Supplementary document

Kevin C. Zhou, Colin Cooke, Jaehee Park, Ruobing Qian,
Roarke Horstmeyer, Joseph A. Izatt, Sina Farsiu

1 Homographic rectification

Here, we describe the procedure for 2D backprojection of a point in the camera image plane in the camera's intrinsic coordinate system relative to the projection center, $\mathbf{r}_{\mathbf{im}}^{\mathbf{cam}} = (x_{im}^{cam}, y_{im}^{cam}, f_{ph})$, onto the object plane in the world reference frame. This procedure ignores 3D height variation, which is accounted for in a separate step (see main text). For clarity, *cam* superscripts indicate that the variables are defined in the camera's reference frame, and *obj* and *im* subscripts reference the object and image planes, respectively.

1. Rotation in the image plane by θ:

$$(x_{im}^{cam}, y_{im}^{cam}) \leftarrow (x_{im}^{cam}, y_{im}^{cam}) R(\theta), \tag{1}$$

   where $R(\theta)$ is a 2D rotation matrix. Let $\mathbf{r}_{\mathbf{im}}^{\mathbf{cam}}$ be accordingly updated.

2. Backprojection to the object plane:

$$\mathbf{r}_{\mathbf{obj}}^{\mathbf{cam}} = \frac{Z}{\hat{\mathbf{n}}_{\mathbf{obj}}^{\mathbf{cam}} \cdot \mathbf{r}_{\mathbf{im}}^{\mathbf{cam}}} \mathbf{r}_{\mathbf{im}}^{\mathbf{cam}}, \tag{2}$$

   where $\cdot$ denotes a dot product and $\hat{\mathbf{n}}_{\mathbf{obj}}^{\mathbf{cam}} = (n_x, n_y, n_z)$ is the unit normal vector of the object plane defined in the camera's reference frame. For example, $\hat{\mathbf{n}}_{\mathbf{obj}}^{\mathbf{cam}} = (0, 0, -1)$ when the image and object planes are parallel.

3. Coordinate change via 3D rotation from camera coordinates, $\mathbf{r}_{\mathbf{obj}}^{\mathbf{cam}}$, to world coordinates, $\mathbf{r}_{\mathbf{obj}} = (x_{obj}, y_{obj})$ (i.e., by angle, $\cos^{-1}(-n_z)$, about axis, $(-n_y, n_x, 0)$):

$$\begin{aligned} x_{obj} &= \frac{Z(1+n_z)\left((n_y^2 + n_z - 1)x_{im} - n_x n_y y_{im}\right)}{n_z(n_x^2 + n_y^2)\hat{\mathbf{n}}_{\mathbf{obj}}^{\mathbf{cam}} \cdot \mathbf{r}_{\mathbf{im}}^{\mathbf{cam}}}, \\ y_{obj} &= \frac{Z(1+n_z)\left((n_x^2 + n_z - 1)y_{im} - n_x n_y x_{im}\right)}{n_z(n_x^2 + n_y^2)\hat{\mathbf{n}}_{\mathbf{obj}}^{\mathbf{cam}} \cdot \mathbf{r}_{\mathbf{im}}^{\mathbf{cam}}}. \end{aligned} \tag{3}$$

   In practice, this equation is numerically unstable, as it involves dividing $1 + n_z$ by $n_x^2 + n_y^2$, which are both 0 when the image and object planes are parallel. We instead use its second-order Taylor expansion at $n_x =$

$0, n_y = 0$, which is valid by our lateral-translation-dominant assumption ($|n_x|, |n_y| \ll |n_z| \approx 1$).

$$x_{obj} \approx \frac{Z}{f_{ph} n_z}\left(x_{im} + \frac{x_{im}(n_x x_{im} + n_y y_{im})}{f_{ph}} + \frac{f_{ph}^2 n_x (n_x x_{im} + n_y y_{im}) + 2x_{im}(n_x x_{im} + n_y y_{im})^2}{2f_{ph}^2}\right), \quad (4)$$

$$y_{obj} \approx \frac{Z}{f_{ph} n_z}\left(y_{im} + \frac{y_{im}(n_x x_{im} + n_y y_{im})}{f_{ph}} + \frac{f_{ph}^2 n_x (n_x x_{im} + n_y y_{im}) + 2y_{im}(n_x x_{im} + n_y y_{im})^2}{2f_{ph}^2}\right). \quad (5)$$

Note that the zero-order terms correspond to the usual camera-centric perspective projection expressions.

4. Addition of camera lateral position, $\mathbf{R} = (X, Y)$:

$$x_{obj} \leftarrow x_{obj} + X, \; y_{obj} \leftarrow y_{obj} + Y. \quad (6)$$

This backprojection procedure onto a common object plane is done for each camera image.

