Lumosaic: Hyperspectral Video via Active Illumination and Coded-Exposure Pixels

Supplementary Document

Table of Contents

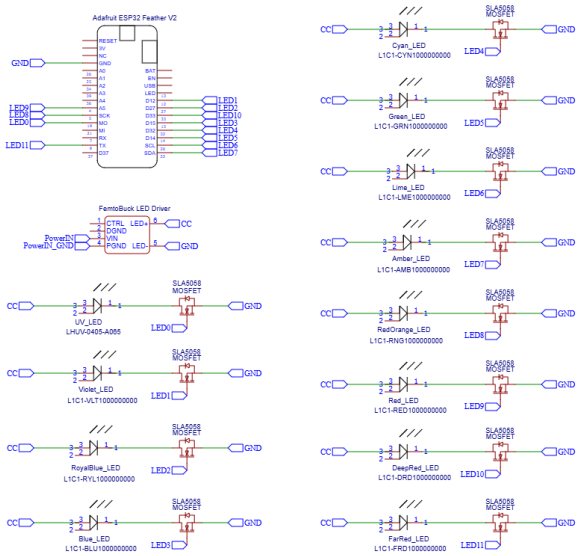

Figure 2. The hardware schematic of Lumosaic's illumination module. The ESP32 microcontroller coordinates the LED driver array and issues synchronization pulses to the CEP camera.

1. Imaging System Prototype

Lumosaic combines actively modulated LED illumination with a coded-exposure-pixel (CEP) camera to achieve dense spatio–spectro–temporal encoding suitable for hyperspectral video capture. This section details the system's implementation and calibration procedure.

1.1. Active Illumination Module

We developed a custom high-speed illumination module to enable time-varying spectral excitation synchronized with the CEP camera (Figure 2). The module integrates 12 high-power narrowband LEDs (Lumileds Luxeon C series) with full-width-at-half-maximum (FWHM) spectral bandwidths of approximately 20–30 nm, spanning the visible range. The LEDs are controlled by a constant-current, high-speed switching driver capable of switching at rates exceeding 100 kHz.

The imaging system is orchestrated by an Adafruit ESP32 Feather v2 microcontroller, which generates digital control signals for both the illumination module and the CEP camera. A single microsecond-resolution clock drives all timing events, ensuring sub-frame synchronization between light modulation and sensor exposure.

In our implementation, each hyperspectral video frame consists of 158 sub-frames; each sub-frame lasts 150 μs, corresponding to a total exposure window of 23.7 ms. Because individual LEDs differ in radiant power and the camera exhibits wavelength-dependent sensitivity, we non-uniformly allocate sub-frame counts per LED to approximately equalize the integrated spectral energy delivered per frame. LEDs with lower radiance receive proportionally more consecutive sub-frames, whereas brighter ones receive fewer. This adaptive exposure scheduling improves both channel balance and spectral dynamic range. The allocation used in our experiments is summarized in Table 2.

1.2. Coding Scheme Design

Figure 3 illustrates the 3×4 spatial–spectral mosaic tile that we achieve by coordinating illumination with pixel exposures. The pattern is executed left-to-right from the top-left corner to the bottom-right. The mapping between LEDs and tile positions spectrally and temporally distributes LEDs with adjacent spectra in order to minimize correlations between nearby measurements.

Table 2. The LED exposure allocation per video frame compensates for the CEP camera's spectral sensitivity and for relative intensity variations across LEDs. Each LED's sub-frame activations occur contiguously within the frame.

LED Name	Relative Allocation	Time per Frame (µs)
UV	5.70%	1,350
Violet	3.16%	750
Royal Blue	3.16%	750
Blue	3.16%	750
Cyan	5.70%	1,350
Green	6.96%	1,650
Lime	5.06%	1,200
Amber	25.32%	6,000
Red Orange	8.23%	1,950
Red	7.59%	1,800
Deep Red	6.96%	1,650
Far Red	18.99%	4,500
Total	100.00%	23,700

Figure 3. The 3×4 mosaic tile that forms the basis of Lumosaic's spatial-spectral coding.

1.3. Calibration

To ensure accurate spectral recovery, we jointly calibrate LED intensity scaling factors and camera gain for each spectral channel. Calibration minimizes reconstruction error over ColorChecker measurements captured under controlled illumination.

Measurements. We first measure the spectral power distribution of each LED at a fixed position using a calibrated spectroradiometer (Konica Minolta CS-2000), yielding high-resolution emission profiles $E_l(\lambda)$ from 380–780 nm at 1-nm intervals. We repeat this process for an incandescent bulb to generate an emission profile $B(\lambda)$.

To characterize the ColorChecker, we measure the spectral radiance of each patch $\mathcal{C}'_p(\lambda)$ under illumination from the incandescent bulb using the same spectroradiometer. We compute the spectral reflectance of each patch $\mathcal{C}_p(\lambda)$ by dividing the measured radiance by the incident illumination:

$$\mathcal{C}_p(\lambda) = \mathcal{C}'_p(\lambda) \,/\, B(\lambda). \tag{1}$$

We measure the camera's spectral sensitivity $\mathcal{S}$ using a monochromator (Image Engineering camSPECS XL) with 39 interference filters, producing calibrated spectral response curves at a resolution of 5 nm. These curves are quantized by spectral binning into $\Lambda = 41$ channels spanning 380–780 nm in 10-nm increments, resulting in spectral vectors $\mathbf{E}_l \in \mathbb{R}^\Lambda$, $\boldsymbol{\mathcal{C}}_p \in \mathbb{R}^\Lambda$, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^\Lambda$.

We then capture five repeated measurements of the ColorChecker using Lumosaic and average them to reduce noise. The resulting image is then demosaiced to produce $L = 12$ sub-images, one for each LED. For each sub-image, we sample a region within each of the 24 ColorChecker patches and compute the average intensity. This yields an empirical camera response matrix $\mathbf{M}^{\text{real}} \in \mathbb{R}^{L\times 24}$ where each entry represents the average response of the camera to a specific LED and ColorChecker patch.

