Learning to Autofocus: Supplement

Charles Herrmann[1] Richard Strong Bowen[1] Neal Wadhwa[2] Rahul Garg[2]
Qiurui He[2] Jonathan T. Barron[2] Ramin Zabih[1,2]
[1]Cornell Tech [2]Google Research
{cih, rsb, rdz}@cs.cornell.edu {nealw, rahulgarg, qiurui, barron, raminz}@google.com

1. Baseline Algorithms

Here we document the algorithms taken from prior work that we use as baselines for our proposed model.

1.1. Contrast-Based Baseline Algorithms

As a point of comparison for our proposed model, we implemented a number of contrast-based autofocus algorithms (or equivalently, patch-based depth-from-defocus algorithms) and evaluated them as baselines on our task. When selecting what baselines to implement, we prioritized top-performing techniques according to a relatively recent survey paper [14]. Given a focal stack of images $\{I\}$ we compute a contrast score ϕ for each I, and we return the index into the focal stack that maximizes ϕ.

Intensity Variance [8]: The variance of the intensity values of the entire image.

$$\phi = \mathrm{Var}(I) \tag{1}$$

Intensity Coefficient of Variation [8]: The coefficient of variation of the intensity values of the entire image, which is the standard deviation of the intensity values divided by their mean. Similar metrics are sometimes referred to in past work as "normalized variance".

$$\phi = \frac{\sqrt{\mathrm{Var}(I)}}{\mu(I)} \tag{2}$$

Total Variation (L1) [11, 15]: The total absolute difference between the intensity value of all pixels and their (4-connected) neighbors:

$$\phi = \sum_{x,y} |I[x,y] - I[x+1,y]| + |I[x,y] - I[x,y+1]| \tag{3}$$

Total Variation (L2) [15]: The total squared difference between the intensity value of all pixels and their (4-connected) neighbors. This is sometimes referred to as "gradient energy":

$$\phi = \sum_{x,y} (I[x,y] - I[x+1,y])^2 + (I[x,y] - I[x,y+1])^2 \tag{4}$$

Energy of Laplacian [17]: The image is convolved by a discrete Laplace operator, and the response are squared and summed.

$$\phi = \sum_{x,y} \Delta[x,y]^2, \quad \Delta = I * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

Laplacian Variance [13]: The image is convolved by a discrete Laplace operator, and the global variance of the response is computed.

$$\phi = \mathrm{Var}(\Delta[x,y]), \quad \Delta = I * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

Sum of Modified Laplacian [12]: The image is convolved by a 1D discrete Laplace operator in x and y, and the absolute values of each filter response are summed.

$$\phi = \sum_{x,y} \Delta[x,y], \quad \Delta = |I * L_x| + |I * L_y|$$
$$L_x = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad L_y = L_x^{\mathrm{T}} \tag{7}$$

Diagonal Laplacian [19]: This is the same as the "sum of modified Laplacian" approach, but augmented with diagonal Laplacian filters as well.

$$\phi = \sum_{x,y} \Delta[x,y]$$
$$\Delta = |I * L_x| + |I * L_y| + |I * L_{xy}| + |I * L_{yx}|$$
$$L_{xy} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad L_{yx} = L_{xy}^{\mathrm{T}} \tag{8}$$

Mean Gradient Magnitude [18]: The mean gradient magnitude, where the gradient is computed using the norm of the response of Sobel filters. This is sometimes referred to as "Tenengrad".

$$\phi = \frac{1}{n} \sum_{x,y} \sqrt{\nabla_x[x,y]^2 + \nabla_y[x,y]^2} \tag{9}$$

$$\nabla_x = I * \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \; \nabla_y = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Gradient Count [8]: The total number of edges in the image whose magnitude is above some threshold t, where the gradient magnitude is again computed using Sobel filters.

$$\phi = \frac{1}{n} \sum_{x,y} [|\nabla_x[x,y]| > t] + [|\nabla_y[x,y]| > t] \tag{10}$$

Gradient Magnitude Variance [13]: The global variance of gradient magnitudes, where gradients are again computed using Sobel filters.

$$\phi = \mathrm{Var}\left(\sqrt{\nabla_x[x,y]^2 + \nabla_y[x,y]^2} \right) \tag{11}$$

Percentile Range: The difference between the $100-p$'th percentile and the p'th percentile of intensity values in the image. When $p = 0$, this is the difference between the maximum and minimum pixel intensities in the image.

$$\phi = \mathrm{percentile}(I, 100-p) - \mathrm{percentile}(I, p) \tag{12}$$

Histogram Entropy [8]: The Shannon entropy of all pixel intensities in the image.

$$\phi = -\sum_i \mathbf{n}_i \log(\mathbf{n}_i), \quad \mathbf{n} = \mathrm{hist}(I) \tag{13}$$

DCT Energy Ratio [3]: The squared sum of all DCT coefficients of the image without the DC component, divided by the squared DC component.

$$\phi = \frac{\left(\sum_{u,v} D[u,v]^2\right) - D[0,0]^2}{D[0,0]^2}, \quad D = \mathrm{DCT}(I) \tag{14}$$

DCT Reduced Energy Ratio [10]: The squared sum of the 5 lowest order DCT coefficients (excluding the DC component) divided by the squared DC component.

$$\phi = \frac{D[0,1]^2 + D[1,0]^2 + D[0,2]^2 + D[1,1]^2 + D[2,0]^2}{D[0,0]^2} \tag{15}$$

Modified DCT [9]: The total filter response of the image convolved with a checkerboard-like filter, which is somewhat related to the DCT of the image.

$$\phi = \sum_{x,y} \left(I * \begin{bmatrix} +1 & +1 & -1 & -1 \\ +1 & +1 & -1 & -1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \end{bmatrix} \right) [x,y] \tag{16}$$

Wavelet Sum [23]: The sum of the absolute value of the high-frequency components of level ℓ of the wavelet decomposition of the image. In our experiments, we use CDF9/7 wavelets [4].

$$\phi = \sum_{x,y} \left| W^{(\ell)}_{LH}[x,y] \right| + \left| W^{(\ell)}_{HL}[x,y] \right| + \left| W^{(\ell)}_{HH}[x,y] \right|$$
$$\left(W^{(\ell)}_{LL}, W^{(\ell)}_{LH}, W^{(\ell)}_{HL}, W^{(\ell)}_{HH} \right) = \mathrm{CDF9/7}(I, \ell) \tag{17}$$

Wavelet Variance [23]: The variance of the high-frequency components of level ℓ of the wavelet decomposition of the image.

$$\phi = \mathrm{Var}\left(W^{(\ell)}_{LH}[x,y] \right) + \mathrm{Var}\left(W^{(\ell)}_{HL}[x,y] \right) + \mathrm{Var}\left(W^{(\ell)}_{HH}[x,y] \right) \tag{18}$$

Wavelet Ratio [22]: The ratio of the squared norm of the high-frequency components of level ℓ of the wavelet decomposition of the image to the squared norm of the low-frequency components.

$$\phi = \frac{\sum_{x,y} W^{(\ell)}_{LH}[x,y]^2 + W^{(\ell)}_{HL}[x,y]^2 + W^{(\ell)}_{HH}[x,y]^2}{\sum_{x,y} W^{(\ell)}_{LL}[x,y]^2} \tag{19}$$