2 Rectification to an arbitrary perspective

Orthorectification is just a special case of a more general rectification procedure (Fig. 2d in the main text), where we instead warp to an arbitrary camera-centric reference frame, specified by its vanishing point, $\mathbf{R_{ref}}$, projection center height, Z_{ref}, and an orientation such that the image and object planes are parallel. Given the i^{th} camera image, whose extrinsics are similarly specified by $\mathbf{R_i}$, Z_i, and $\hat{\mathbf{n}}_{\mathbf{im,i}}$, the vector that warps a point, $\mathbf{r_{obj}}$, to the reference frame is given by

$$\mathbf{r_{rectify}}(\mathbf{r_{obj}}) = \frac{h}{Z_i}\frac{Z_i - Z_{ref}}{Z_{ref} - h}(\mathbf{r_{obj}} - \mathbf{R_i}) + \frac{h}{Z_{ref} - h}(\mathbf{R_i} - \mathbf{R_{ref}}). \quad (7)$$

If we allow $Z_{ref} \to \infty$, the result is consistent with Eqs. 4 and 5 in the main text, and we recover the orthorectification case, as expected.

3 Effect of ignoring thin lens equation

Conventional large-scale 3D computer vision assumes focusing at infinity so that $f_{ph} \approx f_{eff}$. If we make this assumption for our close-range, mesoscopic application, our height estimates become biased. To appreciate the magnitude of this bias, we combine Eqs. 5 and 6 from the main text to obtain

$$h_i^{biased} = -f_{ph} \Delta r_i |\mathbf{r_{obj}} - \mathbf{R_i}|^{-1} M_i^{-1}. \quad (8)$$

Sample	Vert.	Horiz.
Cut cards	17.5	-9.6
PCB	20.2	-20.0
Helicopter	19.8	-14.6
Painting	17.0	-15.5
Tuning sample	22.3	-17.9

Table 1: Estimates of the camera principal point position relative to the center (pixels).

This equation differs from Eq. 8 from the main text by an extra term that is negligible when objects are far away ($M_i \to 0$), but significant at close range. For our samples, a typical value for M_i was 0.08, which gives a $\sim$7.4% error. The closer the range (the larger M_i), the larger this error.

4 COLMAP [1] hyperparameters

We used the OpenCV fisheye camera model, shared among all images in the same sequences, with the focal length supplied via EXIF and the principal point (i.e., the projection center position) optimizable, as in our method. Exhaustive feature matching and guided matching were enabled, and all image-size-related settings were set so that COLMAP maintained the 1512×2016 input size. Finally, among Delaunay meshing hyperparameters, we set max_proj_dist=1 and quality_regularization=2. The camera model and meshing hyperparameters were tuned using the same independent sample we used to select the CNN architecture. For all other hyperparameters, we used the defaults.

5 Camera undistortion estimates

As mentioned in the main text, it is very important to correct for camera distortions, which are position-dependent relative magnifications. Fig. 1 shows the radial camera undistortion estimates for each sample under the piecewise linear model. Since there can be an arbitrary constant global magnification, in our implementation we arbitrarily normalized distortions to their maximum values. This procedure does not constrain our results, as an error in global magnification can be compensated by a global height shift, and vice versa. We also allowed the camera principal point (i.e., the projection center position) to vary; the results are shown in Table 1 for all four samples in the main text plus the hyperparameter tuning sample. Note that the units are in pixels *after* we downsampled the original camera images by $2\times$.

Overall, the results are qualitatively consistent across different samples, with small deviations due to reconstruction error and/or changing distortion properties as a function of sensor or lens repositioning for autofocus.

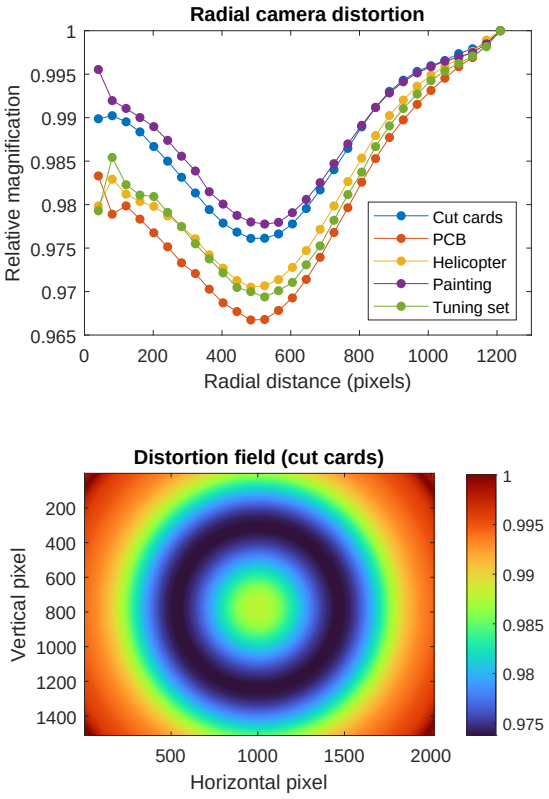

Figure 1: Top: radial undistortion for each sample. Bottom: sample 2D undistortion profile estimated using the cut cards sample. The 2D undistortion profiles estimated from the other samples are similar.

