and $m_{12}r_2(d)$, respectively:

$$\begin{aligned} \mathrm{k}_1^d(x,y) &= \mathcal{K}(x,y;r_1(d)) \\ \mathrm{k}_2^d(x,y) &= \mathcal{K}(x,y;m_{12}r_2(d)) \end{aligned} \tag{A3}$$

where $\mathcal{K}(x,y;r)$ is the pillbox blur kernel with radius r

$$\mathcal{K}(x,y;r) = \begin{cases} \frac{1}{\pi r^2} & \text{if } x^2+y^2 \le r^2 \\ 0 & \text{otherwise.} \end{cases} \ . \tag{A4}$$

We use the matlab function `fspecial('disk', |r|)` to compute $\mathcal{K}(x,y;r)$.

A2.2. Expressions for controlled focus

From Eq. (A1) and (A2) we know the difference in blur kernel radii between the two images is

$$r_1(d) - m_{12}r_2(d) = A\left(1 - \frac{f_1}{f_2}\right) \ . \tag{A5}$$

Our tiny blur assumption requires the defocus blur to differ by two pixels for all depths

$$r_1(d) - m_{12}r_2(d) = 2 \ . \tag{A6}$$

Therefore given focus setting f_1, we choose f_2 to satisfy

$$\frac{f_1}{f_2} = 1 - \frac{2}{A} \ . \tag{A7}$$

A2.3. Analytical expressions for Figure 5 in [9]

We give analytical expressions for the curves plotted in Figure 5. These curves characterize the depth variance and operating range of Local DFD method for a Nexus5 cellphone camera. We reproduce that figure in Figure A4 for the reader's convenience.

Analytical expression for the depth variance $\sigma_{\mathbf{p}}^{-2}$ We first derive the depth variance $\sigma_{\mathbf{p}}$ for patch centered at pixel $\mathbf{p}$ in the first image from second-order Taylor expansion of the likelihood function in Eq. (4), assuming that no illumination change occurs to the patch (*i.e.*, the scalar factor $\alpha = 1$).

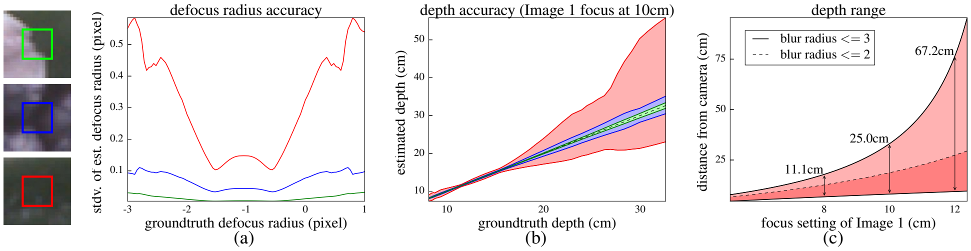

Figure A4: Reproduction of Figure 5 in [9]. (a) Predicting depth uncertainty for three 9 × 9-pixel patches taken from the flower photo in Figure 1 (outlined in color on the left). For each patch, we calculate the standard deviation of the defocus kernel's maximum likelihood (ML) estimate as a function of the ground-truth kernel. This amounts to computing the second derivative of Eq. (4) at the ML depth. The plots confirm our intuition that defocus estimation should be much more precise near an edge (green patch) than on patches with weak (blue) or no (red) texture. (b) Taking the Nexus N5's lens parameters into account, it is possible to convert the plot in (a) into a prediction of actual distance errors to expect from DFD on those patches. (c) Enforcing the tiny blur condition. We first focus at the desired distance (point on the x axis) and then set the second focus to maximize the condition's working range (see Section A2.3 for the analytic expression). The plots show the working ranges of Schechner and Kiryati's optimality condition (red) and of ours (red and pink).

Below we show that computation of the depth variance $\sigma_{\mathbf{p}}^{-2}$ amounts to computing the second derivative of the likelihood function at the maximum likelihood estimate of depth and flow $(\mathrm{d}_{\mathbf{p}}^*, \mathbf{v}_{\mathbf{p}}^*)$:

$$\begin{aligned} \sigma_{\mathbf{p}}^{-2} &= \left. \frac{\partial^2 - \log \Pr(\mathrm{d}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}} \mid \mathrm{i}_1, \mathrm{i}_2)}{\partial \mathrm{d}_{\mathbf{p}}^2} \right|_{\mathrm{d}_{\mathbf{p}} = \mathrm{d}_{\mathbf{p}}^*} && \text{(A8)} \\ &= \sigma_i^{-2} \sum_{\mathbf{q} \in \Omega(\mathbf{p})} \left[\left(\mathrm{i}_1 * \left. \frac{\partial \mathrm{g}_1^d}{\partial d} \right|_{d = \mathrm{d}_{\mathbf{p}}^*} \right) (\mathbf{q}) - \left(\mathrm{i}_2 * \left. \frac{\partial \mathrm{g}_2^d}{\partial d} \right|_{d = \mathrm{d}_{\mathbf{p}}^*} \right) (\mathbf{q} + \boldsymbol{v}) \right]^2 . && \text{(A9)} \end{aligned}$$

Derivation From a second-order Taylor expansion, the likelihood function in Eq. (4) can be approximated by

$$\begin{aligned} &- \log \Pr \left(\mathrm{d}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}} \mid \mathrm{i}_1, \mathrm{i}_2 \right) = \\ &- \log \Pr \left(\mathrm{d}_{\mathbf{p}}^*, \mathbf{v}_{\mathbf{p}}^* \mid \mathrm{i}_1, \mathrm{i}_2 \right) + \left(\left. \frac{\partial^2 - \log \Pr(\mathrm{d}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}} \mid \mathrm{i}_1, \mathrm{i}_2)}{\partial \mathrm{d}_{\mathbf{p}}^2} \right|_{\mathrm{d}_{\mathbf{p}} = \mathrm{d}_{\mathbf{p}}^*} \right) (\mathrm{d}_{\mathbf{p}} - \mathrm{d}_{\mathbf{p}}^*)^2 + (\mathbf{v}_{\mathbf{p}} - \mathbf{v}_{\mathbf{p}}^*)^{\mathrm{T}} \mathbf{H}_{\mathbf{v}}^{-1} (\mathbf{v}_{\mathbf{p}} - \mathbf{v}_{\mathbf{p}}^*) \ , \end{aligned} \tag{A10}$$

where $\mathbf{H}_{\mathbf{v}}$ denotes the 2×2 hessian matrix with regard to $\mathbf{v}_{\mathbf{p}}$ at $(\mathrm{d}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}})$.[1] We get Eq. (A8) by comparing Eq. (A10) to Eq. (5).

In Eq. (4), assuming no illumination change occurs (*i.e.*, $\alpha = 1$), the likelihood function is

$$- \log \Pr(\mathrm{d}_{\mathbf{p}} = d, \mathbf{v}_{\mathbf{p}} = \boldsymbol{v} \mid \mathrm{i}_1, \mathrm{i}_2) = \frac{1}{2} \sigma_i^{-2} \sum_{\mathbf{q} \in \Omega(\mathbf{p})} [(\mathrm{i}_1 * \mathrm{g}_1^d)(\mathbf{q}) - \alpha \cdot (\mathrm{i}_2 * \mathrm{g}_2^d)(\mathbf{q} + \boldsymbol{v})]^2. \tag{A11}$$

where i_1 and i_2 denotes the pair of corresponding patches in the input images under the hypothesis $(d, \boldsymbol{v})$.

By the convolutional theorem and Parseval's theorem, we can express the likelihood equivalently in the Fourier domain as

$$- \log \Pr(\mathrm{d}_{\mathbf{p}} = d, \mathbf{v}_{\mathbf{p}} = \boldsymbol{v} \mid \mathrm{i}_1, \mathrm{i}_2) = \frac{1}{2} \sigma_i^{-2} \sum_{\omega} |\mathrm{I}_1(\omega) \mathrm{G}_1^d(\omega) - \mathrm{I}_2(\omega) \mathrm{G}_2^d(\omega)|^2. \tag{A12}$$

where I_1 and I_2 denote the Fourier transform of i_1 and i_2. The Fourier transform of the unbiased defocus-equalization filters g_1^d and g_2^d are denoted by G_1^d and G_2^d,[2] which are real numbers since g_1^d and g_2^d are both symmetric.

[1] The approximation does not have a first-order term due to Fermat's theorem.

[2] We denote Fourier transforms of images and kernels in capital letters.

Taking second-order derivative with regard to depth d on both sides of Eq. (A12), we have

$$\frac{-\partial^2 \log \Pr(\mathrm{d}_{\mathbf{p}} = d, \mathbf{v}_{\mathbf{p}} = \boldsymbol{v} \mid \mathrm{i}_1, \mathrm{i}_2)}{\partial d^2} = \frac{1}{2}{\sigma_i}^{-2} \sum_{\omega} \frac{\partial^2 |\mathrm{I}_1(\omega)\mathrm{G}_1^d(\omega) - \mathrm{I}_2(\omega)\mathrm{G}_2^d(\omega)|^2}{\partial d^2}. \tag{A13}$$

At each frequency ω,[3]:

$$\frac{\partial^2 |\mathrm{I}_1\mathrm{G}_1^d - \mathrm{I}_2\mathrm{G}_2^d|^2}{\partial d^2} = |\mathrm{I}_1|^2 \frac{\partial^2 (\mathrm{G}_1^d)^2}{\partial d^2} + |\mathrm{I}_2|^2 \frac{\partial^2 (\mathrm{G}_2^d)^2}{\partial d^2} - 2\mathrm{real}(\mathrm{I}_1\overline{\mathrm{I}_2}) \frac{\partial^2 (\mathrm{G}_1^d\mathrm{G}_2^d)}{\partial d^2} \tag{A14}$$

The term $\overline{\mathrm{I}_2}$ denotes complex conjugate of I_2.