Computing Calibration Coefficients. We determine a vector of LED scaling factors $\boldsymbol{\alpha} \in \mathbb{R}^L$ by simulating a theoretical camera response matrix $\mathbf{M}^{\text{sim}}_{l,p}$ from a simplified version of our image formation model:

$$\mathbf{M}^{\text{sim}}_{l,p} = \sum_{\lambda=1}^{\Lambda} \alpha_l \cdot (\mathbf{E}_l)_\lambda \cdot (\boldsymbol{\mathcal{C}}_p)_\lambda \cdot \boldsymbol{\mathcal{S}}_\lambda. \tag{2}$$

To calculate $\boldsymbol{\alpha}$, we minimize the non-negative least-squares reconstruction error between $\boldsymbol{\alpha}$-scaled simulated intensities and our averaged measurements on the ColorChecker:

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}\geq 0} \sum_{l,m} ([\sum_{\lambda=1}^{\Lambda} \alpha_l \cdot (\mathbf{E}_l)_\lambda \cdot (\boldsymbol{\mathcal{C}}_p)_\lambda \cdot \boldsymbol{\mathcal{S}}_\lambda] - \mathbf{M}^{\text{real}}_{l,p})^2. \tag{3}$$

The optimized α_l values are used to scale the LED emission curves, producing calibrated spectra $E'_l(\lambda) = \alpha_l E_l(\lambda)$ used in all forward modeling and reconstruction experiments.

2. Motion Compensation

Because each sub-image corresponds to a distinct time interval within the total exposure window, scenes containing motion may exhibit spatial misalignment across spectral channels. We prevent reconstruction artifacts by applying a procedure that warps all sub-images within a frame to a common temporal reference.

Each sub-image Y^l_i in a given frame i is assigned a timestamp according to its midpoint exposure timestamp. In other words, if Y^l_i is taken between t^s_l and t^e_l, we assign it a timestamp $t_l = (t^s_l + t^e_l)/2$. The timestamps are normalized by the total duration of the frame, taking into account the CEP camera's 6 ms readout time t_{readout}:

$$t'_l = \frac{t_l}{[\sum_l (t^e_l - t^s_l)] + t_{\text{readout}}}. \tag{4}$$

This normalization provides a consistent temporal coordinate $t'_l \in [0, 1]$ for every sub-image, facilitating motion interpolation across time.

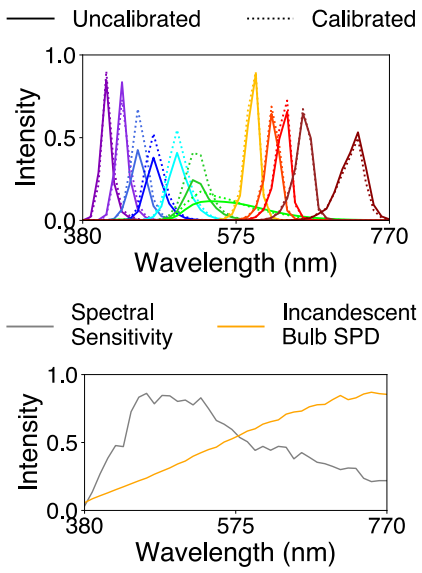

Figure 4. **(top)** LED spectral power distributions before and after calibration via optimized scaling factors α_λ. **(bottom)** Camera spectral sensitivity $\mathcal{S}(\lambda)$ and incandescent bulb spectral power distribution (SPD).

We use the Real-Time Intermediate Flow Estimation (RIFE) model [17] to estimate bidirectional flows between sub-images corresponding to the same LED across adjacent frames. RIFE jointly predicts interpolated flow fields and intermediate frames for arbitrary normalized timesteps $t' \in [0, 1]$. We specifically use RIFE v4.6 [18], which supports arbitrary-timestep warping and high-resolution inference.

We temporally align all sub-images in a frame to the lime-colored LED sub-image as the reference because its illumination occurs near the frame's midpoint and its wavelength lies near the spectral center of the LED set. Suppose we are aligning the sub-image Y_i^l for any LED other than the lime-colored one. If LED l is scheduled before the reference ($t'_l < t'_{lime}$), we estimate the interpolated flow fields and warped image between its sub-image in the current frame Y_i^l and in the next frame Y_{i+1}^l. The normalized timestep t' provided to RIFE is determined by the distance between t'_l and t'_{lime}, $t' = t'_{lime} - t'_l$. If LED l is scheduled after the reference ($t'_l > t'_{lime}$), we use its sub-image in the previous frame Y_{i-1}^l and in the current frame Y_i^l, and the normalized timestep t' provided to RIFE is $t' = (1 - t'_l) + t'_{lime}$. We use the warped sub-images produced by RIFE as input to our downstream reconstruction model.

3. Hyperspectral Video Frame Reconstruction

To generate the full hyperspectral cube from each coded video frame, we apply hyperspectral reconstruction to the densely coded spectral mosaics. This section describes the reconstruction model and its application to the coded video frames.

3.1. Model Architecture

Our reconstruction network is based on the Holistic Attention Network (HAN) [31], which combines residual learning with hierarchical attention to model inter-channel correlations. The architecture comprises multiple residual groups, each containing convolutional blocks enhanced by layer- and channel-level attention mechanisms. This design effectively captures spatial context and spectral dependencies across the encoded mosaic.

We adapt HAN to our hyperspectral reconstruction setting with three key modifications:

1. We adjust the first and final convolutional layers to handle our input of 12 LED sub-images and output of 33 spectral channels[1].
2. Because our input and output share the same spatial resolution, we remove HAN's original upsampling module.
3. To reduce GPU memory consumption and enable training on a single RTX A6000 or TITAN RTX, we reduce the number of residual groups from 20 to 18.

The final model contains approximately 57.1M trainable parameters. A detailed `torchsummary`-style architecture overview is shown in Figure 5.

