Supplemental Material MantissaCam: Learning Snapshot High-dynamic-range Imaging with Perceptually-based In-pixel Irradiance Encoding

Haley M. So, Julien N. P. Martel, Piotr Dudek, and Gordon Wetzstein

1 Encoding Details

We mathematically compare the mantissa and modulo encodings. The image formation model of the MantissaCam is

$$I_{\text{sensor}}(x,y) = q\left(\text{mod}\left(\log_{\tilde{\alpha}}\left(I(x,y)\right), I_{\text{max}}\right)\right) + \eta, \quad (1)$$

while the image formation model for the Modulo Camera is

$$I_{\text{sensor}}(x,y) = q\left(\text{mod}\left(I(x,y), I_{\text{max}}\right)\right) + \eta, \quad (2)$$

As shown, the Mantissa encoding can be thought of as first taking the log of a signal before taking the modulo while the Modulo encoding is the modulo of the linear signal.

2 Pipeline Details

2.1 Mantissa Dataset creation

There are several ways to encode the mantissa. When working with synthetic data, the simplest way is to just take the $\log$ of the signal and then take the modulo of the resulting value. Only after the pixel saturates do we take the $\log$ to simulate the mantissa. When we saturate the pixel, the subsequent wrap will require twice the intensity to wrap again. To create the training dataset, we create the mantissa image and the corresponding winding number image. For each pixel ij,

$$\text{mantissa}_{ij} = I_{\text{sensor}_{ij}} = \begin{cases} I_{ij}, & \text{if } I_{ij} < I_{max} \\ \log_{\tilde{\alpha}}(I_{ij}) \% I_{max}, & \text{otherwise.} \end{cases} \quad (3)$$

$$\text{winding number}_{ij} = W_{ij} = \begin{cases} 0, & \text{if } I_{ij} < I_{max} \\ \lfloor \log_{\tilde{\alpha}}(I_{ij}) \rfloor + 1, & \text{otherwise.} \end{cases} \quad (4)$$

where $\%$ denotes the modulo operation and $\lfloor \cdot \rfloor$ denotes the floor function. For our dataset and experiments, we set $\alpha = 2$. To reconstruct the signal given the winding number,

$$\widetilde{I_{ij}} = \begin{cases} I_{\text{sensor}_{ij}}, & \text{if } W_{ij} = 0 \\ I_{\text{max}} \cdot \alpha^{I_{\text{sensor}_{ij}}/I_{\text{max}} + W_{ij} - 1}, & otherwise. \end{cases} \quad (5)$$

- *https://www.computationalimaging.org/publications/mantissacam*

2.2 Network Architecture

In this subsection, we describe the architecture for the single pass winding number prediction network (also see Figure 1). The extracted edges from the edge prediction network, along with the mantissa image, are fed into the network via feature extraction by a 7×7 convolutional layer, an instance norm, ReLU, and a non-local block for the extracted edge features. These images are then concatenated and sent through a squeeze-and-excitation block to perform dynamic channel-wise feature recalibration. The base network is an attention unet, pioneered by Oktay et al. [1]. The backbone is the U-Net where the expanding path has attention gates added, along with the skip connections. Skip connections allow features extracted from the contracting path to be used in the expanding path. The attention block places more emphasis on the features of the skip connections.

3 Training Procedures

3.1 RGB training on HDR images

For training our network for RGB, we trained the edge network for 400 epochs on the synthetic data at a learning rate of .0001 using an ADAM optimizer in Pytorch. From a dataset of 593 images, we randomly split it into 400 training images and 193 test images. We augment the training images by scaling the HDR image and calculating the corresponding mantissa and winding numbers.

3.2 Training Procedure for SCAMP-5 Prototype

We retrain the edge prediction network for the captured grayscale dataset as described in the paper. Both UnModNet and our method use the same edge prediction network. The other parts of the respective pipelines are retrained on the captured dataset using a similar procedure as used for the synthetic data described above.

3.3 Baseline Comparison

Graph Cuts: Graph Cuts was implemented following the original ModuloCam paper [2] using the same