To further expand the above expression, we compute the second-order derivatives of terms $|\mathrm{G}_1^d|^2$, $|\mathrm{G}_2^d|^2$ and $\mathrm{G}_1^d\mathrm{G}_2^d$ as

$$\begin{aligned} \frac{\partial^2 |\mathrm{G}_1^d|^2}{\partial d^2} &= 2\left(\frac{\partial \mathrm{G}_1^d}{\partial d}\right)^2 + 2\mathrm{G}_1^d \frac{\partial^2 \mathrm{G}_1^d}{\partial d^2}, \\ \frac{\partial^2 |\mathrm{G}_2^d|^2}{\partial d^2} &= 2\left(\frac{\partial \mathrm{G}_2^d}{\partial d}\right)^2 + 2\mathrm{G}_2^d \frac{\partial^2 \mathrm{G}_2^d}{\partial d^2}, \\ \frac{\partial^2 \mathrm{G}_1^d\mathrm{G}_2^d}{\partial d^2} &= 2\frac{\partial \mathrm{G}_1^d}{\partial d}\frac{\partial \mathrm{G}_2^d}{\partial d} + \mathrm{G}_1^d \frac{\partial^2 \mathrm{G}_2^d}{\partial d^2} + \mathrm{G}_2^d \frac{\partial^2 \mathrm{G}_1^d}{\partial d^2}. \end{aligned} \tag{A15}$$

Plugging Eq. (A15) into Eq. (A14) we get

$$\frac{\partial^2 |\mathrm{I}_1\mathrm{G}_1^d - \mathrm{I}_2\mathrm{G}_2^d|^2}{\partial d^2} = 2\left|\mathrm{I}_1 \frac{\partial \mathrm{G}_1^d}{\partial d} - \mathrm{I}_2 \frac{\partial \mathrm{G}_2^d}{\partial d}\right|^2 + \mathrm{real}\left((\overline{\mathrm{I}_1} \frac{\partial^2 \mathrm{G}_1^d}{\partial d^2} - \overline{\mathrm{I}_2} \frac{\partial^2 \mathrm{G}_2^d}{\partial d^2}) \cdot (\mathrm{I}_1\mathrm{G}_1^d - \mathrm{I}_2\mathrm{G}_2^d) \right). \tag{A16}$$

At the ground-truth depth (*i.e.* $d = d^*$), $\mathrm{I}_1\mathrm{G}_1^d - \mathrm{I}_2\mathrm{G}_2^d$ vanishes and thus the first term of Eq. (A16) becomes 0. Applying this simplification and plugging the result back into Eq. (A13) we get

$$\left.\frac{\partial^2 - \log \Pr(\mathrm{d}_{\mathbf{p}} = d, \mathbf{v}_{\mathbf{p}} = \boldsymbol{v})}{\partial d^2}\right|_{d=\mathrm{d}_{\mathbf{p}}^*} = \sigma_i^{-2} \sum_{\omega} \left.\left|\mathrm{I}_1(\omega) \frac{\partial \mathrm{G}_1^d(\omega)}{\partial d} - \mathrm{I}_2(\omega) \frac{\partial \mathrm{G}_2^d(\omega)}{\partial d}\right|^2\right|_{d=\mathrm{d}_{\mathbf{p}}^*}. \tag{A17}$$

By transforming Eq. (A17) back to the image domain and applying the convolutional theorem and Parseval's theorem, we derive Eq. (A9) immediately. □

Figure 5(a): The x-axis is the defocus radius r_1 of the first patch whose ground-truth depth is d^*. The y-axis is the depth variance in units of pixels, and denoted by σ_r. According to Eq. (A1),

$$\sigma_r = A f_1 \sigma_{\mathbf{p}}. \tag{A18}$$

Figure 5(b): The x-axis corresponds to the ground-truth object distances $z(r^*)$. The y-axis corresponds to the estimated object distances. We plot the interval of distance corresponding to defocus radii $r \in [r^* - \sigma_r, r^* + \sigma_r]$, and compute them with

$$z(r) = \left(\frac{1}{F} - \frac{1}{f_1} + \frac{r}{Af_1}\right)^{-1}. \tag{A19}$$

according to Eq. (A1).

Figure 5(c): The x-axis corresponds to the in-focus distance of the first focus setting $z(0) = (1/F - 1/f_1)^{-1}$. The y-axis corresponds to object distances. Given defocus radius r and focus setting f_1, it is computed by Eq. (A19). The red and pink zone plots our operating range in distance $[z(1), z(-3)]$, while the red zone plots the operating range under Schechner and Kiryati's condition [7] $[z(1), z(-2)]$.

[3] In Eq. (A14)-(A17) and Eqs. (A24)-(A28), all terms depend on the frequency ω. We omit dependence on frequency ω for brievity.

A3. Proof of Proposition 1 : Defocus Equalization Filters

***Proposition* 1** *If* i_1 *and* i_2 *are fronto-parallel image patches related by a 2D translation* $\boldsymbol{v}$ *and an intensity scaling* α *that is sufficiently close to one, the image error*

$$(\mathrm{i}_1 * \mathrm{g}_1^d)(\mathbf{p}) - \alpha \cdot (\mathrm{i}_2 * \mathrm{g}_2^d)(\mathbf{p} + \boldsymbol{v}) \tag{A20}$$

follows the same distribution as the noise in i_1 *and* i_2. *The defocus-equalization filters* g_1^d *and* g_2^d *are defined as*

$$\begin{aligned} \mathrm{g}_1^d =& \mathcal{F}^{-1}\left[\ \mathcal{F}[\mathrm{k}_2^d] \ / \sqrt{\mathcal{F}[\mathrm{k}_1^d]^2 + \mathcal{F}[\mathrm{k}_2^d]^2}\ \right] \\ \mathrm{g}_2^d =& \mathcal{F}^{-1}\left[\ \mathcal{F}[\mathrm{k}_1^d] \ / \sqrt{\mathcal{F}[\mathrm{k}_1^d]^2 + \mathcal{F}[\mathrm{k}_2^d]^2}\ \right] \end{aligned} \tag{A21}$$

where $\mathcal{F}[], \mathcal{F}^{-1}[]$ *denote the Fourier transform and its inverse.*

Proof sketch Let h be the hidden, defocus-free version of the first patch. Assuming the illumination changes by α between the two patches, the second hidden image is $\alpha^{-1}\mathrm{h}$. Denote the defocus blur kernels associated with the two input patches at the ground-truth depth by k_1 and k_2. The two patches are given by

$$\mathrm{i}_1 = \quad \mathrm{h} * \mathrm{k}_1 + \mathrm{n}_1 \ , \tag{A22}$$

$$\mathrm{i}_2 = \quad \alpha^{-1}\mathrm{h} * \mathrm{k}_2 + \mathrm{n}_2 \ , \tag{A23}$$

where n_1 and n_2 are i.i.d. zero mean Gaussian noise variables of variance σ_i^2.

Transforming Eqs (A22)-(A23) to the Fourier domain we get

$$\mathcal{F}[\mathrm{i}_1] = \mathrm{I}_1 = \quad \mathrm{HK}_1 + \mathrm{N}_1 \ , \tag{A24}$$

$$\mathcal{F}[\mathrm{i}_2] = \mathrm{I}_2 = \quad \alpha^{-1}\mathrm{HK}_2 + \mathrm{N}_2 \ . \tag{A25}$$

Now we compute the Fourier transform of the image error defined in Eq. (A20), under depth hypothesis d:

$$\mathrm{I}_1\mathrm{G}_1^d - \alpha\mathrm{I}_2\mathrm{G}_2^d \ = \mathrm{H}(\mathrm{K}_1\mathrm{G}_1^d - \mathrm{K}_2\mathrm{G}_2^d) + (\mathrm{N}_1\mathrm{G}_1^d - \alpha\mathrm{N}_2\mathrm{G}_2^d), \tag{A26}$$

which is a Gaussian random variable.

At each frequency, the mean of the Gaussian equals zero

$$\mathrm{H}(\mathrm{K}_1\mathrm{G}_1^d - \mathrm{K}_2\mathrm{G}_2^d) = 0 \tag{A27}$$

according to the definition of defocus-equalization filters. The variance of Eq. (A26) $(|\mathrm{G}_1^d|^2 + \alpha^2|\mathrm{G}_2^d|^2)\sigma_i^2$ satisfies

$$\lim_{\alpha \to 1} \left(|\mathrm{G}_1^d|^2 + \alpha^2|\mathrm{G}_2^d|^2\right)\sigma_i^2 = (|\mathrm{G}_1^d|^2 + |\mathrm{G}_2^d|^2)\sigma_i^2 = \sigma_i^2. \tag{A28}$$

Thus for each frequency, Eq. (A26) follows a Gaussian distribution of mean zero and variance σ_i^2. Transforming back to the image domain, Eq. (A20) also follows a Gaussian distribution of zero mean and variance σ_i^2. □

A4. The global DFD Optimization

In the following we provide a detailed analysis of the global DFD optimization problem, which allows us to solve the discrete-continuous problem by block descent on four subsets of variables:

(1) spline weights $\mathbf{w}$ and segment label of pixels s;

(2) patch-based depth map and flow field $\mathcal{P} = (\mathrm{d}, \mathbf{v})$, depth- and flow-specific parameters of the spline model $\mathbf{D}, \mathbf{U}, \mathbf{V}$, and the segment labels of control points $\mathbf{t}$;

(3) occlusion relationships between each pair of segments $\mathbf{O}$; and

(4) feature vectors of the control points $\mathbf{C}$.

