Supplementary: The Domain Transform Solver

Akash Bapat and Jan-Michael Frahm
Department of Computer Science, The University of North Carolina at Chapel Hill
{akash, jmf}@cs.unc.edu

In Sec.(1), we describe the similarities of our optimization objective to that of the Fast Bilateral Solver (FBS) [1]. In Sec.(2), we provide qualitative and quantitative results for the image colorization task. Additionally, we colorize smartphone images, which are increasingly becoming high-resolution. Finally, in Sec.(3), we provide visualizations for synthetic defocus using our method and highlight the advantages of our edge-aware technique.

1. Comparison with FBS solution

Here, we highlight the relation of our objective to the optimization function of FBS. Recall the optimal solution for our method, DTS, for the simplified objective (Eq. (2) in the main paper) is:

$$z_i = \frac{\lambda \bar{z}_{N_i} + \omega_i c_i t_i}{\lambda + \omega_i c_i}. \tag{1}$$

On the other hand, FBS's optimization objective is

$$\min_z \frac{\lambda_{FBS}}{2} \sum_{i,j} W_{i,j} \left(z_i - z_j\right)^2 + \sum_i c_i \left(z_i - t_i\right)^2. \tag{2}$$

Inspecting the derivative of Eq. (2) at the minimum, we obtain:

$$z_i = \frac{\lambda_{FBS}\, \bar{z}_{N_i} + \dfrac{c_i t_i}{\sum_j W_{i,j}}}{\lambda_{FBS} + \dfrac{c_i}{\sum_j W_{i,j}}}. \tag{3}$$

This optimal solution is, in fact, the same as that of the DTS (see Eqn. (2) from the main paper). However, DTS uses a different solving scheme that only approximates the solution. The domain transform is isometric only when the domain is 1D [2], but we use the pixel space as our domain which is 2D. Hence, DTS must iteratively solve for the ideal solution in the X and Y directions independently. On any given pass, the lateral neighbors of each swept pixel are not considered, and thus only an approximate solution is obtained. In addition, Eq.(1) assumes $\frac{\partial \bar{z}_{N_i}}{\partial z_i} = 0$, which is exact when $W_{ii} = 0$. While filtering using moving average in the domain transform space, however, this is only approximately maintained for pixels with neighborhoods of sufficient weight.

1.1. Derivation for domain transform with L_2 norm

In the following, we provide a detailed derivation for Eqn.(4) and (5) from the paper. Eqn.(4) follows from the definition of domain transform, that the L_2 distance in DT space for two close by points x and $x + h$ is same as L_2 in space (h^2) and in color $((I(x + h) - I(x))^2$. This by definition results in Eqn.(4) which is repeated here.

$$\left(DT(x+h) - DT(x)\right)^2 \overset{\text{def}}{=} h^2 + \sum_{k=1}^{c} \left(I(x+h) - I(x)\right)^2$$

Rearranging the above,

$$\left(\frac{DT(x+h) - DT(x)}{h}\right)^2 = 1 + \sum_{k=1}^{c} \left(\frac{I(x+h) - I(x)}{h}\right)^2$$

taking limit, $h \to 0$

$$\left(DT^{'}(x)\right)^2 = 1 + \sum_{k=1}^{c} \left(I^{'}(x)\right)^2$$

$$DT^{'}(x) = \sqrt{1 + \sum_{k=1}^{c} \left(I'(x)\right)^2}$$

Integrating and assuming $DT(0) = 0$

$$DT(u) = \int_0^u \sqrt{1 + \sum_{k=1}^{c} \left(I'(x)\right)^2 dx}$$

2. Colorization

Levin *et al.* [3] presented a method to convert a grayscale image into a color image using a few color strokes as input. Now we show in detail how we can accomplish the same task in a more computationally efficient manner using our DTS framework. Fig. 1a shows the grayscale image that we convert into the YCbCr color space to extract the Y channel. We use this Y channel as our guide image to compute the bilateral weights $W_{i,j}$. Fig. 1b shows the images with user annotated color strokes, which we convert to YCbCr to extract their Cb and Cr channels. We then apply our DTS twice with each Cb and Cr as target, to estimate the edge-aware and completely filled Cb/Cr channels.

Our method performs the computation at 0.267s/megapixel, which is more than a 3x speedup in comparison to FBS [1]. Recall that FBS uses a grid for optimization. Hence we can get even higher speed-ups when the image resolution is large and the blur windows are small. Fig. 2 shows examples taken from Levin *et al*. [3] with the user annotations, our result, and results from Levin *et al*. [3]. Note that our colorization results are visually indistinguishable from theirs, yet computed significantly faster.

Fig. 3 illustrates how the color propagates as a function of the iterations of gradient descent: (a-g) depicts illustrates the colorization result at 2, 5, 10, 20, 50, 70, 90 iterations, and (h) shows our final result. Note that (e-h) look the same, and further iterations produce only slight changes. This shows that we can adjust the accuracy/time trade-off and stop at 50-70 iterations if time is important. All of the input images were obtained from the author's website [7].

These colorization images are less than 1k$\times$1k in resolution and do not fully exploit the highly parallel nature of our algorithm. To highlight the parallelism of our method, we captured 3k$\times$2k images using an iPhone 6 Plus. We sampled 20% of the pixels at random to create a target image for our method, (Fig. 4a). Since these images are of high resolution, in addition to our result (Fig. 4b), we also show results using a parallel implementation of HFBS [4] which matches our PSNR but is 3.64x slower than our method (Fig. 4c). We will refer to this case as HFBS(equal quality).