Downsample block	Upsample block
	$2\times$ bilinear upsample
3×3 conv, k filters, stride=2	3×3 conv, k filters, stride=1
Batch normalization	Batch normalization
Leaky ReLU	Leaky ReLU
3×3 conv, k filters, stride=1	1×1 conv, k filters, stride=1
Batch normalization	Batch normalization
Leaky ReLU	Leaky ReLU

Table 2: Upsample and downsample blocks used in our encoder-decoder architectures. Convolutions use 'same' padding mode.

6 Full comparison of undistortion models

Here, we show the full comparison of height map reconstructions on all four samples using a 4th, 16th, 32nd and 64th order even polynomials (2, 8, 16, and 32 coefficients, respectively) as well as our piecewise linear radial undistortion model (with 30 line segments). These results are shown in Figs. 2-5. The second rows of these figures have saturated color ranges to better appreciate the distortion-induced artifacts, which typically manifest as rings and affect the polynomial models, but not the piecewise linear model.

7 CNN architectures

We use slightly modified versions of the encoder-decoder CNNs (no skip connections) from the original DIP paper [2]. Specifically, we composed CNNs using the downsample and upsample blocks summarized in Table 2. We designed different symmetric architectures by varying the number of upsample/downsample pairs and number of filters, k, per block. For example, the architecture specified by the list, $[16, 32, 64]$, consists of three sequential downsample blocks with $k = 16, 32,$ and 64 filters, respectively, followed by three sequential upsample blocks with $k = 64, 32,$ and 16 filters. For all reconstructions in the main text, we used $[16, 16, 16, 32, 32]$, which we arrived at by comparing reconstruction quality on an independent sample (see next section).

8 Full comparison of CNN-based and TV regularization

Here, we show height map reconstructions for all four samples in the main text, as well as the independent hyperparameter-tuning sample, using four different CNN architectures:

- Architecture 1: $[16, 16, 32, 32]$, 69,424 parameters
- Architecture 2: $[16, 16, 16, 16]$, 28,080 parameters

- Architecture 3: $[16, 16, 16, 32, 32]$, 76,912 parameters (used in main text)
- Architecture 4: $[16, 16, 16, 16, 16]$, 35,568 parameters

The motivation for these choices was to test two different methods of compression in the CNN – using fewer parameters and using more downsampling blocks. Note that in all cases, the number of parameters is significantly fewer than the number of camera-centric height map pixels (60-70 million, depending on number of images in the sequence). We chose architecture 3 for all the reconstruction figures in the main text (unless otherwise specified), because it balances loss of resolution and reduction of artifacts in the background of the tuning sample.

We also show five different levels of isotropic total variation (TV) regularization,

$$TV(h(x,y)) = \sum_{x,y} \sqrt{|\nabla_x h(x,y)|^2 + |\nabla_y h(x,y)|^2}, \tag{9}$$

where the directional gradients are approximated by finite differences. The total loss is therefore MSE + λTV, where we used $\lambda = 0.003, 0.01, 0.03, 0.1,$ and 0.3.

The comparisons are shown in Figs. 6-10. All height reconstructions within the same figure share the same color range. In general, the compression-based CNN regularization has more desirable properties as the regularization strength is tuned, such as flat backgrounds regardless of regularization strength.

9 Effect of number of iterations

Optimizing for 10000 iterations, as we did for all CNN-regularized reconstructions for consistency, may be conservative in some cases, depending on the sample. Fig. 11 shows height map reconstruction results for all four main samples at 500, 1000, 3000, and 10000 iterations. In particular, a few 1000 iterations may be sufficient for the cut cards and PCB sample, while insufficient for the helicopter and painting sample. In fact, for the cut cards sample, running for far fewer iterations may be beneficial to avoid artifacts both in the background and card surfaces. Indeed, early stopping is a well known strategy to avoid overfitting in neural networks, and DIP is no exception [2].

References

[1] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[2] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