The first two subsections prove the equivalence of our objective function $[\mathrm{E}_{\mathrm{DFD}}(\mathcal{P}, \mathcal{S}(\mathcal{M}), \mathcal{Z}) + \mathrm{E}_{\mathrm{prior}}(\mathcal{M}, \mathcal{Z})]$ in Eq. (1) of the paper with two alternative expressions. These theoretical results builds the foundation for two key steps of our block descendant algorithm. This leads to the algorithm we give in Section A4.3, also shown as Algorithm 1 in [9].

A4.1. Optimization as MRF minimization

In this subsection, we consider the global DFD optimization over the spline weights $\mathbf{w}$ and pixel segmentation s, while fixing the remaining variables $\mathrm{d}, \mathbf{v}, \mathbf{C}, \mathbf{D}, \mathbf{U}, \mathbf{V}$ and $\mathbf{t}$.

We start by introducing an auxiliary variable s^*, which is defined as the labels of the front-most segments of patches. Given the patch-based segment labels s^* and the pixel-based segment labels s, we can define the front-most segment of patches as

$$\Omega_f(\mathbf{p}) = \{\mathbf{q} : \mathbf{q} \in \Omega(\mathbf{p}) \text{ and } \mathrm{s}^*_{\mathbf{p}} = \mathrm{s}_{\mathbf{q}}\} \ . \tag{A29}$$

The following proposition shows that the global optimization over $(\mathbf{w}, \mathrm{s})$ is equivalent to an MRF minimization problem when ignoring the dependence of s^* on s.

***Proposition* 2** *The global DFD optimization problem of Eq.* (1)

$$\min_{\mathbf{w},\mathrm{s}} \ [\,\mathrm{E}_{\mathit{DFD}} + \mathrm{E}_{\mathit{prior}}\,] \quad \textit{s.t.}\ \forall \mathbf{q}, \begin{cases} \mathbf{w}_{\mathbf{q}n} \geq 0, \\ \mathbf{w}_{\mathbf{q}n} = 0 \ \textit{if}\, \mathrm{s}_{\mathbf{q}} \neq \mathbf{t}_n \\ \sum_n \mathbf{w}_{\mathbf{q}n} = 1 \end{cases} \tag{A30}$$

is equivalent to the Markov Random Field minimization problem

$$\min_{\mathrm{s}} \sum_{\mathbf{q}} \underbrace{\mathrm{D}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}})}_{\textit{data term}} + \sum_{\mathbf{p},\mathbf{q} \in \mathcal{N}_8} \underbrace{\mathrm{V}_{\mathbf{pq}}(\mathrm{s}_{\mathbf{p}}, \mathrm{s}_{\mathbf{q}})}_{\textit{smoothness term}} \ . \tag{A31}$$

The data term and the smoothness term can be written respectively in the form

$$\mathrm{D}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) = \min_{\mathbf{w}_{\mathbf{q}}} \left[e^{(0)}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) + \sum_n \mathbf{w}_{\mathbf{q}n} [\log(\mathbf{w}_{\mathbf{q}n}) + e^{(1)}_{\mathbf{q}n}(\mathrm{s}_{\mathbf{q}}) + \mathbf{w}_{\mathbf{q}n} e^{(2)}_{\mathbf{q}mn}(\mathrm{s}_{\mathbf{q}})] \right] \quad \textit{s.t.} \begin{cases} \mathbf{w}_{\mathbf{q}n} \geq 0, \\ \mathbf{w}_{\mathbf{q}n} = 0 \ \textit{if}\, \mathrm{s}_{\mathbf{q}} \neq \mathbf{t}_n \\ \sum_n \mathbf{w}_{\mathbf{q}n} = 1 \end{cases} \tag{A32}$$

$$\mathrm{V}_{\mathbf{pq}}(\mathrm{s}_{\mathbf{p}}, \mathrm{s}_{\mathbf{q}}) = \begin{cases} \lambda_b & \textit{if}\, \mathrm{s}_{\mathbf{p}} \neq \mathrm{s}_{\mathbf{q}} \\ 0 & \textit{if}\, \mathrm{s}_{\mathbf{p}} = \mathrm{s}_{\mathbf{q}} \end{cases} \ . \tag{A33}$$

In the MRF data term, the functions $e^{(0)}_{\mathbf{q}}$, $e^{(1)}_{\mathbf{q}n}$, $e^{(2)}_{\mathbf{q}mn}$ are indepent of $\mathbf{w}$ and s when fixing the labels s^* of the front-most segment of all patches.

Proposition 2 shows that we can solve the minimization problem in Eq. (A30) in two layers. In the inner layer, we can compute for each pixel $\mathbf{q}$ the MRF data term of all possible segmentation labels $\mathrm{s}_{\mathbf{q}}$ by solving the minimization problem in Eq. (A31). In the outer layer, we can solve the MRF minimization problem in Eq. (A31) to obtain the pixel-based segmentation labels.

To define the MRF terms concretely, we introduce three additional auxiliary functions of $\mathrm{s}_{\mathbf{q}}$

$$N_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) = |\{\mathbf{p} : \mathbf{q} \in \Omega_f(\mathbf{p})\}| = |\{\mathbf{p} : \mathbf{q} \in \Omega(\mathbf{p}) \text{ and } \mathrm{s}^*_{\mathbf{p}} = \mathrm{s}_{\mathbf{q}}\}| \ , \tag{A34}$$

$$\bar{\mathrm{d}}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) = \frac{1}{N_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}})} \sum_{\mathbf{p}:\mathbf{q} \in \Omega_f(\mathbf{p})} \mathrm{d}_{\mathbf{p}} \ , \quad \bar{\mathbf{v}}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) = \frac{1}{N_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}})} \sum_{\mathbf{p}:\mathbf{q} \in \Omega_f(\mathbf{p})} \mathbf{v}_{\mathbf{p}} \ . \tag{A35}$$

With these terms we have the following corollary:

***Corollary* 1** *The functions $e^{(1)}_{\mathbf{q}}$, $e^{(2)}_{\mathbf{q}n}$, $e^{(3)}_{\mathbf{q}mn}$ in Eq.* (A32) *are given by*

$$e^{(0)}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}}) = \frac{1}{2} \sum_{\mathbf{p}:\mathbf{q} \in \Omega_f(\mathbf{p})} \left(\sigma_d^{-2}(\mathrm{d}_{\mathbf{q}} - \bar{\mathrm{d}}_{\mathbf{q}})^2 + \sigma_v^{-2}|\mathbf{v}_{\mathbf{q}} - \bar{\mathbf{v}}_{\mathbf{q}}|^2\right) + (81 - N_{\mathbf{q}})\tau_o \ , \tag{A36}$$

$$e^{(1)}_{\mathbf{q}n}(\mathrm{s}_{\mathbf{q}}) = \frac{N_{\mathbf{q}}}{2} \left(\sigma_d^{-2}(\bar{\mathrm{d}}_{\mathbf{q}} - \mathbf{D}_n\mathbf{q})^2 + \sigma_v^{-2}\,|\bar{\mathbf{v}}_{\mathbf{q}} - [\mathbf{U}_n\mathbf{q}, \mathbf{V}_n\mathbf{q}]|^2\right) + 2\lambda_s \sum_m (\delta_{mn}\psi_{\mathbf{q}mn} + (1 - \delta_{mn})\tau_s) + \lambda_i|\mathbf{f}_{\mathbf{q}} - \mathbf{C}_n|^2 \ , \tag{A37}$$

$$e^{(2)}_{\mathbf{q}mn}(\mathrm{s}_{\mathbf{q}}) = (\lambda_f - \frac{N_{\mathbf{q}}}{2})\psi_{\mathbf{q}mn} \ . \tag{A38}$$

Here the term $\psi_{\mathbf{q}mn}$ is defined in Eq. (12) *as* $\frac{1}{2}\sigma_d^{-2}(\mathbf{D}_m\mathbf{q} - \mathbf{D}_n\mathbf{q})^2 + \frac{1}{2}\sigma_{\mathbf{v}}^{-2}(\mathbf{U}_m\mathbf{q} - \mathbf{U}_n\mathbf{q})^2 + \frac{1}{2}\sigma_{\mathbf{v}}^{-2}(\mathbf{V}_m\mathbf{q} - \mathbf{V}_n\mathbf{q})^2$.

We prove the proposition and the corollary with three lemmas.