Mean Wavelet Log-Ratio : This is a baseline of our own design in which we modify the "Wavelet Ratio" model to compute a local log-ratio between the high-frequency and low-frequency energy at each spatial location in one level of a wavelet decomposition, and then compute the mean of those log-ratios. We add 1 to the denominator to prevent numerical issues.

$$\phi = \frac{1}{n} \sum_{x,y} \log \left(\frac{W^{(\ell)}_{LH}[x,y]^2 + W^{(\ell)}_{HL}[x,y]^2 + W^{(\ell)}_{HH}[x,y]^2}{W^{(\ell)}_{LL}[x,y]^2 + 1} \right) \tag{20}$$

Eigenvalue Trace [21]: The image is reduced to a matrix where each column is a vector containing the intensity values of each non-overlapping patch (here, of size 4×4) in the image. The trace of the sample covariance of that matrix is then used as a measure of sharpness.

$$\phi = \mathrm{trace}(\mathrm{cov}(\mathrm{im2col}(I, 4))) \tag{21}$$

Mean Local Ratio [7]: A local measure of contrast is computed at each pixel by considering the ratio of each pixel intensity to a local average, and the overall contrast is computed as the average of those ratios (rectified if they are below 1) across the image. The numerator and denominator of each ratio are incremented by 1 to avoid numerical issues.

$$\phi = \frac{1}{n}\sum_{x,y} \max\left(\frac{\text{blur}(I,\sigma)[x,y]+1}{I[x,y]+1}, \frac{I[x,y]+1}{\text{blur}(I,\sigma)[x,y]+1}\right) \tag{22}$$

Where $\text{blur}(I,\sigma)$ applies a Gaussian blur of standard deviation σ to image I.

Mean Local Log-Ratio: This is a baseline of our own design in which we modify the "Mean Local Ratio" technique above, by using the geometric mean of ratios instead of the arithmetic mean.

$$\phi = \exp\left(\frac{1}{n}\sum_{x,y}\left|\log\left(\frac{I[x,y]+1}{\text{blur}(I,\sigma)[x,y]+1}\right)\right|\right) \tag{23}$$

Mean Local Norm-Dist-Sq: This is another baseline of our own design, in which we modify the "Mean Local Ratio" technique to use normalized squared distance (similar to a Coefficient of Variation) instead of ratios, which improves performance.

$$\phi = \frac{1}{n}\sum_{x,y}\frac{(I[x,y]-\text{blur}(I,\sigma)[x,y])^2}{\text{blur}(I,\sigma)[x,y]^2+1} \tag{24}$$

1.2. Dual-Pixel / Stereo Baseline Algorithms

Because our images are taken from a dual pixel (DP) sensor, our focal stack can be thought of as a stack of left and right images in a stereo pair $\{(L,R)\}$. When a patch is in focus, the left and right DP images should resemble each other. It is therefore possible to construct simple autofocus algorithms by taking each left/right image pair (L,R) in a DP focal stack, compute some measure of mismatch between those two images f, and return the focal index that minimizes that loss. In this section, we describe the baseline algorithms we use for this approach. Because patches of the the left and right DP images may have drastically different global brightnesses due to lens shading (especially when the patches are taken from the periphery of the entire image frame), these stereo-like algorithms must be invariant to global transformations of the input images. For this reason, we center each image by its mean and divide by its standard deviation before computing all stereo measures:

$$\hat{L} = \frac{L-\mu(L)}{\text{Var}(L)}, \quad \hat{R} = \frac{R-\mu(R)}{\text{Var}(R)} \tag{25}$$

This has no effect on some models (such as census and rank transformations) but is critical for other models.

Census Transform (Hamming) [24]: We apply the census transformation to the left and right DP images, wherein each pixel is represented by an 8-length binary vector representing whether or not the pixel is greater than each of its 8 neighbors. We score each pair according to the total Hamming distance between the two census-transformed images.

$$f = \sum_{x,y}\|\text{census}(L)[x,y]-\text{census}(R)[x,y]\|_0$$
$$\text{census}(I)[x,y] = \Big[I[x+\Delta_x,y+\Delta_y] > I[x,y] \Big| \Delta_x\in[-1,0,1],\Delta_y\in[-1,0,1],\Delta_x\neq\Delta_y\neq 0\Big] \tag{26}$$

Rank Transform (L1) [24]: We apply the rank transformation (the 0-norm of the census transformation) to the left and right DP images, and score each pair according to the L1 distance between the two rank-transformed images.

$$f = \sum_{x,y}\|\text{rank}(L)[x,y]-\text{rank}(R)[x,y]\|_1 \tag{27}$$
$$\text{rank}(I)[x,y] = \|\text{census}(I)[x,y]\|_0 \tag{28}$$

Ternary Census [16]: We apply the ternary census transformation to the left and right DP images, wherein each pixel is represented by an 8-length ternary vector representing if the pixel is greater than, less than, or close to (according to some threshold ϵ) each of its 8 neighbors. We then score each pair according to the total L1 distance between the two census-transformed images.

$$f = \sum_{x,y}\left\|\text{census}^3(L)[x,y]-\text{census}^3(R)[x,y]\right\|_1$$
$$\text{census}^3(I)[x,y] = \Big[\text{tsgn}\left(I[x+\Delta_x,y+\Delta_y]-I[x,y]\right) \Big| \Delta_x\in[-1,0,1],\Delta_y\in[-1,0,1],\Delta_x\neq\Delta_y\neq 0\Big]$$
$$\text{tsng}(x,\epsilon) = \text{sgn}(x)\,[|x|>\epsilon] \tag{29}$$

Normalized Cross-Correlation [1, 6]: NCC is just the inner product of these two normalized images, with its sign flipped such that minimization results in maximum cross-correlation. This is equivalent to minimizing the normalized sum of squared distances between the two images.

$$f = -\left\langle \hat{L},\hat{R}\right\rangle \tag{30}$$

Normalized SAD [6]: The sum of absolute deviations between the two normalized images.

$$f = \sum_{x,y}\left|\hat{L}[x,y]-\hat{R}[x,y]\right| \tag{31}$$

Normalized Envelope (L1) [2]: Pixel matching techniques can be made invariant to the discrete sampling of the sensor by adapting them to operate on smooth upper and lower envelopes of image intensities. Here we compute an upper and lower envelope of the left and right images, and from them compute the total L1 distance between the extents of the left and right envelopes.

$$\begin{aligned} f = \textstyle\sum_{x,y} & \left|\max\left(0, \hat{L}_{\text{lo}}[x,y] - \hat{R}_{\text{hi}}[x,y]\right)\right| \\ & + \left|\max\left(0, \hat{R}_{\text{lo}}[x,y] - \hat{L}_{\text{hi}}[x,y]\right)\right| \end{aligned} \tag{32}$$

$$\hat{L}_{\text{lo}} = \text{min2}\left(\text{blur2}\left(\hat{L}\right)\right), \quad L_{\text{hi}} = \text{max2}\left(\text{blur2}\left(\hat{L}\right)\right)$$

where $\text{max2}(\cdot)$ is a 2×2 "max" filter (i.e. max pooling), $\text{min2}(\cdot)$ is a 2×2 "min" filter (i.e. min pooling), and $\text{blur2}(\cdot)$ is a 2×2 box filter (i.e. average pooling). R_{lo} and $\hat{R}_{\text{hi}}$ are defined similarly.