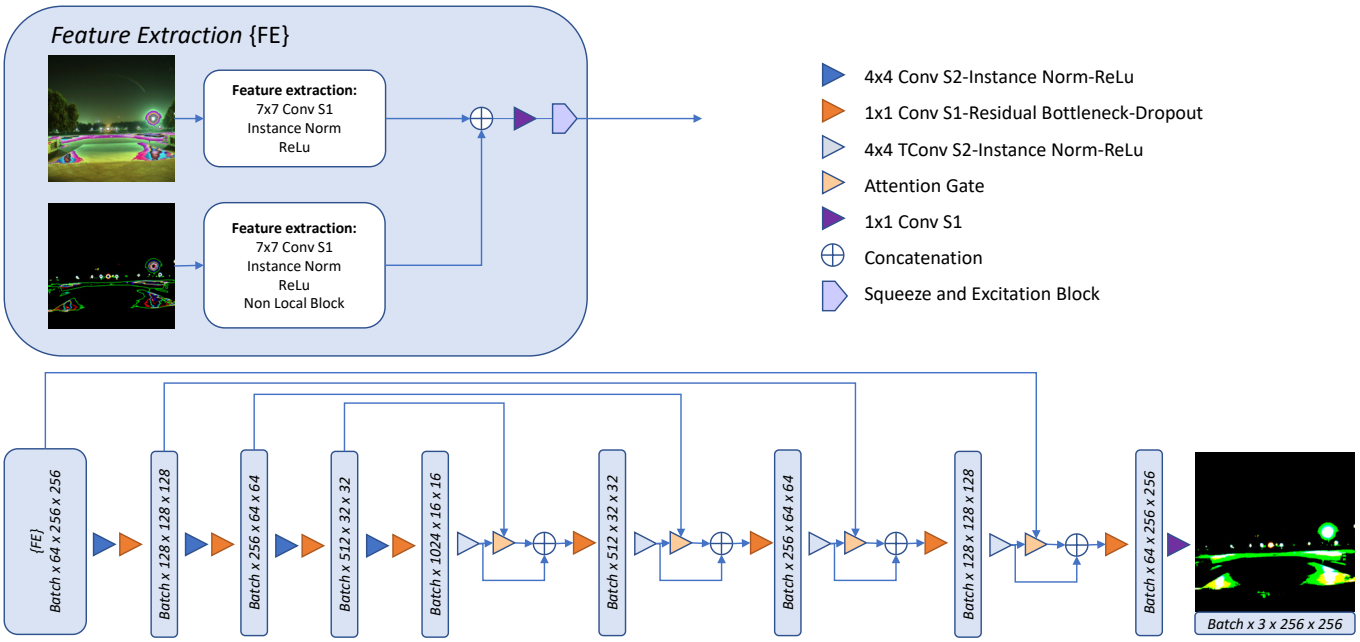

Fig. 1. Attention UNet architecture of the winding number prediction network.

custom potential function. Reaching out to the authors confirmed the method, which can be successful for some clearly wrapped modulo images, however requires delicate parameter tuning for each of the many layer unwrappings of each image. We chose a set of parameters to best unwrap the whole set. PSNR, SSIM, and MS-SSIM scores were comparable to those found in UnModNet [3] when they implemented the MRF algorithm.

Modulo Encoding with UnModNet: We retrained UnModNet, the state of the art for unwrapping modulo images, with the same training process and same dataset as in [3] and results were comparable to those reported in the paper. In areas of dense wrappings, the pipeline struggles to stop unwrapping, leading to patches of white.

Mantissa Encoding with UnModNet: One of our baseline experiment is to use the pipeline of UnModNet with the forward imaging model of MantissaCam. We trained the pipeline using the same training procedure as described above. We noted the layer-by-layer unwinding did not work well with the reconstruction from the mantissa encoding as errors in winding number manifest in exponentially bad errors. Indeed, missing a wrap results in much worse errors in MantissaCam (because of the exponential function used when reconstructing) than in ModuloCam, resulting in huge artifacts. Besides, the nature of the layer-by-layer unwrapping is prone to propagating errors.

Modulo Encoding with Our Network: To combat propagation of errors in unwinding, we directly predict the winding number in a single pass through an attention-unet instead of predicting a mask. Again, we train using the same training procedure as UnModNet. Results are promising, however, the network still struggles when the modulo image has very tight wrappings (of the order of 1–2 pixels width).

Mantissa Encoding with Our Network: Introducing the mantissa allows us to spatially spread out the wraps as we get to higher and higher irradiance levels. This leads to preservation of more detail. Results comparing these methods, excluding the UnModNet for the mantissa, are shown in the paper and in the additional figures in Section 4.

3.4 Additional Implementation Details

Inference time for our method is much faster than for the UnModNet or graph cuts due to the single pass architecture, as opposed to the iterative unwrapping that can unwrap as high as the default of 15 max iterations.

Metrics: We compare the full reconstructed HDR image with the ground truth HDR image to calculate PSNR and Q-Score (2). We then tonemap both the ground truth and the predicted HDR images, all using the Reinhard Algorithm with gamma = 1, intensity = 1. The tonemapped images are then compared to calculate SSIM, MS-SSIM, and LPIPS values. The standard equations for PSNR and SSIM are shown below.