3.2. Patch-wise Reconstruction

To reduce GPU memory usage and enable efficient training, we reconstruct each frame in smaller overlapping patches. The demosaiced sub-images are partitioned into overlapping $66 \times 64 \times 12$ patches using a sliding window with a stride of $(30, 32)$ and no padding. The stride ensures that the top-left corner of each patch aligns with the 3×4 spatial–spectral mosaic grid. When the windows do not completely cover the image, additional patches are appended along the right and bottom boundaries.

The HAN network independently processes each patch to reconstruct its corresponding hyperspectral cube. This patch-wise approach allows large full-resolution frames to be reconstructed without exceeding GPU memory limits while maintaining local spectral consistency across overlapping regions.

[1]Two extrapolated boundary channels are included during training to improve reconstruction near spectral limits but are excluded from evaluation.

```
Layer (type:depth-idx)                                  Output Shape              Param #
==========================================================================================
HANDSA                                                  [14, 33, 66, 64]          --
  UnshuffleToSpatiallyPreservingDemosaic: 1-1           [14, 12, 66, 64]          --
    PixelUnshuffle2D: 2-1                               [14, 12, 22, 16]          --
  HAN: 1-2                                              [14, 33, 66, 64]          --
    Sequential: 2-2                                     [14, 128, 66, 64]         --
      Conv2d: 3-1                                       [14, 128, 66, 64]         13,952
    Sequential: 2-3                                     --                        --
      ResidualGroup: 3-2 to 3-11                        [14, 128, 66, 64]         54,999,200
      Conv2d: 3-12                                      [14, 128, 66, 64]         147,584
    LAM_Module: 2-4                                     [14, 1408, 66, 64]        1
      Softmax: 3-13                                     [14, 11, 11]              --
    Conv2d: 2-5                                         [14, 128, 66, 64]         1,622,144
    CSAM_Module: 2-6                                    [14, 128, 66, 64]         1
      Conv3d: 3-14                                      [14, 1, 128, 66, 64]      28
      Sigmoid: 3-15                                     [14, 1, 128, 66, 64]      --
    Conv2d: 2-7                                         [14, 128, 66, 64]         295,040
    Sequential: 2-8                                     [14, 33, 66, 64]          --
      Conv2d: 3-16                                      [14, 33, 66, 64]          38,049
==========================================================================================
Total params: 57,115,999 | Trainable: 57,115,999 | Non-trainable: 0
Total mult-adds (T): 3.35
Input size (MB): 9.94 | Fwd/bwd pass (MB): 22726.58 | Params (MB): 228.46
Estimated Total Size (MB): 22964.98
```

Figure 5. A summary of our modified HAN-based reconstruction model.

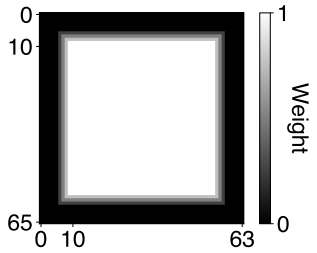

Figure 6. A visualization of the spatial weighting kernel $\mathbf{K}$ used for merging overlapping patch predictions.

3.3. Aggregation of Patch-wise Results

After reconstructing all patches, we reassemble them into the full-resolution hyperspectral frame using weighted averaging to minimize boundary artifacts. Each reconstructed patch is multiplied by a predefined spatial weighting kernel $\mathbf{K} \in [0,1]^{66\times64}$ that assigns higher weights near the center and lower weights near the edges (see Figure 6). This spatial weighting mitigates discontinuities in overlapping regions.

The final full-frame hyperspectral reconstruction is obtained as:

$$\mathbf{R}' = \frac{\mathrm{Fold}(\mathbf{K}_{66\times64\times33} \odot \mathrm{HAN}(\mathrm{Unfold}(\mathbf{Y})))}{\mathrm{Fold}(\mathbf{K}_{66\times64\times33})}, \quad (5)$$

where $\odot$ multiplication and fractional division are applied element-wise. $\mathbf{K}_{66\times64\times33}$ is the repetition of $\mathbf{K}$ in the spectral dimension. $\mathrm{Unfold} : \mathbb{R}^{640\times480} \to \mathbb{R}^{L\times66\times64}$ and $\mathrm{Fold} : \mathbb{R}^{L\times66\times64\times33} \to \mathbb{R}^{640\times480\times33}$ are inverse-like sliding-window operations used to extract and reassemble the L overlapping patches, respectively. $\mathrm{HAN} : \mathbb{R}^{66\times64} \to \mathbb{R}^{66\times64\times33}$ denotes the model used for patch-wise hyperspectral reconstruction.

4. Supplementary Results: Simulations

4.1. Additional Details

During training, we extract random patches from each simulated measurement. To ensure correct alignment between the illumination schedule and the pixel-wise coded-exposure pattern, the top-left corner of every patch is constrained to coincide with the top-left corner of a 3×4 mosaic tile. This guarantees that every extracted patch contains an integer number of complete mosaic repetitions and avoids boundary inconsistencies during reconstruction. Each batch is formed by sampling patches independently—potentially from different hyperspectral images—to improve spectral diversity and reduce overfitting. When noise robustness is evaluated, we add independent Gaussian noise with standard deviation $\sigma \in \{0, 5, 10, 15, 20\}\%$ of the maximum signal value before feeding it to the network. All corresponding hyperspectral ground-truth patches are consistently normalized and extracted at identical spatial coordinates. This pipeline ensures that the model learns reconstruction mappings that faithfully reflect the physical sensing process used on the real hardware.

4.2. Comparisons with Baselines

Figure 7 presents qualitative comparisons of Lumosaic with a HAN backbone on test scenes from the CAVE [43], KAUST [24], and ARAD [1] datasets. Reconstructed outputs from QDO [23] and MST++ [7] are provided as state-of-the-art comparisons. Across all the scenes presented, Lumosaic recovers finer spectral details and spatial textures while avoiding color bleeding commonly observed in the baselines.