Normalized Envelope (L2) [2]: Similarly, we can compute the total squared distance between the extents of the left and right envelopes.

$$\begin{aligned} f = \textstyle\sum_{x,y} & \max\left(0, \hat{L}_{\text{lo}}[x,y] - \hat{R}_{\text{hi}}[x,y]\right)^2 \\ & + \max\left(0, \hat{R}_{\text{lo}}[x,y] - \hat{L}_{\text{hi}}[x,y]\right)^2 \end{aligned} \tag{33}$$

1.3. Single-Slice Baseline Algorithms

The baseline methods above infer the in-focus index by either maximizing contrast ϕ (for contrast-based methods) or minimizing stereo mismatch f (for dual-pixel methods). Hence, they all require the knowledge of the entire focal stack before making a prediction.

However, the DP algorithms can be extended to predict the in-focus index with just one input DP image pair, if we can establish the relationship between left/right disparity d and ideal focus distance z^*. We list a few such algorithms below.

SSD Disparity: We use the block matching approach of [20] to estimate disparity. In order to convert the disparity of a patch to a focal depth, we fit a linear model that estimates focal depth from the median patch disparity. The linear model is robustly estimated from all training patches using RANSAC. This methods computes depth over $1.5\times$ reduced field of view and we report results only on patches contained within that field of view. A narrower field of view is not unfair to the baseline as PSF variations and focal breathing are worse near the periphery.

Learned Depth: We use the neural network based approach of [5] to predict depth from dual-pixel images. The model from [5] predicts depth maps up to an unknown affine transform, which we estimate by solving a least squares problem that minimizes the $L2$ distance between the affine transformed depth map and the disparity from [20] that are known to be linearly related. We use the same fitting described in SSD Disparity and restrict evaluation to the same $1.5\times$ reduced field of view.

ZNCC Disparity with Calibration: We compute the zero-normalized cross correlation between the input DP image pair (L, R) (using Equation 25) to get $(\hat{L}, \hat{R})$. Then, we compute disparity between $\hat{L}$ and $\hat{R}$ [1, 6] and apply a precomputed calibration to convert disparity to focal distance. Specifically, to compute disparity d, we do the following

$$d = \underset{\delta}{\text{argmax}} \left\langle \hat{L}[x,y], \hat{R}[x+\delta, y] \right\rangle \tag{34}$$

for integer δ in a small range around zero. We then refine d to get sub-pixel resolution by fitting a quadratic near the peak and finding its supremum.

Under paraxial and thin-lens approximations, and assuming constant aperture and focal length, signed disparity d and ideal focus distance z^* are related by an affine transform [20]:

$$d = C\left(\frac{1}{z} - \frac{1}{z^*}\right) \tag{35}$$

where C is a calibration constant and z is the lens's current focus distance.

The assumption that C is a constant breaks down for real lenses as they do not satisfy the paraxial and thin-lens approximations. In fact, the value of C varies significantly across the field of view, due to optical aberration, vignetting, changes in optical blur kernels, etc., as shown in [20]. The camera device we use embeds a factory calibration table that specifies the measured C values sparsely across the field of view. We obtain the value of C for each input patch by bilinearly interpolating the low-resolution calibration table.

With the knowledge of disparity d, calibration coefficient C, and current focus distance z, we can easily solve for z^* in Equation 35.

2. Generalization to other phones

To show that our technique generalizes, we use the data captured in the paper to create a new test set using the "left" camera, which has a different calibration and PSF.

This left test set contains the same scenes as the test set in the original paper; however, the overall attributes of the set may be different. The "left" phone is positioned in front of the "center" phone by 1.1 cm (on the z-axis, it is +1.1cm closer to objects in the scene). In addition, the computed

	Algorithm	higher is better $= 0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
D1	Learned Depth† [5]	0.070	0.206	0.340	0.564	7.224	11.010
D1	SSD Disparity† [20]	0.068	0.200	0.333	0.550	7.377	10.951
D1	ZNCC Disparity	0.046	0.136	0.224	0.379	9.436	13.138
D1	Our model	0.105	0.322	0.513	0.807	2.912	3.867

Table 1. Evaluating techniques on the "left" version of the test set. This tests whether the technique generalizes to other phones. Note our model still outperforms the baselines and that the performance went down for all techniques indicating that the "left" version of the test set is harder. See text for explanation. A † indicates that patches within a $1.5\times$ reduced field of view were used.

depth has an overall lower confidence than that of the center camera since fewer cameras see all the pixels captured by the left camera. This problem is particularly apparent on the left side of the capture. In addition, because we keep the same confidence threshold as used for the center camera, fewer patches will be generated. In general, it may be difficult to compare the raw numbers from the test set using the center camera and the test set using the left camera.

As shown in Table 1, all techniques report slightly lower numbers. This indicates that the "left" test-set may be more difficult than the "center" test-set due to the aforementioned changes. Despite this, our model still outperforms the baselines. Additionally, several simple techniques, like adding calibration data to the model or a brief fine-tuning stage for each camera, could be easily added to our approach and potentially lead to improved per-device performance.

For this run, ZNCC Disparity uses calibration for the "left" camera, and linear models to convert to focal depths for SSD Disparity and Learned Depth were estimated using the training data patches from the "left" camera.

3. Multi-step problem

In Figure 3, we obtain improved results on the multi-step problem.

4. Light and Dark Scenes

In Figure 8 in the main paper, we presented examples on particularly dark images. In Table 2, we present the full numeric breakdowns of the performance of single-index algorithms on scenes with a normal amounts of light versus scenes with low light.

To capture these, we placed the rig in a fixed position and then captured two focal stacks: one with the light on and then one with the light turned off. As a result, these captures should be perfectly registered and should be the identical besides the presence or absence of light. We then used the ground truth depth from the light image to eliminate any possible mistakes that the SFM pipeline would have with the darker images.

5. Example Images

5.1. Single slice as input

In Figures 1, 2, 3, 4, we provide a random selection of inputs (among those inside the 1.5x crop center, so that the PD baselines are present) and the predictions from all baselines and our models. The "Input" is what the algorithm is given. The focal stack identification key is directly above the row. The title of each focal slice is: the name of the algorithm, the index, ("Err" followed by the number of indices away from the ground truth).

5.2. Focal stack as input

In Figures 5, 6, 7, 8, we provide a random selection of inputs in the test set. The diagram contains an "Input" category; however, this is simply to display another element of the focal stack. All of these algorithms receive the full focal stack as input. The focal stack identification key is directly above the row. The title of each focal slice is: the name of the algorithm, the index, ("Err" followed by the number of indices away from the ground truth).

References

[1] Daniel I. Barnea and Harvey F. Silverman. A class of algorithms for fast digital image registration. *Transactions on Computers*, 1972.

[2] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *TPAMI*, 1998.

[3] Chun-Hung Shen and H. H. Chen. Robust focus measure for low-contrast images. *International Conference on Consumer Electronics*, 2006.