***Lemma* 1** *Let* $\mathbf{w}$ *be a* $N \times 1$ *weight vector corresponding to a convex combination,* i.e. $\sum_n \mathbf{w}_n = 1$ *and* $\mathbf{w} \geq 0$*. For any scalar* x *and* $N \times 1$ *vector* $\mathbf{x}$*, the following equality holds:*

$$(x - \sum_n \mathbf{w}_n \mathbf{x}_n)^2 = \sum_n \mathbf{w}_n (x - \mathbf{x}_n)^2 - \frac{1}{2} \sum_{m,n} \mathbf{w}_m \mathbf{w}_n (\mathbf{x}_m - \mathbf{x}_n)^2 \quad . \tag{A39}$$

Proof sketch We start by expanding the left handside of Eq. (A39)

$$(x - \sum_n \mathbf{w}_n \mathbf{x}_n)^2 = (\sum_n \mathbf{w}_n x - \sum_n \mathbf{w}_n \mathbf{x}_n)^2 = \left[\sum_n \mathbf{w}_n (x - \mathbf{x}_n) \right]^2 \quad . \tag{A40}$$

Since $\mathbf{w}$ is a weight vector corresponding to a convex combination, for any $N \times 1$ vector $\mathbf{z}$, we have

$$\sum_n \mathbf{w}_n \mathbf{z}_n^2 - (\sum_n \mathbf{w}_n \mathbf{z}_n)^2 = \sum_n \mathbf{w}_n (\mathbf{z}_n - \sum_m \mathbf{w}_m \mathbf{z}_m)^2 \quad . \tag{A41}$$

By applying Eq. (A41) with $\mathbf{z} = x - \mathbf{x}$ we can rewrite the right handside of Eq. (A40) as

$$(x - \sum_n \mathbf{w}_n \mathbf{x}_n)^2 = \sum_n \mathbf{w}_n (x - \mathbf{x}_n)^2 - \sum_n \mathbf{w}_n (x - \mathbf{x}_n - (x - \sum_m \mathbf{w}_m \mathbf{x}_m))^2 \quad . \tag{A42}$$

Since $\sum_m \mathbf{w}_m = 1$, we have

$$\begin{aligned} x - \mathbf{x}_n - \sum_m \mathbf{w}_m \mathbf{x}_m &= x - \sum_m \mathbf{w}_m \mathbf{x}_n - (x - \sum_m \mathbf{w}_m \mathbf{x}_m) = \sum_m \mathbf{w}_m \mathbf{x}_m - \mathbf{x}_n \\ &= \sum_n \mathbf{w}_n (x - \mathbf{x}_n)^2 - \sum_n \mathbf{w}_n (\sum_m \mathbf{w}_m \mathbf{x}_m - \mathbf{x}_n)^2 \quad . \end{aligned} \tag{A43}$$

By applying Eq. (A41) with $\mathbf{z} = \mathbf{x}$ we can expand the second term on the right handside of Eq. (A42) as

$$\sum_n \mathbf{w}_n (\sum_m \mathbf{w}_m \mathbf{x}_m - \mathbf{x}_n)^2 = \sum_n \mathbf{w}_n \mathbf{x}_n^2 - (\sum_n \mathbf{w}_n \mathbf{x}_n)^2 \tag{A44}$$

Since $\sum_m \mathbf{w}_m = 1$, we have

$$\sum_n \mathbf{w}_n \mathbf{x}_n^2 = \sum_n (\sum_m \mathbf{w}_m) \mathbf{w}_n \mathbf{x}_n^2 = \sum_{mn} \mathbf{w}_m \mathbf{w}_n \mathbf{x}_n^2 \tag{A45}$$

Using the fact that $\sum_{mn} \mathbf{w}_m \mathbf{w}_n \mathbf{x}_n^2 = \sum_{mn} \mathbf{w}_n \mathbf{w}_m \mathbf{x}_m^2$, we can expand the right handside of Eq. (A44) as

$$\begin{aligned} \sum_n \mathbf{w}_n (\sum_m \mathbf{w}_m \mathbf{x}_m - \mathbf{x}_n)^2 =& \frac{1}{2} \sum_{mn} \mathbf{w}_m \mathbf{w}_n \mathbf{x}_m^2 + \frac{1}{2} \sum_{mn} \mathbf{w}_m \mathbf{w}_n \mathbf{x}_n^2 - \left(\sum_m \mathbf{w}_m \mathbf{x}_m \right) \left(\sum_n \mathbf{w}_n \mathbf{x}_n \right) \\ =& \frac{1}{2} \sum_{mn} \mathbf{w}_m \mathbf{w}_n (\mathbf{x}_m^2 + \mathbf{x}_n^2 - 2 \mathbf{x}_m \mathbf{x}_n) \\ =& \frac{1}{2} \sum_{mn} \mathbf{w}_m \mathbf{w}_n (\mathbf{x}_m - \mathbf{x}_n)^2 \quad . \end{aligned} \tag{A46}$$

By plugging Eq. (A46) into Eq. (A42) we obtain Eq. (A39). □

***Lemma* 2** *The local prior term* $\sum_{\mathbf{p}} \sum_{\mathbf{q} \in \Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}}$ *of the bottom-up likelihood* E_{DFD} *is given by:*

$$\begin{aligned} \sum_{\mathbf{p}} \sum_{\mathbf{q} \in \Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} =& \sum_{\mathbf{q}} \sum_{\mathbf{p}:\mathbf{q} \in \Omega_f(\mathbf{p})} \left[\frac{1}{2} \sigma_d^{-2} (\mathrm{d}_{\mathbf{p}} - \bar{\mathrm{d}}_{\mathbf{q}})^2 + \frac{1}{2} \sigma_v^{-2} |\mathbf{v}_{\mathbf{p}} - \bar{\mathbf{v}}_{\mathbf{q}}|^2 \right] + \sum_{\mathbf{q}} (81 - N_{\mathbf{q}}) \tau_o \\ &+ \sum_{\mathbf{q}} N_{\mathbf{q}} \sum_n \mathbf{w}_{\mathbf{q}n} \left[\frac{1}{2} \sigma_d^{-2} (\mathbf{D}_n \mathbf{q} - \bar{\mathrm{d}}_{\mathbf{q}})^2 + \frac{1}{2} \sigma_v^{-2} |[\mathbf{U}_n \mathbf{q}, \mathbf{V}_n \mathbf{q}] - \bar{\mathbf{v}}_{\mathbf{q}}|^2 \right] \\ &- \sum_{\mathbf{q}} \sum_{mn} \frac{1}{2} \mathbf{w}_m \mathbf{w}_n N_{\mathbf{q}} \psi_{\mathbf{q}mn} \quad . \end{aligned} \tag{A47}$$

Proof sketch We first swap the two sums on the left handside of Eq. (A47)

$$
\begin{aligned}
\sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} &= \sum_{\mathbf{q}} \sum_{\mathbf{p}:\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} \\
&= \sum_{\mathbf{q}} \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \left[\frac{1}{2\sigma_d^2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 + \frac{1}{2\sigma_{\mathbf{v}}^2}|\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2\right] + \sum_{\mathbf{q}} \left[\sum_{\mathbf{p}:\mathbf{q}\in\Omega(\mathbf{p})} \tau_o - \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \tau_o\right] \qquad \text{(A48)} \\
&= \sum_{\mathbf{q}} \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \left[\frac{1}{2\sigma_d^2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 + \frac{1}{2\sigma_{\mathbf{v}}^2}|\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2\right] + \sum_{\mathbf{q}} (81 - N_{\mathbf{q}})\tau_o
\end{aligned}
$$

The contribution relevant to the depth of pixel $\mathbf{q}$ is

$$
\sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} (\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 = N_{\mathbf{q}}\mathrm{d}'^2_{\mathbf{q}} - 2(\sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \mathrm{d}_{\mathbf{p}})\mathrm{d}'_{\mathbf{q}} + \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \mathrm{d}^2_{\mathbf{p}} = N_{\mathbf{q}}(\bar{\mathrm{d}}_{\mathbf{q}} - \mathrm{d}'_{\mathbf{q}})^2 + \sum_{\Omega_f(\mathbf{p})} \mathrm{d}^2_{\mathbf{p}} - N_{\mathbf{q}}\bar{\mathrm{d}}^2_{\mathbf{q}} \ . \qquad \text{(A49)}
$$

The last step is obtained by completing the squares.

Further, we incorporate the hard constraint $\mathrm{d}'_{\mathbf{q}} = \sum_n \mathbf{w}_{\mathbf{q}n}\mathbf{D}_n\mathbf{q}$ in Eq. (9) into the first term of Eq. (A49), apply Lemma 1 to depth, and get

$$
(\bar{\mathrm{d}}_{\mathbf{q}} - \mathrm{d}'_{\mathbf{q}})^2 = (\bar{\mathrm{d}}_{\mathbf{q}} - \sum_n \mathbf{w}_{\mathbf{q}n}\mathbf{D}_n\mathbf{q})^2 = \sum_n \mathbf{w}_{\mathbf{q}n}(\bar{\mathrm{d}}_{\mathbf{q}} - \mathbf{D}_n\mathbf{q})^2 - \frac{1}{2}\sum_{mn} \mathbf{w}_m\mathbf{w}_n(\mathbf{D}_m\mathbf{q} - \mathbf{D}_n\mathbf{q})^2 \qquad \text{(A50)}
$$

Also note that

$$
\sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} \mathrm{d}^2_{\mathbf{p}} - N_{\mathbf{q}}\bar{\mathrm{d}}^2_{\mathbf{q}} = \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} (\mathrm{d}_{\mathbf{p}} - \bar{\mathrm{d}}_{\mathbf{q}})^2 \ . \qquad \text{(A51)}
$$

Plugging Eq. (A50) and Eq. (A51) into Eq. (A49), we obtain

$$
\sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} (\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 = N_{\mathbf{p}}\left[\sum_n \mathbf{w}_{\mathbf{q}n}(\bar{\mathrm{d}}_{\mathbf{q}} - \mathbf{D}_n\mathbf{q})^2 - \frac{1}{2}\sum_{mn} \mathbf{w}_m\mathbf{w}_n(\mathbf{D}_m\mathbf{q} - \mathbf{D}_n\mathbf{q})^2\right] + \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} (\mathrm{d}_{\mathbf{p}} - \bar{\mathrm{d}}_{\mathbf{q}})^2 \ . \qquad \text{(A52)}
$$