To explore the quality/speed trade-off for HFBS we lower the number of iterations to match DTS's run time and then examine the quality of the output of HFBS result (Fig. 4d). We will refer to this as HFBS(equal runtime). This results in visually noticeable artifacts, and this deterioration in quality is also reflected by lower PSNR and SSIM [6] scores for the Cb and Cr channels, as seen in Table 1. In the table, we show SSIM and PSNR scores for our results for the above three cases. Our DTS results and HFBS (equal quality) have similar SSIM and PSNR results. This is also reflected in visually indistinguishable results in (Fig. 4c), (Fig. 4b), Fig.(5 b,c) and Fig.(6 b,c), but HFBS is much slower. When we compare our results with HFBS (equal runtime) in Fig.(4d), we can notice speckles and spots, zooms of which are shown in Fig.(5 b,d) and Fig.(6 b,d).

3. Synthetic defocus from depth

A good synthetic defocus effect requires the estimated depth to align well with the color edges. Figs. 7–9 show near- and far-focus renderings using depthmaps refined using DTS and estimated using MC-CNN [8] for scenes from the Middlebury dataset [5]. The ground-truth defocus was created using the ground-truth depth. The Figs. 7–9 show that our edge-aware optimization framework removes any jarring artifacts and ringing and creates visually pleasing synthetic defocus. These artifacts are the most noticeable at object boundaries and occur due to inaccurate depth estimates in the MC-CNN defocus results.

References

[1] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer, 2016. 1, 2

[2] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, volume 30, page 69. ACM, 2011. 1

[3] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, volume 23 (3), pages 689–694. ACM, 2004. 1, 2, 3, 4

[4] Amrita Mazumdar, Armin Alaghi, Jonathan T Barron, David Gallup, Luis Ceze, Mark Oskin, and Steven M Seitz. A hardware-friendly bilateral solver for real-time virtual reality video. In *Proceedings of High Performance Graphics*, page 13. ACM, 2017. 2, 3, 5

[5] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014. 2

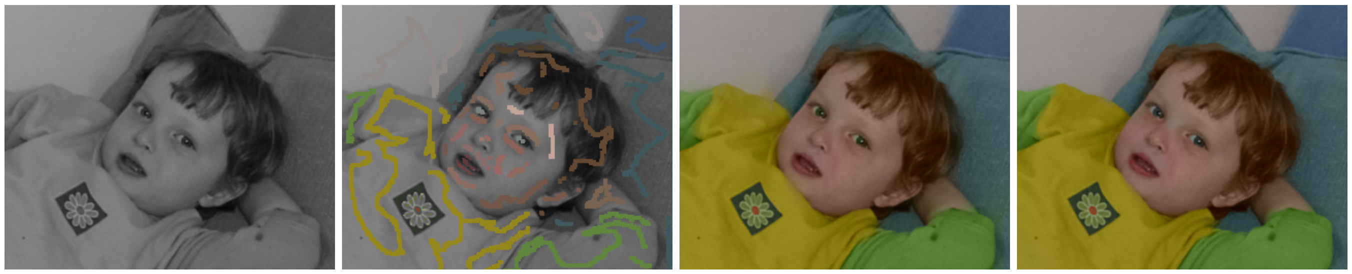
[6] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[7] Weiss Yair. Colorization Using Optimization webpage. `http://www.cs.huji.ac.il/~yweiss/Colorization/`. Online; accessed 12 March 2018. 2

[8] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 2

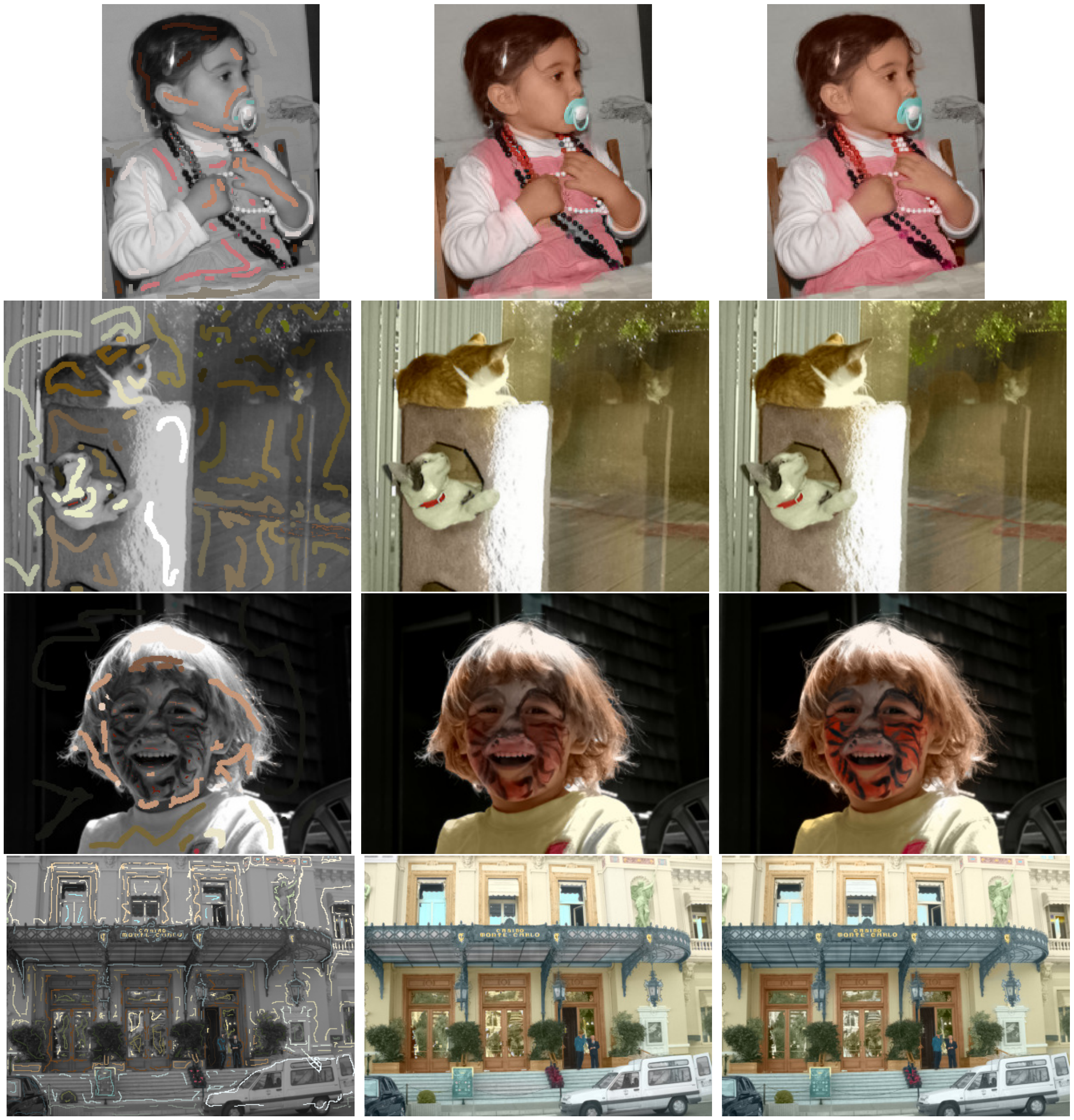
	SSIM (Cb, Cr)		PSNR (Cb, Cr)		Time ms/megapixels
	Fig.(5)	Fig.(6)	dB Fig.(5)	dB Fig.(6)	
DTS (ours)	0.981, 0.983	0.979, 0.975	45.41, 45.60	44.65, 44.00	19.10
HFBS equal quality [4]	0.979, 0.976	0.973, 0.965	45.79, 45.01	45.10, 44.20	69.54
HFBS equal runtime [4]	0.948, 0.944	0.915, 0.884	43.60, 42.91	42.30, 41.18	19.09

