Disambiguating Monocular Depth Estimation with a Single Transient: Supplemental Information

Anonymous ECCV submission

Paper ID 3668

1 Ablation study on number of SID bins

We conducted an ablation study on the effect of the number of SID bins [2] on both runtime and RMSE. We performed this analysis using SPAD data with a signal-to-background (SBR) of 100, simulated on the test set of NYU Depth v2. We used DenseDepth [1] for our MDE CNN. Only the histogram matching portion was timed, not the CNN nor the denoising pipeline.

# of sid bins	RMSE	Approx. Time/image (sec)
70	0.351	0.24
140	0.346	0.63
210	0.345	1.12
280	0.345	1.84

Fig. 1: Effect of number of SID bins on RMSE and runtime. The marginal improvement in RMSE is offset by the increase in runtime as the number of bins grows.

2 Ablation study on effect of reflectance estimation

We conducted an ablation study on whether the use of a reflectance estimate has an impact on the runtime and quality of the solution. We performed this analysis using SPAD data with a signal-to-background (SBR) of 100, simulated on the test set of NYU Depth v2 and using DenseDepth [1] for our MDE CNN. Only the histogram matching portion was timed, not the CNN nor the denoising pipeline.

Intensity-weighted histogram	Intensity-aware pixel movement	Avg. RMSE	Time per image (sec)
Yes	Yes	0.346	4.6
	No	0.346	0.6
No	Yes	0.444	4.7
	No	0.444	0.6

Fig. 2: Effect of reflectance modeling on RMSE and runtime. When the SPAD is simulated with the reflectance info but no reflectance estimate is used to generate a weighted histogram from the CNN depth map, the results are significantly worse. Furthermore, once the pixel movement matrix has been computed, the pixel movement procedure need not take into account the weights of the pixels being moved, since doing so provides no improvement and can take appreciably longer than a vectorized implementation that does not take pixel weights into account.

3 Pseudocode, pixel shifting, and dither artifacts

We give pseudocode for our algorithm here. In the first part of our algorithm we compute the pixel shifting matrix mapping the histogram h_s (computed from the initial depth map and reflectance estimate)to h_t (computed from the captured transient).

Algorithm 1 Find Pixel Movement MOVE TO SUPPLEMENT!

```
procedure FindPixelMovement($h_s$ of length $M$, $h_t$ of length $N$)
    Initialize $T$ as an $M \times N$ array of zeros.
    for $m$ in $1, \ldots, M$ do
        for $n$ in $1, \ldots, N$ do
            $p_s \leftarrow \sum_{i=1}^{n-1} T[m, i]$
            $p_t \leftarrow \sum_{i=1}^{m-1} T[i, n]$
            $T[m, n] \leftarrow \min(h_s[m] - p_s, h_t[n] - p_t)$
        end for
    end for
    return $T$
end procedure
```

Given this pixel movement matrix T, we apply the appropriate movements to the initial depth map I. The pixels of the image I take depth bin values in $\{0, \ldots, K-1\}$.

Algorithm 2 Move Pixels

```
procedure MovePixels(input image $I$ size $M \times N$, pixel movement matrix $T$ of size $K \times K$)
    for $k$ in $0, \ldots, K-1$ do
        $p[k,:] \leftarrow T[k,:] / \sum_{i=1}^{K} T[k,i]$
    end for
    for $m$ in $1, \ldots, M$ do
        for $n$ in $1, \ldots, N$ do
            Sample $k'$ according to $p[I[m,n],:]$.
            $I[m,n] \leftarrow k'$.
        end for
    end for
    return $I$
end procedure
```

Because the pixel shifting process in Algorithm 2 contains a sampling step, it is possible for *dither artifacts* to appear in the output image I, as shown in figure 3. Specifically, when there are multiple possible output depth bins for a given input depth bin, and a large region of equal depth in the input image, the randomness in the pixel shifting algorithm will distribute the pixels of large, equal-depth region in the input across the multiple possible output depth bins in a random fashion.

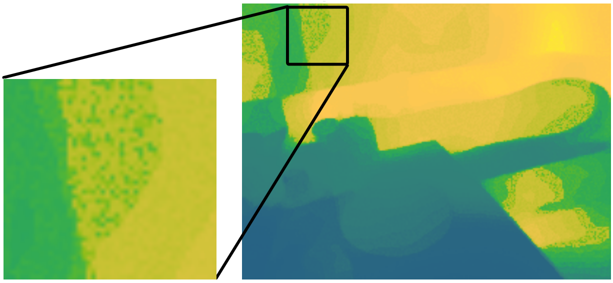

Fig. 3: Example of dither artifacts. Sometimes, when our histogram matching is applied to images with large regions of similar depths, dither artifacts will occur.

4 Additional results on NYU Depth v2

Figures **??**–**??** show additional results for our method on the NYU Depth v2 dataset when the depth estimate is initialized with the DenseDepth [1] (Figures **??**–**??**), DORN [2] (Figures **??**–**??**) and MiDaS [3] (Figures **??** – **??**) monocular depth estimators.

We compare the output of the network z_0, the median-rescaled network output (where the depth map z_0 is scaled pixel-wise by a scalar $\frac{\text{median}(z_{GT})}{\text{median}(z_0)}$, z_{GT}

being the ground truth depth map), the network output matched to the ground truth depth histogram, and the output of our histogram matching method under a signal-to-background ratio (SBR) of 100. We use the luminance of the RGB image as our reflectance map for both SPAD simulation and histogram matching. We show absolute difference maps and also give the root-mean-square error (RMSE) for each example.

5 Additional results for hardware prototype

Figures **??**–**??** show all the captured results when the depth estimate is initialized with the MiDaS [3] (Figures **??**–**??**), DenseDepth (Figures **??**–**??**), and DORN (Figures **??**–**??**). We compare the output of the network z_0, the mean-rescaled network output where the depth map z_0 has been scaled pixel-wise by the scalar $\frac{\text{median}(h_{target})}{\text{median}(z_0)}$ (h_{target} is the processed SPAD transient), and the output of our method. As our laser is red, we use the R channel of the RGB image as our reflectance map. We show absolute difference maps and also give the root-mean-square-error (RMSE) for each example.

Black pixels in the ground truth depth correspond to locations where our scanner was unable to produce an accurate depth estimate (this can occur for a variety of reasons including dark albedo and surface specularity). These pixels are masked off and not used in the RMSE calculation, and appear as an absolute difference of 0 in the difference maps.

References

1. Alhashim, I., Wonka, P.: High quality monocular depth estimation via transfer learning. arXiv:1812.11941v2 (2018)
2. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proc. CVPR (2018)
3. Lasinger, K., Ranftl, R., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv:1907.01341 (2019)