



# ChromaFlash: Snapshot Hyperspectral Imaging Using Rolling Shutter Cameras

DHRUV VERMA, University of Toronto, Canada

IAN RUFFOLO, University of Toronto, Canada

DAVID B. LINDELL, University of Toronto, Canada

KIRIAKOS N. KUTULAKOS, University of Toronto, Canada

ALEX MARIAKAKIS, University of Toronto, Canada

Hyperspectral imaging captures scene information across narrow, contiguous bands of the electromagnetic spectrum. Despite its proven utility in industrial and biomedical applications, its ubiquity has been limited by bulky form factors, slow capture times, and prohibitive costs. In this work, we propose a generalized approach to snapshot hyperspectral imaging that only requires a standard rolling shutter camera and wavelength-adjustable lighting. The crux of this approach entails using the rolling shutter as a spatiotemporal mask, varying incoming light quicker than the camera's frame rate in order for the captured image to contain rows of pixels illuminated at different wavelengths. An image reconstruction pipeline then converts this coded image into a complete hyperspectral image using sparse optimization. We demonstrate the feasibility of this approach by deploying a low-cost system called ChromaFlash, which uses a smartphone's camera for image acquisition and a series of LEDs to change the scene's illumination. We evaluated ChromaFlash through simulations on two public hyperspectral datasets and assessed its spatial and spectral accuracy across various system parameters. We also tested the real-world performance of our prototype by capturing diverse scenes under varied ambient lighting conditions. In both experiments, ChromaFlash outperformed state-of-the-art techniques that use deep learning to convert RGB images into hyperspectral ones, achieving snapshot performance not demonstrated by prior attempts at accessible hyperspectral imaging.

CCS Concepts: • Human-centered computing → Ubiquitous and mobile devices; • Computing methodologies → Hyperspectral imaging; Computational photography.

Additional Key Words and Phrases: hyperspectral imaging, smartphone cameras, rolling shutter, coded imaging, image reconstruction

## ACM Reference Format:

Dhruv Verma, Ian Ruffolo, David B. Lindell, Kiriakos N. Kutulakos, and Alex Mariakakis. 2024. ChromaFlash: Snapshot Hyperspectral Imaging Using Rolling Shutter Cameras. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 3, Article 132 (September 2024), 31 pages. <https://doi.org/10.1145/3678582>

## 1 INTRODUCTION

Cameras have transitioned from mere devices for capturing memories and performing surveillance to supporting advanced computer vision tasks. The ubiquity of cameras has facilitated exciting applications, including augmented

---

Authors' addresses: **Dhruv Verma**, University of Toronto, Toronto, Canada, dhruvverma@cs.toronto.edu; **Ian Ruffolo**, University of Toronto, Toronto, Canada, ian.ruffolo@cs.toronto.edu; **David B. Lindell**, University of Toronto, Toronto, Canada, lindell@cs.toronto.edu; **Kiriakos N. Kutulakos**, University of Toronto, Toronto, Canada, kyros@cs.toronto.edu; **Alex Mariakakis**, University of Toronto, Toronto, Canada, mariakakis@cs.toronto.edu.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2474-9567/2024/9-ART132

<https://doi.org/10.1145/3678582>

reality [9], remote vital sign monitoring [65], body motion tracking [1], and assistive technologies [46]. These diverse applications predominantly rely on the capabilities of standard RGB cameras, which mimic the human color vision system by faithfully reproducing scenes with a dense array of red, green, and blue color filters. Because of how they are designed, RGB cameras do not conventionally capture the intricate spectral characteristics of objects and surfaces, which can provide insights into their material composition and contribute to a better understanding of the scene. This is particularly important for application domains requiring high precision, such as robotics and machine vision [68], agriculture [36], digital forensics [38], and healthcare [23].

This shortcoming has given rise to hyperspectral imaging systems that are able to capture the reflectance properties of each point in a scene across numerous finely divided spectral bands, revealing distinctions between objects and surfaces that may appear identical to the naked eye or RGB cameras. Traditional hyperspectral imaging systems rely on a 2D sensor, thereby requiring a spatial or temporal scanning process for acquiring the 3D hyperspectral information. Achieving this functionality requires sophisticated equipment that can cost above \$25,000 USD, prohibiting the widespread adoption of these devices [50]. Furthermore, scanning makes traditional hyperspectral imaging systems impractical for scenarios involving moving cameras (e.g., head-mounted displays, robots) and dynamic target objects (e.g. moving eyes of a patient [23], produce on conveyor belts [61]).

Prior work in ubiquitous computing has attempted to alleviate problems associated with the form factor and accessibility of hyperspectral imaging by proposing mobile, wearable, and low-cost implementations [21, 50, 60]. However, each of them fails to address critical technical challenges including spatial resolution [60], acquisition time [21], and robustness to variation in camera parameters [50]. Literature in computer vision has made notable progress in developing snapshot hyperspectral imaging techniques leveraging compressed sensing and coded imaging [66], yet most of these methods either necessitate highly specialized components or bulky hardware.

We propose a generalized approach to accessible snapshot hyperspectral imaging. Most RGB cameras employ a rolling shutter to expose the sensor's pixels to incoming light row by row within a given frame. Since the exposure of each row is on the order of microseconds, this mechanism enables opportunities for high-speed imaging and optical sensing [4, 51]. Meanwhile, the wavelength of incoming light from the scene can be varied using either passive filters or active illumination. We harness these affordances to capture a 2D-coded image with spectral information interleaved between pixel rows. This coded image can then be reconstructed into a 3D hyperspectral image using sparse optimization techniques. However, using a multispectral flash in conjunction with a rolling shutter camera introduces a few technical challenges. Since the rolling shutter operates row by row, the spectral information is spread across that dimension according to factors such as the sensor's frame rate, the shutter speed, the frequency at which the incident light is changed, and the order of the light's wavelengths. These factors lead to an intricate trade-off between spread and noise.

To demonstrate the feasibility of our proposed approach, we implemented a smartphone-based system called ChromaFlash that uses a standard smartphone camera with a low-cost multispectral flash consisting of multiple narrowband LEDs. ChromaFlash's multispectral flash is similar to the True Tone flash<sup>1</sup> found on recent iPhone models, which cycles through multiple wavelengths at a high rate to enhance image quality according to the surrounding light. Leading up to our ChromaFlash system, our work makes the following contributions:

- **Rolling shutter-coded hyperspectral imaging:** We introduce a novel imaging approach that uses a rolling shutter camera and rapidly changing incident light to efficiently capture 3D hyperspectral information into 2D images with row-wise interleaving of spectral information. We explain the underlying theory of the image formation process and outline the steps required to reconstruct complete hyperspectral images from these coded images.
- **Discovering optimal parameters of operation:** Several parameters impact the accuracy of our approach to hyperspectral reconstruction. For ChromaFlash specifically, these include the frequency at which the

<sup>1</sup><https://www.apple.com/newsroom/2017/09/iphone-8-and-iphone-8-plus-a-new-generation-of-iphone/>

illumination is varied, the number of LEDs illuminated together, the number of coded image frames used, and sensor noise. We conducted a series of simulated experiments using publicly available hyperspectral datasets [34, 63] to examine how these parameters affect reconstruction performance across various scenes. These simulations helped us quantify theoretical performance limits, trade-offs, and trends before implementing ChromaFlash.

- **Real-world implementation and evaluation:** We demonstrate the feasibility of our approach by implementing ChromaFlash using a smartphone camera paired with a compact multispectral LED array attachment. We assessed its real-world performance by capturing diverse scenes under varying ambient lighting conditions, comparing it against the state-of-the-art RGB-to-hyperspectral model MST++ [13]. These tests contextualize our work within existing research and showcase its robustness and versatility in practical scenarios.
- **Open-source hardware and software:** To facilitate replication and further development, we provide a list of materials, hardware schematics, and source code for our multispectral flash implementation, Android camera application, and image processing pipeline (<https://github.com/chai-toronto/chromaflash/>).

## 2 RELATED WORK

In our overview of related work, we first cover conventional methods of hyperspectral imaging. We then describe how techniques in computational imaging have been used to optimize hyperspectral image acquisition. We conclude by discussing various approaches that have been taken to make hyperspectral imaging more accessible.

### 2.1 Conventional Hyperspectral Imaging Techniques

The goal of hyperspectral imaging is to generate a 3D image wherein the first two dimensions are spatial (width and height) and the third dimension is spectral (wavelengths). Conventional hyperspectral imaging techniques typically involve scanning the scene across either the spectral or spatial dimensions in order to generate this spectral image. These methods can be further categorized as either active or passive, where passive imaging relies on natural scene lighting and active imaging utilizes controlled light sources to illuminate the scene with the desired spectra.

Spatial scanning techniques are typically passive. Whiskbroom scanning uses a 1D camera sensor to scan through a scene pixel-by-pixel, whereas pushbroom scanning uses a 2D camera sensor to scan through a scene line-by-line [19]. Before the light reaches the camera sensor, it is passed through a series of refractive lenses and then either a prism or a diffraction grating to disperse the light into multiple wavelengths. Another passive approach that implicitly provides spatial scanning involves placing a custom color filter array directly in front of the camera sensor [59]. These filter arrays consist of narrowband filters arranged in a known pattern, similar to the Bayer filter pattern used for RGB cameras. Recovering the final spectral image requires a demosaicing algorithm to interpolate missing spectral information in neighboring pixels.

Wavelength scanning techniques [32] can be either passive or active. The passive approach involves capturing a scene through optical bandpass filters that change over time, either mechanically with motors that switch distinct filters [18] or electronically with a tunable filter like the Fabry-Perot interferometer [54]. The active approach involves sequentially changing the illuminant spectrum over time using narrowband lasers or LEDs.

The aforementioned methods must adhere to the Nyquist-Shannon sampling theorem, sampling pixels and wavelengths at a sufficient rate to characterize the spectral image before advancing to a new frame. To achieve that minimum rate, these methods must sacrifice temporal resolution at the expense of reasonable spatial and spectral resolution, making them most suitable for static or slow-moving scenes. Our proposed approach relies on principles of compressed sensing to capture the spectral image in a single camera sensor frame. By utilizing computational algorithms, we reconstruct the signal from a significantly reduced number of samples in both the

spatial and spectral domains. This approach enables us to overcome the limitations of traditional methods and achieve efficient spectral imaging.

## 2.2 Computational Hyperspectral Imaging

Computational hyperspectral involves applying techniques like compressed sensing and image processing to extract hyperspectral information from a subset of measurements. Turning to computational methods circumvents the temporal limitations of scanning methods, leading to hyperspectral imaging systems with faster acquisition times. As such, prior work has leveraged computational hyperspectral imaging systems to capture spectral videos [33]. Below, we discuss some of these methods based on the underlying optical phenomenon they leverage.

**2.2.1 Refraction-based Systems.** Refraction-based cameras utilize the principles of refraction and dispersion to extract spectral information. Computed tomographic imaging spectrometry (CTIS) is one of the earliest computational approaches in this space [30]. CTIS utilizes a diffraction grating to split incident light rays into multiple spectral projections on an image plane. It is similar to spatial scanning approaches in that it splits a spectral image into spatial bands; however, CTIS captures all of these bands at once rather than scanning them over time and uses algorithms to reconstruct the target spectral image. Still, this method has its own spatial resolution limits, as the reconstructed image typically has less than 10% of the original sensor resolution. A significant breakthrough in this field came with the introduction of coded aperture snapshot spectral imagers (CASSI) [6, 56]. CASSI involves directing the scene through collimating optics onto a binary-coded aperture placed in front of a dispersive element. This arrangement spreads the spectral information over a small spatial neighborhood on the sensor, improving the distribution of wavelengths across the entire image. Capturing a set of coded images also makes the sensing problem less challenging and enables robust reconstruction of spectral information. Unfortunately, such systems require precisely customized and calibrated aperture masks, the cost of which has hindered the use of refraction-based systems outside of laboratory settings.

**2.2.2 Diffraction-based Systems.** Diffraction-based cameras make use of diffractive optical elements (DOEs) to bend and spread light waves through an opening. The DOE serves as a lens that results in spatially shifted projections variant to the spectral wavefront. Therefore, they can be modeled as the spectral integration of a convolution between the target spectral image and a point spread function for each wavelength, allowing for the spectral image to be recovered via deconvolution. Baek et al. [7] demonstrated such an arrangement by attaching a prism to a standard DSLR camera. The resulting system had a high spatial resolution, equal to that of the camera sensor; however, the spectral information was only encoded at the edges of the scene, resulting in a highly ill-conditioned inverse problem and lower spectral accuracy. Others have used a diffuser as the DOE in their system [39]. Prisms and diffusers are much more compact than the optical coding elements that are used in refractive-based systems, making diffraction-based systems inherently smaller. To further improve compactness, Jeon et al. [28] developed a custom DOE that uniquely encoded spectral information in a spectrally rotating point spread function. Fabricating the diffractive element required sophisticated processes such as photolithography and reactive-ion etching on a silicon wafer, making it difficult to mass produce. Moreover, even the most advanced DOE designs still pose severely ill-conditioned sensing problems due to the lack of spatial codification.

**2.2.3 Upsampling from RGB.** To circumvent the need for any additional hardware, several works have attempted to approximate spectral information directly from a single RGB image. Some of these approaches have mapped RGB values to spectra directly by basis functions derived from a large database [20] or smooth parametric models [2]. Unlike other methods discussed, these approaches do not attempt to capture the spectral properties of a specific scene; rather, they propose a general mapping from RGB to further spectra. Others have tried to reconstruct spectral images from RGB by optimizing ill-posed inverse problems. These methods can be split into two categories — prior-based and data-driven methods — that take advantage of the redundancy in image data.