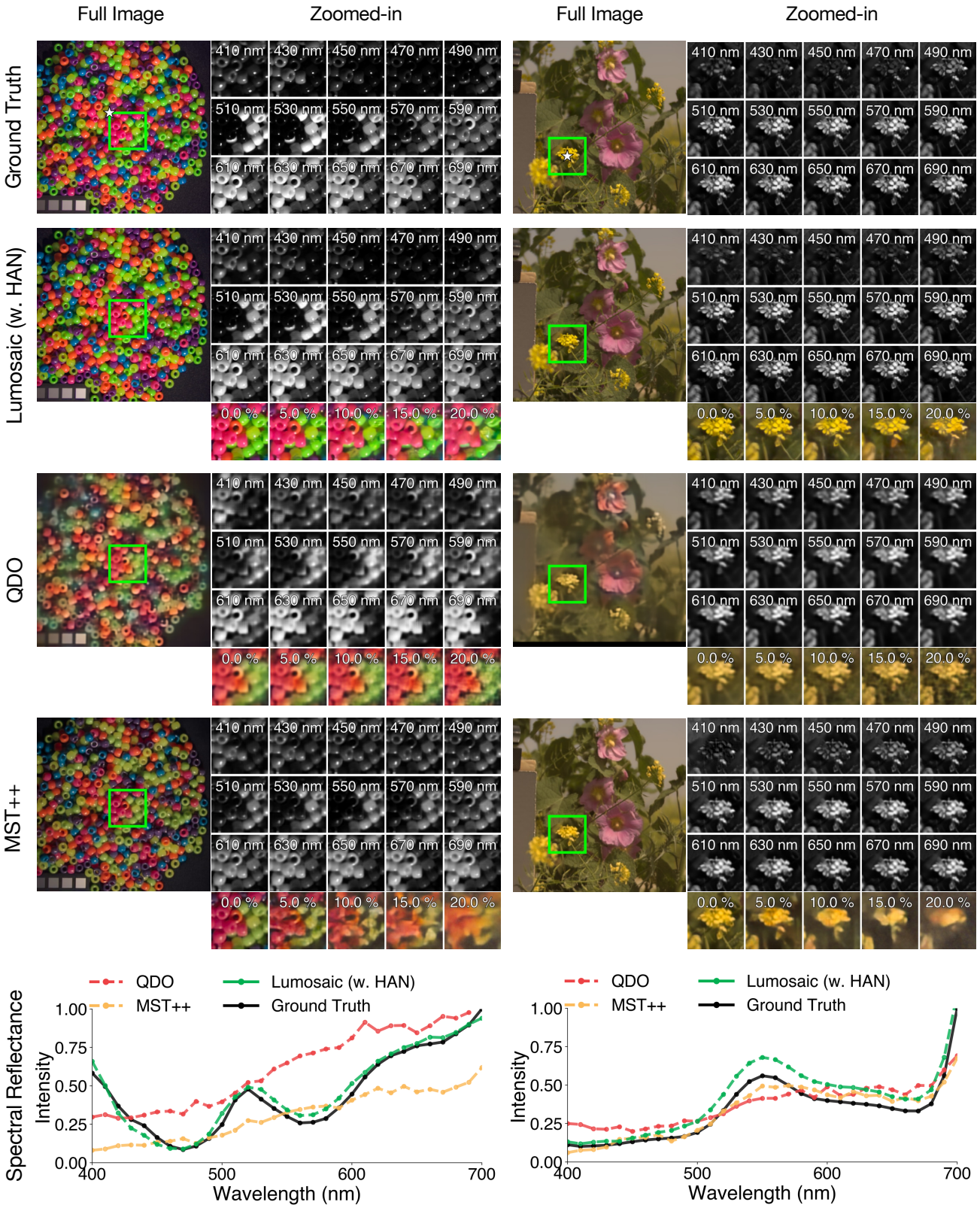

Figure 7. **Qualitative comparisons on public datasets.** Representative reconstructions from the test set comparing QDO [23], MST++ [7], and Lumosaic. Each column shows the reconstructed sRGB image, selected hyperspectral channels, and zoomed-in sRGB crops under varying Gaussian noise levels (annotated with the corresponding standard deviation). We also present spectral validation plots at the bottom of the figure for two specific locations, marked with white stars in the first row of sRGB images.

4.3. High-Frequency Spectral Recovery Test

We assess Lumosaic's ability to recover high-frequency spectral features using a synthetic rainbow scene (Figure 8) designed to stress-test spectral resolving capability beyond what natural scene datasets typically demand. The scene is created by sweeping Gaussian-shaped spectral reflectance profiles with a FWHM of 20 nm from the bottom to the top of a 512×512 image. The gradient spans the 400–700 nm range, so each row's central wavelength changes by (700-400 nm) / 512 px = 0.59 nm/px. This produces a high-frequency spectral gradient with sharp transitions between neighboring wavelength bands that are rarely encountered in natural scenes. As a result, it provides a controlled out-of-distribution benchmark that reveals failure modes in systems relying heavily on natural-scene priors.

Physical Sampling Limits. Before discussing results, we clarify the fundamental constraints governing Lumosaic's spectral resolving power. Lumosaic acquires 12 narrowband LED measurements (20-30 nm FWHM) and upsamples to 31 channels (10-nm binning) using learned priors. The 10-nm spectral binning of the reconstruction model limits the resolvable spectral bandwidth to features with FWHM exceeding approximately 20 nm per the Nyquist sampling criterion. The physical bandwidths of the LEDs themselves impose an additional constraint, as they determine the finest spectral modulation the illumination can encode in the measurement domain.

The learned model improves spectral conditioning beyond what direct inversion of 12 measurements would allow, but it cannot synthesize spectral information beyond these physical sampling limits. Narrower-band spectral analysis would require hardware improvements such as narrower LED emitters or denser spectral sampling, and it is physically precluded by the current implementation. These constraints do not undermine Lumosaic's core claims, which target the visible range at 10-nm resolution under video-rate acquisition, but they establish the regime within which spectral recovery should be evaluated.