[4] Albert Cohen, Ingrid Daubechies, and J-C Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 1992.

[5] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. *ICCV*, 2019.

[6] Marsha Jo Hannah. *Computer Matching of Areas in Stereo Images.* PhD thesis, 1974.

[7] Franz Helmli and Stefan Scherer. Adaptive shape from focus with an error estimation in light microscopy. *International Symposium on Image and Signal Processing and Analysis*, 2001.

[8] Eric Krotkov. Focusing. *IJCV*, 1988.

[9] Sang Yong Lee, Yogendera Kumar, Ji Man Cho, Sang Won Lee, and Soo-Won Kim. Enhanced autofocus algorithm using robust focus measure and fuzzy reasoning. *Transactions on Circuits and Systems for Video Technology*, 2008.

[10] Sang Yong Lee, Jae Tack Yoo, and Soo-Won Kim. Reduced energy-ratio measure for robust autofocusing in digital camera. *Signal Processing Letters*, 2009.

[11] Harsh Nanda and Ross Cutler. Practical calibrations for a real-time digital omnidirectional camera. Technical report, In Technical Sketches, Computer Vision and Pattern Recognition, 2001.

	Algorithm	# of steps	higher is better $= 0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
D1	ZNCC Disparity with Calibration	1	0.064	0.181	0.286	0.448	8.879	12.911
		2	0.100	0.278	0.426	0.617	6.662	10.993
D1	Learned Depth† [5]	1	0.108	0.289	0.428	0.586	7.176	11.351
		2	0.172	0.433	0.618	0.802	3.876	7.410
I1	Our model	1	0.115	0.318	0.597	0.691	4.321	6.737
		2	0.138	0.377	0.567	0.807	2.855	4.088
D1	Our model	1	0.164	0.455	0.653	0.885	2.235	3.112
		2	**0.201**	**0.519**	**0.723**	**0.916**	**1.931**	**2.772**

Setting		Algorithm	higher is better $= 0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
Light	D1	SSD Disparity† [20]	0.079	0.228	0.355	0.528	6.732	9.577
	D1	Learned Depth† [5]	0.094	0.264	0.401	0.576	6.262	9.376
	D1	ZNCC Disparity	0.064	0.188	0.304	0.486	7.222	10.179
	D1	Our model	0.126	0.369	0.578	0.832	2.654	3.563
Dark	D1	Learned Depth† [5]	0.061	0.178	0.286	0.442	9.104	12.793
	D1	SSD Disparity† [20]	0.055	0.162	0.252	0.396	9.343	12.669
	D1	ZNCC Disparity	0.056	0.167	0.272	0.443	7.972	11.080
	D1	Our model	0.112	0.323	0.497	0.729	3.479	4.957

Table 2. Performance for scenes in high and low light. Note that our technique is the most resistant to dark scenes. A † indicates that patches within a $1.5\times$ reduced field of view were used.

[12] Shree K. Nayar and Yasuo Nakagawa. Shape from focus. *TPAMI*, 1994.

[13] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martínez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. *ICPR*, 2000.

[14] Said Pertuz, Domenec Puig, and Miguel García. Analysis of focus measure operators in shape-from-focus. *Pattern Recognition*, 2012.

[15] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992.

[16] Fridtjof Stein. Efficient computation of optical flow using the census transform. *Pattern Recognition*, 2004.

[17] Murali Subbarao, Tae-Sun Choi, and Arman Nikzad. Focusing techniques. *Optical Engineering*, 1993.

[18] Jay Martin Tenenbaum. *Accommodation in Computer Vision*. PhD thesis, Stanford University, 1971.

[19] Andrea Thelen, Susanne Frey, Sven Hirsch, and Peter Hering. Improvements in shape-from-focus for holographic reconstructions with regard to focus operators, neighborhood-size, and height value interpolation. *TIP*, 2009.

[20] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *SIGGRAPH*, 2018.

[21] Chong-Yaw Wee and Raveendran Paramesran. Measure of image sharpness using eigenvalues. *Information Sciences*, 2007.

[22] Hui Xie, Weibin Rong, and Lining Sun. Wavelet-based focus measure and 3-d surface reconstruction method for microscopy images. *IROS*, 2006.

[23] Ge Yang and B. J. Nelson. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. *IROS*, 2003.

[24] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. *ECCV*, 1994.