Prior-based methods leverage statistical information in spectral images like sparsity, spatial structure similarity, and spectral correlation to seek plausible solutions in a more constrained space. Data-driven methods exploit deep learning models trained on large-scale datasets with paired RGB and hyperspectral images to find solutions. Various model architectures have been used to this end, such as convolutional neural networks (CNNs) [31, 62] and generative adversarial networks (GANs) [3, 35]. Although deep learning approaches surpass prior-based methods in terms of reconstruction accuracy, their generalizability to scenes that deviate from their training datasets remains questionable. Most of these models also do not generalize well when transferred to cameras with unknown color responses. More importantly, the spectral accuracy of these methods is limited by the metamericistic input of the RGB camera, meaning the same RGB pixel could map to different spectral curves.

While computational hyperspectral imaging addresses many limitations associated with conventional methods, challenges regarding hardware complexity, generalizability, and spectral accuracy remain. ChromaFlash takes inspiration from the aforementioned work and utilizes simpler hardware to make computational hyperspectral imaging more accessible.

### 2.3 Towards Accessible Hyperspectral Imaging

Recent advancements in hyperspectral imaging have brought forth low-cost and miniaturized solutions, making it more accessible than ever before. Within the ubiquitous computing community, there has been a particular interest in spectroscopy, driving efforts to create cost-effective solutions for various applications. These applications encompass food and beverage analysis [26, 29, 37], medication management [16], and blood glucose monitoring [60]. However, spectroscopy captures spectral information at a single point, lacking the detailed spatial data provided by hyperspectral imaging more broadly. Goel et al. were among the first to develop a low-cost active hyperspectral camera through their work on HyperCam [21], which utilized an off-the-shelf camera sensor and narrow-band LEDs. Their system was operationally similar to a wavelength scanner, sequentially capturing frames in 17 visible and NIR spectral bands at 9 fps. Cao et al. [15] demonstrated the ability to capture hyperspectral videos using a simple setup consisting of a prism and a checkerboard mask. In the realm of do-it-yourself approaches, Zhao et al. [67] were able to achieve hyperspectral imaging using a random aperture mask created using a consumer-level color printer and a CNN model to reconstruct the spectral image. Salesin et al. [47] introduced a similar system based on polarization-induced spectral filters.

In addition to dedicated hyperspectral imaging devices, recent research has sought to push the limits of the existing cameras in everyday consumer devices. Researchers have explored integrating simple optical components like prisms [7], diffractive gratings [27], and diffusers [28] with standard DSLR cameras to collect data that is later reconstructed into spectral images. Oh et al. [41] harnessed the inherent variation in spectral sensitivities across different digital cameras to estimate the spectral reflectance of scenes. Furthermore, the SpeCam project [64] utilized a smartphone's display and front-facing camera as an active illumination source and optical sensor, respectively, to enable surface color and material sensing applications. Stuart et al. [52] developed a 3D-printed attachment that transforms a smartphone into a line-scan hyperspectral imager.

These developments have led to several open-source and commercially available options for consumer hyperspectral imaging. However, devices like the Specim IQ<sup>2</sup> and Monarch II<sup>3</sup> are based on scanning methods that limit their practical usage for mobile devices. Similarly, devices like Lumos [60] and Spectricity<sup>4</sup> rely on sophisticated custom hardware that may restrict their efficacy to certain classes or generations of devices. In contrast, our generalized approach could be theoretically implemented with any rolling shutter camera by varying the wavelength of incoming light from the scene using either passive filters or active illumination light. With a

<sup>2</sup><https://www.specim.com/iq/>

<sup>3</sup><https://www.unispectral.com/product/monarch-ii-camera/>

<sup>4</sup><https://spectricity.com/product/>

simple hardware attachment and a software update, our approach can be easily deployed on the cameras found on smartphones and IoT devices to provide novel optical sensing capabilities.

### 3 THEORY OF OPERATION

Our generalized approach and the working principle of ChromaFlash rely on the fact that rolling shutter cameras do not capture an entire scene at once; rather, they scan the scene over a brief period. Changing how light is sampled in the scene during this sequential capture results in a unique coding scheme that enables the simultaneous capture of spatial and spectral data on the 2D camera sensor. This section describes this process in more detail, providing details about rolling shutter exposure, the theoretical image formation model of our technique, and the subsequent reconstruction process.

#### 3.1 Rolling Shutter Exposure

In a rolling shutter camera, the exposure of each pixel row happens  $T_{RS}$  seconds after the exposure of the preceding row<sup>5</sup>. If the topmost row is exposed at time  $T_0$ , then the subsequent row  $r$  starts becoming exposed at time  $T_r = T_0 + rT_{RS}$ . As a consequence, each row corresponds to a distinct temporal sample of the scene. Row  $r$  captures the incoming light from the scene starting at time  $T_r$  and continues for the exposure duration  $T_{EXP}$  seconds, also known as the shutter speed of the camera.  $T_{RS}$  is typically smaller than  $T_{EXP}$ , implying that a band of  $N_{EXP} = T_{EXP}/T_{RS}$  rows is exposed to the incoming light at any given instant. This configuration enables the rolling shutter to function as a spatiotemporal binary mask that regulates the incoming light.

#### 3.2 Image Formation Model

Spectral reflectance refers to the ratio between the radiant flux reflected by a surface and the radiant flux incident upon it. In simpler terms, it represents how much light at different wavelengths is reflected by an object or surface. Spectral reflectance is often measured across a range of wavelengths to characterize the reflective properties of materials. In an RGB image, spectral reflectance from every point within the scene is combined with the spectral response of the incident illumination and integrated across three broad spectral channels of the color filter array: red, green, and blue. In contrast, a hyperspectral image seeks to represent the spectral reflectance from every point within the scene according to many more spectral channels without such integration.

A hyperspectral image can be represented as a 3D cube denoted by  $H(x, y, \lambda) \in \mathbb{R}^{X \times Y \times \Lambda}$ . Here,  $X$ ,  $Y$ , and  $\Lambda$  represent the horizontal, vertical, and spectral axes respectively. Each point within the cube corresponds to the intensity value of wavelength  $\lambda$  at a spatial location  $(x, y)$  on the camera sensor. Hyperspectral images typically have  $\Lambda > 10$ , ensuring significantly greater spectral fidelity compared to RGB images. According to the Lambertian assumption [43], the relationship between a hyperspectral image and an RGB image can be expressed as:

$$I_c(x, y) = \int F_c(\lambda) L(\lambda) H(x, y, \lambda) d\lambda \quad (1)$$

For each channel  $c \in \{\text{red, green, blue}\}$ ,  $I_c$  represents the intensity of each channel in the linear RGB image.  $F_c(\lambda)$  denotes the spectral sensitivity function of the color filter array, indicating how sensitive each color filter channel is to that particular wavelength. Finally,  $L(\lambda)$  is the spectrum of the illumination incident on the scene.

Since a rolling shutter camera serves as a spatiotemporal binary mask, modulating the illumination spectrum received by the camera sensor over time can reveal spectral information along spatial dimensions. This procedure is illustrated by the timing diagram on the left side of Figure 1. Spectral modulation can either be done actively by rapidly changing the scene's illumination with spectrally distinct light sources, or it can be done passively by rapidly changing spectrally distinct filters in front of the camera sensor. The outcome of this process is a

<sup>5</sup>Some rolling shutter sensors might delay columns instead of rows. For simplicity and without loss of generality, we assume a sensor with an inter-row delay.

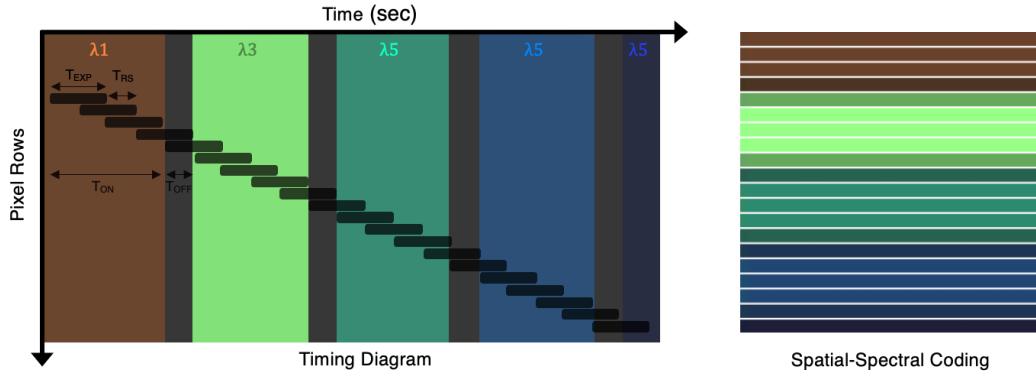


Fig. 1. The timing diagram on the left shows how modulating the illumination spectrum received by the camera sensor within a single frame produces a spatial-spectral coded mask on the right. The columns of the timing diagram represent different wavelengths received by the camera, while the horizontal black bars represent exposure during the camera's rolling shutter operation.

coded image where spectral information is spatially interleaved across successive bands of rows as shown on the right side of Figure 1. In an active imaging approach utilizing an RGB camera, as we use for ChromaFlash, each channel of the coded image  $J_c(x, y)$  can be described as the following integral equation:

$$J_c(x, y) = \iint F_c(\lambda) S(x, y, t) L(\lambda, t) H(x, y, \lambda) dt d\lambda \quad (2)$$

$H(x, y, \lambda)$  represents the hyperspectral image,  $S(x, y, t) \in \{0, 1\}^{X \times Y \times T}$  denotes the binary mask function describing the rolling shutter exposure,  $F_c(\lambda)$  denotes the spectral sensitivity function of the RGB camera sensor, and  $L(\lambda, t)$  signifies the time-varying spectral response of the active illumination. The equation remains similar for a passive imaging approach besides two differences. First, there is an additional multiplicative term  $F_+(\lambda, t)$  within the integral representing the time-varying spectral response of additional filters placed in front of the sensor, and second, the spectral response for the ambient light  $L(\lambda)$  remains fixed with respect to time.

This model can be represented more compactly in a discrete vector-and-matrix form. Let  $\mathbf{h} \in \mathbb{R}^{XY\Lambda \times 1}$  be the target hyperspectral image vector and  $\mathbf{j} \in \mathbb{R}^{XY3 \times 1}$  be the captured RGB coded image vector. The spatial-spectral mask  $\mathbf{M} \in \mathbb{R}^{XY\Lambda \times XY\Lambda}$  is calculated by integrating  $L(\lambda, t)$  and  $S(x, y, t)$  over time for an active imaging approach or  $L(\lambda)$ ,  $F_+(\lambda, t)$ , and  $S(x, y, t)$  over time for a passive imaging approach. We can represent the RGB camera sensor's spectral sensitivity function  $F_c(\lambda)$  as  $\mathbf{F} \in \mathbb{R}^{XY3 \times XY\Lambda}$ . The coding matrix  $\Theta \in \mathbb{R}^{XY3 \times XY\Lambda}$  is the product of  $\mathbf{F}$  and  $\mathbf{M}$ . Therefore, the continuous image formation model described above can be represented in a discrete matrix form as:

$$\mathbf{j} = \mathbf{F}\mathbf{M}\mathbf{h} = \Theta\mathbf{h} \quad (3)$$

Our objective is to reconstruct the original hyperspectral image  $\mathbf{h}$  from the captured coded image  $\mathbf{j}$ .

### 3.3 Ambient Light Correction

In a passive configuration wherein filters are swapped in front of the camera sensor, the working principle described thus far assumes that ambient illumination  $L(\lambda)$  is fixed in time and its spectral characteristics are known. Ideally, the illumination should be from a broadband source like sunlight or white light to cover the spectrum and achieve non-zero terms for  $L(\lambda)$  in Equation 1. Assuming the spectral characteristics of the

illuminant are not already known, they can be calculated by having a diffuse reference object (e.g., Spectralon<sup>6</sup> or white print paper) with known spectral reflectance properties in the scene.

In an active configuration that switches between spectrally distinct light sources, the working principle assumes that the total illumination reaching the camera sensor solely originates from the known active illumination source  $L(\lambda, t)$  and is reflected back by the scene. However, the presence of additional ambient illumination complicates this assumption with an additive component. Therefore, the desired coded image  $\mathbf{j}$  can be formulated as:

$$\mathbf{j} = \mathbf{j}_{\text{measured}} - \mathbf{j}_{\text{ambient}} \quad (4)$$

$\mathbf{j}_{\text{measured}}$  denotes the image captured in the presence of both active illumination and ambient light, while  $\mathbf{j}_{\text{ambient}}$  represents the image captured solely under ambient illumination. The latter quantity can either be captured as a distinct image or reconstructed from the coded image itself by having some image rows without active illumination. This equation assumes that the light reaching the camera sensor remains within its dynamic range (i.e., no saturation).

### 3.4 Hyperspectral Reconstruction

Reconstructing the complete hyperspectral image from the coded image is particularly challenging because the system of equations is severely under-determined, meaning there are far fewer measurements (rows in  $\Theta$ ) than unknowns (columns in  $\Theta$ ). Traditional least-squares inversion in such cases yields an infinite number of possible solutions that satisfy the observed measurements in  $\mathbf{j}$  as well as the coding operation  $\Theta$ , most of which are incorrect. To overcome this issue and narrow down the search space for solutions, we employ regularized least-squares inversion.

By leveraging sparsity constraints and employing sparse reconstruction techniques using  $\ell_1$ -minimization, we can efficiently find the most sparse solution of  $\mathbf{h}$  that still satisfies the observed measurements in  $\mathbf{j}$  and coding operation  $\Theta$ . This optimization problem is formulated and solved as:

$$\mathbf{h}_{\text{opt}} = \arg \min_{\mathbf{s.t. } \mathbf{h} \geq 0} \underbrace{\|\mathbf{j} - \Theta \mathbf{h}\|_2^2}_{\text{data term}} + \underbrace{\alpha \|\nabla_{xy\lambda} \mathbf{h}\|_1 + \beta \|\Delta_\lambda \mathbf{h}\|_2}_{\text{prior terms}} \quad (5)$$

In this equation, the data term ensures that the reconstruction matches the coded measurements as closely as possible. The prior terms impose additional regularization constraints to facilitate reconstruction. The 3D-TV operator  $\nabla_{xy\lambda}$  encourages solutions that are spatially and spectrally smooth by promoting sparsity of gradients in all three dimensions, while the Laplace operator  $\Delta_\lambda$  is exclusively applied to the spectral dimension to encourage a smooth transition across the spectrum. The coefficients  $\alpha$  and  $\beta$  are tunable parameters that help balance the data and prior terms.