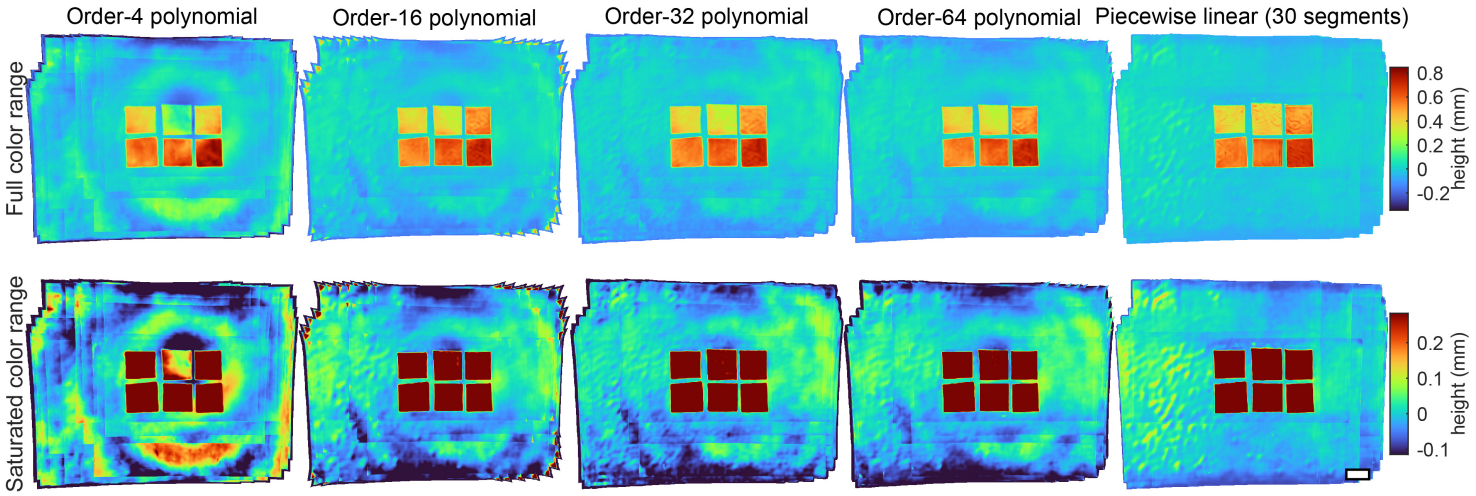

Figure 2: Comparison of height map reconstructions of the cut cards sample under various undistortion models. The second row uses a saturated color range to emphasize artifacts. Scale bar, 1 cm.

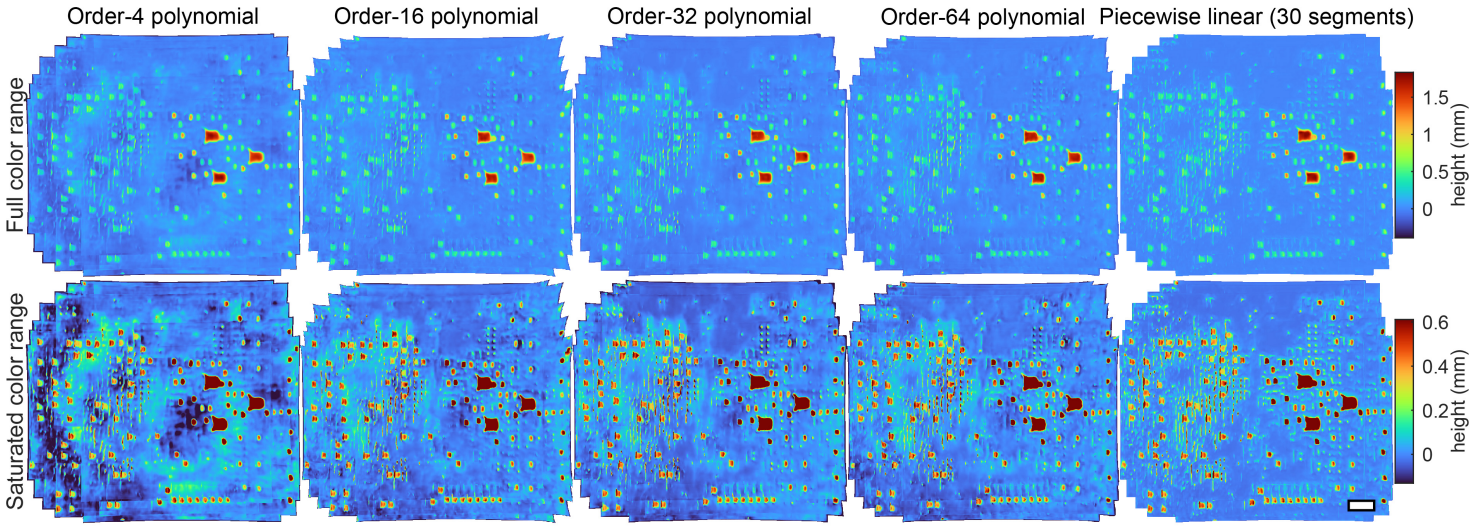

Figure 3: Comparison of height map reconstructions of the PCB sample under various undistortion models. The second row uses a saturated color range to emphasize artifacts. Scale bar, 1 cm.