Similar relationships can be derived for the two flow fields:

$$
\begin{aligned}
\sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} |\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2 = N_{\mathbf{p}}&\left[\sum_n \mathbf{w}_{\mathbf{q}n}|\bar{\mathbf{v}}_{\mathbf{q}} - [\mathbf{U}_n\mathbf{q}, \mathbf{V}_n\mathbf{q}]|^2 - \frac{1}{2}\sum_{mn} \mathbf{w}_m\mathbf{w}_n\left((\mathbf{U}_m\mathbf{q} - \mathbf{U}_n\mathbf{q})^2 + (\mathbf{V}_m\mathbf{q} - \mathbf{V}_n\mathbf{q})^2\right)\right] \qquad \text{(A53)} \\
&+ \sum_{\mathbf{p}:\mathbf{q}\in\Omega_f(\mathbf{p})} |\mathbf{v}_{\mathbf{p}} - \bar{\mathbf{v}}_{\mathbf{q}}|^2 \ .
\end{aligned}
$$

Plugging Eq. (A52) and (A53) into Eq. (A48) we get derive Eq. (A47). □

***Lemma* 3** *The segment-specific smoothing term* E_{smo} *can be rewritten as*

$$
\mathrm{E}_{smo}(\mathbf{D}, \mathbf{U}, \mathbf{V}, \mathbf{w}, \mathbf{t}) = 2\sum_{\mathbf{q}}\sum_n \mathbf{w}_{\mathbf{q}n}\sum_m [\delta_{mn}\psi_{\mathbf{q}mn}(\mathbf{D}, \mathbf{U}, \mathbf{V}) + (1 - \delta_{mn})\tau_s]. \qquad \text{(A54)}
$$

Proof sketch For each pixel $\mathbf{q}$

$$
\sum_{mn}(\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n})\delta_{mn}\psi_{\mathbf{q}mn} = \sum_{mn} \mathbf{w}_{\mathbf{q}m}\delta_{mn}\psi_{\mathbf{q}mn} + \sum_{mn} \mathbf{w}_{\mathbf{q}n}\delta_{mn}\psi_{\mathbf{q}mn}. \qquad \text{(A55)}
$$

Since m and n are symmetric in δ_{mn} and $\psi_{mn\mathbf{q}}$,

$$
\sum_{mn}(\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n})\delta_{mn}\psi_{\mathbf{q}mn} = \sum_m\sum_n \mathbf{w}_{\mathbf{q}m}\delta_{nm}\psi_{\mathbf{q}nm} + \sum_m\sum_n \mathbf{w}_{\mathbf{q}n}\delta_{mn}\psi_{\mathbf{q}mn} = 2\sum_n \mathbf{w}_{\mathbf{q}n}\sum_m \delta_{mn}\psi_{\mathbf{q}mn}. \qquad \text{(A56)}
$$

Similarly, we also have

$$\sum_{mn}(\mathbf{w}_{\mathbf{q}m}+\mathbf{w}_{\mathbf{q}n})(1-\delta_{mn})\tau_s = 2\sum_{n}\mathbf{w}_{\mathbf{q}n}\sum_{m}(1-\delta_{mn})\tau_s. \tag{A57}$$

Summing Eq. (A56) and Eq. (A57) over all pixels $\mathbf{q}$ we get Eq. (A54). □

Proof sketch of Proposition 2 In the top-down term $\mathrm{E}_{\text{prior}}$, the discontinuity penalty term E_{bnd} is equivalent to the MRF smoothness function V defined in Eq. (A33). All the other four terms contributes to the data term. For the segmentation-specific smoothness term, we use the equivalent expressions proposed in Lemma 3. We use the original expression of the flatness term and the image coherence term. In the bottom-up term E_{DFD}, the local likelihood term is independent of $\mathbf{w}$ or s, and therefore can be omitted. We use the equivalent expression in Lemma 2 for the local prior term. Summing the above terms and minimizing over the spline weights of each pixel $\mathbf{w}_{\mathbf{q}}$ gives the MRF data term defined in Eq. (A32). □

A4.2. Optimization as robust estimation

Now we turn to minimizing $\mathrm{E}_{\text{DFD}} + \mathrm{E}_{\text{prior}}$ over the plane's segment labels $\mathbf{t}$, the patch-based depth map d and flow fields $\mathbf{v} = (\mathrm{u}, \mathrm{v})$ and the depth and flow parameters of planes in the spline model $\mathbf{D}, \mathbf{U}, \mathbf{V}$. We show below such an optimization problem corresponds to an robust estimation problem.

Proposition 3 *The optimization problem*

$$\min_{\substack{\mathrm{d,u,v}\\ \mathbf{D},\mathbf{U},\mathbf{V},\mathbf{t}}} \ \mathrm{E}_{DFD} + \mathrm{E}_{prior} \tag{A58}$$

is equivalent to the following problem

$$\min_{\substack{\mathrm{d,u,v}\\ \mathbf{D},\mathbf{U},\mathbf{V}}} g^{(0)}(\mathrm{d,u,v}) + g^{(1)}(\mathrm{d,u,v},\mathbf{D},\mathbf{U},\mathbf{V}) + g^{(2)}(\mathbf{D},\mathbf{U},\mathbf{V}) \tag{A59}$$

where the objective function of the optimization problem is composed of two truncated quadratic terms

$$g^{(0)}(\mathrm{d,u,v}) = \sum_{\mathbf{p}}\min(\mathrm{Q}_{\mathbf{p}}(\mathrm{d}_{\mathbf{p}},\mathrm{u}_{\mathbf{p}},\mathrm{v}_{\mathbf{p}}),\tau_i) \tag{A60}$$

$$g^{(2)}(\mathbf{D},\mathbf{U},\mathbf{V}) = \lambda_s\sum_{mn}\min\Big(\sum_{\mathbf{q}}(\mathbf{w}_{\mathbf{q}m}+\mathbf{w}_{\mathbf{q}n})\psi_{\mathbf{q}mn}(\mathbf{D},\mathbf{U},\mathbf{V}),\sum_{\mathbf{q}}(\mathbf{w}_{\mathbf{q}m}+\mathbf{w}_{\mathbf{q}n})\tau_s\Big). \tag{A61}$$

and a a quadratic term

$$\begin{aligned} g^{(1)}(\mathrm{d,u,v},\mathbf{D},\mathbf{U},\mathbf{V}) = {} & \frac{1}{2}\sigma_d^{-2}\left[\sum_{\mathbf{p}}\sum_{\mathbf{q}\in\Omega_f(\mathbf{p})}\Big(\sum_n \mathrm{w}_n\mathbf{D}_n\mathbf{q}-\mathrm{d}_{\mathbf{p}}\Big)^2 + \lambda_f\sum_{\mathbf{q}}\sum_{mn}\mathbf{w}_{\mathbf{q}m}\mathbf{w}_{\mathbf{q}n}(\mathbf{D}_m\mathbf{q}-\mathbf{D}_n\mathbf{q})^2\right] \\ & + \frac{1}{2}\sigma_v^{-2}\left[\sum_{\mathbf{p}}\sum_{\mathbf{q}\in\Omega_f(\mathbf{p})}\Big(\sum_n \mathrm{w}_n\mathbf{U}_n\mathbf{q}-\mathrm{u}_{\mathbf{p}}\Big)^2 + \lambda_f\sum_{\mathbf{q}}\sum_{mn}\mathbf{w}_{\mathbf{q}m}\mathbf{w}_{\mathbf{q}n}(\mathbf{U}_m\mathbf{q}-\mathbf{U}_n\mathbf{q})^2\right] \\ & + \frac{1}{2}\sigma_v^{-2}\left[\sum_{\mathbf{p}}\sum_{\mathbf{q}\in\Omega_f(\mathbf{p})}\Big(\sum_n \mathrm{w}_n\mathbf{V}_n\mathbf{q}-\mathrm{v}_{\mathbf{p}}\Big)^2 + \lambda_f\sum_{\mathbf{q}}\sum_{mn}\mathbf{w}_{\mathbf{q}m}\mathbf{w}_{\mathbf{q}n}(\mathbf{V}_m\mathbf{q}-\mathbf{V}_n\mathbf{q})^2\right] \end{aligned} \tag{A62}$$

Proof sketch of Proposition 3 The entropy term E_{ent}, the image coherence term E_{img}, and the discontinuity penalty term of the top-down likelihood $\mathrm{E}_{\text{prior}}$ does not depend on the optimized variables. In addition we note that $\mathbf{t}$ only affects E_{smo} Therefore the optimization problem defined in Eq. (A59)

$$\min_{\substack{\mathrm{d,u,v}\\ \mathbf{D},\mathbf{U},\mathbf{V}}}\left[\sum_{\mathbf{p}}\min(\mathrm{Q}_{\mathbf{p}}(\mathrm{d,u,v}),\tau_i) + \sum_{\mathbf{p}}\sum_{\mathbf{q}\in\Omega(\mathbf{p})}\mathrm{L}_{\mathbf{qp}}(\mathrm{d,u,v},\mathbf{D},\mathbf{U},\mathbf{V}) + \lambda_f\mathrm{E}_{\text{flat}}(\mathbf{D},\mathbf{U},\mathbf{V}) + \lambda_s\min_{\mathbf{t}}\mathrm{E}_{\text{smo}}(\mathbf{D},\mathbf{U},\mathbf{V})\right] \tag{A63}$$

```
Algorithm 1: Global DFD
  input  : initial control points $\mathbf{C}$, feature map $\mathbf{f}$, patch likelihoods $\mathrm{Q}$
  output: patch-based depth and flow $\mathcal{P} = (\mathrm{d}, \mathbf{v})$, spline parameters $\mathcal{M} = (\mathbf{C}, \mathbf{D}, \mathbf{U}, \mathbf{V}, \mathbf{w})$, scene segmentation $\mathcal{Z} = (\mathrm{s}, \mathbf{t}, \mathbf{O})$,
          pixel-based depth and flow $\mathcal{S}(\mathcal{M}) = (\mathrm{d}', \mathbf{v}')$
1 initialize $\mathcal{S} = (\mathbf{0}, \mathbf{0}), \mathbf{D} = \mathbf{U} = \mathbf{V} = 0,\ \mathbf{t} = \mathbf{0}$
2 repeat
3     update s and $\mathbf{w}$ jointly by solving MRF
4     update $\mathbf{O}$ by thresholding
5     update $\mathcal{P}$, $\mathbf{D}, \mathbf{U}, \mathbf{V}$ and $\mathbf{t}$ jointly by solving IRLS
6     update $\mathbf{C}_n = \sum_{\mathbf{p}}(\mathbf{w}_{\mathbf{p}n}\mathbf{f}_{\mathbf{p}}) / \sum_{\mathbf{p}} \mathbf{w}_{\mathbf{p}n}$
7 until convergence
8 compute $\mathcal{S}(\mathcal{M})$ from spline parameters $\mathcal{M}$ using Eq. (8).
```