The effectiveness of this reconstruction process depends on multiple factors, including the distribution of spectral information across the scene and the noise present during the image formation process. The problem is undersampled to the point of potentially violating the Nyquist-Shannon sampling theorem, but principles from compressed sensing theory like mutual incoherence and the restricted isometry property can be applied to facilitate successful reconstruction [14, 17]. In short, it is crucial that the data and prior terms are incoherent (i.e., not spatially or spectrally correlated) to efficiently capture the sparsity structure of the signal. This can be achieved by introducing randomness into the coding matrix  $\Theta$ , such as by randomizing the temporal order of light sources utilized in an active imaging system.

---

<sup>6</sup><https://en.wikipedia.org/wiki/Spectralon>

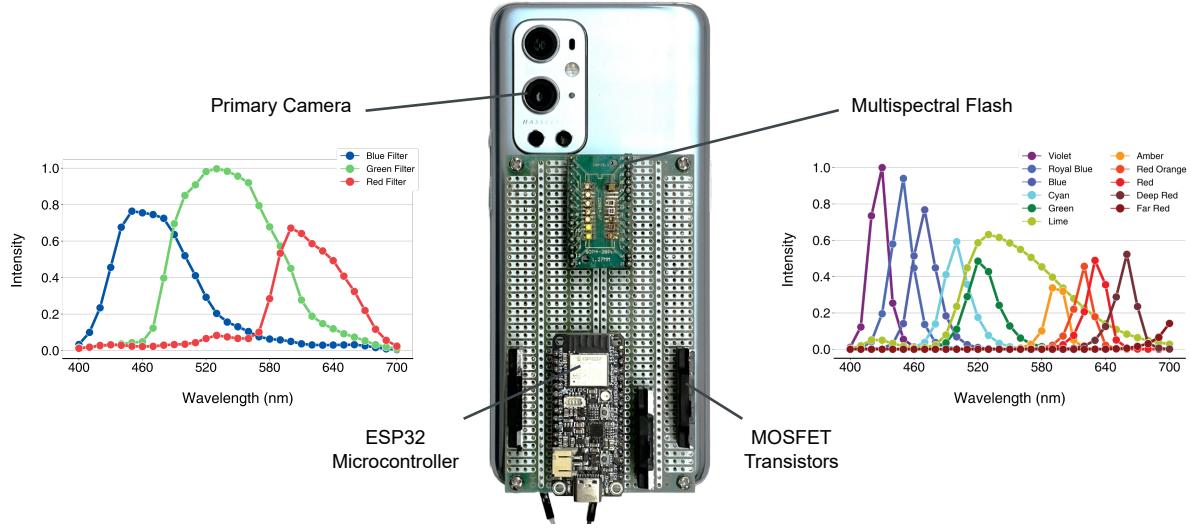


Fig. 2. ChromaFlash comprises a multispectral flash mounted on the back of a OnePlus 9 Pro smartphone. The smartphone’s camera spectral response is depicted on the left, while the spectral power distribution of the LEDs is shown on the right. Communication between the multispectral flash and the smartphone is enabled by an ESP32 microcontroller. MOSFET transistor arrays facilitate high-speed LED switching via control signals from the ESP32 board.

### 3.5 Post-Processing

The reconstruction process involves interpolating information across pixel rows with image priors informed by natural scenes. While the random sampling of spectral bands facilitates this process, interpolation accuracy can be diminished in regions of the image where measurements for specific spectral bands are sparse, resulting in noticeable band-like artifacts (see column 3 in Figure 8). While the reconstructed output typically maintains high accuracy on smooth surfaces, this effect is particularly pronounced in areas with high spatial-spectral details, such as those with numerous edges, texture, or rapidly changing colors. These visual artifacts can be remedied using traditional image filtering techniques or data-driven models.

## 4 IMPLEMENTATION

Our generalized approach to accessible hyperspectral imaging requires a rolling shutter camera, a means of manipulating the light that is either incident to the scene or received by the camera, and a software pipeline for capturing and reconstructing spatial-spectral coded images. ChromaFlash is a smartphone-specific implementation that uses LEDs to illuminate the scene in different wavelengths of light.

### 4.1 Hardware

As shown in Figure 2, our hardware setup comprises a smartphone camera and a purpose-built multispectral flash that is affixed to the smartphone. We describe each of these components below.

**4.1.1 Smartphone Camera.** ChromaFlash uses the primary rear-facing camera on a OnePlus 9 Pro. The device operates on Android 13 and is powered by a Qualcomm Snapdragon 888 octa-core processor with 12 GB RAM and 256 GB storage. The camera supports autofocus and auto white balance procedures. The camera sensor on this device, the Sony IMX 789, has an image capture resolution of 12 MP ( $4000 \times 3008$ ) and a maximum frame rate

of 120 fps. It employs an electronic rolling shutter, allowing for speeds up to  $T_{RS} = 10 \mu\text{s}$  per row, with exposure durations as short as  $T_{EXP} = 67 \mu\text{s}$ . For capturing RGB images, the camera sensor utilizes a standard RGGB Bayer filter array, which we characterized using the camSPECS XL<sup>7</sup> measurement tool. The left side of Figure 2 shows the spectral sensitivity of our camera sensor.

**4.1.2 Multispectral Flash.** We fabricated a multispectral flash that attaches to the back of a smartphone. This flash unit includes 11 unique narrowband LEDs (Lumileds Luxeon C Color) exhibiting spectral bandwidths typically ranging between 20 and 30 nm full width at half maximum. We prioritized LEDs with adequate brightness and peak wavelengths spaced roughly 30 nm apart across the visible spectrum<sup>8</sup>. The right side of Figure 2 illustrates the spectral power distribution for the LEDs we selected. The flash unit also includes a white LED (Lumileds Luxeon C 4000K) to capture the scene under broadband illumination, which we use for image post-processing described later.

The flash unit is controlled by an ESP32 microcontroller (Adafruit ESP32 Feather V2) that communicates with the smartphone via Bluetooth. When the smartphone sends a signal to the ESP32, the board rapidly cycles through the LEDs according to a predetermined sequence. The flash unit includes n-type MOSFET arrays (Sanken Electric SLA5085) to enable such high-speed switching operations. Because randomizing the illumination sequence of LEDs facilitates reconstruction, the sequence is deliberately pseudo-randomized such that each LED illuminated the scene for the same cumulative duration, but the duration is split and shuffled across the sequence randomly.

## 4.2 Software

As shown in Figure 3, our software pipeline involves acquiring coded images, processing them, and then reconstructing them into hyperspectral images. Since current smartphone camera APIs do not allow for precise synchronization between the start of camera exposure and external hardware like our multispectral flash, ChromaFlash uses a post-hoc alignment step to determine where the LED sequence starts within an image.

**4.2.1 Coded Image Capture.** We developed a custom Android application using the Camera2 API to facilitate the acquisition of spatial-spectral coded images. Like a standard camera app, pressing the shutter button within our app initiates the image capture process. This process starts by sending an activation signal to the ESP32 that controls the multispectral flash. The app then uses a simple thresholding-based algorithm to identify the optimal exposure, coupled with the phone's in-built autofocus algorithm to adjust focus for the entire frame. Once this is determined, the app initiates RAW image capture requests. The app collects three images in a single request: one with the flash activated to rapidly cycle through narrowband LEDs (multispectral mode), one with the flash deactivated (ambient mode), and one under white LED light (white mode). The first image is the only one strictly required for ChromaFlash, while the other two facilitate ambient light correction and post-processing respectively. The app allows for more images to be taken in the multispectral mode to gather a greater number of spectral measurements for each pixel. It also provides control over multiple camera settings (e.g., ISO, exposure time, number of image capture requests) and flash settings (e.g., capture mode, LED on/off duration).

**4.2.2 Pre-Processing.** Once the images have been captured and stored locally on the smartphone, they are transferred to a desktop workstation for subsequent image signal processing steps. These steps include black level adjustment, RGB demosaicing using bilinear interpolation, and normalization. Additionally, we perform ambient light correction at this stage by subtracting the image taken in ambient mode  $j_{\text{ambient}}$  from the captured coded image  $j_{\text{measured}}$  to obtain corrected coded image  $j$ .

<sup>7</sup><https://www.image-engineering.de/products/equipment/measurement-devices/588-camspecs-express>

<sup>8</sup>Ultraviolet or infrared LEDs could be used to expand ChromaFlash's spectral coverage provided that a sensor with extended spectral sensitivity is used.

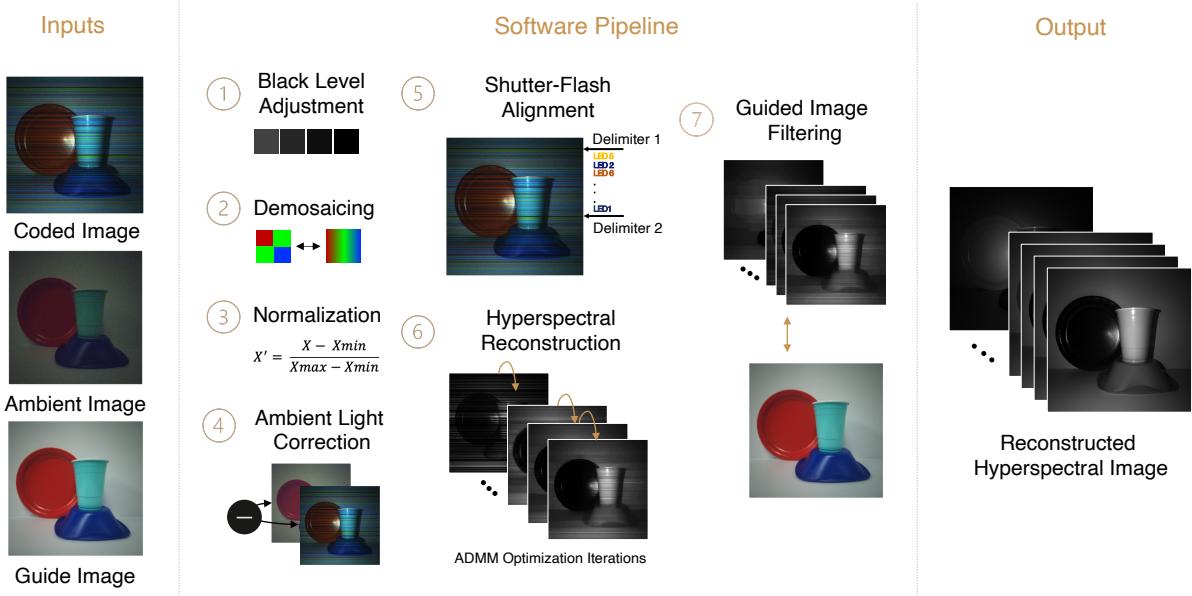


Fig. 3. The software pipeline of ChromaFlash. The system uses at most three inputs: (1) a coded image captured using a multispectral flash, (2) an image taken under ambient light for ambient light correction, and (3) an image taken under a white LED for guided image filtering; only the first is strictly required for reconstruction. After applying various pre-processing steps, ChromaFlash determines where the LED sequence started in the image by detecting a fixed illumination delimiter, after which it uses ADMM to generate the complete hyperspectral image cube. Finally, a guided image filter is used to reduce visual artifacts.

**4.2.3 Temporal Alignment of Rolling Shutter and Multispectral Flash.** To perform reconstruction from the coded images, it is imperative to precisely determine the coding matrix  $\Theta$  in Equation 3. In the context of ChromaFlash, this involves ascertaining two key parameters: (1) the camera's spectral sensitivity  $F$ , and (2) the spatial-spectral mask  $M$  formed by the interplay of the rolling shutter and the LED sequence. The camera's spectral sensitivity is a property of the color filter array and remains fixed throughout the capture process. Similarly, the behavior of the rolling shutter is set by the app and remains constant. However, determining the spatial-spectral mask also necessitates identifying the correspondence between different rows of the coded image and the specific LEDs that were illuminated at the time. This task becomes straightforward when the LEDs and rolling shutter can coordinate with one another, but this is difficult to achieve with a smartphone camera given the limited amount of control provided by device manufacturers. We instead use a post-hoc alignment procedure to temporally align the rolling shutter operation and the LED sequence.

We use a fixed illumination delimiter (blue LED, no LED, red LED) to demarcate the start of the flash's sequence. This delimiter is intentionally activated for a longer duration compared to parts of the flash's sequence, creating thicker bands that serve as reference points for alignment. We detect these bands by first calculating the average intensity of each row in grayscale and the corresponding blue and red color channels. We then identify the contiguous set of rows with the lowest brightness to determine the center of the delimiter. Since multiple delimiters appear in the same image at a known spacing, we leverage this fact to verify the detected bands. Once the delimiters have been successfully identified, the rest of the spatial-spectral mask can be inferred according to the known LED sequence.

**4.2.4 Hyperspectral Reconstruction.** Upon acquiring the coded image  $\mathbf{j}$  and computing the coding matrix  $\Theta$ , we use the alternating direction method of multipliers (ADMM) algorithm<sup>9</sup> [10] to solve the optimization problem and produce the reconstructed hyperspectral image  $\mathbf{h}_{\text{opt}}$  specified in Equation 5. We set the regularization weights to  $\alpha = [0.01, 0.01, 0.005]$  and  $\beta = 0.01$ , the ADMM tolerance value to  $\rho = 1$ , and run the algorithm for 25 iterations. To make the reconstruction process tractable for large images, we downsample by pooling  $4 \times 4$  patches and then cropping the resulting images to a resolution of  $752 \times 752$  pixels. We reconstruct hyperspectral images at 10 nm intervals between 400–700 nm, resulting in 31 spectral channels.