Comparison with Baselines. As shown in Figure 8 (A-D), the baseline QDO [23] and MST++ [7] methods exhibit noticeable spectral smoothing and blending artifacts, failing to reproduce abrupt transitions between neighboring wavelength bands. In contrast, Lumosaic reconstructs these features while maintaining acceptable spatial fidelity and substantially better spectral localization across most of the gradient. This improvement stems from its deterministic mosaic-based sensing strategy, which captures densely sampled, non-multiplexed spectral measurements rather than entangling multiple wavelengths within a single coded exposure. By reducing reliance on strong learned priors and providing cleaner spectral cues directly in the measurement domain, Lumosaic achieves more accurate and reliable reconstruction of fine spectral structure.

Influence of Training Data Composition. Reconstruction sharpness depends on the spectral distribution of the training data. To investigate this, we introduce two types of synthetic spectra representing narrow-band features:

1. **Single-peak profiles:** Gaussian functions with center wavelengths spanning 400–700 nm and FWHM values of $\{10, 20, 30, 40, 50\}$ nm, sampled at 1-nm intervals.
2. **Double-peak profiles:** Pairs of Gaussians with center separations of $\{10, 20, 30, 40, 60, 80\}$ nm, constrained to lie within 400–700 nm.

All profiles are integrated into 10-nm bins via trapezoidal integration and normalized to unit maximum. For training purposes, we use these synthetic spectra to create uniform, textureless hyperspectral images by repeating each profile spatially. We evaluate versions of Lumosaic on the rainbow scene after training on $\{0\%, 33\%, 66\%, 100\%\}$ synthetic spectra mixed with natural scenes from our dataset corpus.

As shown in Figure 8 (D-G), moderate synthetic augmentation (33%) provides the best trade-off between generalization and spectral precision, producing sharper and well-localized peaks that closely align with the ground-truth spectra. Models trained exclusively on natural scenes exhibit lower sensitivity to subtle spectral variations, while heavier synthetic mixing (66% and 100%) offers limited additional resolving power and can slightly bias the reconstructions toward narrowband profiles.

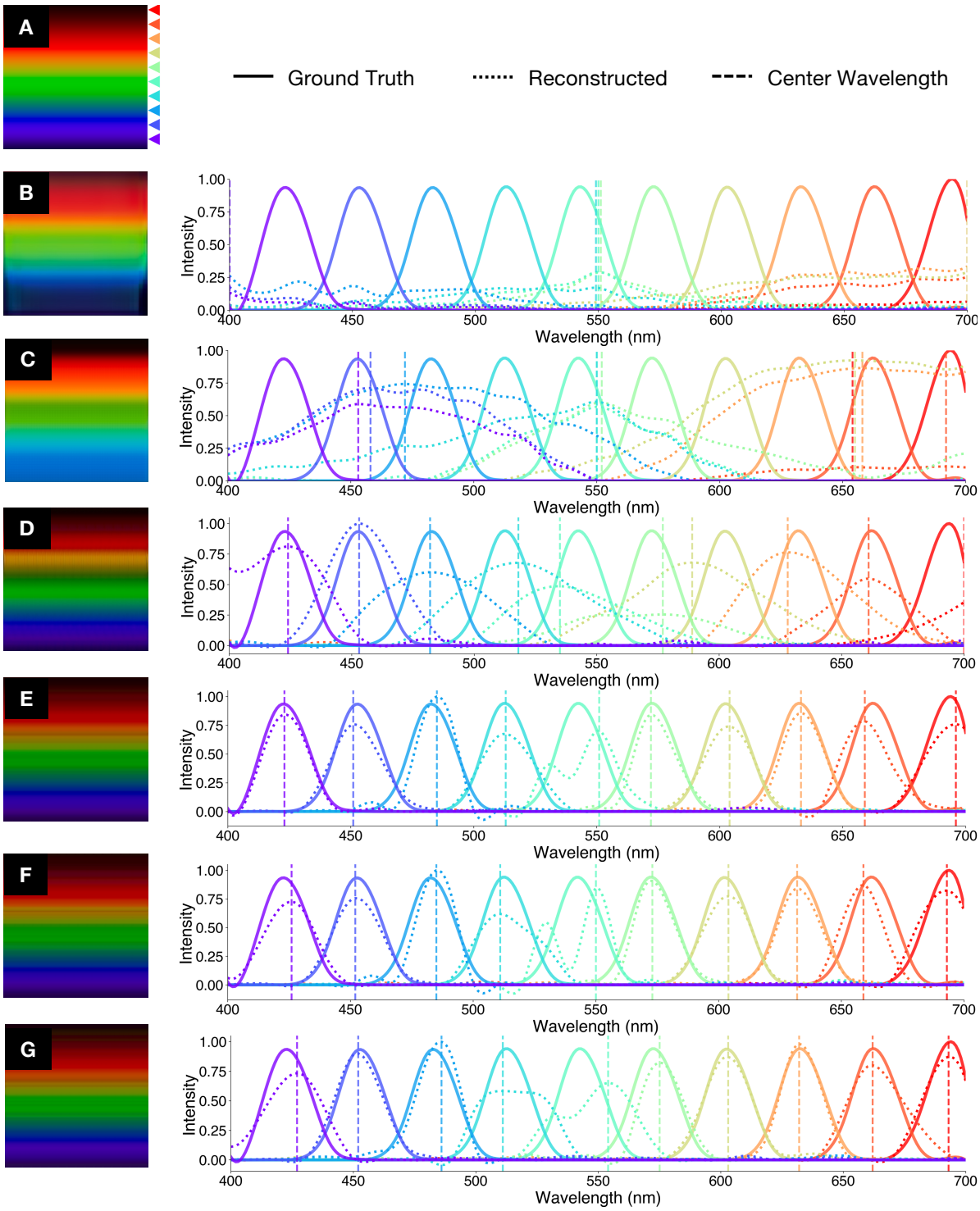

Figure 8. **Spectral recovery test on a high-frequency rainbow scene. (A)** The ground-truth rainbow scene rendered in sRGB, with triangles used to mark center wavelength positions. **(B–D)** Reconstructed sRGB images and reconstructed spectral reflectance curves interpolated at 1-nm intervals for QDO [23], MST++ [7], and Lumosaic respectively. The vertical dashed lines correspond to the reconstructed central wavelengths. **(D-F)** Lumosaic results under increasing proportions of high-frequency synthetic spectra in the training set (0%, 33%, 66%, 100%).