Let I be the reference HDR image and $\widetilde{I}$ be the reconstructed HDR image. Let I_t and $\widetilde{I}_t$ be the tonemapped I and $\widetilde{I}$. The peak-signal-to-noise (PSNR) is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\max(I)^2}{\text{MSE}(I, \widetilde{I})}\right) \tag{6}$$

where the mean-squared-error (MSE) is:

$$\text{MSE}(I, \widetilde{I}) = \frac{1}{mn} \sum_{i}^{m-1} \sum_{j}^{n-1} \left(I(i,j) - \widetilde{I}(i,j) \right)^2 \quad (7)$$

for a monochrome image. For RGB images, the MSE is calculated for each channel and averaged. The structural similarity (SSIM) between two images is calculated by

$$\text{SSIM}(I_t, \widetilde{I}_t) = \frac{(2\mu_{I_t}\mu_{\widetilde{I}_t} + c_1)(2\sigma_{I_t\widetilde{I}_t} + c_2)}{(\mu_{I_t}^2 + \mu_{\widetilde{I}_t}^2 + c_1)(\sigma_{I_t}^2 + \sigma_{\widetilde{I}_t}^2 + c_2)} \quad (8)$$

where μ_{I_t}, $\mu_{\widetilde{I}_t}$,σ_{I_t}, $\sigma_{\widetilde{I}_t}$, and $\sigma_{I_t\widetilde{I}_t}$ are averages, variances, and covariance of I_t and $\widetilde{I}_t$. $c_1 = k_1 L^2$, $c_2 = k_2 L^2$ stabilize the denominator. $k_1$and k_2 are small constants $ll1$, 0.01 and 0.03 by default. L is the DR of the pixel-values. While PSNR (↑) is a measure of absolute closeness, the SSIM and MS-SSIM are perception based models that measure the perceived similarities of luminance, contrast, and structure. The Learned Perceptual Image Patch Similarity (LPIPS) (↓) metric evaluates the distance between images. Lower numbers means more perceptually similar. HDR Visual Difference Predictor (HDR-VDP3) Quality Scores (↑) predicts image quality degradation with respect to the reference image. PSNR and HDR-VDP3 Q-scores were calculated on the full HDR reference and reconstructed images. SSIM, MSSIM, and LPIPS metrics were computed on tonemapped images. Additional spatial maps are shown in Figure 2.

4 Additional Details of Experimental Results

Currently, mantissa images cannot be directly captured in SCAMP-5. However we implemented a procedure on SCAMP-5 to capture modulo images as described in the main paper.

We also implemented a bracketed exposure procedure directly on the camera in order to get reference HDR images. Exposure bracketing is performed by capturing 5 images doubling the exposure time between each exposure, starting from a configurable short exposure time.

5 Additional Results

See Figures 4 and 5 for additional results. From left to right, each row shows the modulo image, the graph cuts method, UnModNet + modulo, Ours + modulo, the mantissa image, Ours + mantissa, and the ground truth image. All tonemapped images follow the tone-mapping described in Section 3. Additionally, we performed a study on the effects of noise. Our networks and comparisons were not trained on noisy images, so as we increase additive Gaussian noise, the PSNRs decrease, as shown in figure 3. However, if the networks are trained with real data, they are able to capture the effects of noise, as demonstrated by the results from our reconstruction algorithm on the captured images with our prototype. Figures 6–10 show additional results for captured data with the SCAMP-5.

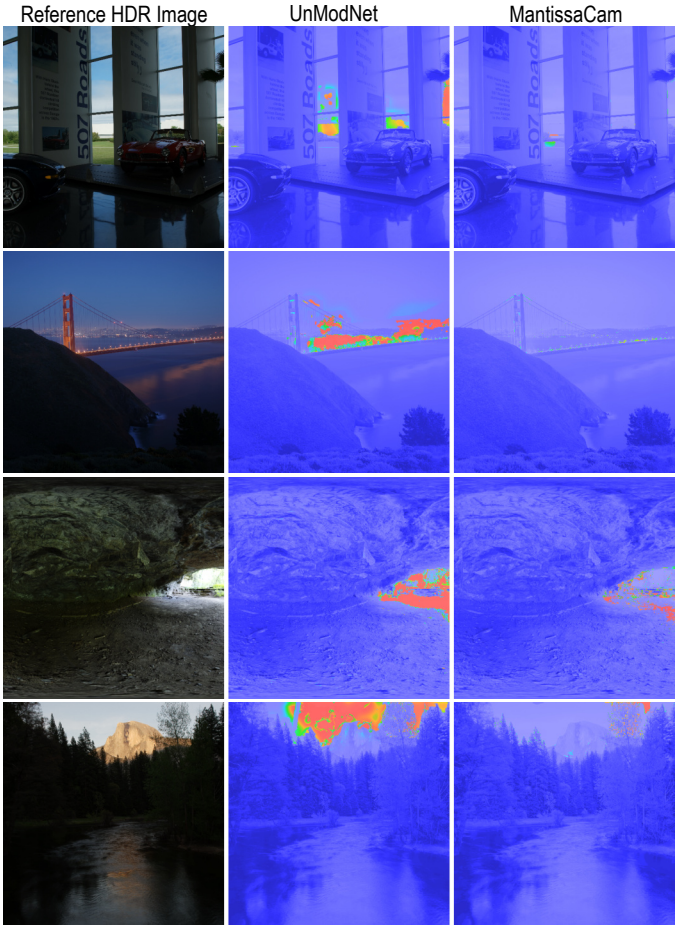

Fig. 2. Additional spatial maps. From left to right, we show the tonemapped ground truth, the HDR-VDP3 spatial map for UnModNet's reconstruction, and the spatial map for our reconstruction. The maps show the contrast-normalized per-pixel difference weighted by the probability of detection. Red corresponds to a large perceived difference and blue a low perceived difference.