Table 1: The table lists the SSIM and PSNR scores for Cb anc Cr channels for the high-resolution images. HFBS can match the performance of DTS (ours) but, takes 3.64x more time. If we limit the time taken by HFBS to that of DTS, HFBS has worse PSNR and SSIM scores.



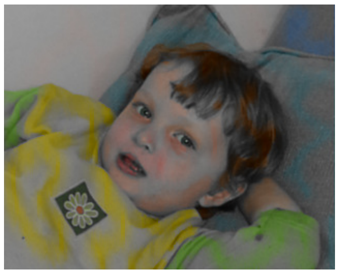
(a) Grayscale image (b) Marked color strokes (c) Our result (d) Levin *et al*. [3]

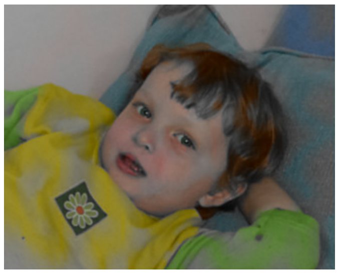
Figure 1: Colorization: (a) input grayscale image (our reference image), (b) user-annotated strokes (our target image), (c) our result using DTS, and (d) result of Levin *et al*. [3]. Note that (c) and (d) look the same with only small differences in the hair.

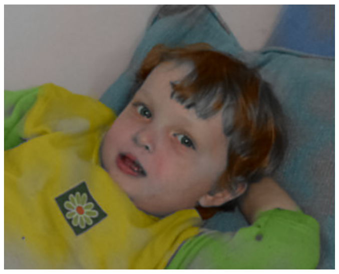


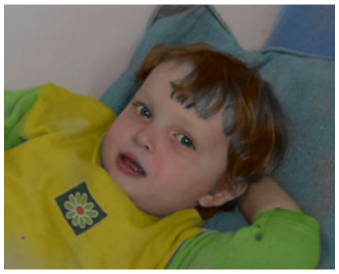
(a) User annotated scribbles (b) DTS (ours) (c) Levin *et al*. [3]

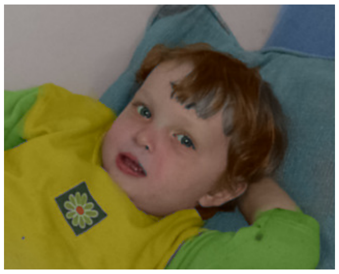
Figure 2: Colorization: (a) user-annotated color scribbles, (b) our results, and (c) results using Levin *et al*. [3].