5. Additional Results: Real World

5.1. Static Scenes

Figure 9 further demonstrates Lumosaic's hyperspectral reconstruction capabilities for static scenes. As shown, Lumosaic correctly reconstructs fine-grained spatial details in each scene: the names of major countries and bodies of water on the globe, the box label and thin branches surrounding the butterfly, and the teeth of the figurine. The spectral information is also accurate, as clearly illustrated by the distinct colors of the countries on the globe.

5.2. Metamerism Analysis

We evaluate Lumosaic's ability to distinguish between metameric objects by imaging a genuine, pigment-based ColorChecker and its printed photocopy. Figure 10 showcases Lumosaic's ability to resolve metameric ambiguities. Both targets appear visually identical under sRGB rendering, yet their spectral profiles differ notably across multiple patches. The genuine ColorChecker exhibits smooth, well-defined reflectance spectra characteristic of pigmented surfaces, whereas the printed copy shows irregular spectral peaks due to ink absorption and printer gamut limitations.

5.3. Dynamic Scenes

Figure 11 and Figure 12 further demonstrate Lumosaic's capabilities as a hyperspectral video reconstruction system for dynamic scenes. In the scene with the rotating figurine, all printed elements remain properly resolved throughout the video despite continuous rotational motion. In the hand gesture scene, there is slight ghosting when the hand motion is fastest, but the content is nonetheless coherent. The scene with effervescent tonic water is particularly well-suited for evaluating hyperspectral video, as tonic water exhibits strong fluorescence for an otherwise transparent liquid. Despite the liquid's transparency, the system is able to render the bubbles with high spatial resolution. Finally, we demonstrate robustness to camera motion through a free-hand panning sequence, in which Lumosaic maintains stable hyperspectral reconstruction despite rapid viewpoint changes.

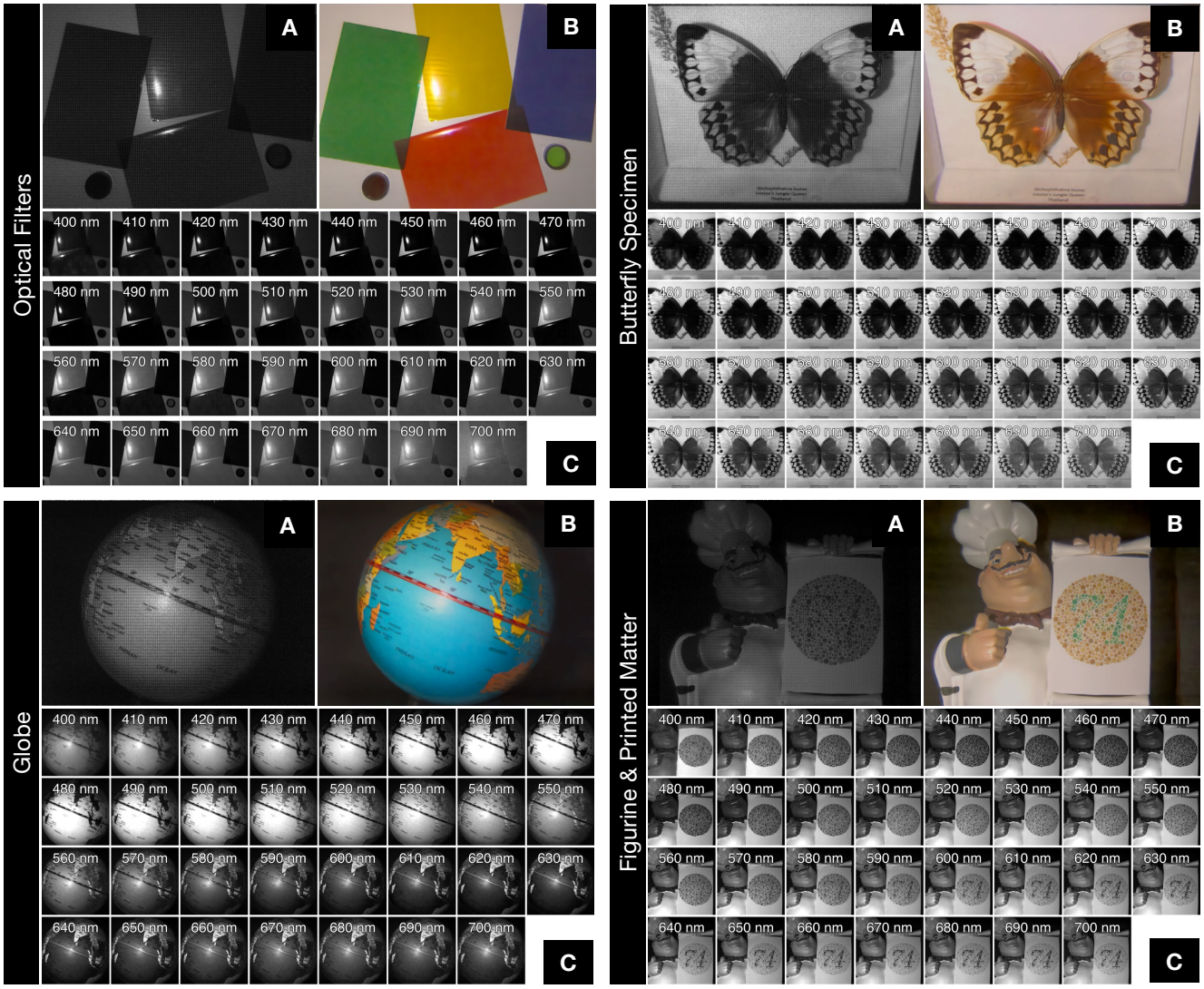

Figure 9. Additional Lumosaic reconstruction results on diverse static scenes: optical filters, a globe, a butterfly specimen, and a figurine with printed matter. Each scene includes **(A)** the input coded image, **(B)** the reconstructed hyperspectral image rendered in sRGB, and **(C)** the 31-channel hyperspectral composites rendered in grayscale for visualization.