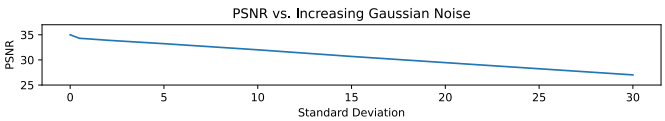

Fig. 3. The effect of Gaussian read noise. The range for image values is [0, 255].

References

[1] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," 2018.

[2] H. Zhao, B. Shi, C. Fernandez-Cull, S. Yeung, and R. Raskar, "Unbounded high dynamic range photography using a modulo camera," in *Proc. ICCP*, 2015, pp. 1–10.

[3] C. Zhou, H. Zhao, J. Han, C. Xu, C. Xu, T. Huang, and B. Shi, "Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging," *Advances in Neural Information Processing Systems*, 2020.

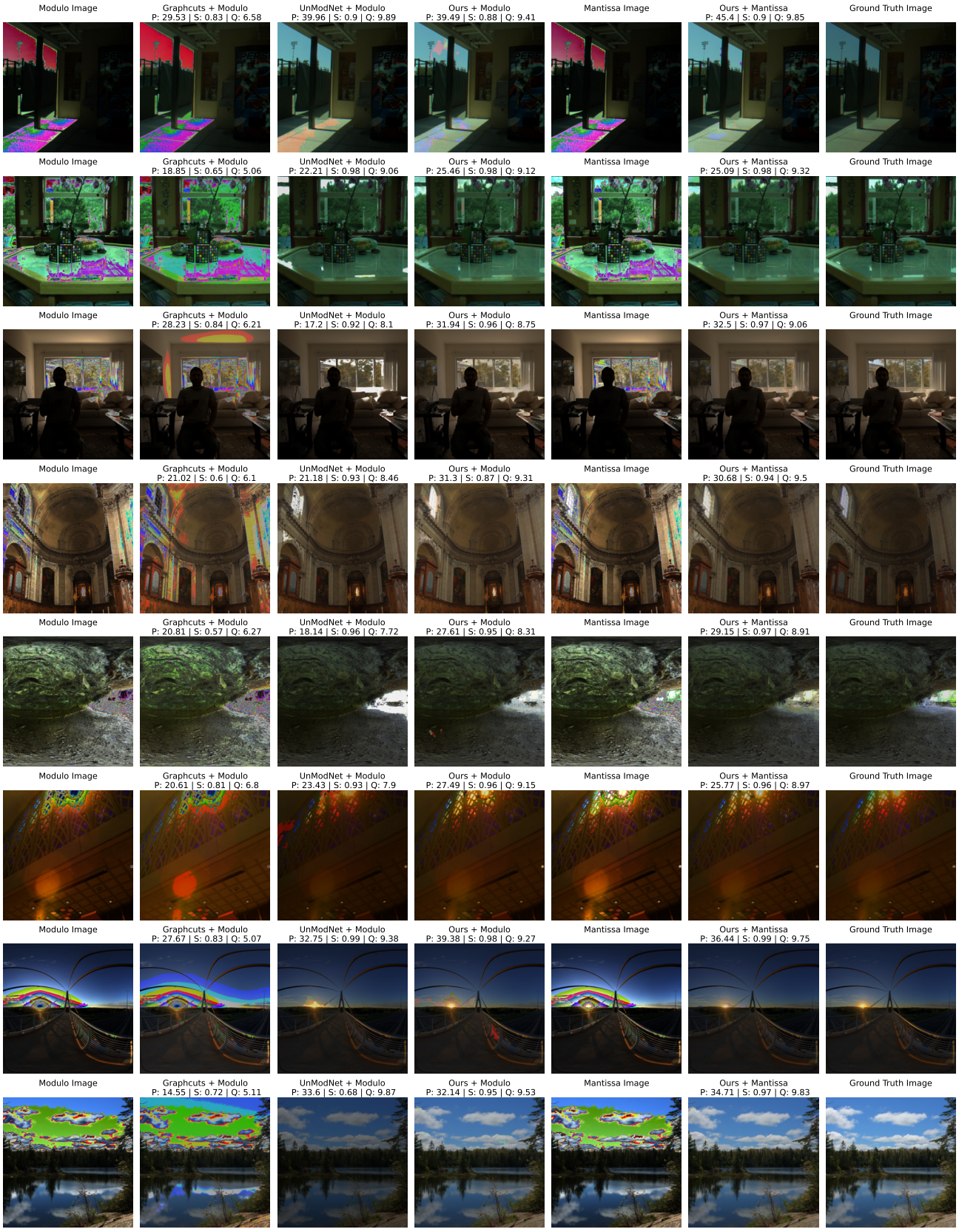

Fig. 4. More results showing the comparison between different baselines and encodings. Ours + mantissa is better able to keep details in the high intensity areas. PSNR (P), SSIM (S), and Q-Scores (Q) are shown about the reconstructed images.