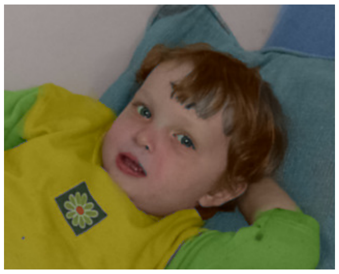

(a) Iteration 2

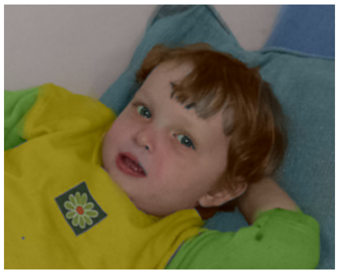

(b) Iteration 5


(c) Iteration 10


(d) Iteration 20


(e) Iteration 50


(f) Iteration 70


(g) Iteration 90

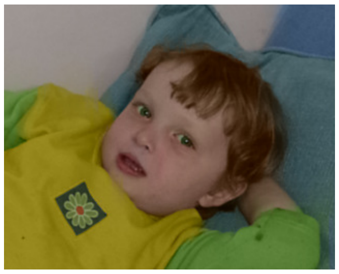
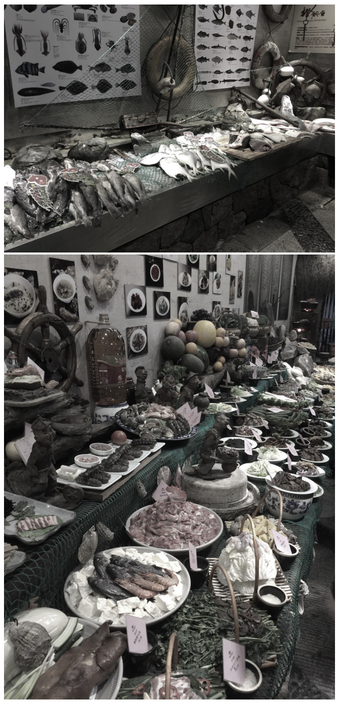
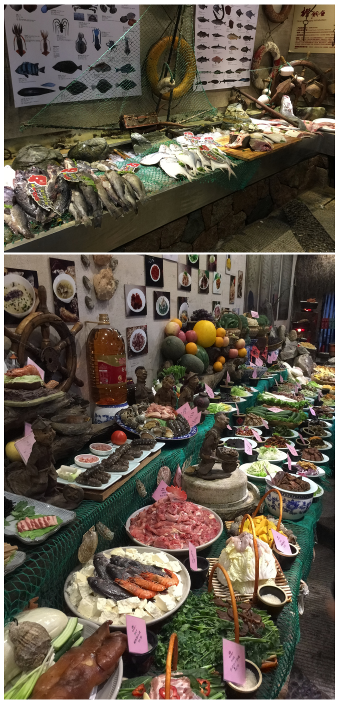
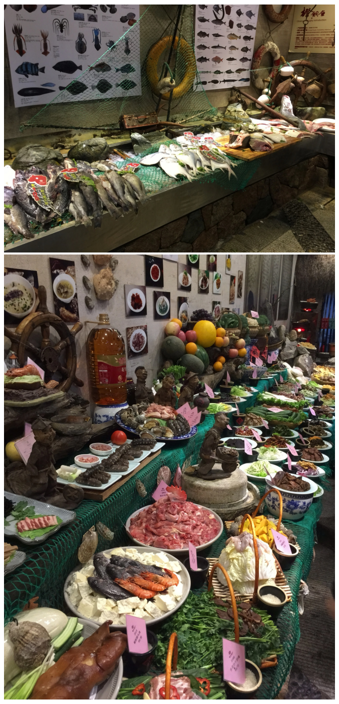

(h) Our result

Figure 3: Propagation of colors: (a-h) shows how the colors from the user-annotated strokes propagate through the image while not crossing strong image edges using DTS. After iteration 50, the changes are small and can be ignored as a trade-off for speed.


(a) 20% randomly sampled color.


(b) DTS (ours)


(c) HFBS [4], equal PSNR.