We implemented the reconstruction algorithm in Python using Pylops [45] for linear optimization and CuPy [42] for GPU-based acceleration. The code was run on a workstation equipped with an Intel Core i9 processor with 64 GB RAM and an NVIDIA GeForce RTX 3060 GPU with 12 GB of memory. Our unoptimized implementation currently requires 97 seconds on average to generate a hyperspectral image; readers may refer to Appendix A for a detailed breakdown of this runtime. We acknowledge that there is still considerable potential for further speedup, but optimizing the code for such efficiency was beyond the scope of our work.

**4.2.5 Post-Processing.** To mitigate visual artifacts and restore spatial details, we apply a guided image filter [24] for edge-preserving smoothing. Derived from a local linear model, the guided filter uses a reference image to identify which edges should be smoothed or preserved. This approach enables us to smooth the band artifacts while preserving the structure of the reconstructed hyperspectral image  $\mathbf{h}_{\text{opt}}$ . For ChromaFlash, the guide is an RGB image of the scene captured under white light. We used OpenCV’s implementation of guided image filtering with a filter radius of 25 pixels and  $\epsilon = 0$ .

## 5 EVALUATION: SIMULATED DATA

Before implementing ChromaFlash as a real-world system, we conducted a series of simulated experiments to assess its feasibility against benchmark hyperspectral datasets containing diverse scenes. These experiments allowed us the flexibility to modify system parameters without being concerned about hardware constraints and engineering challenges. We also used these experiments to assess how key system parameters impact ChromaFlash’s accuracy.

### 5.1 Datasets

We conducted our simulated experiments on the publicly available KAUST [34] and CAVE [63] datasets. The KAUST dataset comprises 409 hyperspectral images obtained from indoor and outdoor environments using the Specim IQ<sup>10</sup> hyperspectral camera; to enable tractable experimentation, we randomly selected 100 images from this dataset. The CAVE dataset comprises 32 hyperspectral images of various indoor objects captured through a monochrome CCD camera paired with a tunable filter. The images in both datasets have a spatial resolution of  $512 \times 512$  and 31 spectral bands across the visible spectrum (400–700 nm) at 10 nm intervals. We calibrated the images to represent the true spectral reflectance of the scenes and individually normalized them between 0 and 1 for computation.

### 5.2 Simulation Design

We designed our simulation to closely replicate the image formation process that ChromaFlash would use in the real world, with the exception that we adjusted the images’ spatial resolution to match those of benchmark hyperspectral datasets. The simulated camera model assumed a standard RGB sensor with a spatial resolution of

<sup>9</sup>ADMM is a popular algorithm that attempts to solve a convex optimization problem by breaking it into smaller pieces, each of which will be easier to handle. We refer the reader to this paper by Baek et al. [7] for a comprehensive explanation of how ADMM is used for image reconstruction.

<sup>10</sup><https://www.specim.com/iq/>

Table 1. The ChromaFlash system parameters that were varied during the simulated experiments. The underlined values indicate the defaults used when other parameters were being manipulated.

System Parameter	Description	Values
LED Duration	The duration for which each LED was kept on.	$(\underline{2}, 4, 8, 16, 32, 48) \times T_{RS}$
LED Multiplexing	The number of LEDs turned on simultaneously for a given exposure band. The LEDs were randomly selected without any potential optimizations other than avoiding repeated combinations.	$\underline{1}, 2, 3, 4, 5, 6, 7$
Number of Captured Frames	The number of frames used for reconstruction, with each frame including a different sequential order of LEDs.	$\underline{1}, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$
Noise Level	The standard deviation of the additive synthetic Gaussian noise relative to the maximum pixel intensity.	$\underline{0\%}, 1\%, 2.5\%, 5\%, 10\%$

512 × 512 that captured data at 30 fps. For the color filter array, we used the spectral sensitivity of the OnePlus 9 Pro characterized using the camSPECS XL measurement device. Although we assumed that all red, green, and blue channels could be measured at every pixel, most cameras use demosaicing algorithms to align color channels from neighboring pixels. Simplifying assumptions in our simulated experiments included uniform spatial fallout, negligible rise and fall times for the LEDs, and the sole consideration of diffuse surface reflectance.

The simulation process involved applying the spatial-spectral mask formed by combining the light and rolling shutter operation matrices to the hyperspectral images as outlined in Section 3.2. Subsequently, we transformed these masked hyperspectral images into the RGB space using the spectral sensitivity of our model camera. This was followed by normalization and the addition of synthetic Gaussian noise to simulate the read noise of a conventional camera sensor.

### 5.3 Experiment Details

Our experiments were designed to examine the importance of various parameters related to the image formation process. Table 1 enumerates these parameters and the corresponding ranges of values that were tested. Although each parameter interacts with one another during the image formation process, we limited our experiments to univariate experiments rather than an exhaustive grid search for tractability.

We also used our experiments to compare the performance of ChromaFlash relative to MST++ [13], a state-of-the-art data-driven model for RGB-to-hyperspectral image reconstruction. Unlike traditional CNN-based approaches, the MST++ architecture utilizes a neural self-attention mechanism and a transformer backbone to effectively model spectral self-similarity and long-range dependencies. We used an implementation of this model that was pre-trained on the ARAD-1K dataset [5]. We did not apply guided image filtering to either ChromaFlash or MST++ so that we could clearly convey the importance of the design factors and ensure a fair comparison.

We assessed the quality of the reconstructed images using standard metrics that describe either their spatial or spectral qualities relative to a reference image. These metrics are described below:

- **Peak signal-to-noise ratio (PSNR):** As a measure of spatial accuracy, PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Higher PSNR values represent better spatial quality.
- **Structural similarity index (SSIM):** SSIM serves as a second measure of spatial similarity, comparing luminance (mean intensity), contrast (standard deviation), and structural information (correlation) between two images. It calculates a value between -1 and 1, with higher values indicating stronger similarity.
- **Spectral angle mapper (SAM):** SAM quantifies the similarity between the spectral response of two pixels as an angular measurement in a multidimensional spectral space. Lower SAM values indicate higher spectral similarity.

We calculated PSNR and SSIM for each spectral channel, reporting the average value across channels. We computed SAM for each pixel in the image and reported the average value across pixels.

## 5.4 Results

**5.4.1 LED Duration.** Longer LED durations lead to more contiguous pixel rows being illuminated with the same light spectrum. Thicker exposure bands increase the amount of coherent spatial information that is captured within a coded image, but they also limit the diversity of spectral information that is gathered within that spatial region. Furthermore, thicker bands are more likely to increase the spacing between successive bands corresponding to the same spectral channel. Varying the LED duration allowed us to explore this potential trade-off between spatial and spectral accuracy.

As shown in Figure 4, we observed the best reconstruction performance for the shortest LED duration in our experiments. When each exposure band was two pixels thick, we observed a PSNR of  $27.67 \pm 5.37$  dB, an SSIM of  $0.89 \pm 0.07$ , and a SAM of  $14.58 \pm 9.41^\circ$  for the KAUST dataset. The corresponding metrics for the CAVE dataset were  $25.95 \pm 3.36$  dB,  $0.88 \pm 0.06$ , and  $22 \pm 7.86^\circ$  respectively. Across both datasets, we noticed that longer LED durations led to a decline in PSNR and SSIM along with an increase in SAM. This trend suggests that the benefit of having more diverse spectral information within a spatial region outweighed the potential disadvantage of losing coherent spatial information. In other words, using thicker bands necessitates the reconstruction algorithm to fill in larger spaces for each spectral channel, posing challenges for algorithms employing hand-crafted priors like the Laplace and 3D-TV operators.

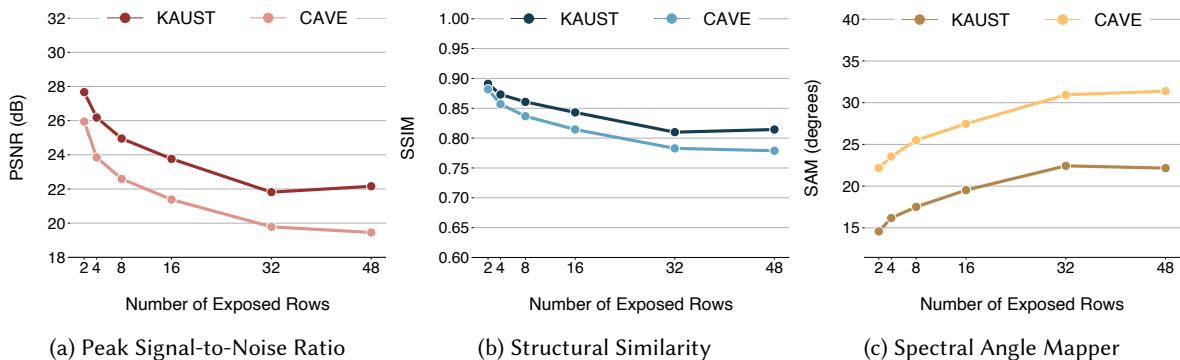


Fig. 4. The influence that varying the LED duration had on the performance of ChromaFlash in our simulated experiments on the KAUST [34] and CAVE [63] datasets.

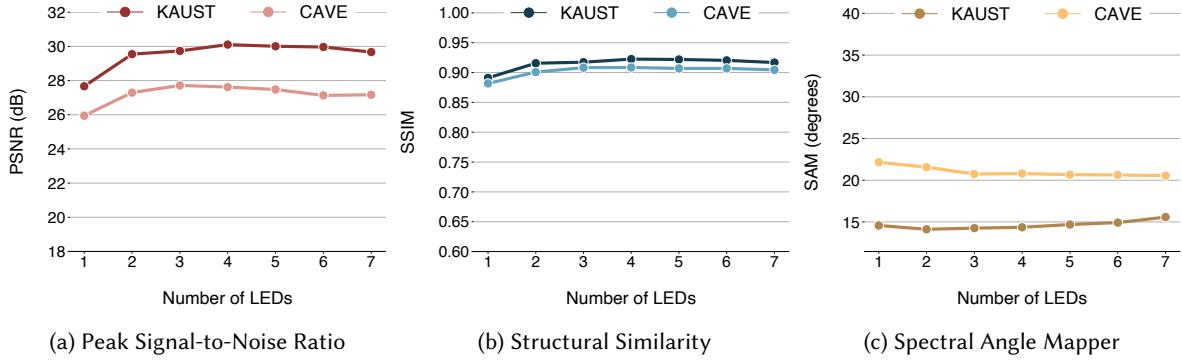


Fig. 5. The influence that LED multiplexing had on the performance of ChromaFlash in our simulated experiments on the KAUST [34] and CAVE [63] datasets.

**5.4.2 LED Multiplexing.** Measuring a scene’s appearance under multiplexed light sources has shown promise for various imaging tasks such as scene relighting [48], color estimation [8], and hyperspectral imaging [44]. In the context of ChromaFlash, multiplexing the LEDs results in each pixel receiving light from a greater number of LEDs and hence, a broader spectral coverage. However, it also removes the ability to associate each pixel’s measurement with the relatively narrow spectral response of individual LEDs. Thus, multiplexing the illumination sources results in a trade-off between spectral coverage and spectral ambiguity when solving the hyperspectral image reconstruction problem.

Figure 5 illustrates this trade-off and demonstrates that illuminating a few LEDs improved both the spatial and spectral accuracy of ChromaFlash relative to the configuration without any multiplexing. The optimal configuration for the KAUST dataset involved having three simultaneously lit LEDs, in which case we observed a 7% increase in PSNR, a 3% increase in SSIM, and a 2% decrease in SAM. The optimal configuration for the CAVE dataset also had three simultaneously lit LEDs, leading to a 7% increase in PSNR, a 3% increase in SSIM, and a 6% decrease in SAM. Beyond three LEDs, ChromaFlash’s performance either did not show any notable differences or degraded since the reconstruction algorithm was unable to disambiguate the diverse spectral responses in the absence of more measurements.

**5.4.3 Number of Captured Frames.** Capturing multiple frames with a distinct illumination pattern across different pixel rows results in a greater number of spectral measurements for each pixel, thereby reducing the ill-posedness of our reconstruction problem. However, gathering more frames comes at the cost of acquisition time.

As shown in Figure 6, the use of multiple frames substantially improved performance. We observed meaningful performance gains up until five frames, corresponding to an acquisition time of 16 ms for a 30 fps camera. Relative to the single-frame snapshot approach, we observed that accumulating measurements across five frames yielded a 9% increase in PSNR, a 3% increase in SSIM, and a 2% decrease in SAM for the KAUST dataset. Using the same number of frames for the CAVE dataset yielded a 14% increase in PSNR, a 5% increase in SSIM, and a 14% decrease in SAM. The observed performance plateau suggests that additional frames provided redundant information, given the settings for the other system parameters.

**5.4.4 Noise Level and Data-Driven Approach.** Sensor noise is an inevitable confound in most real-world imaging systems. In the context of ChromaFlash, operating at short exposures and a high gain results in the amplification of sensor read noise, which can be effectively modeled by a signal-independent zero-mean Gaussian distribution.