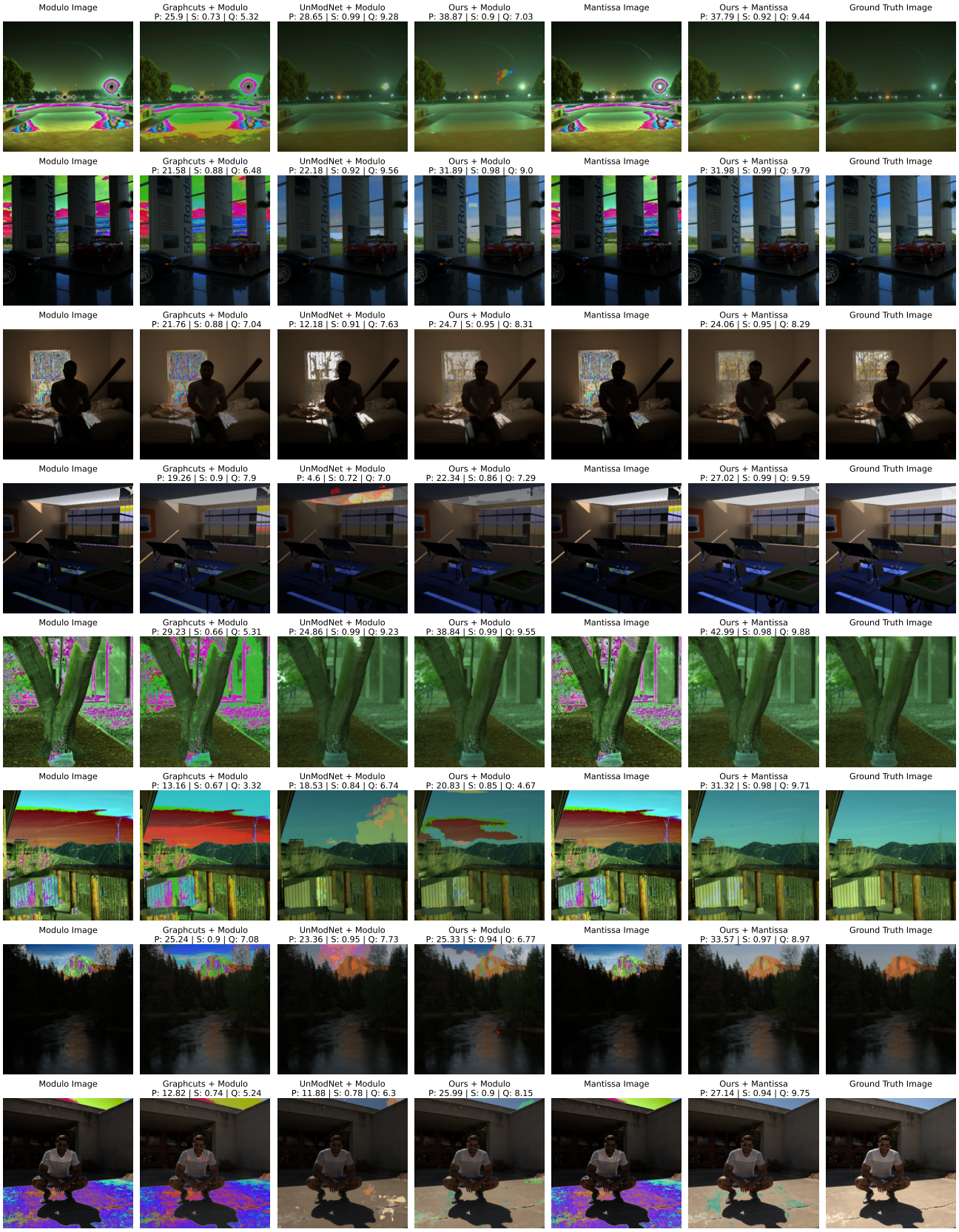

Fig. 5. More results comparing the different reconstruction and encoding methods.