The first term in Eq. (A63) $\sum_{\mathbf{p}} \min(\mathrm{Q}_{\mathbf{p}}, \tau_i)$ corresponds to $g^{(0)}$. The sum of the second term and the third term in Eq. (A63) equals

$$\sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} + \lambda_f \mathrm{E}_{\text{flat}}(\mathbf{D}, \mathbf{U}, \mathbf{V}) = g^{(1)}(\mathrm{d}, \mathrm{u}, \mathrm{v}, \mathbf{D}, \mathbf{U}, \mathbf{V}) + \sum_{\mathbf{p}} (81 - |\Omega_f(\mathbf{p})|\tau_o) \ . \tag{A64}$$

The second term on the right handside of Eq. A64 is contant to the optimized variables. Now we rewrite the last term in in Eq. (A63). Here we use an equivalent representation of $\mathbf{t}$: the set of binary indicators $\{\delta_{mn}\}$ that examines whether nearly planes belong to the same segment. Therefore

$$\min_{\mathbf{t}} \mathrm{E}_{\text{smo}}(\mathbf{D}, \mathbf{U}, \mathbf{V}) = \min_{\delta} \mathrm{E}_{\text{smo}}(\mathbf{D}, \mathbf{U}, \mathbf{V}) \ . \tag{A65}$$

We reorganize the right handside of the above equation noting that the contributions of the binary indicators δ_{mn} are independent of each other

$$\begin{aligned}
\lambda_s \min_{\delta} \mathrm{E}_{\text{smo}}(\mathbf{D}, \mathbf{U}, \mathbf{V}) &= \lambda_s \min_{\delta} \sum_{\mathbf{q}} \sum_{mn} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n})(\delta_{mn}\psi_{\mathbf{q}mn}(\mathbf{D}, \mathbf{U}, \mathbf{V}) + (1 - \delta_{mn}\tau_s)) \\
&= \lambda_s \min_{\delta} \sum_{mn} \left(\delta_{mn} \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \psi_{\mathbf{q}mn}(\mathbf{D}, \mathbf{U}, \mathbf{V}) + (1 - \delta_{mn}) \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \tau_s \right) \\
&= \lambda_s \sum_{mn} \min_{\delta_{mn}} \left(\delta_{mn} \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \psi_{\mathbf{q}mn}(\mathbf{D}, \mathbf{U}, \mathbf{V}) + (1 - \delta_{mn}) \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \tau_s \right) \\
&= g^{(2)}(\mathbf{D}, \mathbf{U}, \mathbf{V}) \ .
\end{aligned} \tag{A66}$$

Summing the three terms we prove the proposition. □

A4.3. Explanation of Algorithm 1

For the readers convenience we show the global DFD optimization algorithm first in Algorithm 1, and define its exact steps as follows.

Step 3: MRF minimization and weight update (Algorithm 2) We first compute a rough estimation of the front-most segment of each patch $\Omega(\mathbf{p})$ in two steps. First, for each control point n which affects a pixel $\mathbf{q} \in \Omega(\mathbf{p})$, we compute the weighted distance from the patch to the point by

$$\mathrm{dist}_{\mathbf{p}n} = \begin{cases} \sigma_d^{-2}(\mathrm{d}_{\mathbf{p}} - \mathbf{D}_n\mathbf{q})^2 + \sigma_v^{-2}[(\mathrm{u}_{\mathbf{p}} - \mathbf{U}_n\mathbf{q})^2 + (\mathrm{v}_{\mathbf{p}} - \mathbf{V}_n\mathbf{q})^2] & \text{if } \exists \mathbf{q} \in \Omega(\mathbf{p}) : \mathbf{w}_{\mathbf{q}n} > 0 \\ \infty & \text{otherwise} \end{cases} \ . \tag{A67}$$

Then we assign each patch-based segment label $s^*_{\mathbf{p}}$ with the segment label of the control point that minimizes $\mathrm{dist}_{\mathbf{p}n}$.

We then compute the optimal spline weight for each pixel $\mathbf{q}$ which solves the optimization problem defined in Eq. (A32):

$$\hat{\mathbf{w}}_{\mathbf{q}n}(i) = \begin{cases} \dfrac{\exp(-e^{(1)}_{\mathbf{q}n} - \sum_m \mathbf{w}_{\mathbf{q}m} e^{(2)}_{\mathbf{q}mn})}{\sum_{n:\mathbf{t}_n = i} \exp(-e^{(1)}_{\mathbf{q}n} - \sum_m \mathbf{w}_{\mathbf{q}m} e^{(2)}_{\mathbf{q}mn})} & \text{if } \mathbf{t}_n = i \\ 0 & \text{otherwise.} \end{cases} \tag{A68}$$

Algorithm 2: MRF minimization (Step 3 of Algorithm 1)

```
for each segment $i$ do
    for each pixel $\mathbf{q}$, compute $e^{(0)}_{\mathbf{q}}(i)$ according to Eq. (A36)
    repeat
        for each pixel $\mathbf{q}$, compute $e^{(1)}_{\mathbf{q}n}(i)$ and $e^{(2)}_{\mathbf{q}mn}(i)$ according to Eq. (A37) and (A38)
        compute $\hat{\mathbf{w}}_{\mathbf{p}}(i)$ according to Eq. (A68)
    until convergence
    compute data cost of MRF for segment label $i$ by Eq. (A32)
set smoothness cost of MRF with Eq. (A33)
compute $\mathrm{s}$ by solving the MRF in Eq. (A31)
update $\mathbf{w}_{\mathbf{q}} = \hat{\mathbf{w}}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}})$.
```

This defines the outer minimization of Eq. (A31) as an Markov random field problem and we solve it with an α-expansion algorithm in the second step.

Step 4: Updating the occlusion relationships The only term relevant to the occlusion relationships in the objective function is the local prior term $\sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}}$. The term can be further decomposed as a sum of contributions $\mathrm{L}_{\mathbf{qp}}$ due to boundaries between segment pairs (i, j):

$$\sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} \approx \sum_{i,j} \mathrm{L}_{ij}(\mathbf{O}_{ij}). \tag{A69}$$

Each term L_{ij} is a ternary function of occlusion relationship $\mathbf{O}_{ij}$ parameterized on the patch-based depth and flow, as well as the spline model

$$\begin{aligned}
\mathrm{L}_{ij}(\mathbf{O}_{ij}) &= \sum_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \mathrm{L}_{\mathbf{qp}} \\
&= \begin{cases}
\sum\limits_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum\limits_{\substack{\mathbf{q}\in\Omega(\mathbf{p})\\ \mathrm{s}_{\mathbf{q}}=i}} [\frac{1}{2\sigma_d^2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 + \frac{1}{2\sigma_v^2}|\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2] + \sum\limits_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum\limits_{\substack{\mathbf{q}\in\Omega(\mathbf{p})\\ \mathrm{s}_{\mathbf{q}}=j}} \tau_o & \text{if } \mathbf{O}_{ij} = 1 \\
\sum\limits_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum\limits_{\mathbf{q}\in\Omega(\mathbf{p})} [\frac{1}{2\sigma_d^2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 + \frac{1}{2\sigma_v^2}|\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2] & \text{if } \mathbf{O}_{ij} = 0 \\
\sum\limits_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum\limits_{\substack{\mathbf{q}\in\Omega(\mathbf{p})\\ \mathrm{s}_{\mathbf{q}}=j}} [\frac{1}{2\sigma_d^2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}'_{\mathbf{q}})^2 + \frac{1}{2\sigma_v^2}|\mathbf{v}_{\mathbf{p}} - \mathbf{v}'_{\mathbf{q}}|^2] + \sum\limits_{\mathbf{p}:\Omega(\mathbf{p})\in\mathcal{B}_{ij}} \sum\limits_{\substack{\mathbf{q}\in\Omega(\mathbf{p})\\ \mathrm{s}_{\mathbf{q}}=i}} \tau_o & \text{if } \mathbf{O}_{ij} = -1
\end{cases}
\end{aligned} \tag{A70}$$

Therefore, we simply update $\mathbf{O}_{ij}$ which minimizes L_{ij}.

Step 5: Robust estimation (Algorithm 3)

We first solve the robust estimation problem in Eq. (A58) with the iterative reweighted least squares algorithm shown in Algorithm 3.