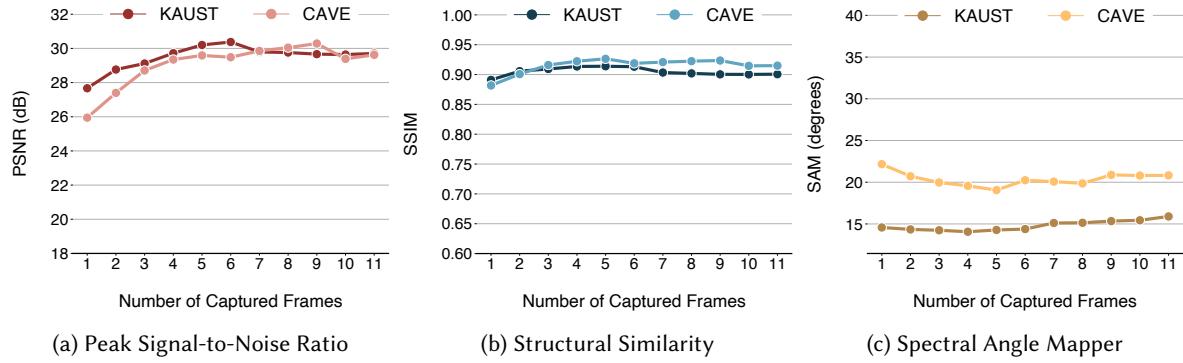


Fig. 6. The influence that increasing the number of captured frames had on the performance of ChromaFlash in our simulated experiments on the KAUST [34] and CAVE [63] datasets.

We quantify the negative impact that synthetic Gaussian noise has on our reconstruction performance across a range of standard deviation values.

Gathering multiple frames is a clear way of mitigating noise, particularly with respect to spatial information as the noise averages out across multiple measurements. Therefore, we examine the impact of noise on both a snapshot and three-frame configuration of ChromaFlash. We also used this experiment to benchmark our approach against the data-driven MST++ model. Since MST++ requires RGB images as input, we synthetically generated them by applying the RGB spectral sensitivity of our camera to the hyperspectral images.

As shown in Figure 7, ChromaFlash significantly outperformed MST++ across noise levels. Under our highest noise setting, MST++ achieved a PSNR of  $22.43 \pm 5.73$  dB, an SSIM of  $0.74 \pm 0.10$ , and a SAM of  $32.88 \pm 6.83^\circ$  for the KAUST dataset and a PSNR of  $20.99 \pm 3.47$  dB, an SSIM of  $0.75 \pm 0.08$ , and a SAM of  $37.89 \pm 6.04^\circ$  for the CAVE dataset. The snapshot configuration of ChromaFlash yielded a 10% increase in PSNR, a 5% increase in SSIM, and a 38% decrease in SAM for the KAUST dataset and a 7% increase in PSNR, a 5% increase in SSIM, and a 26% decrease in SAM for the CAVE dataset. Meanwhile, the multi-frame configuration yielded a 10% increase in PSNR, a 1% increase in SSIM, and a 26% decrease in SAM for the KAUST dataset and a 4% increase in PSNR, a 6%

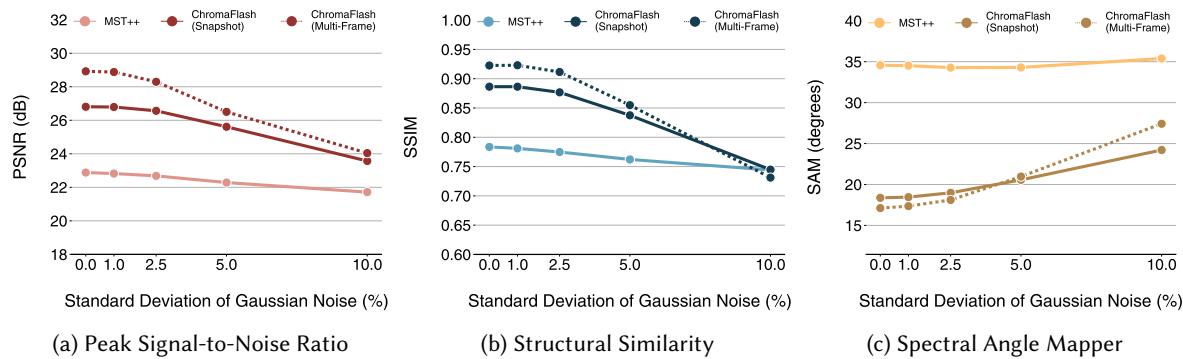
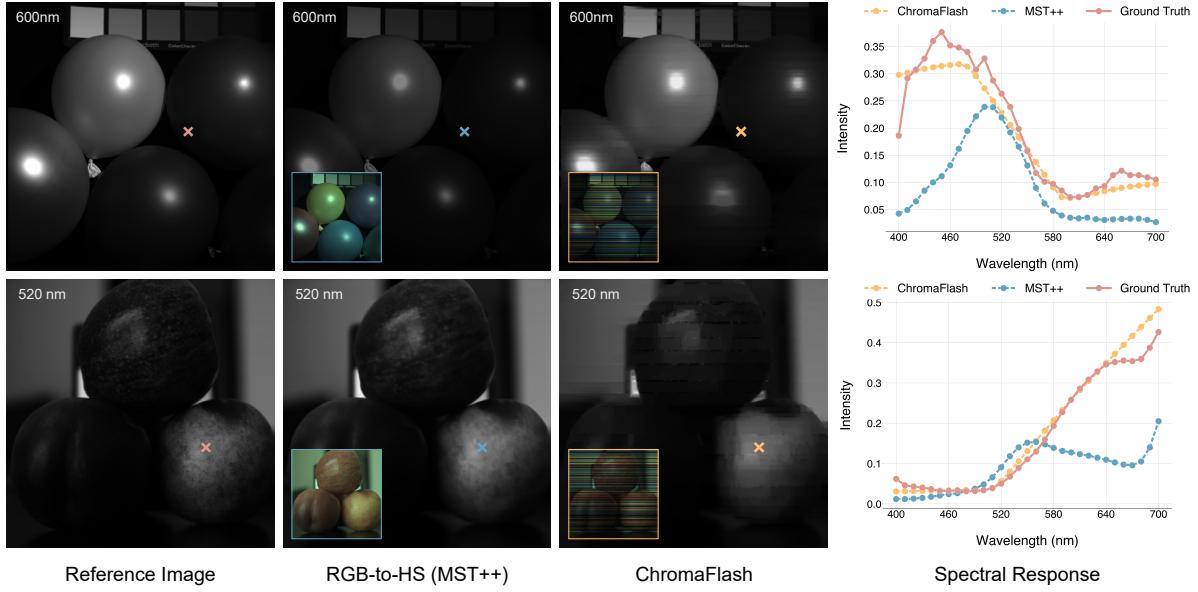


Fig. 7. The influence that varying levels of sensor noise had on the performance of ChromaFlash and MST++ [13] in our simulated experiments on the KAUST [34] and CAVE [63] datasets.



**Fig. 8. Reconstructions from simulated data:** Two examples of reconstructed hyperspectral images from scenes depicting similar shapes but different surface textures. From left to right, the columns show a hyperspectral reference image, the image reconstructed by MST++ [13], the image constructed by ChromaFlash, and the spectral response at the point indicated with a cross ( $\times$ ). For the reconstructed images, the inset shows the image that was fed into the corresponding algorithms.

decrease in SSIM, and a 6% decrease in SAM for the CAVE dataset. Examples of reconstructed images for this experiment are shown in Figure 8. This finding underscores the limited generalizability of RGB-to-hyperspectral approaches and supports a more mechanistic approach like the one presented in this paper.

## 6 EVALUATION: REAL-WORLD DATA

In this section, we discuss the results of our real-world experiments with ChromaFlash.

### 6.1 Data Collection

Our experimental setup is shown in Figure 9. We mounted ChromaFlash on a tripod one foot away from objects that were staged on a table against a white background. To regulate the ambient illumination, we used a dimmer-controlled light source positioned at a considerable distance from the setup and measured the light intensity using a digital light meter. For each scene, we captured the following images:

- **Coded Images (12):** Three images captured using ChromaFlash at each of the four following light levels: 0 lux, 50 lux, 150 lux, and 250 lux. These light levels are akin to complete darkness, dim environments like corridors and parking garages, normally lit environments like homes and warehouses, and bright environments like gyms and cafeterias. We captured multiple frames at each light level to enable the examination of both snapshot and multi-frame configurations.
- **Ambient Light Images (3):** A single image captured at each non-zero light level (50 lux, 150 lux, and 250 lux) without any illumination from ChromaFlash. These images were used for ambient light correction.
- **Reference Image (1):** An RGB image captured with illumination from ChromaFlash’s white LED. This image served as input to MST++ and as a guide for the post-processing step.

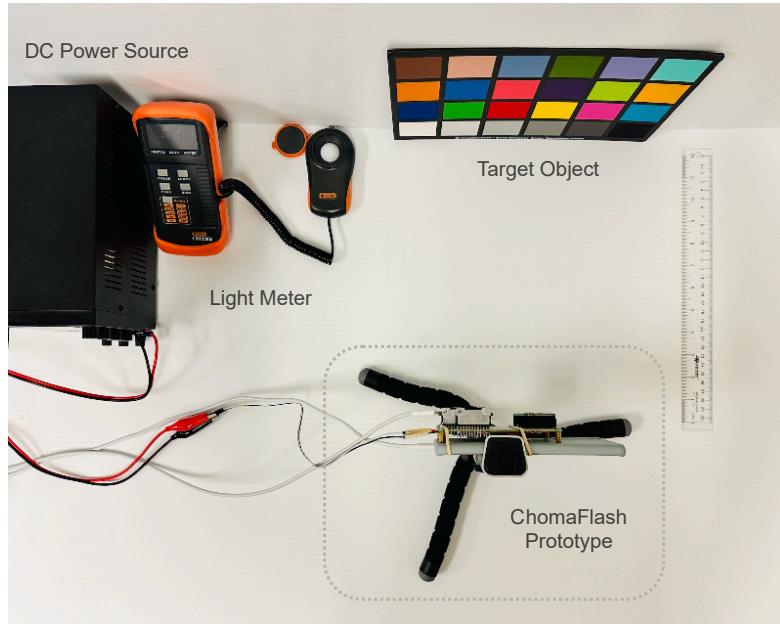


Fig. 9. The experimental setup with ChromaFlash mounted on a tripod one foot away from the target object (ColorChecker in this case). The flash was powered using a DC power supply, and ambient light levels were measured using a digital light meter.

- **Single LED Images (11):** Eleven images captured with the scene being illuminated by each LED in ChromaFlash’s multispectral flash individually. These images served as our ground truth.

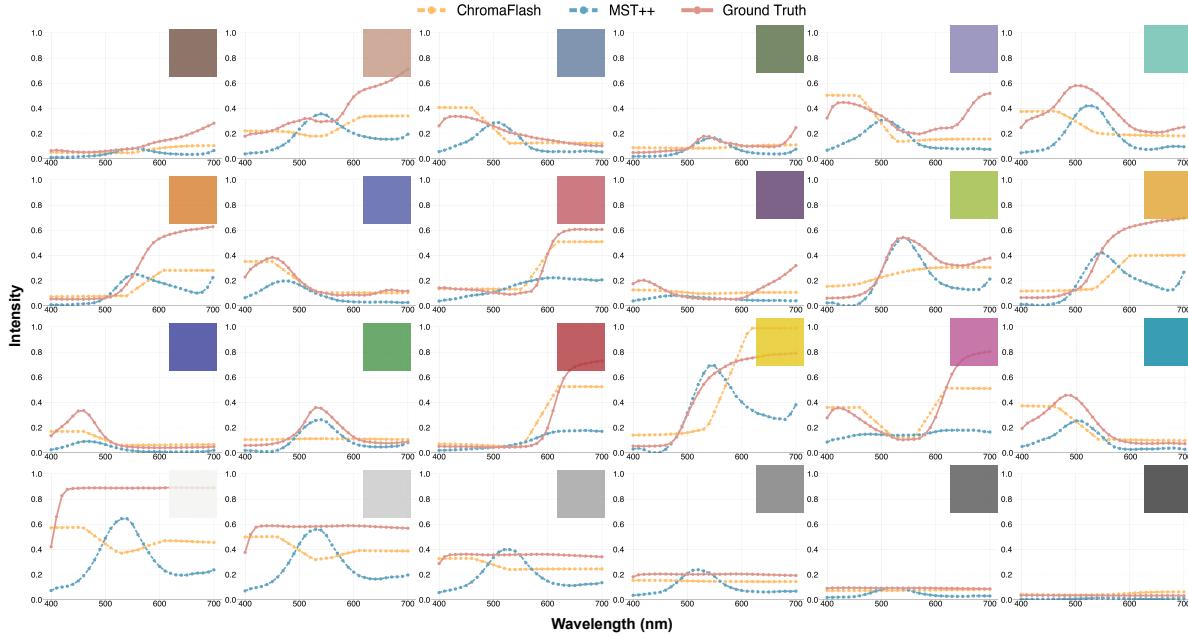
We processed the acquired coded images using ChromaFlash’s reconstruction algorithm to generate hyperspectral images with a spatial resolution of  $752 \times 752$  across 31 channels. Similarly, the reference RGB images were fed to MST++ to generate similarly sized hyperspectral images for comparison.

## 6.2 Experiment Descriptions

The Macbeth ColorChecker<sup>11</sup> comprises 24 color swatches with well-documented spectral characteristics, so we used it as a target object to quantify ChromaFlash’s spectral accuracy according to SAM. We also report ChromaFlash’s spatial accuracy according to PSNR and SSIM as before. Since the 31 channels in the reconstructed images outnumbered the 11 channels in our ground-truth spatial data, we used the LEDs’ spectral characteristics to inform a linearly weighted transformation between the two. Although the comparison could have been made in the RGB space, our chosen approach ensured a more fine-grained evaluation of spatial quality. We also qualitatively inspected the performance of our reconstruction technique on four additional scenes featuring everyday objects – fruits, toys, plasticware, and flowers – that varied in their spatial and spectral characteristics.

For both the Macbeth ColorChecker and the additional scenes, we examined spatial accuracy both with and without guided image filtering to characterize its importance. We also repeat these analyses across different ambient illumination levels to assess the effectiveness of our simple ambient light correction step.

<sup>11</sup><https://www.chromaxion.com/information/colorchecker.html>



**Fig. 10. Reconstructions from real-world data:** The 24 color swatches on a Macbeth ColorChecker and their respective spectral profiles according to ChromaFlash (snapshot configuration without guided image filtering), MST++ [13], and the ColorChecker’s specifications.