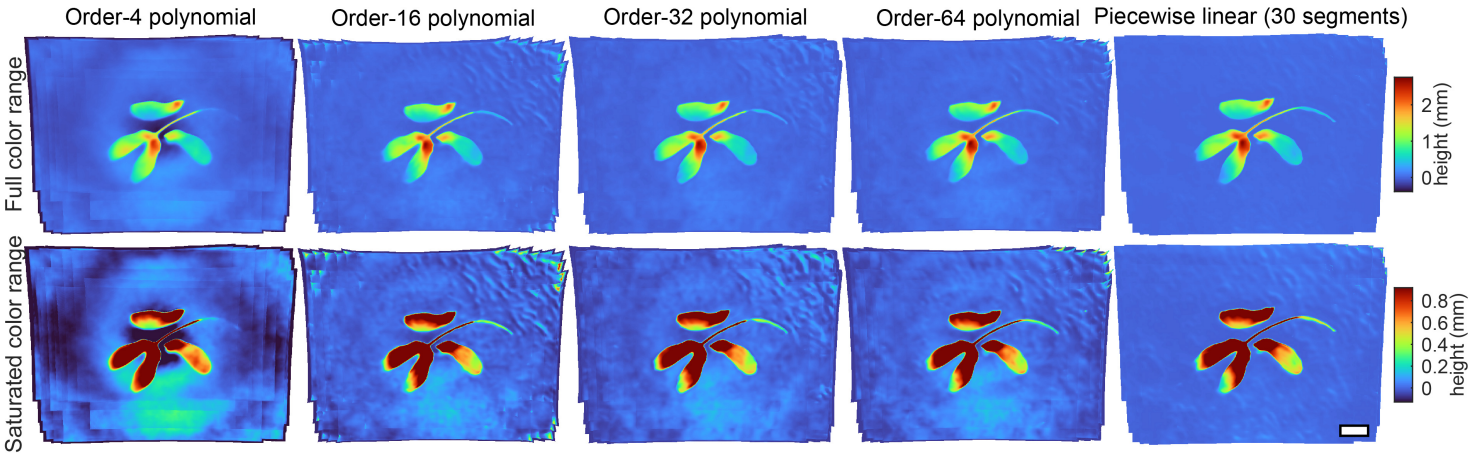

Figure 4: Comparison of height map reconstructions of the helicopter seeds sample under various undistortion models. The second row uses a saturated color range to emphasize artifacts. Scale bar, 1 cm.

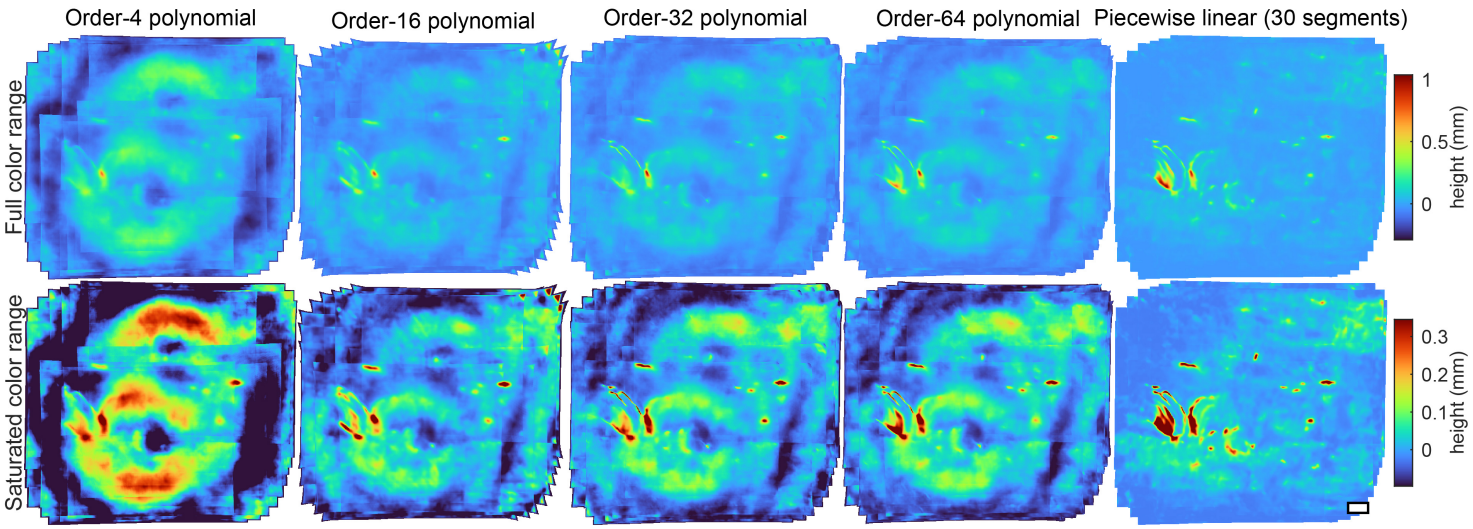

Figure 5: Comparison of height map reconstructions of the painting brush strokes sample under various undistortion models. The second row uses a saturated color range to emphasize artifacts. Scale bar, 1 cm.