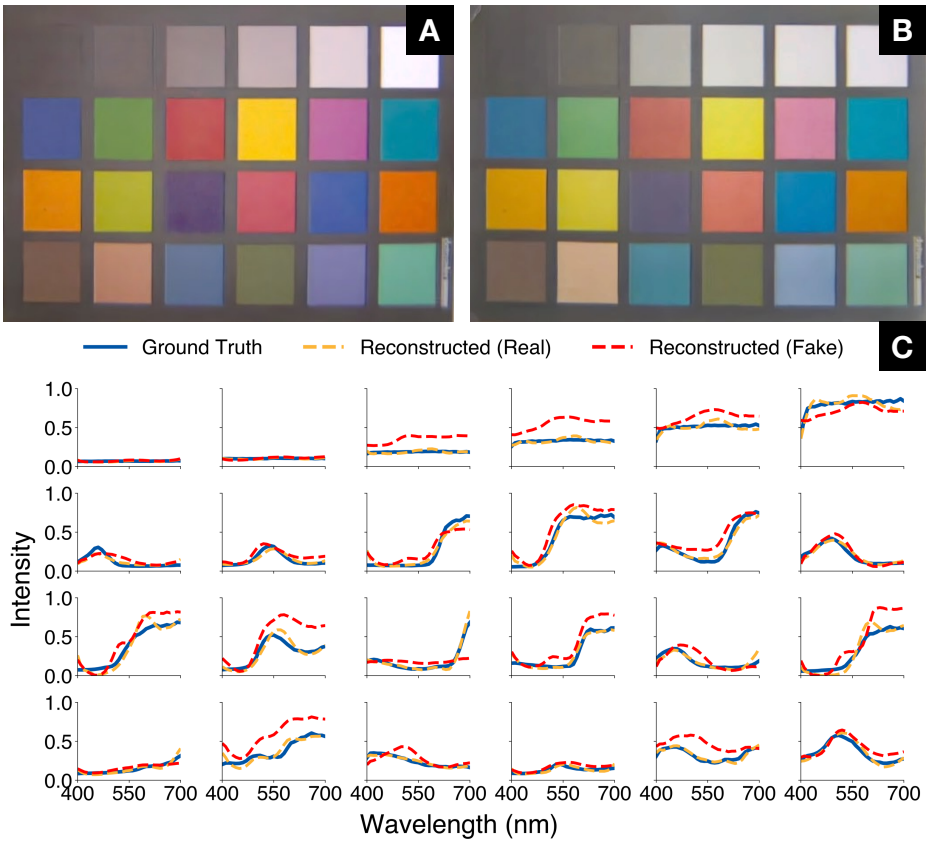

Figure 10. **Metamerism analysis with Lumosaic using genuine and printed ColorChecker targets.** (**A and B**) RGB visualization of the genuine ColorChecker and a printed photocopy under identical illumination. (**C**) Spectral curves illustrating the differences between genuine and printed patches.