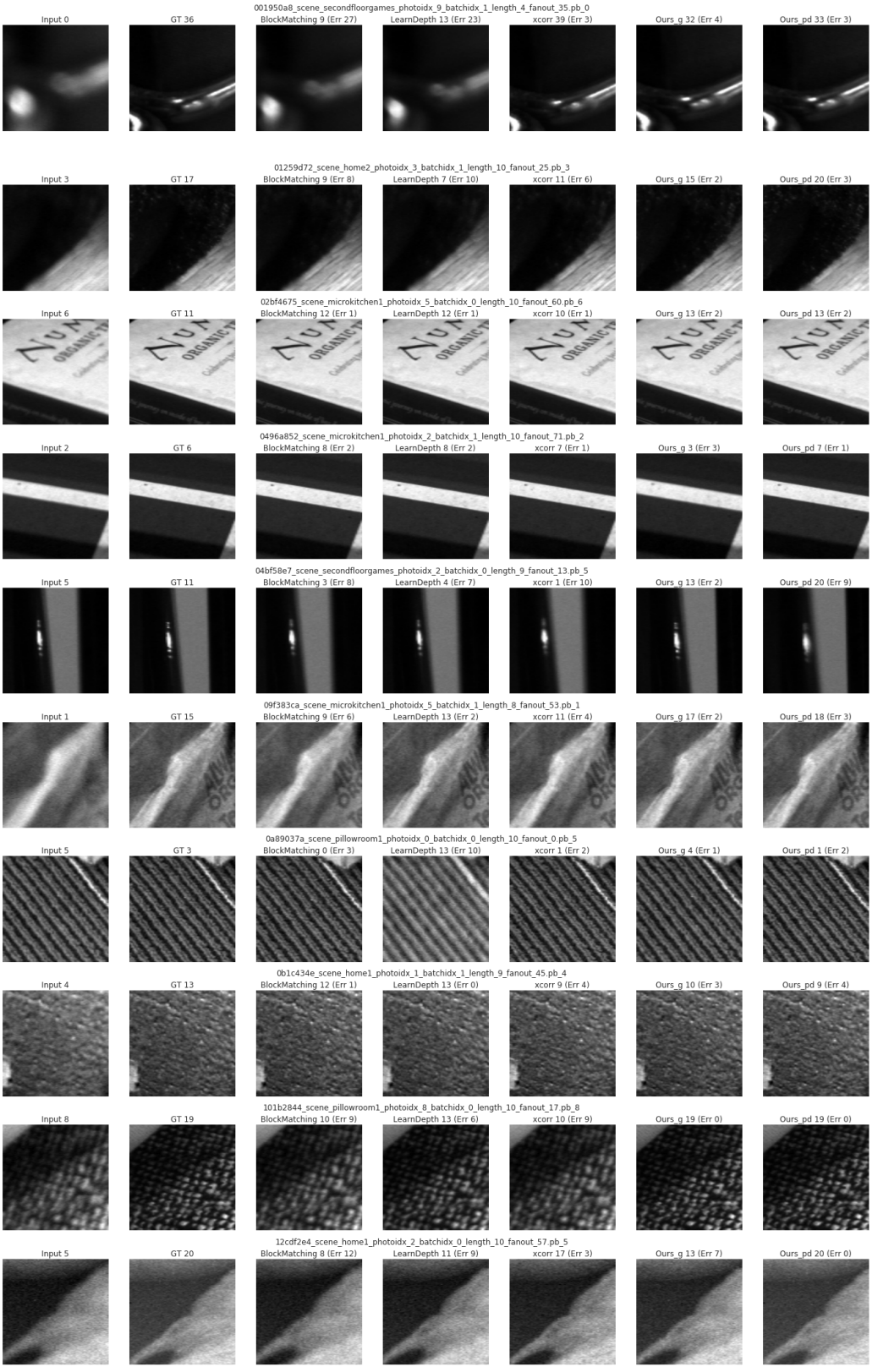

Figure 1. Algorithms given singleindex. Example page 1

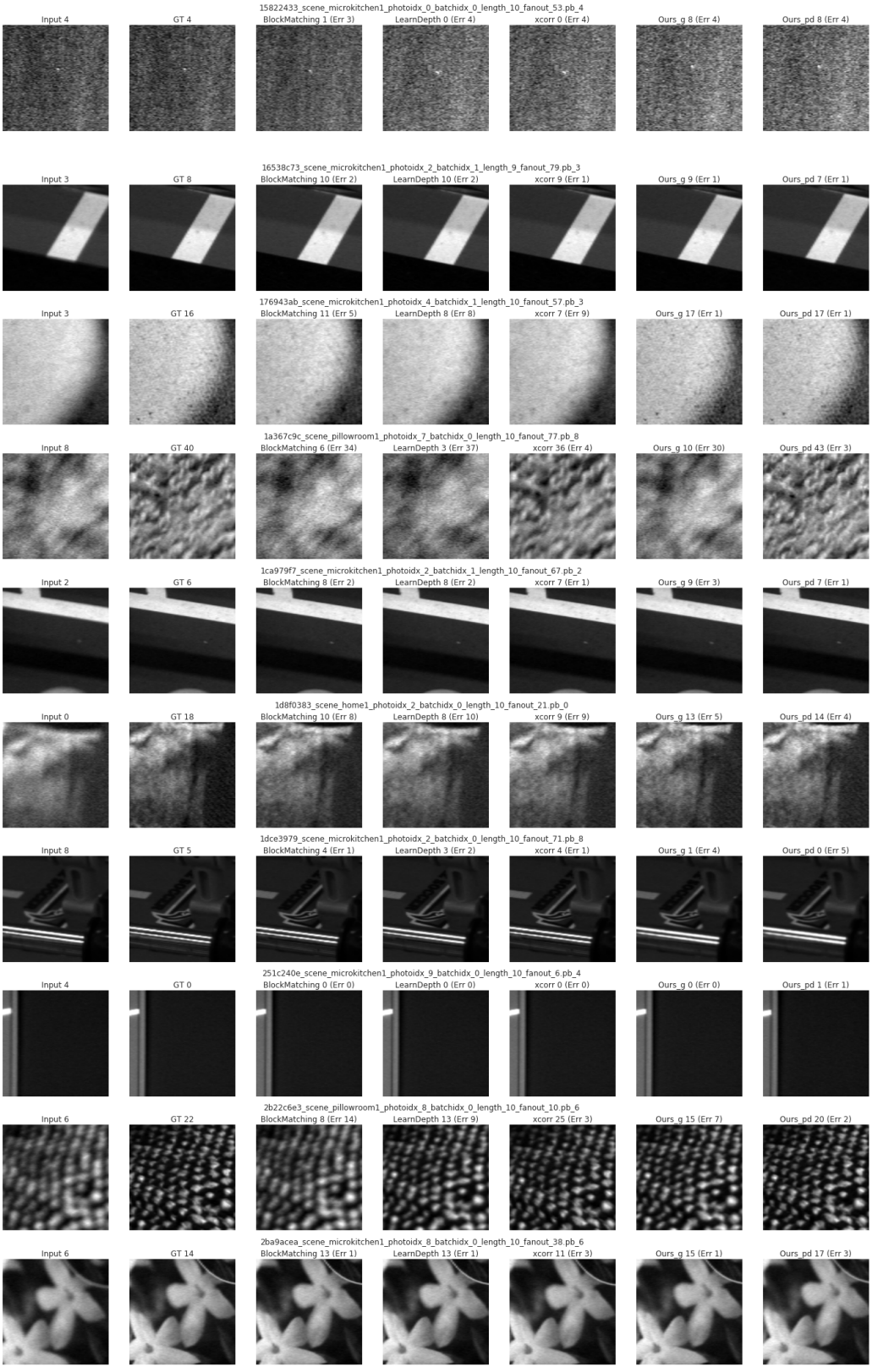

Figure 2. Algorithms given singleindex. Example page 2