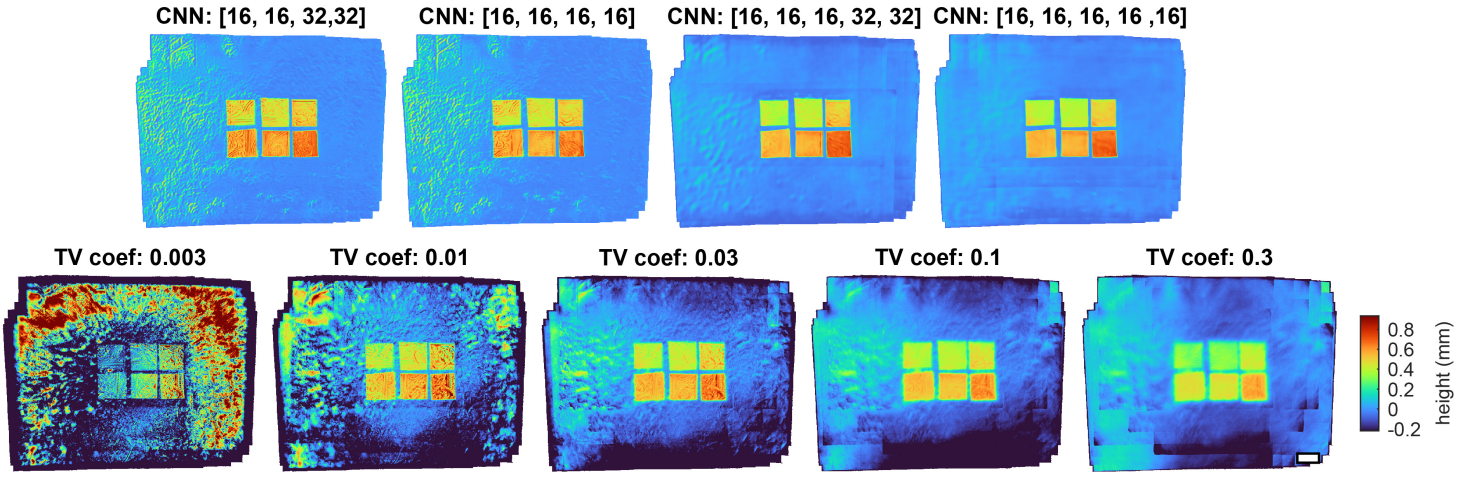

Figure 6: Regularization comparison for the cut cards sample. Scale bar, 1 cm.

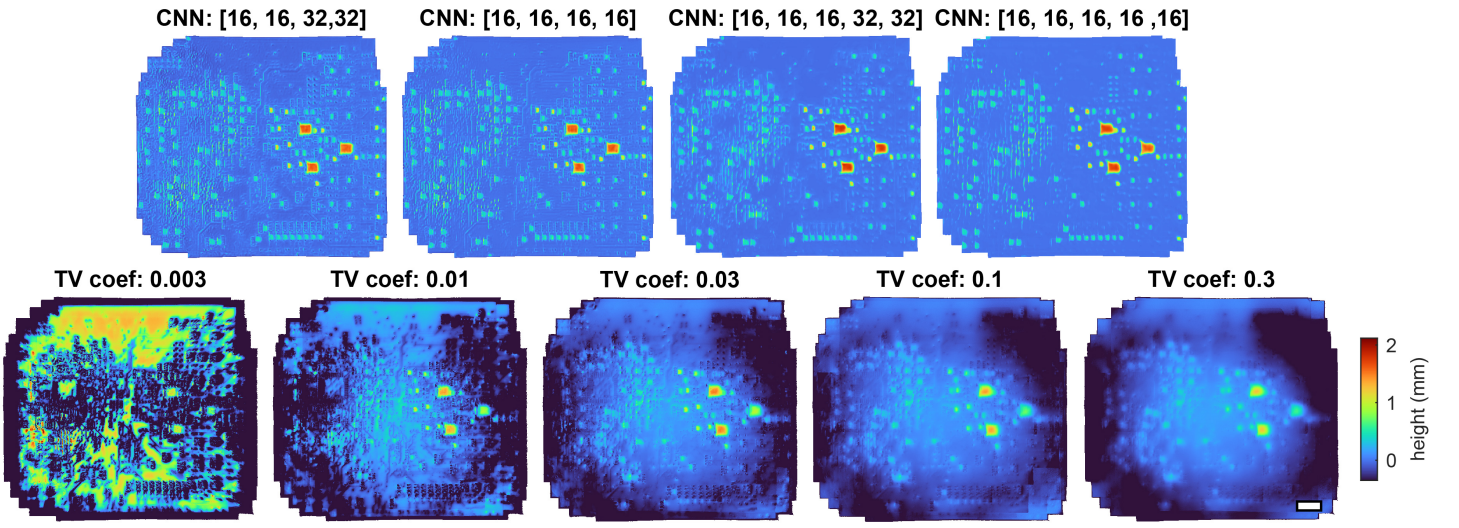

Figure 7: Regularization comparison for the PCB sample. Scale bar, 1 cm.