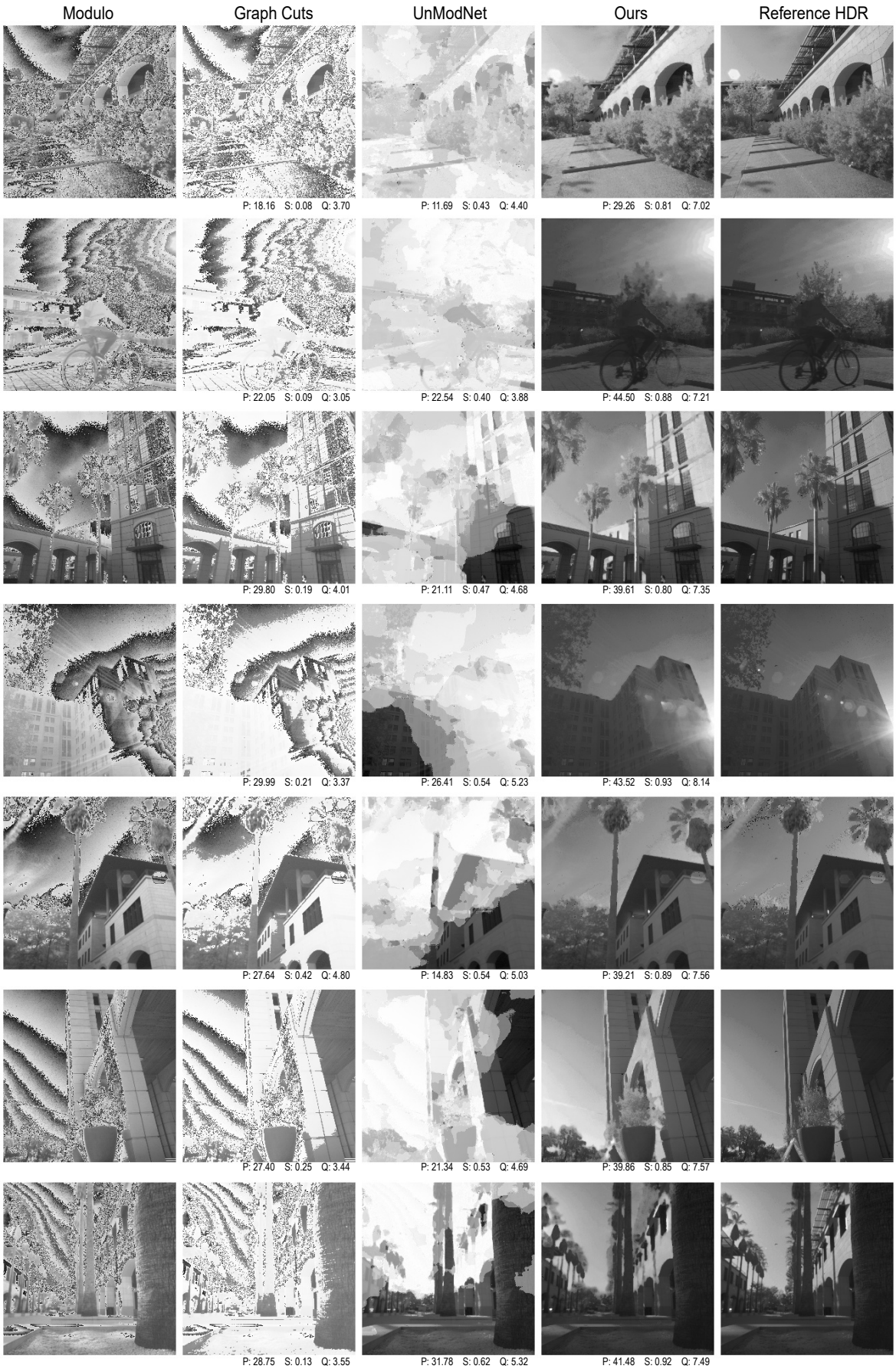

Fig. 6. Comparisons on captured data.