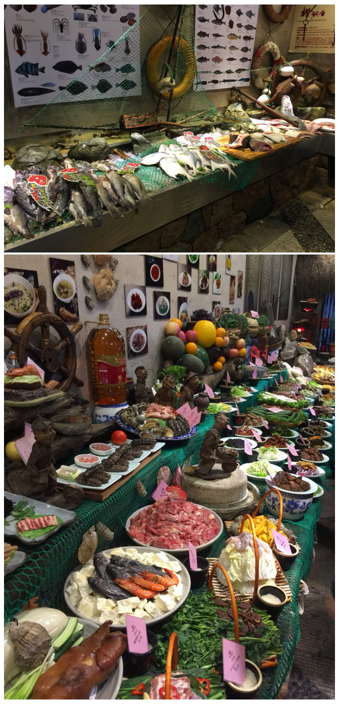
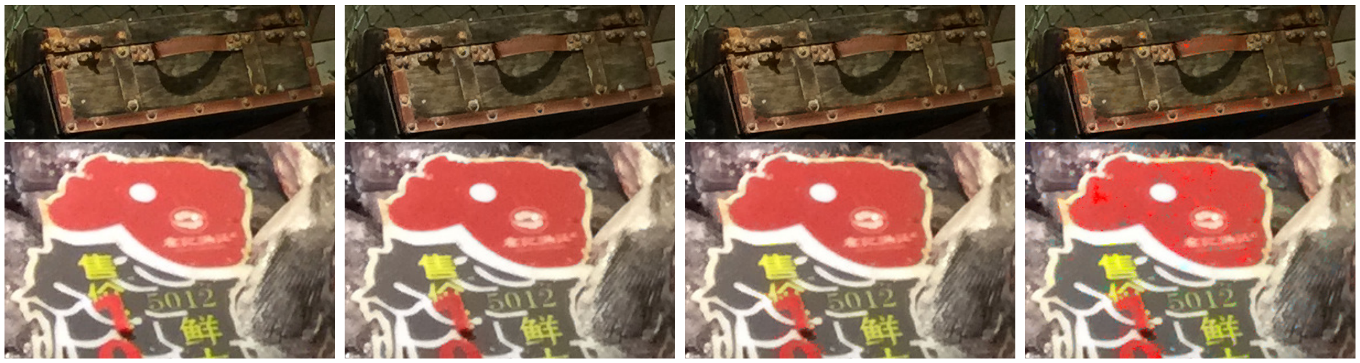
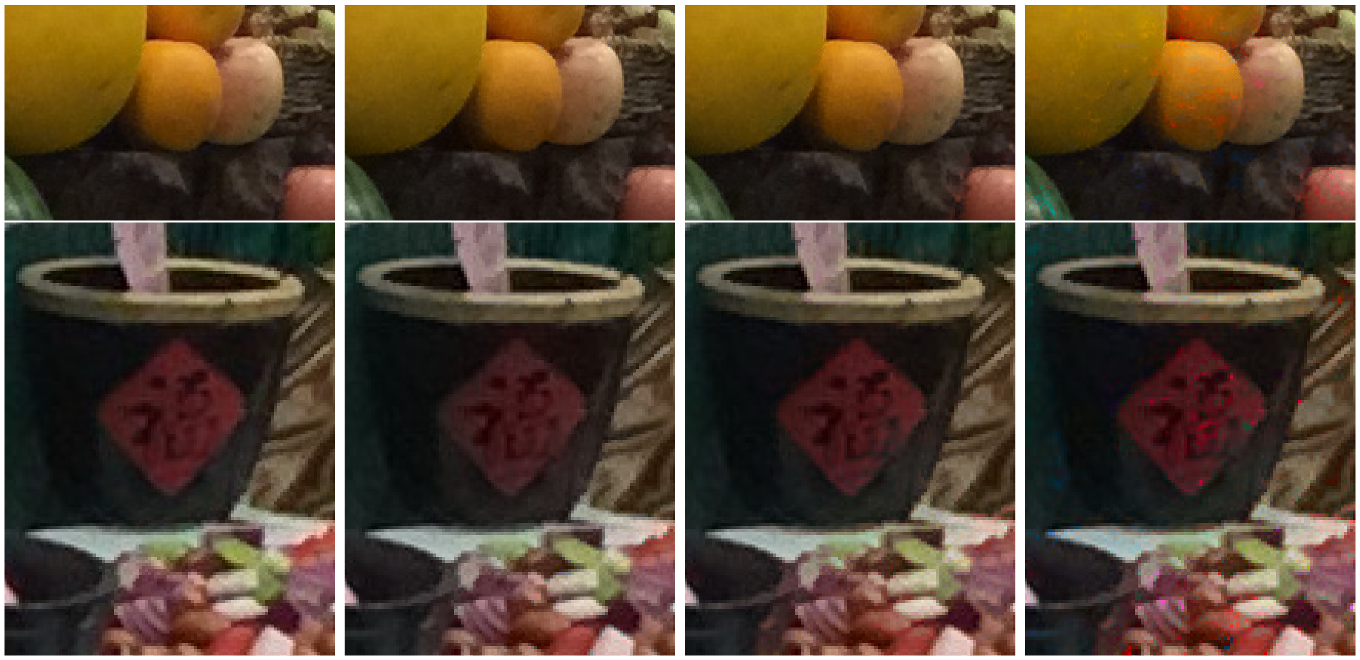

(d) HFBS [4], equal time budget.

Figure 4: Colorization for high-resolution images. (a) The target images retain color in only 20% of the pixels. (b) DTS – our result. (c) HFBS can match the performance of DTS but, takes 3.64x more time. (d) When HFBS is limited to the time of DTS, it has substantially worse PSNR and SSIM scores.



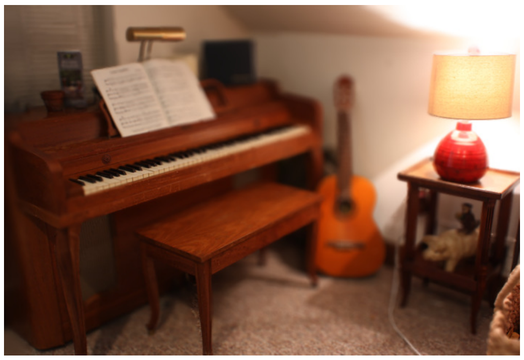
(a) Ground truth. (b) DTS (c) HFBS equal quality (d) HFBS equal time

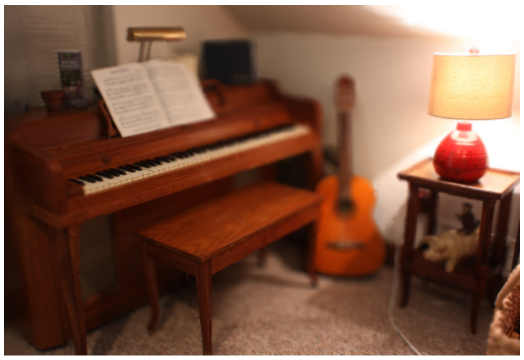
Figure 5: Zoom-ins for Fig. 4 first row: Notice the speckles in (d) on the handle of the chest and red portion of the meat.

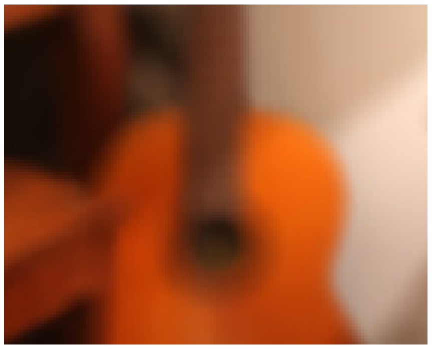


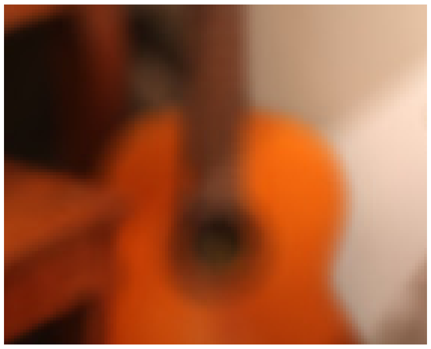
(a) Ground truth. (b) DTS (c) HFBS equal quality (d) HFBS equal time