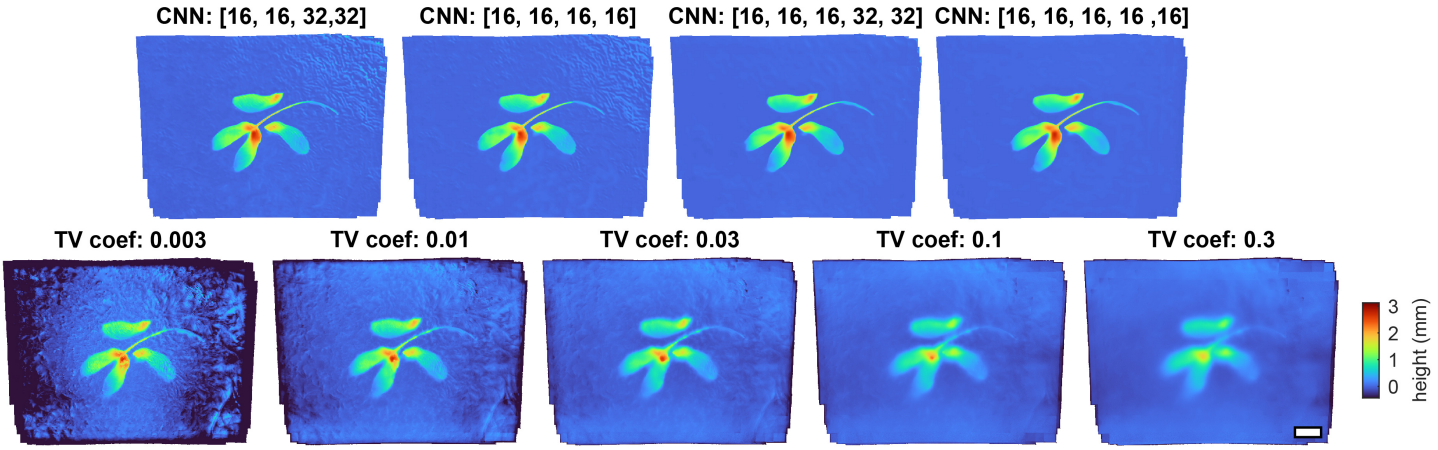

Figure 8: Regularization comparison for the helicopter sample. Scale bar, 1 cm.

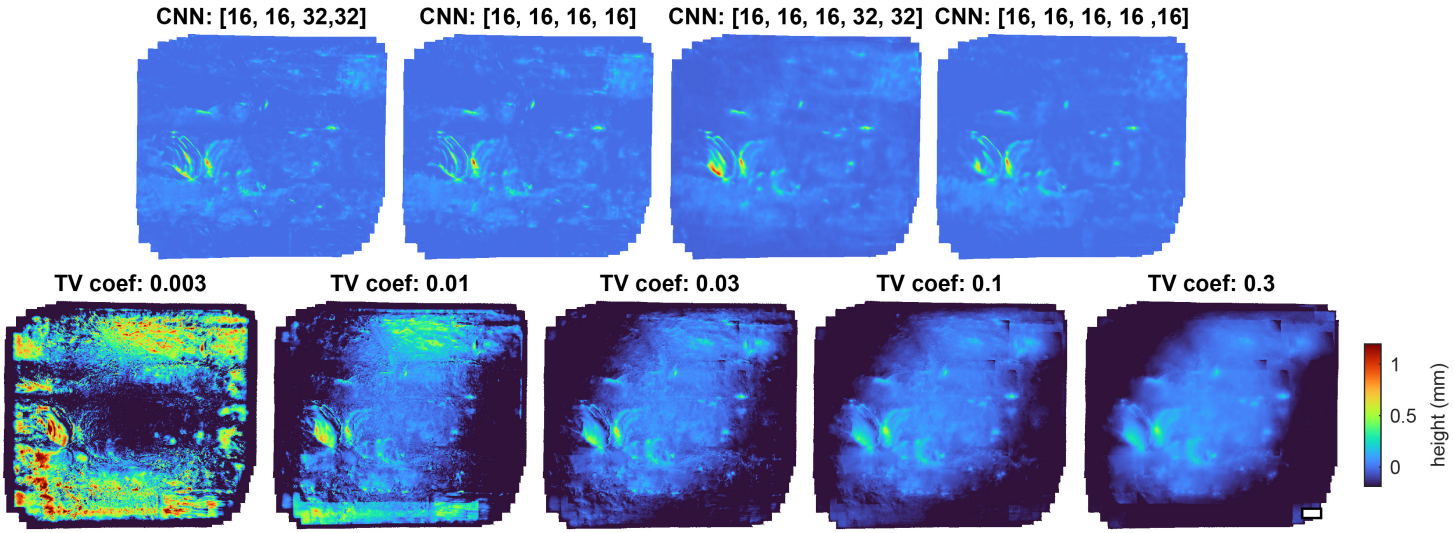

Figure 9: Regularization comparison for the painting brush strokes sample. Scale bar, 1 cm.