Table 2. Summary of spatial and spectral accuracy metrics for ChromaFlash and MST++ [13] in the Macbeth ColorChecker experiment.

	<b>Without Guided Image Filtering</b>	<b>With Guided Image Filtering</b>
<b>MST++</b>	PSNR = $24.20 \pm 4.15$ dB	PSNR = $26.03 \pm 4.14$ dB
	SSIM = $0.93 \pm 0.03$	SSIM = $0.94 \pm 0.04$
	SAM = $26.73 \pm 7.14^\circ$	SAM = $26.99 \pm 7.14^\circ$
<b>ChromaFlash (snapshot)</b>	PSNR = $20.23 \pm 2.58$ dB	PSNR = $25.92 \pm 4.85$ dB
	SSIM = $0.85 \pm 0.10$	SSIM = $0.88 \pm 0.07$
	SAM = $15.45 \pm 7.45^\circ$	SAM = $15.47 \pm 6.93^\circ$
<b>ChromaFlash (3-frame)</b>	PSNR = $22.88 \pm 4.49$ dB	PSNR = $25.95 \pm 4.89$ dB
	SSIM = $0.88 \pm 0.07$	SSIM = $0.89 \pm 0.06$
	SAM = $17.19 \pm 5.73^\circ$	SAM = $16.35 \pm 7.64^\circ$

### 6.3 Results: Macbeth ColorChecker

**6.3.1 Without Guided Image Filtering.** Table 2 summarizes the performance of MST++ and ChromaFlash according to spatial and spectral accuracy metrics on the Macbeth ColorChecker. MST++ achieved a PSNR of  $24.20 \pm 4.15$  dB and an SSIM of  $0.93 \pm 0.03$ . The snapshot and multi-frame of ChromaFlash exhibited slightly inferior performance without guided image filtering, albeit not significantly so. The snapshot configuration achieved a PSNR of

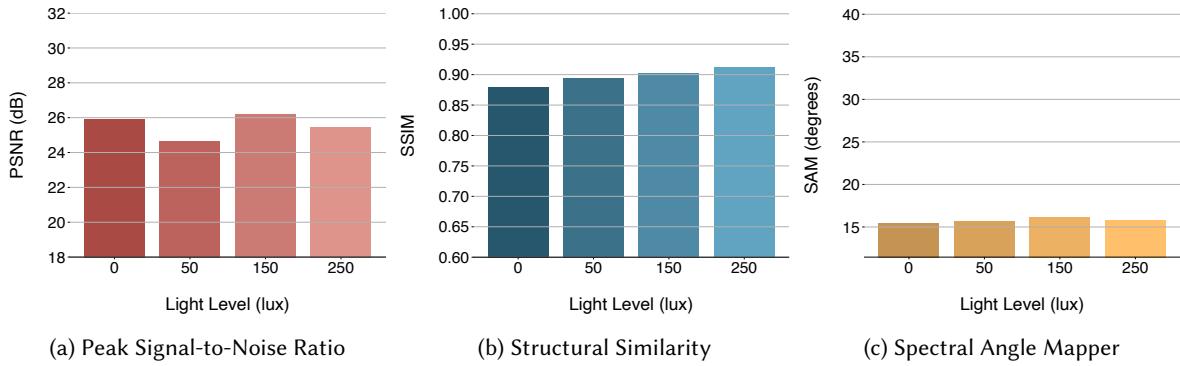


Fig. 11. The influence that varying levels of ambient illumination had on the performance of ChromaFlash (snapshot configuration with guided image filtering) in the Macbeth ColorChecker experiment.

$20.23 \pm 2.58$  dB and an SSIM of  $0.85 \pm 0.10$ , while the multi-frame configuration achieved a PSNR of  $22.88 \pm 4.49$  dB and an SSIM of  $0.88 \pm 0.07$ .

To evaluate the spectral accuracy of our prototype, we examined the spectral response curves obtained from our reconstructions and the ColorChecker datasheet. Figure 10 illustrates the effectiveness of ChromaFlash in reconstructing these spectral profiles. In contrast to MST++, which achieved a SAM of  $26.73 \pm 7.14^\circ$  across all color swatches, the snapshot configuration of ChromaFlash obtained values of  $15.45 \pm 7.45^\circ$  and  $17.19 \pm 5.73^\circ$  with single- and multi-frame configurations respectively. Surprisingly, SAM worsened in the multi-frame configuration, a trend observed in our simulations in the presence of relatively high noise levels.

**6.3.2 With Guided Image Filtering.** Using guided image filtering improved spatial accuracy for all three techniques. According to PSNR, the spatial accuracy of MST++ improved by an average of 1.83 dB, while the single- and multi-frame configurations of ChromaFlash improved by an average of 5.69 dB and 3.07 dB respectively. In fact, this post-processing step brought the average PSNR across all three techniques within 0.11 dB of one another, rendering ChromaFlash on par with MST++ according to that metric. The spatial accuracy of the three techniques did not improve as much according to SSIM, as each technique improved by an average of less than 0.03.

The impact of guided image filtering was also negligible on the spectral accuracy of MST++ and the snapshot configuration of ChromaFlash. SAM improved by an average of  $0.84^\circ$  in the multi-frame configuration of ChromaFlash, but this performance gain fell well within the variance observed across all 24 color swatches.

**6.3.3 Impact of Ambient Illumination.** The application of a simple ambient light correction step resulted in consistent performance across all evaluated light levels. Figure 11 shows how the snapshot configuration of ChromaFlash with guided image filtering performed across all of the tested light levels. In the brightest lighting condition (250 lux), ChromaFlash achieved a PSNR of  $25.45 \pm 2.79$  dB, an SSIM of  $0.91 \pm 0.04$ , and a SAM of  $15.79 \pm 6.74$ . Compared to ChromaFlash's performance in the absence of ambient lighting, these results indicate marginal decreases in PSNR and SAM values while showcasing an improvement in SSIM.

**6.3.4 Takeaways.** Overall, we observed that MST++ slightly outperformed ChromaFlash with respect to spatial accuracy, whereas the converse was true for spectral accuracy. These discrepancies may be attributed to a couple of factors. First, it is important to note that multiple images within the ARAD-1K dataset [5] used to train MST++ include a ColorChecker as part of broader scenes, so the model may be particularly adept at reconstructing images with it. Second, analysis of the spectral curves revealed that green colors between 500–560 nm posed a greater challenge to ChromaFlash's reconstruction ability relative to MST++. We speculate that this is because

ChromaFlash had inadequate sampling in the green region, due in part to the fact that the corresponding LEDs had a wide bandwidth. Conversely, we suspect that MST++ might have benefited from the greater quantum efficiency of the green color filter in RGB cameras. Despite these differences, our findings demonstrate that ChromaFlash delivers comparable results to a data-driven RGB-to-hyperspectral approach without the need for training data.

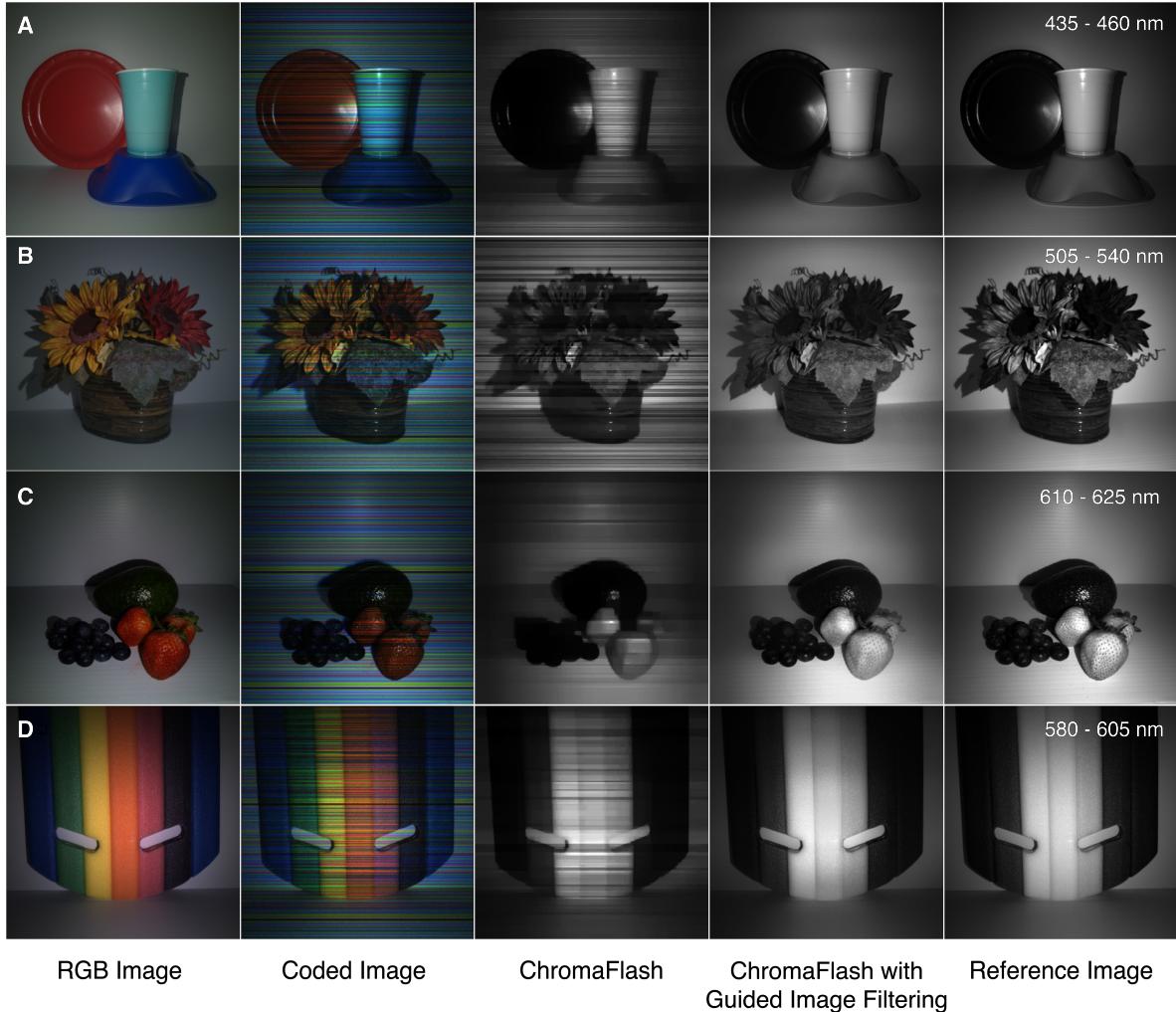
## 6.4 Results: Additional Scenes

**6.4.1 Qualitative Results.** Figure 12 shows the other scenes that we captured using the snapshot configuration of ChromaFlash; Appendix B shows a subset of these scenes across all 11 spectral bands. Although all of the objects are recognizable from the reconstructed images (column 3), their spatial quality is diminished compared to those produced during our simulations. The images exhibited band-like artifacts after reconstruction, primarily due to the interpolation of information coded by the rolling shutter between pixels. However, guided image filtering effectively eliminated such artifacts using a single frame taken under white lighting (column 4). Although this step requires an extra image capture, it substantially enhances spatial quality. In contrast, the reconstructions from MST++ did not exhibit band artifacts and were more spatially consistent without any filtering procedures, but they showed poor spectral accuracy. This visual comparison is provided in Appendix C.

Figure 13 showcases a subset of scenes under varied ambient light levels with ambient light correction applied. As in the experiment with the Macbeth ColorChecker, we observed minimal impact from this potential confound once there was a reference image under ambient lighting to subtract from the original algorithm input.

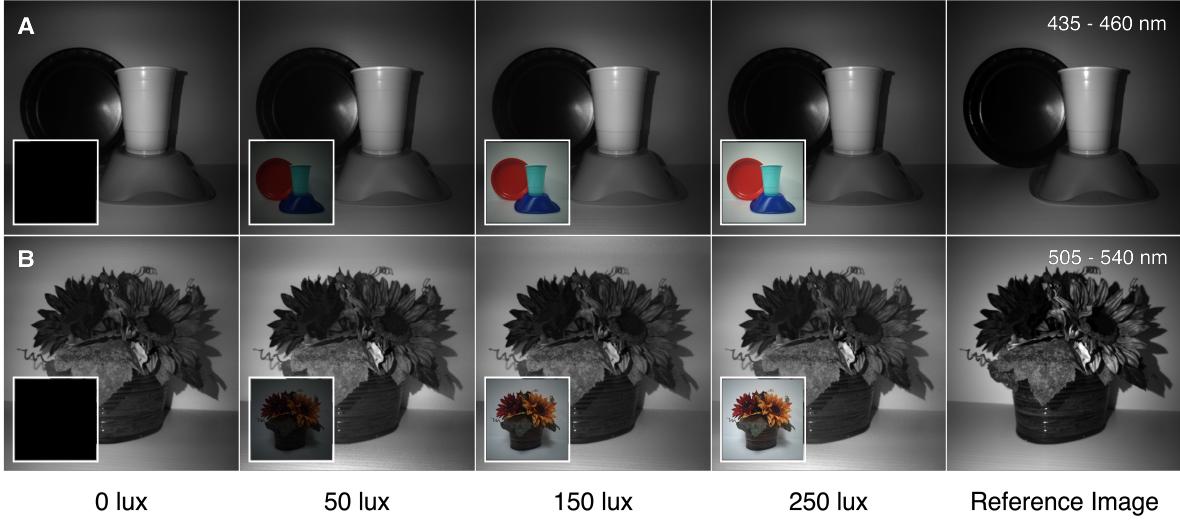
**6.4.2 Takeaways.** There are many factors that may have contributed to the performance discrepancy between the simulations and real-world experiments that we ran. A subset of them are described below:

- **LED Transition Times:** We assumed instantaneous rise and fall times for the LEDs in our simulated experiments. However, the LEDs we used in our implementation took approximately  $40\ \mu s$  to transition between states, causing rows along the fringes of a given band to receive less light than those in the middle. This parameter could have been incorporated in the spatial-spectral mask used to represent the illumination pattern, but doing so perfectly would require perfect synchronization between the LEDs and the rolling shutter. Otherwise, the rows within a given band would be scaled incorrectly. Because of imperfect shutter-flash alignment and the potential blurring between band fringes, the horizontal artifacts were more apparent in our real-world data.
- **Uniform Lighting:** We assumed that each LED uniformly illuminated the entire scene, but LEDs typically emit a narrow cone of high intensity that diminishes further away from the center. Furthermore, the LEDs in our implementation were arranged in a way such that they had slightly different optical axes relative to the camera. The uniformity of real-world lighting also depends on the distance between the LEDs and the scene. At far distances, the unevenness of the lighting becomes more apparent and the difference in the optical axes becomes less important. Conversely, putting the hardware close to the target object can ensure that the majority of the camera frame lies in the flash’s hotspot where the illumination is essentially uniform, although doing so sacrifices the camera’s field of view. In our real-world experiments, the heterogeneity of the lighting ultimately led to poorer reconstruction along the top and bottom of the images where the lighting was less uniform.
- **Object Reflectance:** An additional element that could not have been factored into our simulation was the reflective properties of the target objects. All objects exhibit some combination of diffuse and specular reflection. Diffuse reflection occurs when light hits a rough or non-glossy surface and scatters in many different directions, creating a matte appearance. Specular reflection occurs when light hits a smooth or glossy surface and reflects at a specific angle, creating a shiny appearance with bright spots. Our simulated experiments relied on passive hyperspectral imaging datasets, so the reflective properties of the target



**Fig. 12. Reconstructions from real-world data:** Four scenes depicting (A) plasticware, (B) flowers in a vase, (C) fruits and vegetables, and (D) a multi-colored pool toy. From left to right, the columns show an RGB image of the scene captured under white light, a coded image captured using our ChromaFlash prototype, the reconstructed image from the snapshot configuration at a specific wavelength, the reconstructed output following the guided filtering step, and the ground-truth image for that particular wavelength.

objects only influenced the amount of natural light that was reflected back to the camera. On the other hand, ChromaFlash uses active illumination for hyperspectral image reconstruction, making its accuracy heavily dependent upon how the target objects reflect the multispectral flash's light. In general, we observed more coherent reconstruction along the spatial dimensions in regions with less specular reflection.



**Fig. 13. Reconstructions from real-world data:** Two scenes showcasing (A) plasticware, and (B) flowers in a vase. From left to right, the columns show the final reconstructed images at specific wavelengths from our prototype system, corresponding to different ambient light levels: 0 lux, 50 lux, 150 lux, and 250 lux (depicted in the insets). The rightmost image in each row represents the ground truth for that particular wavelength.

## 7 DISCUSSION

In our discussion, we first describe design considerations and potential applications for ChromaFlash. We then enumerate additional limitations of our generalized approach.

### 7.1 Implementation Recommendations

Our experiments with ChromaFlash revealed important design considerations for both our generalized approach and our specific implementation thereof.

**7.1.1 Hardware Optimizations.** ChromaFlash requires light sources across the spectrum in order to probe different components of the scene’s spectral signature. We used 11 narrowband LEDs that were spaced roughly 30 nm apart to cover most of the visible spectrum (400–700 nm), but the peak wavelengths of the LEDs could be catered towards a single target application if it demands higher accuracy in specific portions of the spectrum. Regardless, using LEDs with narrow bandwidths is critical for accurate reconstruction. If two LEDs have similar characteristics, then the resulting spectral information will be conflated during reconstruction. We observed this phenomenon in two of our experimental results. First, our simulations revealed that LED multiplexing only provided benefits when a few LEDs were illuminated at the same time; once more LEDs were illuminated, the spectra began to overlap and produce a broadband light source. Second, our real-world experiments yielded better results for reds and blues than for greens. This is because green LEDs inherently have a wider bandwidth – a commonly known problem known as the “green gap” [53].

**7.1.2 Software Optimizations.** Our simulations revealed that thinner spatial bands are preferable for ChromaFlash. We varied this parameter by changing the LED duration while assuming a fixed shutter speed setting on the camera. Conversely, we could have changed the camera’s shutter speed parameter while keeping the LED duration fixed, but we chose the former approach since faster shutter speeds provide more precise control over the spatial

resolution of the bands. Nevertheless, the camera’s shutter speed is important for multiple reasons beyond band thickness. If the shutter speed is too fast and the exposure time is very short, the camera sensor will not have enough time to receive sufficient light within a single pixel row, leading to poor signal quality. If the shutter speed is too slow, the light from consecutive LEDs can mix within the same row. Although our simulated experiments showed that some amount of multiplexing is helpful, we posit that such mixing is undesirable since it is harder to determine the contributions of each LED within a given row. With these considerations in mind, it is optimal to have a shutter speed that balances the benefits of moderate LED mixing with the need for sufficient light in a single pixel row. Further investigation could also be put towards optimizing the LED sequence such that the overlapping spectra are relatively orthogonal. While we used a pseudo-random sequence in line with compressed sensing theory, algorithms for selecting the optimal LED multiplexing have been shown to improve performance in comparable tasks [40].

**7.1.3 Calibration.** While ChromaFlash allows developers to empirically specify a number of hardware and software specifications, there are inevitably going to be shortcomings in this process. Camera specifications can be difficult to find online, the spectral characteristics for LEDs may be incorrectly reported, and facets of the image formation process may need to be assumed as we did in our work. These challenges can be overcome by employing a calibration process similar to what is done for professional photography. This procedure would entail capturing an image with a color calibration target like a Macbeth ColorChecker and then calculating a matrix that can be applied post-reconstruction to recover radiometrically accurate results.

## 7.2 Potential Applications

Our experiments alluded to potential applications of ChromaFlash, but we expand on these opportunities here.

**7.2.1 Domain-Specific Applications.** Hyperspectral imaging has been utilized for an array of human-centered applications across domains [22], and ChromaFlash draws these applications closer to end-users without access to traditional hyperspectral cameras. For example, hyperspectral imaging has been used to analyze the chemical composition of produce [61], which can be used for quality control and shelf-life prediction. Similar capabilities have been applied to material sensing. Although many plastics share the same translucent appearance, only a fraction of plastic objects are recyclable. Hyperspectral imaging has been used to discriminate between different types of plastic to help individuals make this determination in the absence of proper labels [25]. In digital forensics, hyperspectral imaging has been used to validate the authenticity of documents and art pieces by analyzing the spectral characteristics of the materials and revealing hidden or altered features. Finally, many medical applications rely on precise color information about tissues, veins, and other parts of the body. One of the most popular applications in this space involves superficial blood vessel localization [49], but devices like pulse oximeters also rely on measuring the relative absorption of different wavelengths of light. This technique has been translated to smartphone applications that estimate blood oxygenation [11, 12] and hemoglobin [57, 58], which could be made more precise with hyperspectral data.

**7.2.2 Form Factors.** Our real-world experiments were conducted such that ChromaFlash was placed close to the target objects. Although this may seem like a restrictive limitation, we foresee many form factors and deployments that would be amenable to these conditions. Since we showed that we can account for ambient lighting, we believe that our instantiation of ChromaFlash could be used indoors provided that the user moves their smartphone close to the target object. A small enclosure could also be placed around the smartphone, allowing users to place objects in a dark compartment that precludes the need for ambient light correction. Factory machines, recycling bins, and household appliances (e.g., refrigerators and vacuum cleaners) could be similarly constructed to analyze objects in relative darkness.

**7.2.3 Hyperspectral Video.** Past hyperspectral imaging systems proposed by mobile and ubiquitous computing researchers [21, 50, 60] can already enable the aforementioned domain-specific applications to a degree. However, ChromaFlash surpasses these systems by enabling snapshot capture. Doing so paves the way for hyperspectral videos that can handle dynamic scenes. This affordance can be useful when either the target object or the camera is moving. Many proposed systems for localizing blood vessels involve head-worn devices, which means that the information shown on the display must update itself as the clinician’s head moves. Without hyperspectral video, the clinician would have to keep their head still in order to have a hyperspectral camera scan the patient. The patients themselves can also move. Although it may be reasonable to assume that an adult patient can keep their arm still while a standard hyperspectral imaging system scans the area, the same may not be true for a baby. Being able to deliver real-time hyperspectral data for every video frame would enable a clinician to track a baby’s blood vessels as they move their arm.

### 7.3 Limitations

Beyond the assumptions we noted throughout our analyses, there are other limitations to our implementation of ChromaFlash. First, our software pipeline does not explicitly address sensor noise. The ADMM algorithm allows for integrating sophisticated denoisers as additional plug-and-play priors [55], providing a hybrid approach that combines the advantages of traditional optimization and deep learning; however, we chose to omit these techniques to keep our approach simple. Second, we only tested ChromaFlash in indoor scenarios where the target object was close to the camera. We believe that these conditions are suitable for many applications, yet testing ChromaFlash at further distances and in the presence of outdoor sunlight remains an area for future exploration. Future work could incorporate ambient illumination into the image formation model itself, eliminating the need for an ambient frame subtraction step. Ambient light could conceptually be treated as another LED in the multispectral flash, with multiple pixel rows in a frame capturing the scene under ambient light using our existing approach. For application contexts that require operation under bright daylight such as drones or farming, a passive implementation of our rolling-shutter coded approach with time-varying filters may be more suitable. Finally, we demonstrated the algorithmic feasibility of reconstructing the hyperspectral images from a single camera frame, but we did not implement a real-time snapshot hyperspectral imaging system. Doing so would require implementing and optimizing our reconstruction algorithm so that each camera frame can be fully processed before the next one is received, but this effort was beyond the scope of our work. Potential solutions to this include employing memory-efficient data structures to streamline reconstruction, implementing pixel quantization to reduce computational complexity, and leveraging parallelization to handle frame-level computations more efficiently.

## 8 CONCLUSION

We demonstrated the possibility of using commodity rolling shutter cameras for accessible, snapshot hyperspectral imaging provided a means of controlling the wavelength of light incident on the sensor. We implemented and evaluated ChromaFlash using a combination of simulated and real-world data to not only compare our approach to a state-of-the-art data-driven technique but also to evaluate the implications of different design parameters and potential confounds. We hope that our work inspires researchers to build off of our blueprint for accessible hyperspectral imaging in order to accelerate the development of ubiquitous computing applications that may benefit from advanced spectral analysis capabilities.

## ACKNOWLEDGMENTS

A. Mariakakis acknowledges the funding support of the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant. K. N. Kutulakos acknowledges NSERC support under the NSERC Discovery

and RTI programs. D. B. Lindell acknowledges the support of the Canada Foundation for Innovation and the Ontario Research Fund. The authors also gratefully acknowledge Michael S. Brown and Ian MacPherson from York University for lending us their consultation and equipment (camSPECs XL and spectroradiometer).

## REFERENCES

- [1] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (*UIST '19*). Association for Computing Machinery, New York, NY, USA, 453–462. <https://doi.org/10.1145/3332165.3347889>
- [2] Naveed Akhtar and Ajmal Mian. 2018. Hyperspectral recovery from RGB images using Gaussian processes. *IEEE transactions on pattern analysis and machine intelligence* 42, 1 (2018), 100–113.
- [3] Aitor Alvarez-Gila, Joost Van De Weijer, and Estibaliz Garrote. 2017. Adversarial networks for spatial context-aware spectral image reconstruction from RGB. In *Proceedings of the IEEE international conference on computer vision workshops*. 480–490.
- [4] Nick Antipa, Patrick Oare, Emrah Bostan, Ren Ng, and Laura Waller. 2019. Video from stills: Lensless imaging with rolling shutter. In *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 1–8.
- [5] Boaz Arad, Radu Timofte, Rony Yahel, Nimrod Morag, Amir Bernat, Yuanhao Cai, Jing Lin, Zudi Lin, Haoqian Wang, Yulun Zhang, Hanspeter Pfister, Luc Van Gool, Shuai Liu, Yongqiang Li, Chaoyu Feng, Lei Lei, Jiaoqiao Li, Songcheng Du, Chaoxiong Wu, Yihong Leng, Rui Song, Mingwei Zhang, Chongxing Song, Shuyi Zhao, Zhiqiang Lang, Wei Wei, Lei Zhang, Renwei Dian, Tiansi Shan, Anjing Guo, Chengguo Feng, Jinyang Liu, Mirko Agarla, Simone Bianco, Marco Buzzelli, Luigi Celona, Raimondo Schettini, Jiang He, Yi Xiao, Jiajun Xiao, Qiangqiang Yuan, Jie Li, Liangpei Zhang, Taesung Kwon, Dohoony Ryu, Hyokyoung Bae, Hao-Hsiang Yang, Hua-En Chang, Zhi-Kai Huang, Wei-Ting Chen, Sy-Yen Kuo, Junyu Chen, Haiwei Li, Song Liu, Sabarinathan, K Uma, B Sathya Bama, and S. Mohamed Mansoor Roomi. 2022. NTIRE 2022 Spectral Recovery Challenge and Data Set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 863–881.
- [6] Gonzalo R Arce, David J Brady, Lawrence Carin, Henry Arguello, and David S Kittle. 2013. Compressive coded aperture spectral imaging: An introduction. *IEEE Signal Processing Magazine* 31, 1 (2013), 105–115.
- [7] Seung-Hwan Baek, Incheol Kim, Diego Gutierrez, and Min H Kim. 2017. Compact single-shot hyperspectral imaging using a prism. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–12.
- [8] Richard L Baer, William D Holland, Jack M Holm, and Poorvi L Vora. 1999. Comparison of primary and complementary color filters for CCD-based digital photography. In *Sensors, Cameras, and Applications for Digital Photography*, Vol. 3650. SPIE, 16–25.
- [9] Ananta Narayanan Balaji, Clayton Kimber, David Li, Shengzhi Wu, Ruofei Du, and David Kim. 2023. RetroSphere: Self-Contained Passive 3D Controller Tracking for Augmented Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 157 (jan 2023), 36 pages. <https://doi.org/10.1145/3569479>
- [10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.
- [11] Nam Bui, Anh Nguyen, Phuc Nguyen, Hoang Truong, Ashwin Ashok, Thang Dinh, Robin Deterding, and Tam Vu. 2017. Pho2: Smartphone based blood oxygen level measurement systems using near-ir and red wave-guided light. In *Proceedings of the 15th ACM conference on embedded network sensor systems*. 1–14.
- [12] Nam Bui, Anh Nguyen, Phuc Nguyen, Hoang Truong, Ashwin Ashok, Thang Dinh, Robin Deterding, and Tam Vu. 2020. Smartphone-based spo2 measurement by exploiting wavelengths separation and chromophore compensation. *ACM Transactions on Sensor Networks (TOSN)* 16, 1 (2020), 1–30.
- [13] Yuanhao Cai, Jing Lin, Zudi Lin, Haoqian Wang, Yulun Zhang, Hanspeter Pfister, Radu Timofte, and Luc Van Gool. 2022. Mst++: Multi-stage spectral-wise transformer for efficient spectral reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 745–755.
- [14] Emmanuel J Candès and Terence Tao. 2005. Decoding by linear programming. *IEEE transactions on information theory* 51, 12 (2005), 4203–4215.
- [15] Xun Cao, Hao Du, Xin Tong, Qionghai Dai, and Stephen Lin. 2011. A prism-mask system for multispectral video acquisition. *IEEE transactions on pattern analysis and machine intelligence* 33, 12 (2011), 2423–2435.
- [16] Pravin V Dhole, Vijay D Dhanger, Sulochana D Shejul, and Bharti W Gawali. 2023. A Survey on Hyperspectral Sensing Techniques for Identification of Fake Pharmaceuticals Medicines. *AITC-2023 and CSSP-2023* (2023), 244.
- [17] David L Donoho, Xiaoming Huo, et al. 2001. Uncertainty principles and ideal atomic decomposition. *IEEE transactions on information theory* 47, 7 (2001), 2845–2862.
- [18] Jason M Eichenholz, Nick Barnett, and Dave Fish. 2010. Sequential filter wheel multispectral imaging systems. In *Applied industrial optics: Spectroscopy, imaging and metrology*. Optica Publishing Group, ATuB2.
- [19] James E Fowler. 2014. Compressive pushbroom and whiskbroom sensing for hyperspectral remote-sensing imaging. In *2014 IEEE international conference on image processing (ICIP)*. IEEE, 684–688.