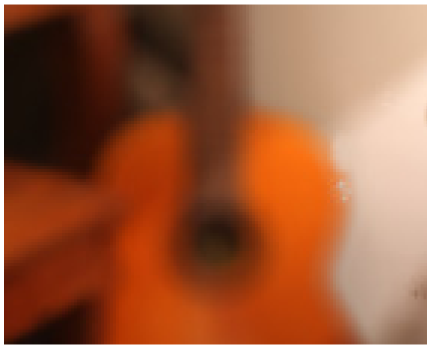
Figure 6: Zoom-ins for Fig. 4 second row: Notice the speckles in (d) reddish spots on the orange, and red spots on the vase.

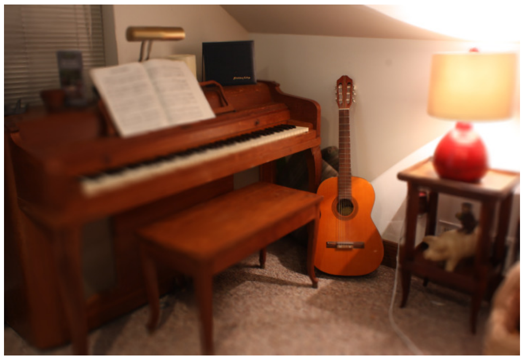

(a) DTS (ours)

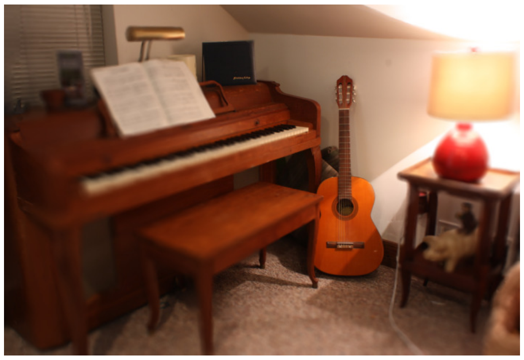

(b) MC-CNN

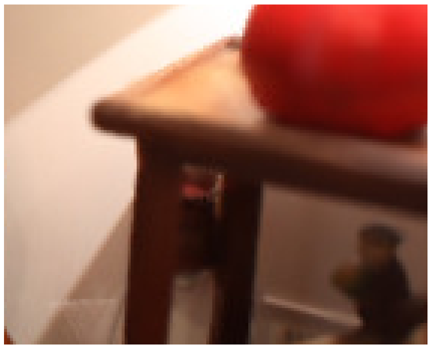

(c) Ground truth, near focus.

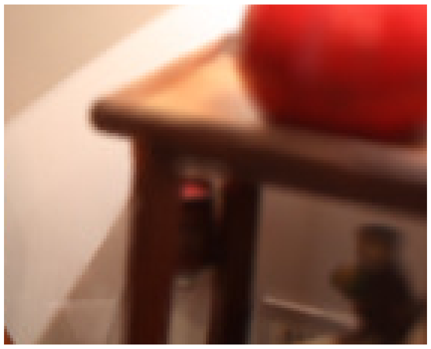

(d) DTS (ours)


(e) MC-CNN


(f) DTS (ours)


(g) MC-CNN


(h) Ground truth, far focus.


(i) DTS (ours)

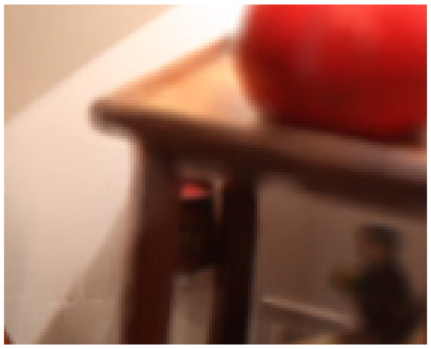
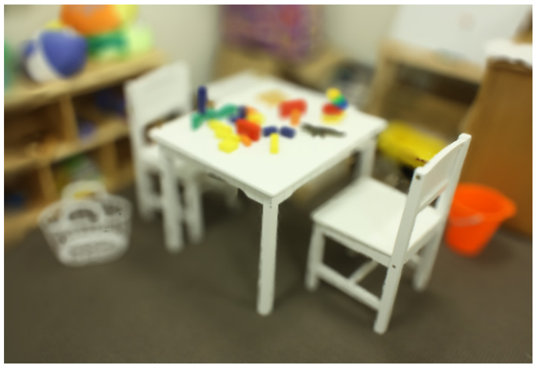
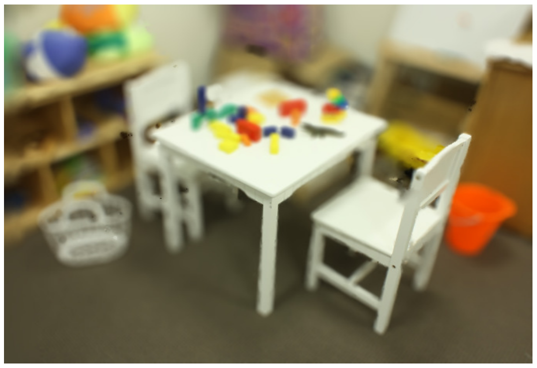
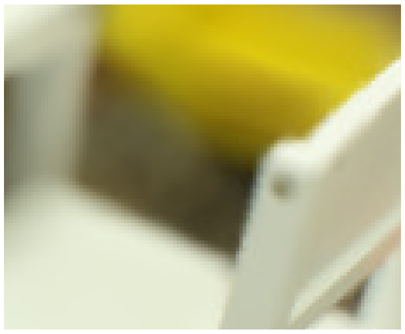
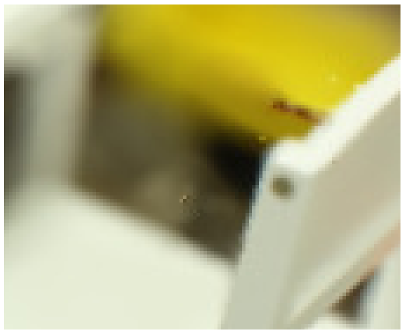