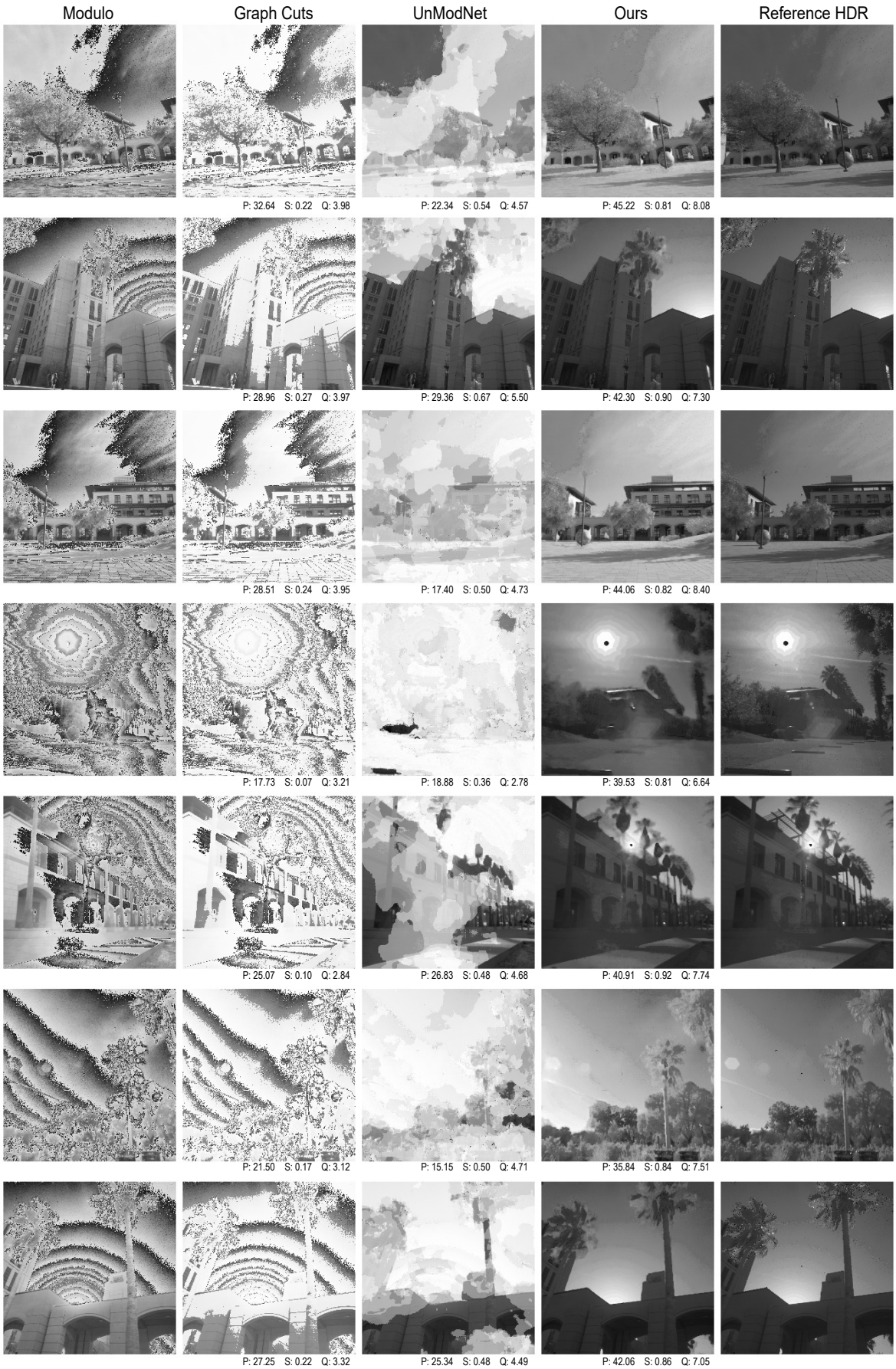

Fig. 7. Comparisons on captured data.

Fig. 8. Comparisons on captured data.