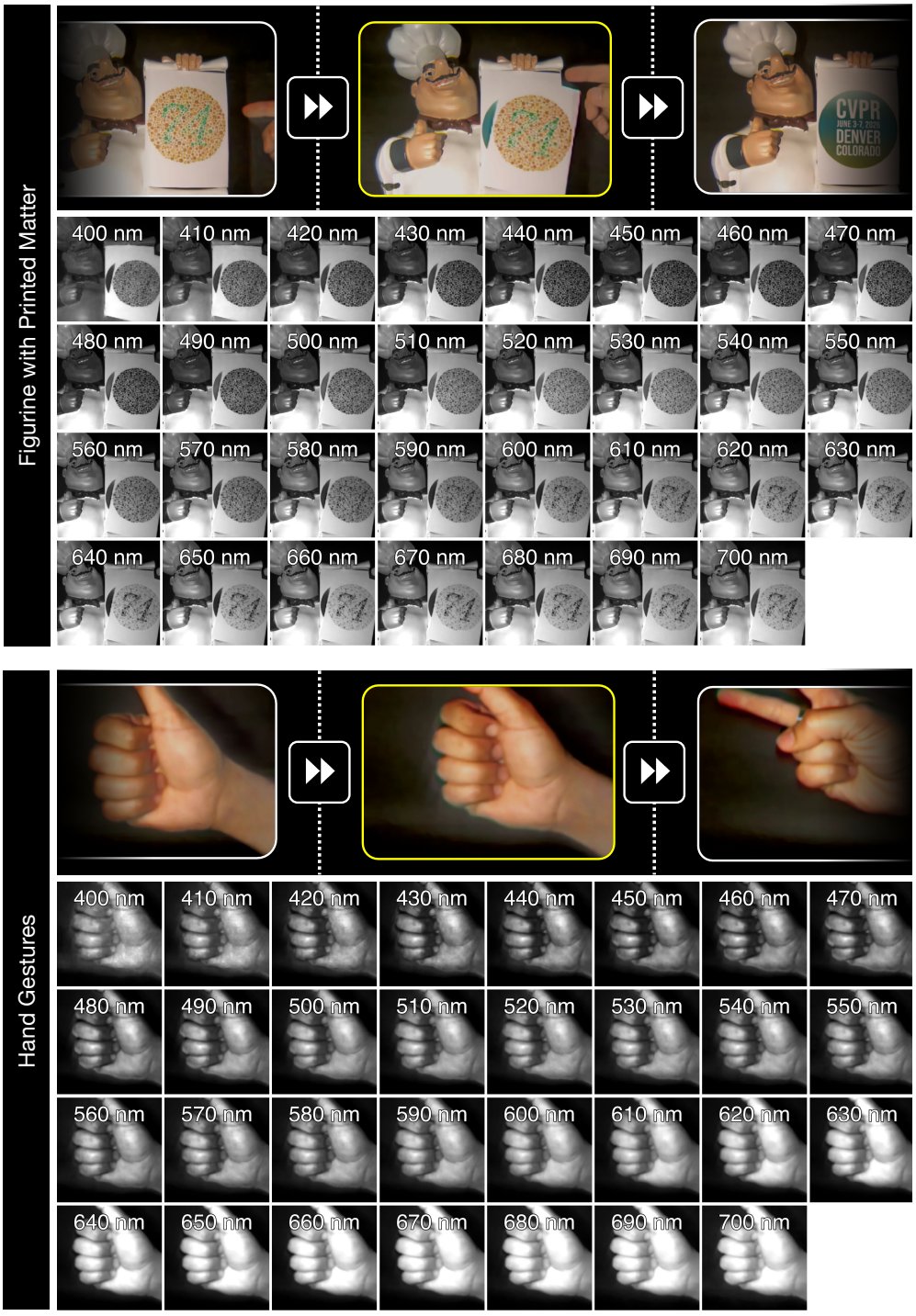

Figure 11. Additional hyperspectral reconstruction results with Lumosaic on scenes exhibiting diverse motion characteristics. **(top)** A figurine with printed elements, including Ishihara patterns and text, undergoing both rotational and falling motion relative to the camera. **(bottom)** Dynamic hand gestures with non-rigid motion. The top row for each scene shows rendered sRGB views from non-consecutive frames of the reconstructed hyperspectral video, while the bottom row shows the corresponding full 31-channel hyperspectral images for the frames outlined in yellow.

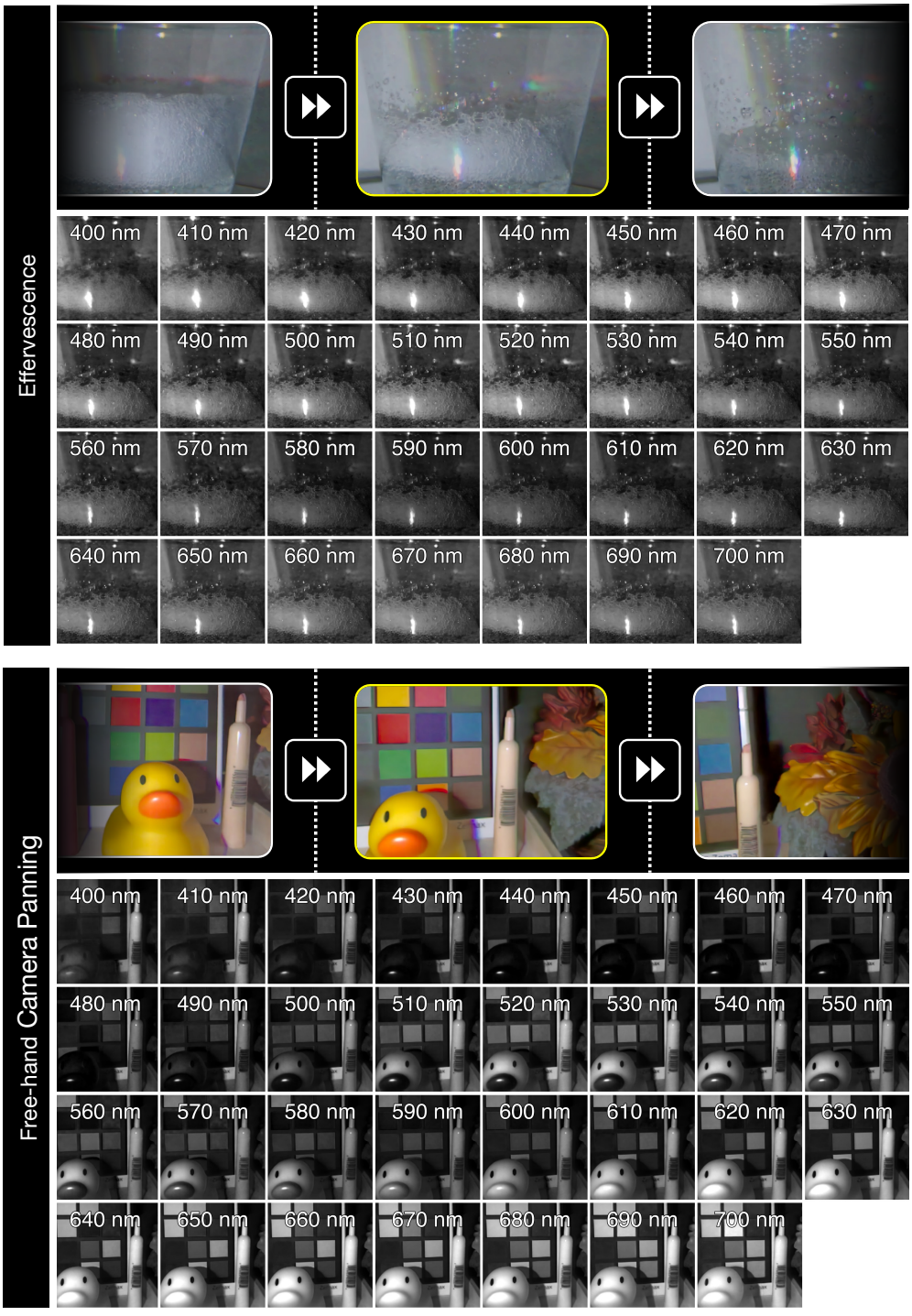

Figure 12. Additional hyperspectral reconstruction results with Lumosaic on scenes exhibiting diverse motion characteristics. **(top)** A transparent plastic cup filled with carbonated tonic water, capturing high-frequency motion due to effervescence. **(bottom)** A complex indoor scene with colorful objects captured while freely panning the imaging system to introduce continuous viewpoint and illumination changes. The top row for each scene shows rendered sRGB views sampled from non-consecutive frames of the reconstructed hyperspectral video, while the bottom row shows the corresponding full 31-channel hyperspectral images for the frames outlined in red.