Specifically we use the following alternative expressions for the two truncated quadratic terms of Eq. (A59)

$$g^{(0)}(\mathrm{d}, \mathrm{u}, \mathrm{v}) = \sum_{\mathbf{p}} \min_{\mathrm{y}_{\mathbf{p}}}(\mathrm{y}_{\mathbf{p}} \mathrm{Q}_{\mathbf{p}}(\mathrm{d}_{\mathbf{p}}, \mathrm{u}_{\mathbf{p}}, \mathrm{v}_{\mathbf{p}}) + (1 - \mathrm{y}_{\mathbf{p}})\tau_i) \tag{A71}$$

$$g^{(2)}(\mathbf{D}, \mathbf{U}, \mathbf{V}) = \lambda_s \sum_{mn} \min_{\delta_{mn}} [\delta_{mn} \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \psi_{\mathbf{q}mn}(\mathbf{D}, \mathbf{U}, \mathbf{V}) + (1 - \delta_{mn}) \sum_{\mathbf{q}} (\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \tau_s]. \tag{A72}$$

The auxilary variable y is introduced to identify inlier patches from patches that cannot be explained by a single depth and translational flow model.

Thus we can solve the robust estimation problem with three iterative steps:

Algorithm 3: Robust estimation (Step 5 of Algorithm 1)

```
initialize all elements of $\mathrm{y}$ and $\delta$ to one
repeat
    update $\mathrm{d}$ and $\mathbf{D}$ by solving the quadratic minimization problem in Eq. (A76)
    update $\mathbf{v} = (\mathrm{u}, \mathrm{v})$ and $\mathbf{U}, \mathbf{V}$ by solving the quadratic minimization problem in Eq. (A77)
    update $\mathrm{y}_{\mathbf{p}}$ for each pixel $\mathbf{p}$ by Eq. (A73)
    update $\delta_{mn}$ for each pair of control points $(m, n)$ by Eq. (A74).
until convergence
compute $\mathbf{t}$ from $\delta_{mn}$ by finding connected component (see text);
```

1. Update y and δ by

$$\mathrm{y}_{\mathbf{p}} = \begin{cases} 1 & \text{if } \mathrm{Q}_{\mathbf{p}}(\mathrm{d}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}}) < \tau_i \\ 0 & \text{otherwise} \end{cases} , \tag{A73}$$

$$\delta_{mn} = \begin{cases} 1 & \text{if the distance between the initial control points m, n are less than 10\% of the image diagonal} \\ & \text{and } \sum_{\mathbf{p}}(\mathbf{w}_{\mathbf{p}m} + \mathbf{w}_{\mathbf{p}n})\varphi_{\mathbf{p}mn} < \sum_{\mathbf{p}}(\mathbf{w}_{\mathbf{p}m} + \mathbf{w}_{\mathbf{p}n})\tau_{\mathrm{s}} \\ 0 & \text{otherwise} \end{cases} ; \tag{A74}$$

2. Let

$$\omega_{\mathbf{q}mn} = \mathbf{w}_{\mathbf{q}m}\mathbf{w}_{\mathbf{q}n} + \delta_{mn}(\mathbf{w}_{\mathbf{q}m} + \mathbf{w}_{\mathbf{q}n}) \ , \tag{A75}$$

update d, $\mathbf{D}$ by solving the quadratic minimization problem

$$\min_{\mathrm{d},\mathbf{D}} \left[\sum_{\mathbf{p}} \mathrm{y}_{\mathbf{p}}\sigma_{\mathbf{p}}^{-2}(\mathrm{d}_{\mathbf{p}} - \mathrm{d}_{\mathbf{p}}^*)^2 + \frac{1}{2}\sigma_d^{-2} \left(\sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega_f(\mathbf{p})} (\mathrm{d}_{\mathbf{p}} - \sum_n \mathbf{w}_n \mathbf{D}_n \mathbf{q})^2 + \sum_{\mathbf{q}} \sum_{mn} \omega_{\mathbf{q}mn}(\mathbf{D}_m\mathbf{q} - \mathbf{D}_n\mathbf{q})^2 \right) \right] ; \tag{A76}$$

3. Update $\mathbf{v} = (\mathrm{u}, \mathrm{v})$ and $\mathbf{U}, \mathbf{V}$ by solving the quadratic minimization problem

$$\min_{\mathrm{u},\mathrm{v},\mathbf{U},\mathbf{V}} \begin{array}{l} \sum_{\mathbf{p}} (\mathbf{v}_{\mathbf{p}} - \mathbf{v}_{\mathbf{p}}^*)^{\mathrm{T}} \mathrm{y}_{\mathbf{p}} \Sigma_{\mathbf{p}}^{-2} (\mathbf{v}_{\mathbf{p}} - \mathbf{v}_{\mathbf{p}}^*) \\ + \frac{1}{2}\sigma_v^{-2} \sum_{\mathbf{p}} \sum_{\mathbf{q}\in\Omega_f(\mathbf{p})} \left[(\mathrm{u}_{\mathbf{p}} - \sum_n \mathbf{w}_n \mathbf{U}_n \mathbf{q})^2 + (\mathrm{v}_{\mathbf{p}} - \sum_n \mathbf{w}_n \mathbf{U}_n \mathbf{q})^2 \right] \\ + \frac{1}{2}\sigma_v^{-2} \sum_{\mathbf{q}} \sum_{mn} \omega_{\mathbf{q}mn} \left[(\mathbf{U}_m\mathbf{q} - \mathbf{U}_n\mathbf{q})^2 + (\mathbf{V}_m\mathbf{q} - \mathbf{V}_n\mathbf{q})^2 \right] \end{array} . \tag{A77}$$

Step 2 and step 3 of the iteration requires solving a sparse linear system of up to $3N + P$ unknowns and $6N + 2P$ unknowns, respectively, where N is the number of control points in the spline model and P is the number of pixels in the image.

Once the iteration converges, we obtain a set of binary labels δ_{mn}, which allows us to compute the planes' segment labels $\mathbf{t}$. To do so, we first construct a graph with each node corresponding to a control point, and connect node m and n with an edge if $\delta_{mn} = 1$. Then we compute connected components of the graph, and thus assign segment labels of the control points as the connected component label of the corresponding node in the graph.

Step 6: Updating feature vector of control points Finally we optimize the objective function over the feature vector of control points $\mathbf{C}$. Since only the coherence term E_{img} involves this variable, we can update it by minimizing E_{img} over $\mathbf{C}$. This can be computed analytically by

$$\mathbf{C}_n = \arg\min_{\mathbf{C}_n} \mathrm{E}_{\text{img}} = \arg\min_{\mathbf{C}_n} \sum_{\mathbf{q}} \mathbf{w}_{\mathbf{q}n} |\mathbf{C}_n - \mathbf{f}_{\mathbf{q}}|^2 = \frac{\sum_{\mathbf{q}}(\mathbf{w}_{\mathbf{q}n}\mathbf{f}_{\mathbf{q}})}{\sum_{\mathbf{q}} \mathbf{w}_{\mathbf{q}n}} . \tag{A78}$$

A5. Implementation

A5.1. Camera Calibration Procedure

For each camera in Table A1 (except Samsung) we follow standard camera calibration procedure to compute the magnification parameters m_{12} and blur parameters

$$r_n(0) = A\left(1 - \frac{f_n}{F}\right) \qquad \Delta r_n = A f_n. \tag{A79}$$

Since r_n is a function of depth, which is the inverse of the object distance, $r_1(0)$ denotes the defocus radius when the object is at infinity.

In particular, we capture a checkerboard pattern at 5 different depths using the various focus settings. We extract corners from the captured images to robustly estimate the relative magnification among the focus settings. This also yields an estimation of the hidden image, from which we calculate the blur kernel radii at each depth by minimizing the image reconstruction error. Then, we compute $r_n(0)$ and Δr_n for each focus setting f_n by robustly fitting a linear model to the measured defocus radii, according to Eq. (A1). Finally, we estimate the circular aperture radius, focal length and all focus settings jointly by minimizing the reconstruction error in the inlier defocus radii.

A5.2. Fitting $\mathrm{Q}_\mathbf{p}$ (Eq. 5)

Now we discuss the fitting algorithm to compute the quadratic approximation $\mathrm{Q}_\mathbf{p}(d, \boldsymbol{v})$ in Eq. (5) of $-\log\Pr(\mathrm{d}_\mathbf{p} = d, \mathbf{v}_\mathbf{p} = \boldsymbol{v} \mid \mathrm{i}_1, \mathrm{i}_2)$.

Initial optical flow. In principle ANN flow estimation method can be used to initialize flow between defocus equalized images. Here we compute an initial flow between the two input images regardless of their defocus blur, with the assumption that the initial flow is close to the actual flow. We use Algorithm 4, a multi-scale method that propagates reliable, local flow estimates, since it does not produce regions of coherent yet erroneous flow estimation. It is particularly similar to the recent work of Fields Flow [3] except for a few minor differences: (1) we initialize flow estimation at coarsest level with Coherent Sensitive Hashing (CSH) [5] rather than KD-tree; (2) we use SSD error (with relative illumination accounted) as the "data cost" function (3) rather than allowing flow of subpixel accuracy, we only consider flow of integer values to restrict the search space; (4) rather than randomly disturbing the flow estimation in the random search step, we exhaustively choose among the 9 flow values surrounding the current flow estimation and (5) we reject outliers only when forward-backward consistency check fails.

Sampling depth-flow hypotheses. We uniformly sample 32 depth values within the operating depth range and flow values within 3 pixels from the initial flow estimated in the last step. For each depth-flow pair $(d_k, \boldsymbol{v}_k)$, we evaluate its likelihood $\Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2)$ according to Eq. (4).