(j) MC-CNN

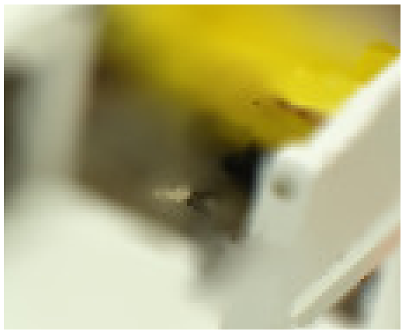
Figure 7: PianoL scene: (a,b) Our and MC-CNN's result where the stool is in focus, (c-e) shows a zoomed-in region. (f,g) The guitar is in focus. In (e) and (j), notice the rough edges at the right side of the guitar and on the left side of the leg of the table, which are reduced in our results (d,i).

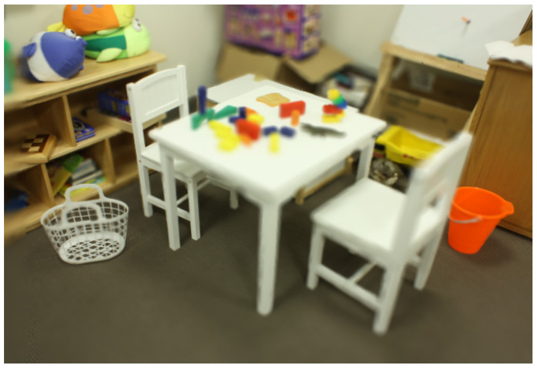

(a) DTS (ours)

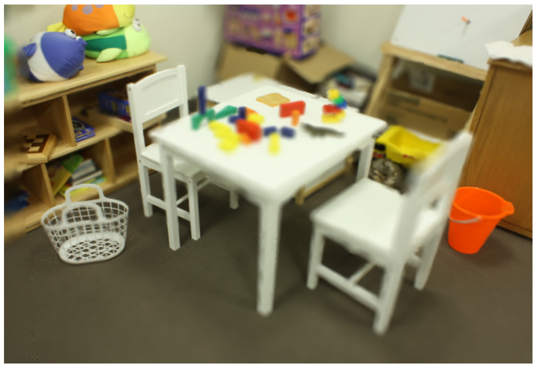

(b) MC-CNN

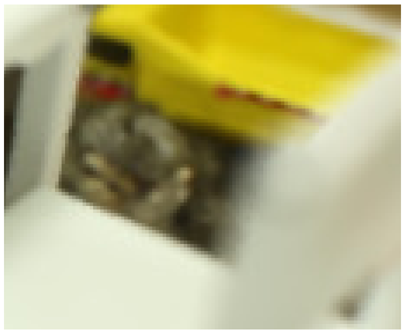

(c) Ground truth, near focus.

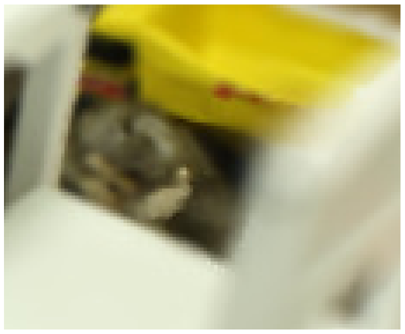

(d) DTS (ours)


(e) MC-CNN


(f) DTS (ours)


(g) MC-CNN


(h) Ground truth, far focus.


(i) DTS (ours)

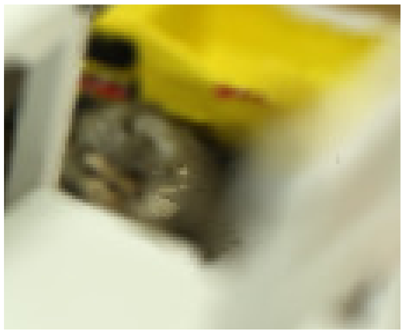
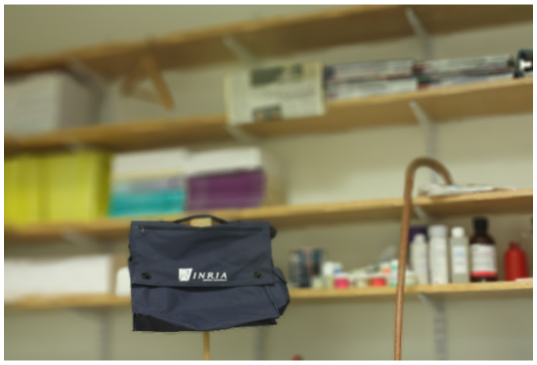
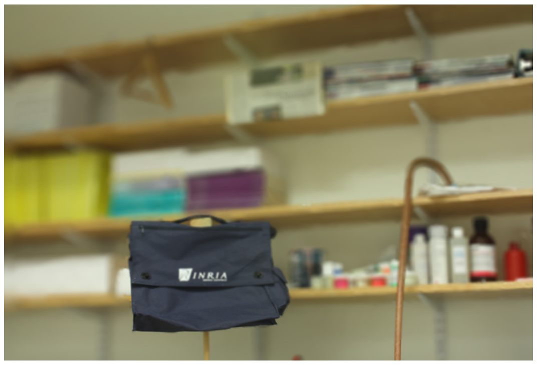
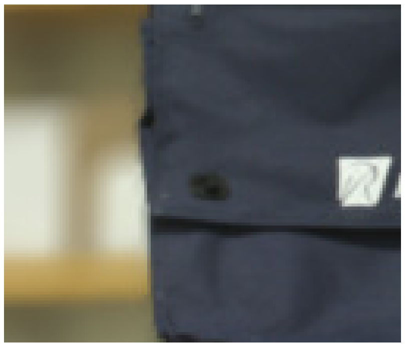
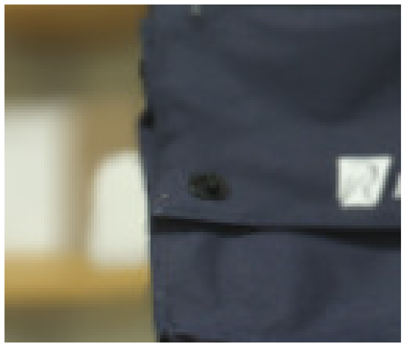