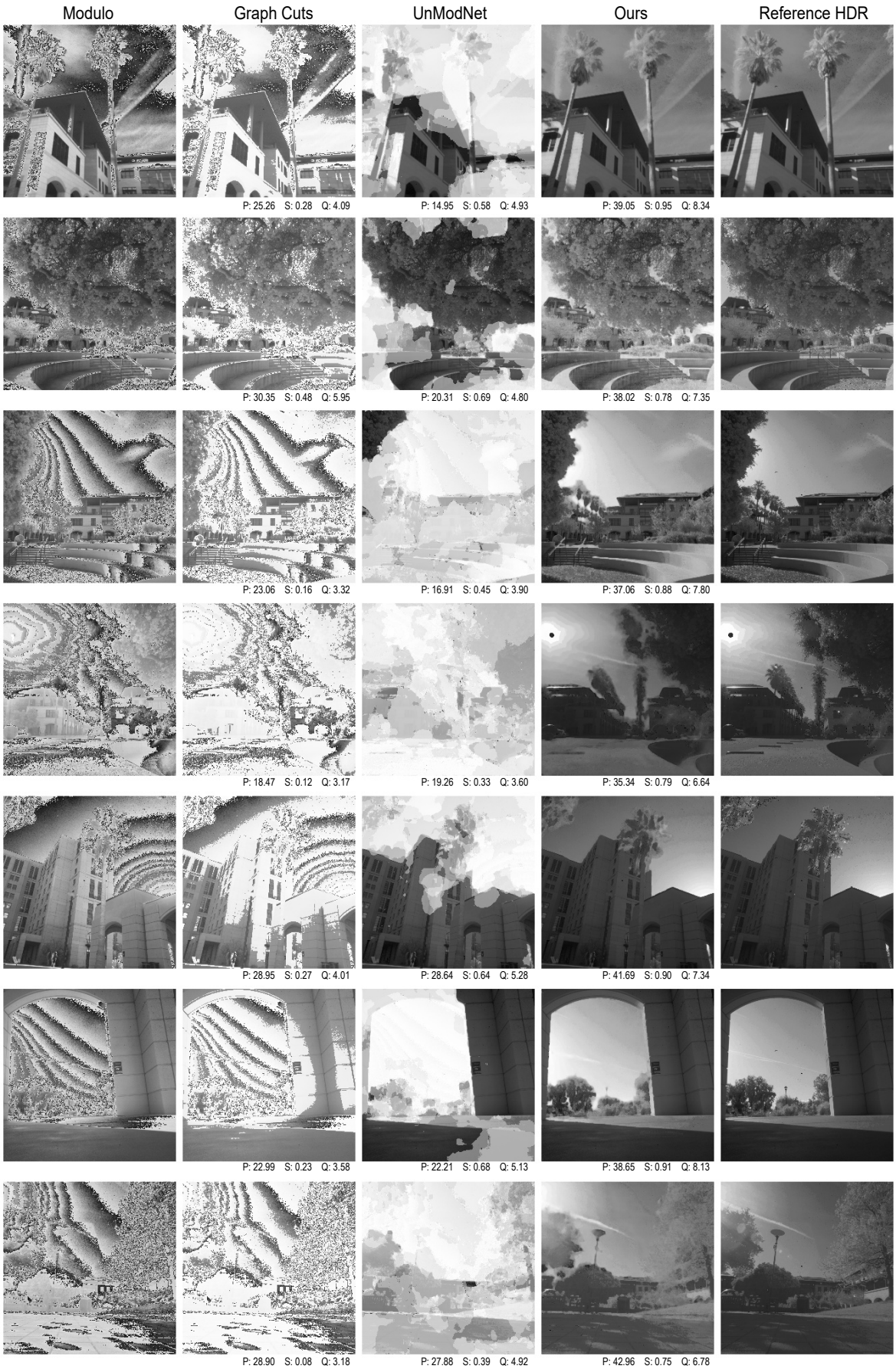

Fig. 9. Comparisons on captured data.

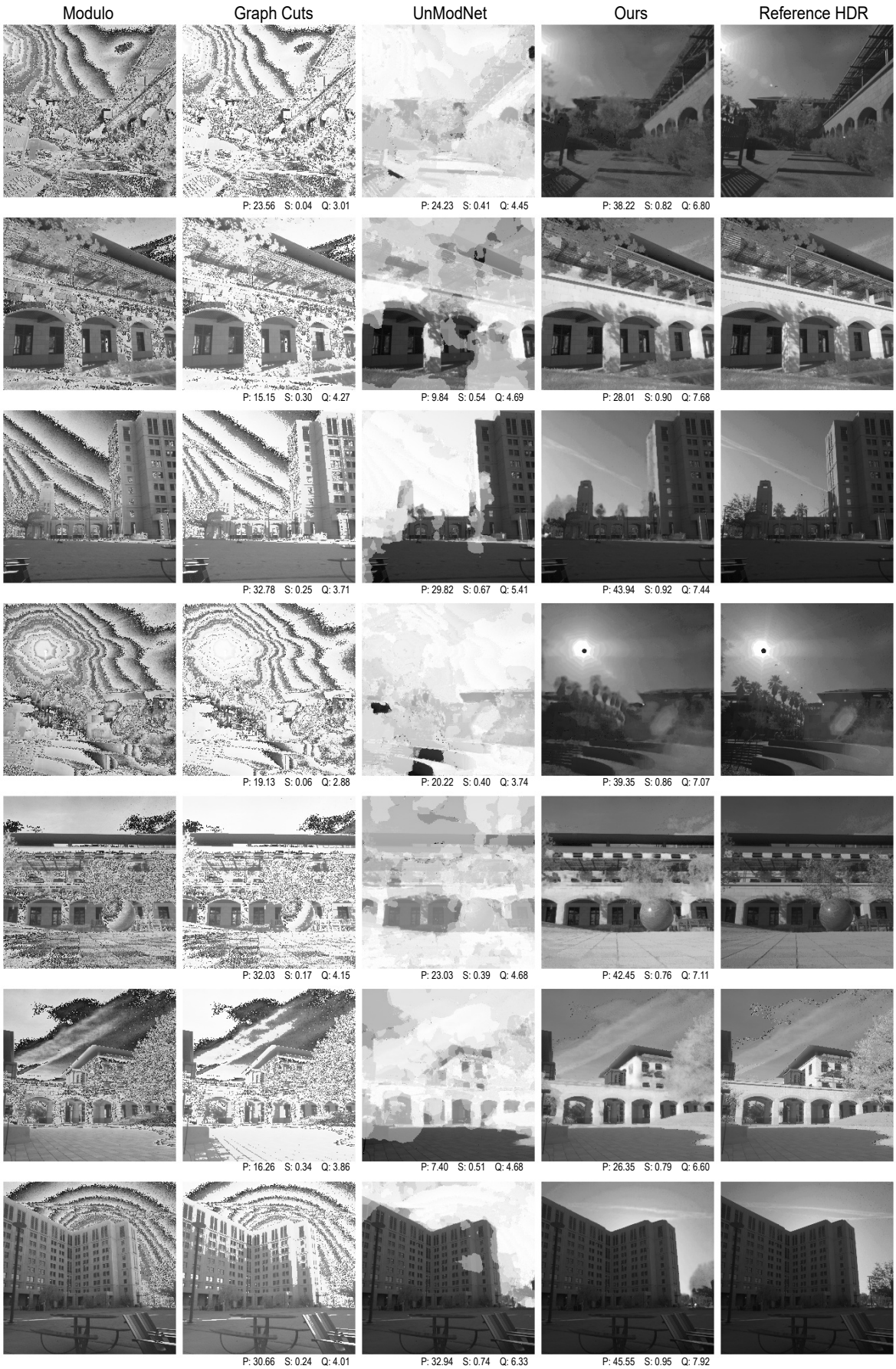

Fig. 10. Comparisons on captured data.