Parameter estimation. The parameters of $\mathrm{Q}_\mathbf{p}(d, \boldsymbol{v})$ are estimated with

$$\mathrm{d}^*_\mathbf{p}, \mathbf{v}^*_\mathbf{p} = \arg\max\nolimits_{d_k, \boldsymbol{v}_k} \Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2) \tag{A80}$$

$$\mathrm{q}_\mathbf{p} = -\log\Pr(\mathrm{d}_\mathbf{p} = \mathrm{d}^*_\mathbf{p}, \mathbf{v}_\mathbf{p} = \mathbf{v}^*_\mathbf{p} | \mathrm{i}_1, \mathrm{i}_2) \tag{A81}$$

$$\sigma^2_\mathbf{p} = \frac{\sum_k \Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2)(d_k - \mathrm{d}^*_\mathbf{p})^2}{\sum_k \Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2)} \tag{A82}$$

$$\mathbf{\Sigma}_\mathbf{p} = \frac{\sum_k \Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2)(\boldsymbol{v}_k - \mathbf{v}^*_\mathbf{p})^\mathrm{T}(\boldsymbol{v}_k - \mathbf{v}^*_\mathbf{p})}{\sum_k \Pr(\mathrm{d}_\mathbf{p} = d_k, \mathbf{v}_\mathbf{p} = \boldsymbol{v}_k \mid \mathrm{i}_1, \mathrm{i}_2)}, \tag{A83}$$

where $\mathbf{\Sigma}_\mathbf{p}$ are 2×2 covariance matrices since $\boldsymbol{v}_k$ are 1×2 row vectors.

A5.3. Image features

In our implementation we define the image feature $\mathbf{f}$ as a 35-dimensional vector map. This is not a critical choice to our method, and may be replaced or augmented by other features, *e.g.* segmentation features utilizing the recent convolutional neural networks based boundary detection method [6].

Algorithm 4: Initial flow estimation

```
input     : two images $\mathrm{i}_1, \mathrm{i}_2$
output    : optical flow map $\mathbf{v}$ that warps from $\mathrm{i}_1$ to $\mathrm{i}_2$
parameter: number of scales $S$
$\mathrm{i}_1^0 = \mathrm{downsample}(\mathrm{i}_1, 2^{-S})$, $\mathrm{i}_2^0 = \mathrm{downsample}(\mathrm{i}_2, 2^{-S})$
compute $\mathbf{v}_F$, $\mathbf{v}_B$ as the forward and backward flow between $\mathrm{i}_1^0$ and $\mathrm{i}_2^0$ using CSH
compute inlier mask $m$ by forward-backward consistency check with a threshold of 3 pixels
for $s = 1 \ldots S$ do
    $\mathrm{i}_1^s = \mathrm{downsample}(\mathrm{i}_1, 2^{s-S})$, $\mathrm{i}_2^s = \mathrm{downsample}(\mathrm{i}_2, 2^{s-S})$
    upsample $\mathbf{v}_F$, $\mathbf{v}_B$ and $m$ with a factor of 2 by nearest-neighbor interpolation
    update $\mathbf{v}_F, \mathbf{v}_B$ as the forward and backward flow between $\mathrm{i}_1^s$ and $\mathrm{i}_2^s$ by 4 propagations and 3 local searches (see [3]
    for the propagation algorithm)
    update $m$ by forward-backward consistency check with a threshold of 3 pixels
$\mathbf{v} = \mathbf{v}_F$
```

The features in our implementation includes the following:

1. **Pixel locations (2D)** We divide the pixels' 2D spatial coordinates by the image diagonal so that the pixel location features are invariant to image size.

2. **Pixel colors (3D)** We convert the input image to the CIE-LAB color space, and normalize each channel so that the standard deviation in each channel is 0.1.

3. **Spectral clustering eigenvectors (30D)** We first compute the bilateral affinity matrix $\mathbf{S}$ where the affinity between pixels $\mathbf{p}$ and $\mathbf{q}$ is defined as

$$\mathbf{S}_{\mathbf{pq}} = \begin{cases} \exp(-\Delta_{\mathbf{pq}}) + \exp(-\Delta_{\mathbf{qp}}) & \text{if } |\mathbf{p} - \mathbf{q}| < 3\sigma_s \\ 0 & \text{otherwise} \end{cases} \tag{A84}$$

$$\Delta_{\mathbf{pq}} = \min(\frac{|\mathrm{l}_{\mathbf{p}} - \mathrm{l}_{\mathbf{q}}|^2}{2\sigma^2_{\mathbf{p}l}} + \frac{|\mathrm{a}_{\mathbf{p}} - \mathrm{a}_{\mathbf{q}}|^2}{2\sigma^2_{\mathbf{p}a}} + \frac{|\mathrm{b}_{\mathbf{p}} - \mathrm{b}_{\mathbf{q}}|^2}{2\sigma^2_{\mathbf{p}b}}, \epsilon) + \frac{|\mathbf{p} - \mathbf{q}|^2}{2\sigma_s^2} \tag{A85}$$

where l, a, b are the three channels in the LAB image. Then we compute the eigenvectors $\boldsymbol{f}$ of the laplacian matrix of $\mathbf{S}$

$$\lambda \mathbf{z} = \underbrace{(\mathrm{Diag}(\mathbf{S1}) - \mathbf{S})}_{\text{Laplacian matrix of } \mathbf{S}} \mathbf{z}. \tag{A86}$$

Each eigenvector $\mathbf{z}$ is associated with an eigenvalue λ. We drop the smallest eigenvector (corresponding to $\lambda = 0$) as it is all one, and take the next 30 smallest eigenvectors. We normalize each eigenvector image so that its standard deviation is $1/30$.

Since it is impractical to compute spectral clustering eigenvectors on megapixel-sized images, we exploit two work-arounds to approximately compute them (see Figure A5):

Joint bilateral upsampling. Since even constructing the affinity matrix becomes prohibitively inefficient for large images, we first downsample our input image to a smaller size and compute the affinity matrix and the eigenvectors at this coarse scale. Then we upsample the eigenvector images to the original size by nearest-neighbor interpolation, and then refine them by performing joint bilateral filtering [10] with the full-resolution input image as guidance.

Multi-scale spectral analysis. At the downsampled scale, we further reduce the computational cost of eigenvectors by using the multi-scale DNCuts algorithm in [2].

Parameters. We downsample the image to a size whose diagonal is approximately 500 pixels. The downsampling ratio is $1/4$, $1/6$, $1/2$, $2/3$ for the Nexus, Canon, Middlebury and Samsung dataset respectively. For computing the affinities, we set the spatial standard deviation σ_s to be $1/125$ of image diagonal, and the color variance $\sigma^2_{\mathbf{p}l}$, $\sigma^2_{\mathbf{p}a}$,$\sigma^2_{\mathbf{p}b}$ to be $16\times$ the variance within the neighborhood of $\mathbf{p}$, and $\epsilon = 16$. For bilateral filtering, we set the color variance to 0.05 and spatial variance to $3\times$ the downsampling rate. We use the default parameters for the DNCuts algorithm (*i.e.* three scales, decimation level two).

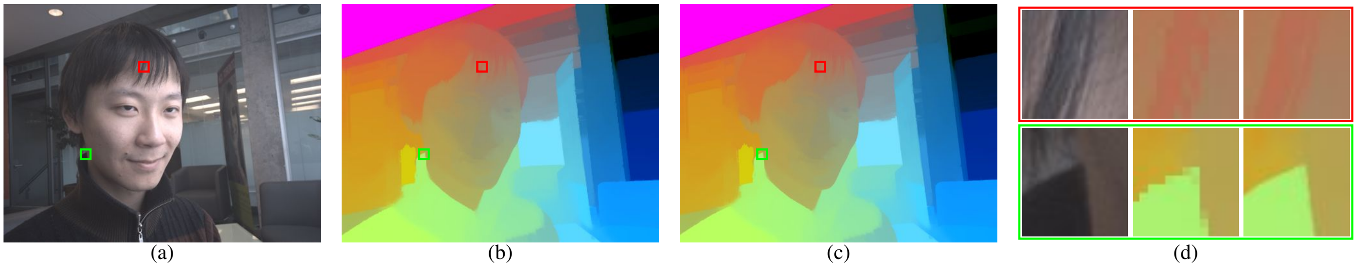

Figure A5: (a) Input image 1. (b) Three smallest eigenvector images (excluding the all one vector) by DNcuts on downsampled image. In the eigenvector images, each of the R, G, and B channel stores a single eigenvector. (c) Upsampled three smallest eigenvector images ($4\times$ the result of DNcuts). (d) Closeups in two windows. Notice that the bilateral upsampling ensures coherence between the feature map and image details.

References

[1] Available at http://www.dgp.toronto.edu/WildDFD. *Depth from Defocus in the Wild: Project webpage. [Online] (2017).* 2

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. IEEE CVPR*, 2014. 18

[3] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proc. IEEE ICCV*, 2015. 16, 17

[4] A. Chakrabarti, Y. Xiong, S. J. Gortler, and T. Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *Proc. IEEE CVPR*, 2015. 1, 2, 3, 4

[5] S. Korman and S. Avidan. Coherency sensitive hashing. In *Proc. IEEE ICCV*, 2011. 16

[6] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries. In *Proc. ECCV*, 2016. 17

[7] Y. Schechner and N. Kiryati. The optimal axial interval in estimating depth from defocus. In *Proc. IEEE CVPR*, 1999. 7

[8] S. Suwajanakorn, C. Hernández, and S. M. Seitz. Depth from focus with your mobile phone. In *Proc. IEEE CVPR*, pages 3497–3506, 2015. 1, 3

[9] H. Tang, S. Cohen, B. Price, S. Schiller, and K. N. Kutulakos. Depth from defocus in the wild. In *Proc. IEEE CVPR*, 2017. 1, 5, 6, 8

[10] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE ICCV*, pages 839–846, 1998. 17

[11] J. Yao, M. Boben, S. Fidler, and R. Urtasun. Real-Time Coarse-to-fine Topologically Preserving Segmentation. In *Proc. IEEE CVPR*, 2015. 1, 2, 3, 4