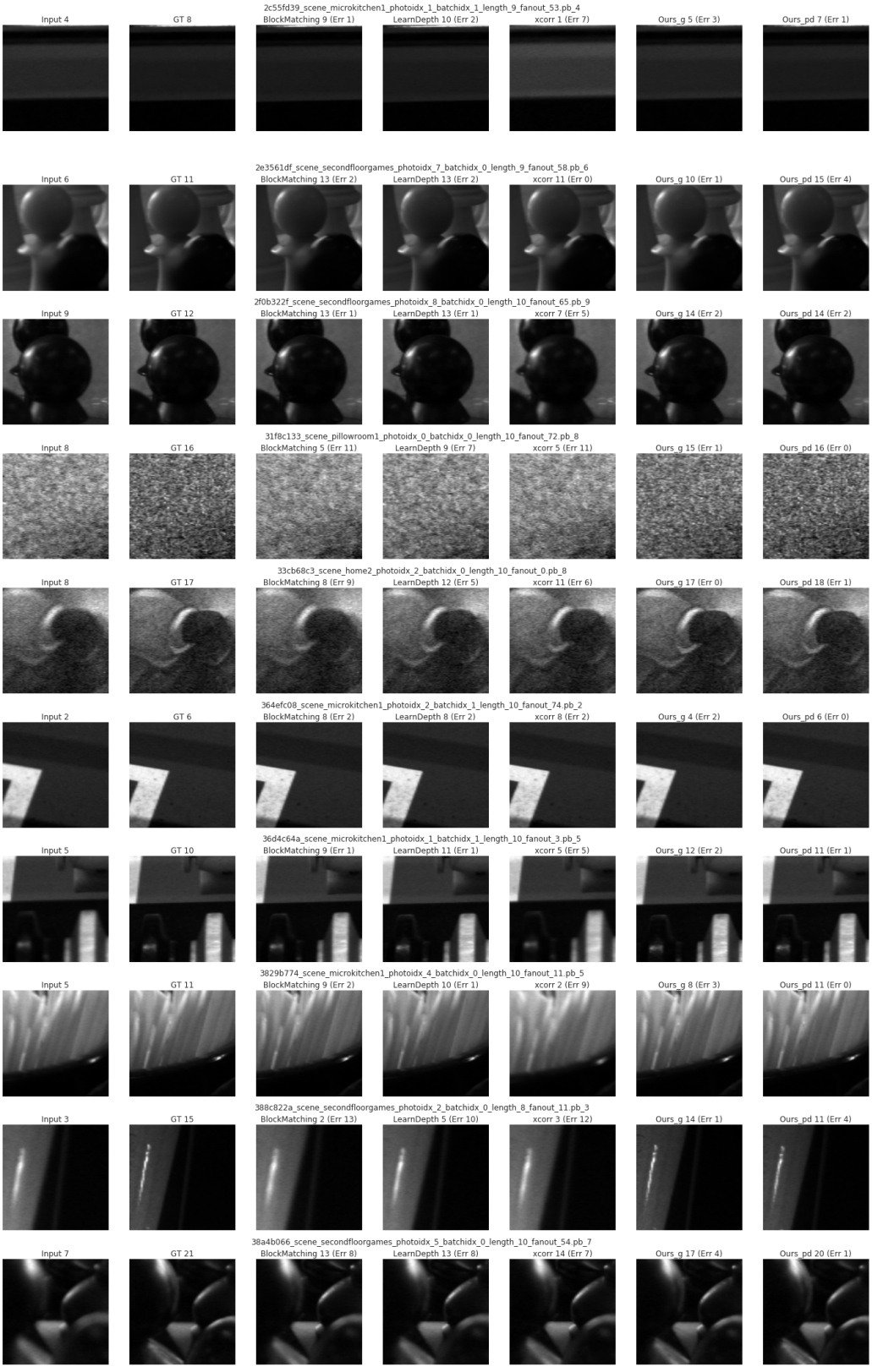

Figure 3. Algorithms given singleindex. Example page 3

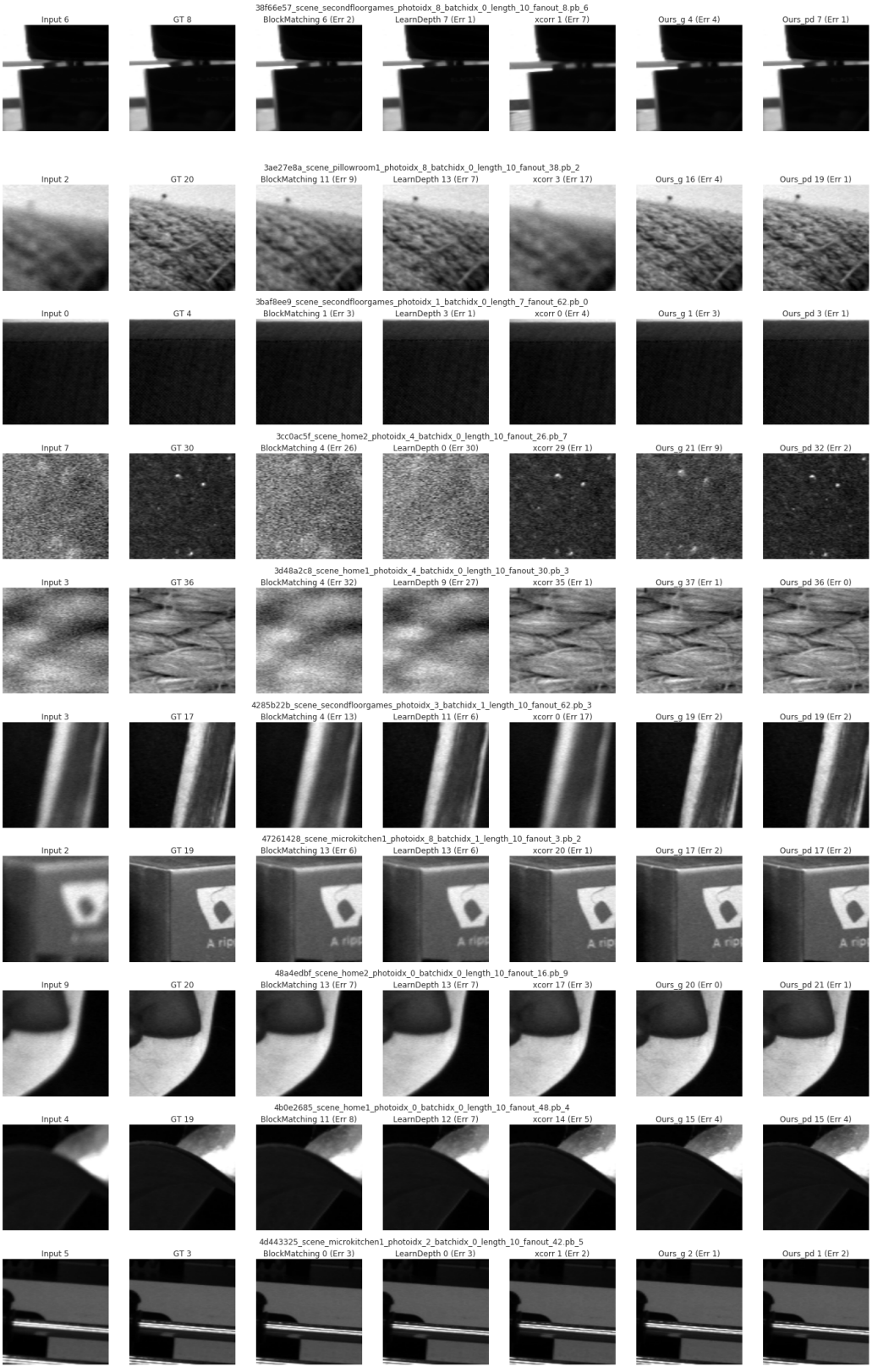

Figure 4. Algorithms given singleindex. Example page 4