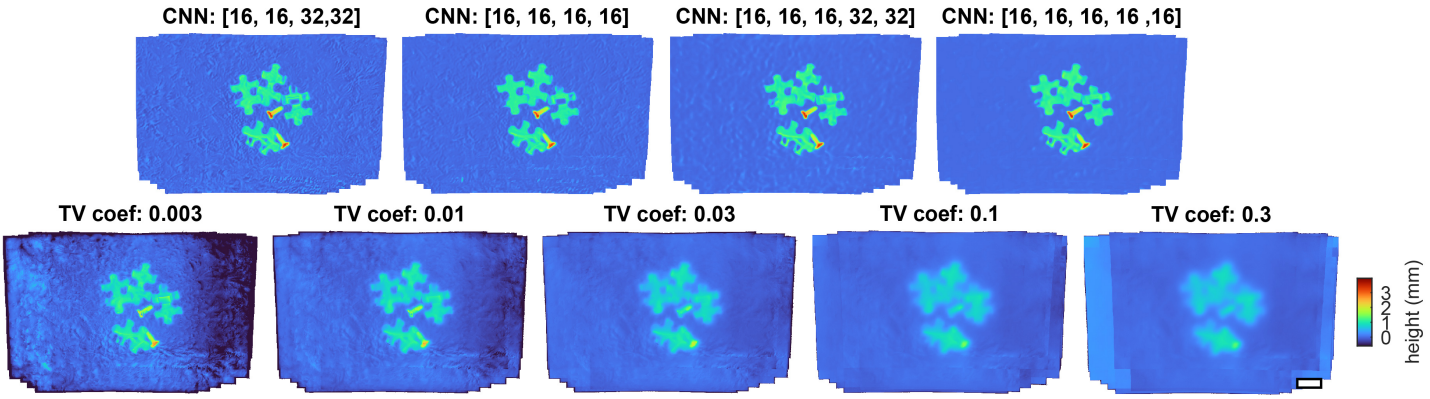

Figure 10: Regularization comparison for the hyperparameter tuning sample (puzzle pieces and screws). The third CNN architecture was chosen to balance resolution and reduction of artifacts in the background. Scale bar, 1 cm.

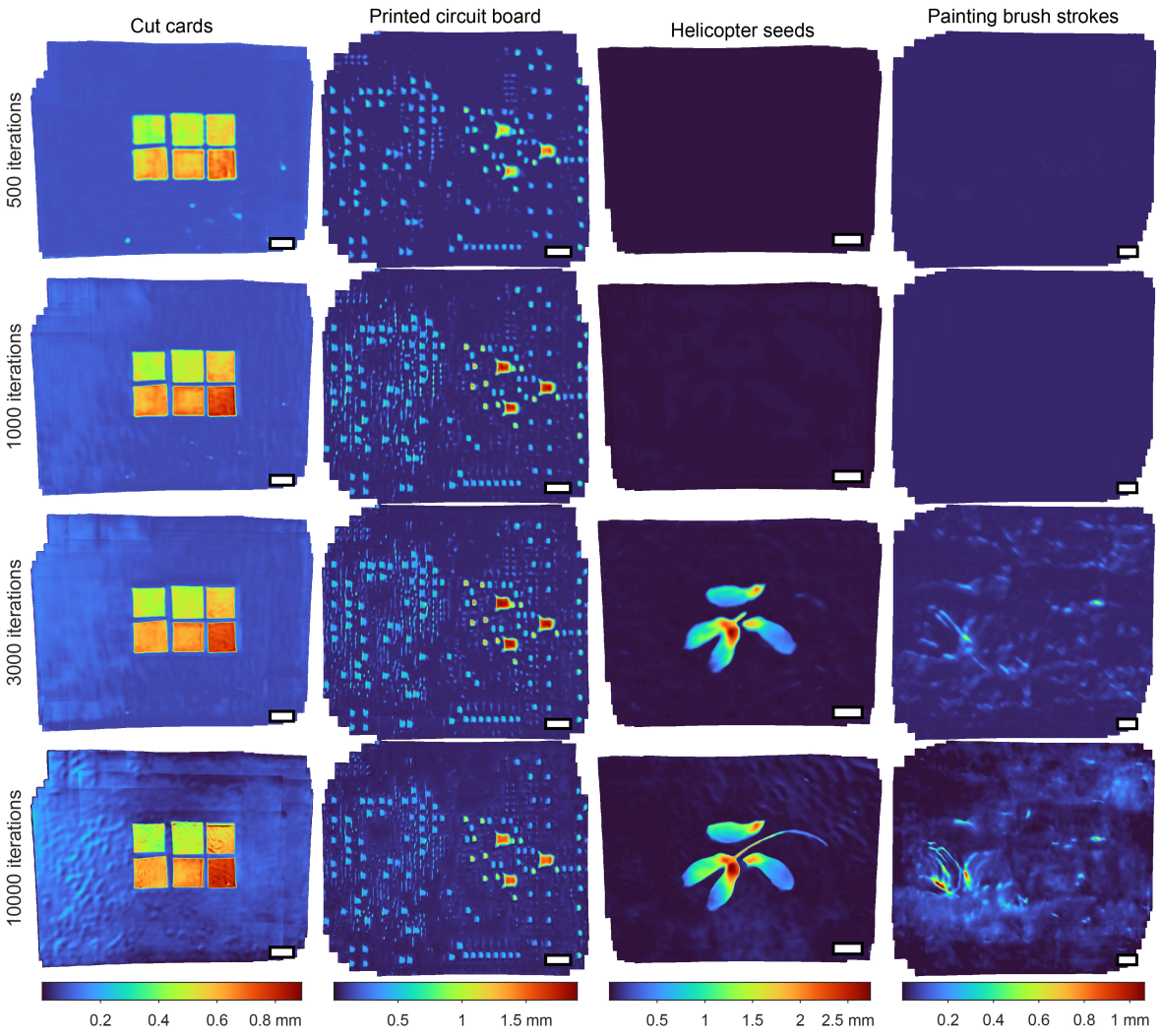

Figure 11: The four main samples optimized for 500, 1000, 3000, and 10000 iterations. Images within a column use the same color ranges. Scale bar, 1 cm.