(j) MC-CNN

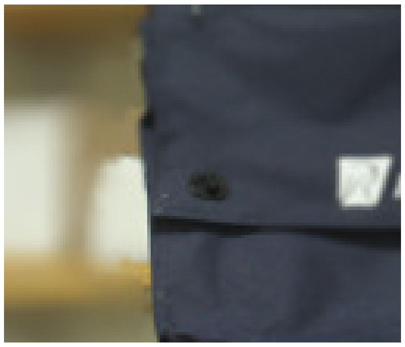
Figure 8: PlaytableP scene: (a,b) Oour and MC-CNN's result where the front of the table is in focus, (c-e) shows a zoomed-in region. (f,g) The orange bucket is in focus. In (e) and (j), notice the jarring changes in blur due to incorrect depth, which is reduced in our results (d,i).

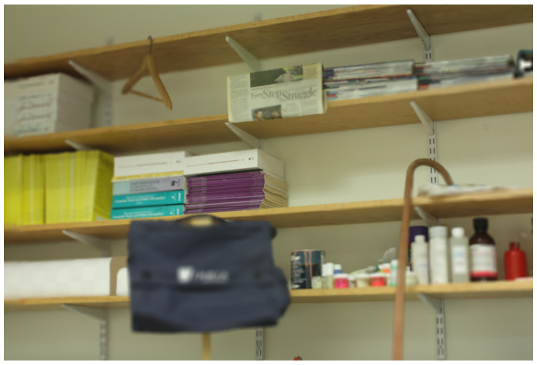

(a) DTS (ours)

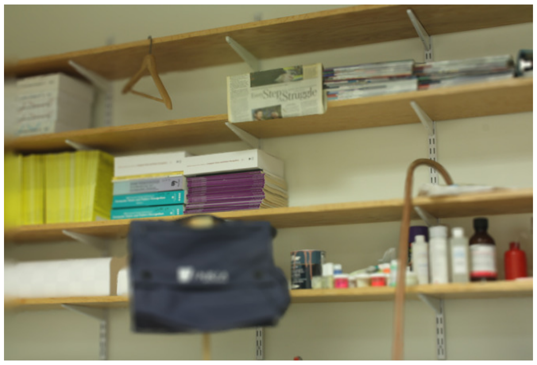

(b) MC-CNN

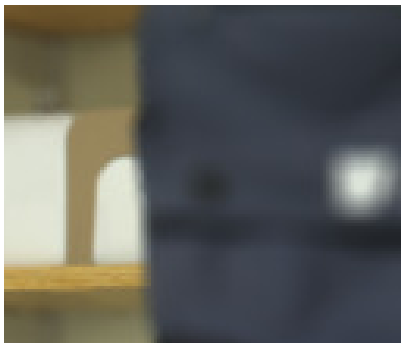

(c) Ground truth, near focus.

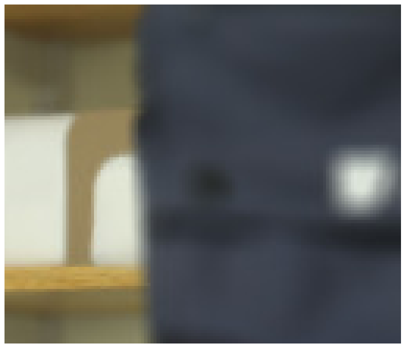

(d) DTS (ours)


(e) MC-CNN


(f) DTS (ours)


(g) MC-CNN


(h) Ground truth, far focus.


(i) DTS (ours)

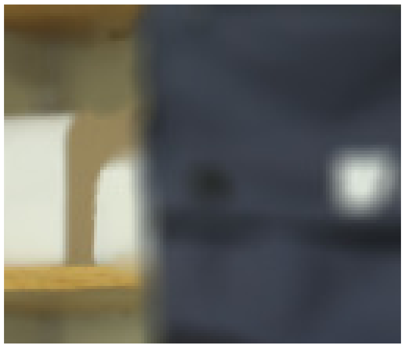

(j) MC-CNN

Figure 9: Shelves scene: (a,b) Our and MC-CNN's result where the blue bag is in focus, (c-e) shows a zoomed-in region. (f,g) The hanger is in focus. In (e) and (j), notice the ringing near the boundary of the bag, which is removed in our results (d,i).