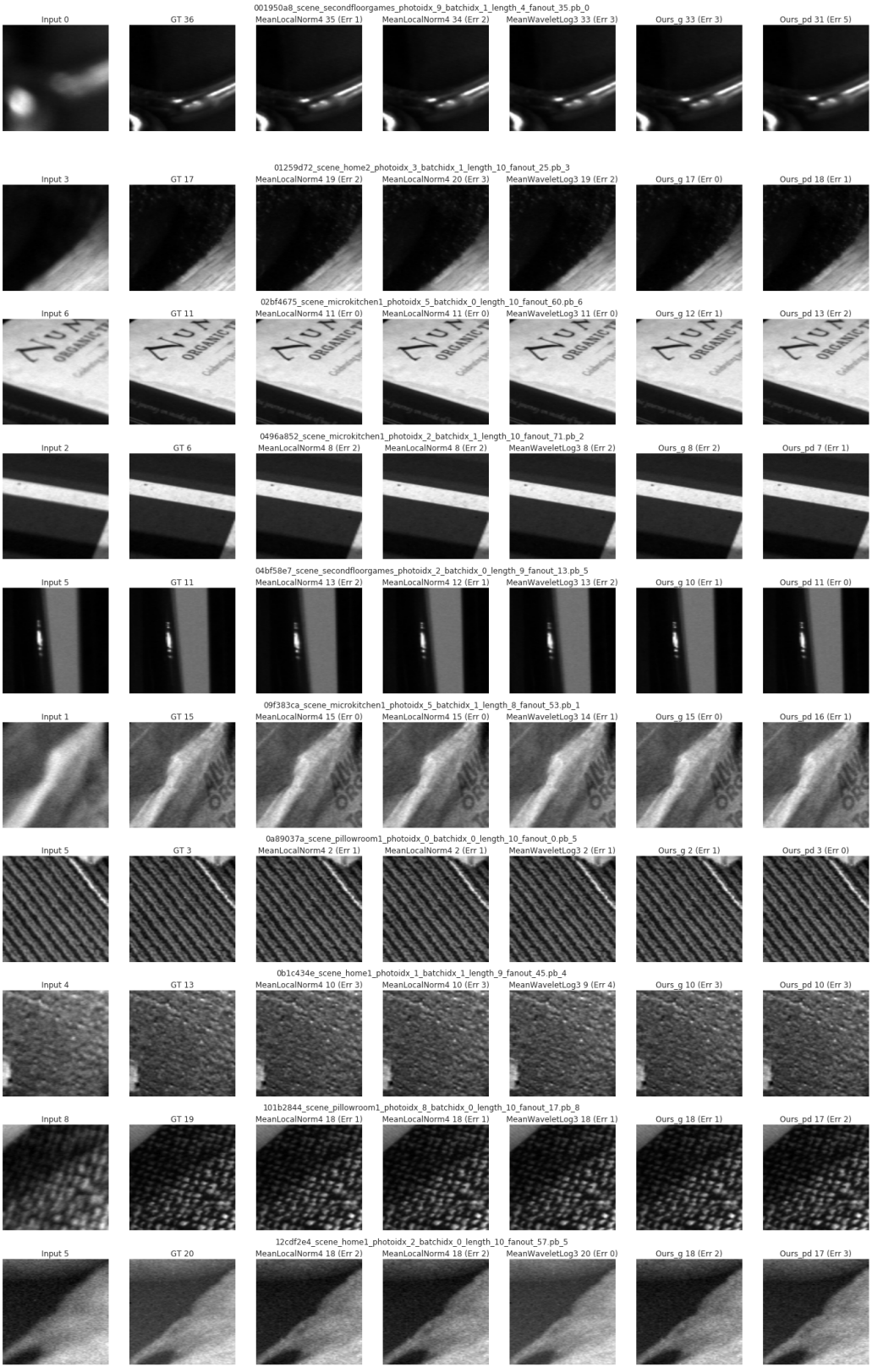

Figure 5. Algorithms given fullfocal. Example page 1

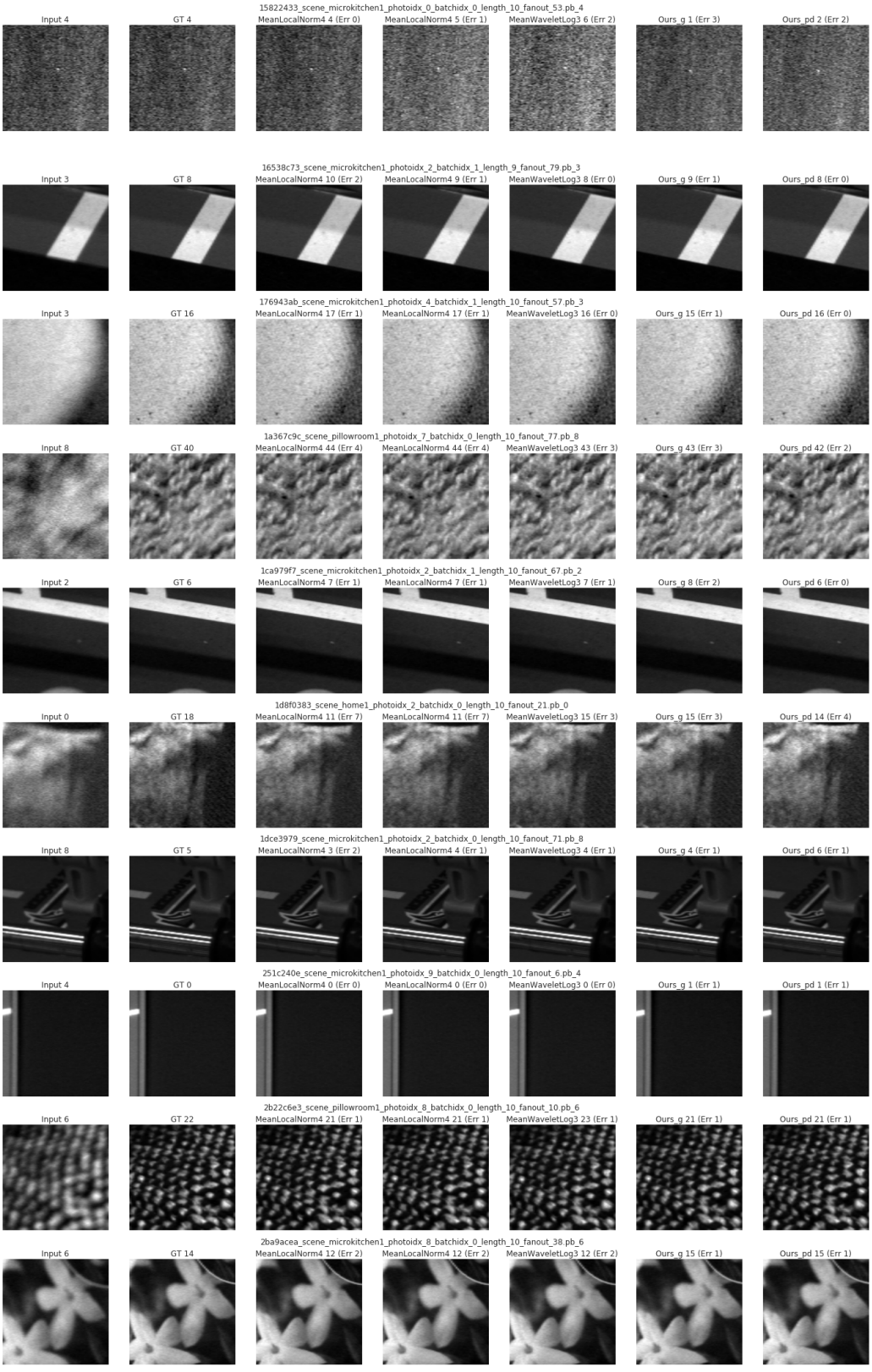

Figure 6. Algorithms given fullfocal. Example page 2

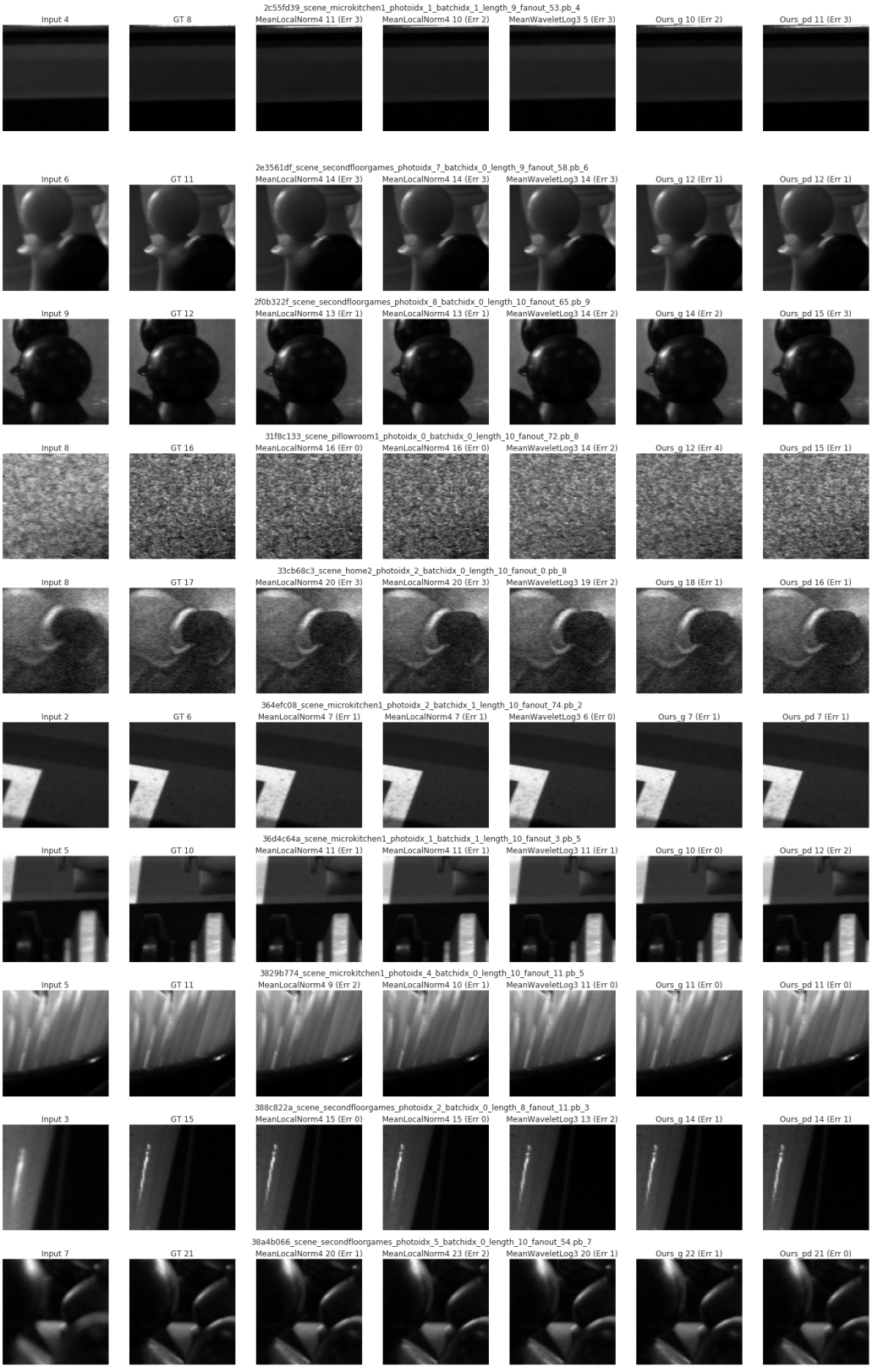

Figure 7. Algorithms given fullfocal. Example page 3

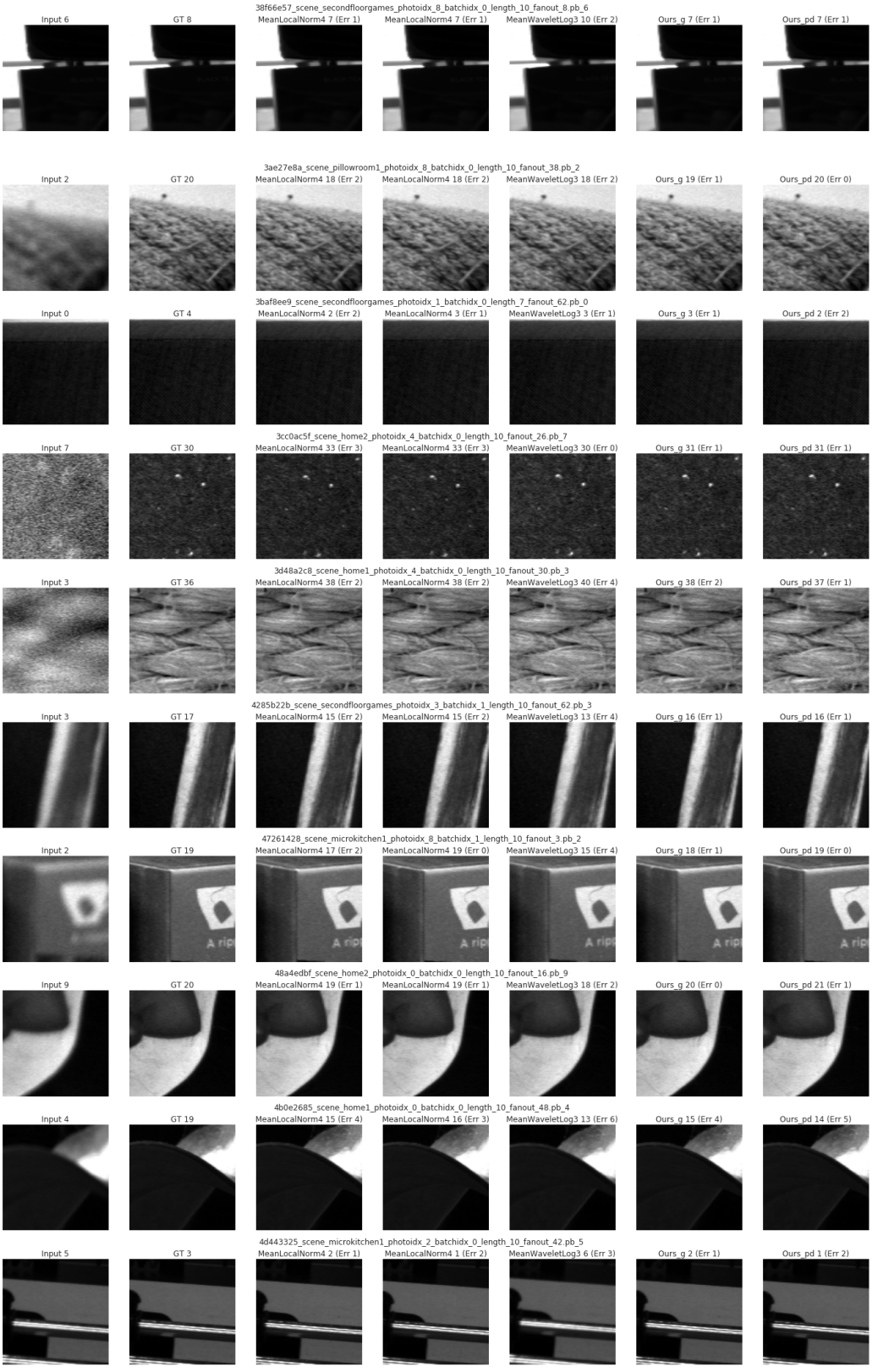

Figure 8. Algorithms given fullfocal. Example page 4