- [20] Biebelo Joslyn Fubara, Mohamed Sedky, and David Dyke. 2020. RGB to spectral reconstruction via learned basis functions and weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 480–481.
- [21] Mayank Goel, Eric Whitmire, Alex Mariakakis, T Scott Saponas, Neel Joshi, Dan Morris, Brian Guenter, Marcel Gavriliu, Gaetano Borriello, and Shwetak N Patel. 2015. HyperCam: hyperspectral imaging for ubiquitous computing applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 145–156.
- [22] Hans Grahn and Paul Geladi. 2007. *Techniques and applications of hyperspectral image analysis*. John Wiley & Sons.
- [23] Xavier Hadoux, Flora Hui, Jeremiah KH Lim, Colin L Masters, Alice Pébay, Sophie Chevalier, Jason Ha, Samantha Loi, Christopher J Fowler, Christopher Rowe, et al. 2019. Non-invasive in vivo hyperspectral imaging of the retina for potential biomarker use in Alzheimer’s disease. *Nature communications* 10, 1 (2019), 4227.
- [24] Kaiming He, Jian Sun, and Xiaou Tang. 2012. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence* 35, 6 (2012), 1397–1409.
- [25] Martin L Henriksen, Celine B Karlsen, Pernille Klarskov, and Mogens Hinge. 2022. Plastic classification via in-line hyperspectral camera analysis and unsupervised machine learning. *Vibrational Spectroscopy* 118 (2022), 103329.
- [26] Haiyan Hu, Qianyi Huang, and Qian Zhang. 2023. BabyNutri: A Cost-Effective Baby Food Macronutrients Analyzer Based on Spectral Reconstruction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–30.
- [27] Haiquan Hu, Hao Zhou, Zhihai Xu, Qi Li, Huajun Feng, Yueting Chen, Tingting Jiang, and Wenbin Xu. 2022. Practical snapshot hyperspectral imaging with DOE. *Optics and Lasers in Engineering* 156 (2022), 107098.
- [28] Daniel S Jeon, Seung-Hwan Baek, Shinyoung Yi, Qiang Fu, Xiong Dun, Wolfgang Heidrich, and Min H Kim. 2019. Compact snapshot hyperspectral imaging with diffracted rotation. (2019).
- [29] Weiwei Jiang, Gabriele Marini, Niels van Berkel, Zhanna Sarsenbayeva, Zheyu Tan, Chu Luo, Xin He, Tilman Dingler, Jorge Goncalves, Yoshihiro Kawahara, et al. 2019. Probing sucrose contents in everyday drinks using miniaturized near-infrared spectroscopy scanners. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–25.
- [30] William R Johnson, Daniel W Wilson, Wolfgang Fink, Mark Humayun, and Greg Bearman. 2007. Snapshot hyperspectral imaging in ophthalmology. *Journal of biomedical optics* 12, 1 (2007), 014036–014036.
- [31] Sriharsha Koundinya, Himanshu Sharma, Manoj Sharma, Avinash Upadhyay, Raunak Manekar, Rudrabha Mukhopadhyay, Abhijit Karmakar, and Santanu Chaudhury. 2018. 2D-3D CNN based architectures for spectral reconstruction from RGB images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 844–851.
- [32] Haebom Lee and Min H Kim. 2014. Building a two-way hyperspectral imaging system with liquid crystal tunable filters. In *Image and Signal Processing: 6th International Conference, ICISP 2014, Cherbourg, France, June 30–July 2, 2014. Proceedings* 6. Springer, 26–34.
- [33] Gareth M León-López, Laura V Galvis Carreno, and Henry Arguello Fuentes. 2018. Temporal colored coded aperture design in compressive spectral video sensing. *IEEE Transactions on Image Processing* 28, 1 (2018), 253–264.
- [34] Yuqi Li, Qiang Fu, and Wolfgang Heidrich. 2021. Multispectral illumination estimation using deep unrolling network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2672–2681.
- [35] Pengfei Liu and Huaici Zhao. 2020. Adversarial networks for scale feature-attention spectral image reconstruction from a single RGB. *Sensors* 20, 8 (2020), 2426.
- [36] S Mahesh, DS Jayas, J Paliwal, and NDG White. 2015. Hyperspectral imaging to classify and monitor quality of agricultural materials. *Journal of Stored Products Research* 61 (2015), 17–26.
- [37] Hidenori Matsui, Takahiro Hashizume, and Koji Yatani. 2018. Al-light: An alcohol-sensing smart ice cube. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 2, 3 (2018), 1–20.
- [38] Binu Melit Devassy and Sony George. 2021. Forensic analysis of beverage stains using hyperspectral imaging. *Scientific reports* 11, 1 (2021), 6512.
- [39] Kristina Monakhova, Kyrollos Yanny, Neerja Aggarwal, and Laura Waller. 2020. Spectral DiffuserCam: lensless snapshot hyperspectral imaging with a spectral filter array. *Optica* 7, 10 (2020), 1298–1307.
- [40] Muyao Niu, Zhuoxiao Li, Zhihang Zhong, and Yinqiang Zheng. 2023. Visibility constrained wide-band illumination spectrum design for seeing-in-the-dark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13976–13985.
- [41] Seoung Wug Oh, Michael S Brown, Marc Pollefeyt, and Seon Joo Kim. 2016. Do it yourself hyperspectral imaging with everyday digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2461–2469.
- [42] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. 2017. CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*. [http://learningsys.org/nips17/assets/papers/paper\\_16.pdf](http://learningsys.org/nips17/assets/papers/paper_16.pdf)
- [43] Michael Oren and Shree K Nayar. 1994. Generalization of Lambert’s reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 239–246.
- [44] Jong-Il Park, Moon-Hyun Lee, Michael D. Grossberg, and Shree K. Nayar. 2007. Multispectral Imaging Using Multiplexed Illumination. In *2007 IEEE 11th International Conference on Computer Vision*. 1–8. <https://doi.org/10.1109/ICCV.2007.4409090>

- [45] Matteo Ravasi and Ivan Vasconcelos. 2020. PyLops—A linear-operator Python library for scalable algebra and optimization. *SoftwareX* 11 (2020), 100361.
- [46] Kyle Rector, Cynthia L. Bennett, and Julie A. Kientz. 2013. Eyes-Free Yoga: An Exergame Using Depth Cameras for Blind & Low Vision Exercise. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington) (ASSETS '13). Association for Computing Machinery, New York, NY, USA, Article 12, 8 pages. <https://doi.org/10.1145/2513383.2513392>
- [47] Katherine Salesin, Dario Seyb, Sarah Friday, and Wojciech Jarosz. 2022. DIY hyperspectral imaging via polarization-induced spectral filters. In *2022 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 1–12.
- [48] Soumyadip Sengupta, Brian Curless, Ira Kemelmacher-Shlizerman, and Steven M Seitz. 2021. A light stage on every desk. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2420–2429.
- [49] Aamir Shahzad, Mohamad Naufal Saad, Nicolas Walter, Aamir Saeed Malik, and Fabrice Meriaudeau. 2014. Hyperspectral venous image quality assessment for optimum illumination range selection based on skin tone characteristics. *Biomedical engineering online* 13, 1 (2014), 1–13.
- [50] Neha Sharma, Muhammad Shahzaib Waseem, Shahrzad Mirzaei, and Mohamed Hefeeda. 2023. MobiSpectral: Hyperspectral Imaging on Mobile Devices. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.
- [51] Mark Sheinin, Yoav Y. Schechner, and Kiriakos N. Kutulakos. 2018. Rolling shutter imaging on the electric grid. In *2018 IEEE International Conference on Computational Photography (ICCP)*. 1–12. <https://doi.org/10.1109/ICCPHOT.2018.8368472>
- [52] Mary B Stuart, Andrew JS McGonigle, Matthew Davies, Matthew J Hobbs, Nicholas A Boone, Leigh R Stanger, Chengxi Zhu, Tom D Pering, and Jon R Willmott. 2021. Low-cost hyperspectral imaging with a smartphone. *Journal of Imaging* 7, 8 (2021), 136.
- [53] Muhammad Usman, Munaza Munsif, Urooj Mushtaq, Abdur-Rehman Anwar, and Nazeer Muhammad. 2021. Green gap in GaN-based light-emitting diodes: in perspective. *Critical Reviews in Solid State and Materials Sciences* 46, 5 (2021), 450–467.
- [54] M Vaughan. 2017. *The Fabry-Perot interferometer: history, theory, practice and applications*. Routledge.
- [55] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. 2013. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*. IEEE, 945–948.
- [56] Ashwin Wagadarikar, Renu John, Rebecca Willett, and David Brady. 2008. Single disperser design for coded aperture snapshot spectral imaging. *Applied optics* 47, 10 (2008), B44–B51.
- [57] Edward Jay Wang, William Li, Doug Hawkins, Terry Gernsheimer, Colette Norby-Slycord, and Shwetak N Patel. 2016. HemaApp: noninvasive blood screening of hemoglobin using smartphone cameras. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 593–604.
- [58] Edward Jay Wang, Junyi Zhu, William Li, Rajneil Rana, and Shwetak Patel. 2017. HemaApp IR: noninvasive hemoglobin measurement using unmodified smartphone cameras and built-in LEDs. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 305–308.
- [59] Xingbo Wang, Jean-Baptiste Thomas, Jon Yngve Hardeberg, and Pierre Goutou. 2013. Discrete wavelet transform based multispectral filter array demosaicking. In *2013 Colour and Visual Computing Symposium (CVCS)*. IEEE, 1–6.
- [60] Amanda Watson, Claire Kendell, Anush Lingamoorthy, Insup Lee, and James Weimer. 2023. Lumos: An Open-Source Device for Wearable Spectroscopy Research. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–24.
- [61] Yuzhen Wei, Feiyue Wu, Jie Xu, Junjing Sha, Zhangfeng Zhao, Yong He, and Xiaoli Li. 2019. Visual detection of the moisture content of tea leaves with hyperspectral imaging technology. *Journal of Food Engineering* 248 (2019), 89–96.
- [62] Zhiwei Xiong, Zhan Shi, Huiqun Li, Lizhi Wang, Dong Liu, and Feng Wu. 2017. Hscnn: Cnn-based hyperspectral image recovery from spectrally undersampled projections. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 518–525.
- [63] F. Yasuma, T. Mitsunaga, D. Iso, and S.K. Nayar. 2008. *Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum*. Technical Report.
- [64] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, David Harris-Birtill, and Aaron Quigley. 2017. Specam: Sensing surface color and material with the front-facing camera of a mobile device. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–9.
- [65] Zitong Yu, Xiaobai Li, and Guoying Zhao. 2021. Facial-video-based physiological signal measurement: Recent advances and affective applications. *IEEE Signal Processing Magazine* 38, 6 (2021), 50–58.
- [66] Xin Yuan, David J Brady, and Aggelos K Katsaggelos. 2021. Snapshot compressive imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine* 38, 2 (2021), 65–88.
- [67] Yuanyuan Zhao, Hui Guo, Zhan Ma, Xun Cao, Tao Yue, and Xuemei Hu. 2019. Hyperspectral imaging with random printed mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10149–10157.
- [68] Tiancheng Zhi, Bernardo R Pires, Martial Hebert, and Srinivasa G Narasimhan. 2019. Multispectral imaging for fine-grained recognition of powders on complex backgrounds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8699–8708.

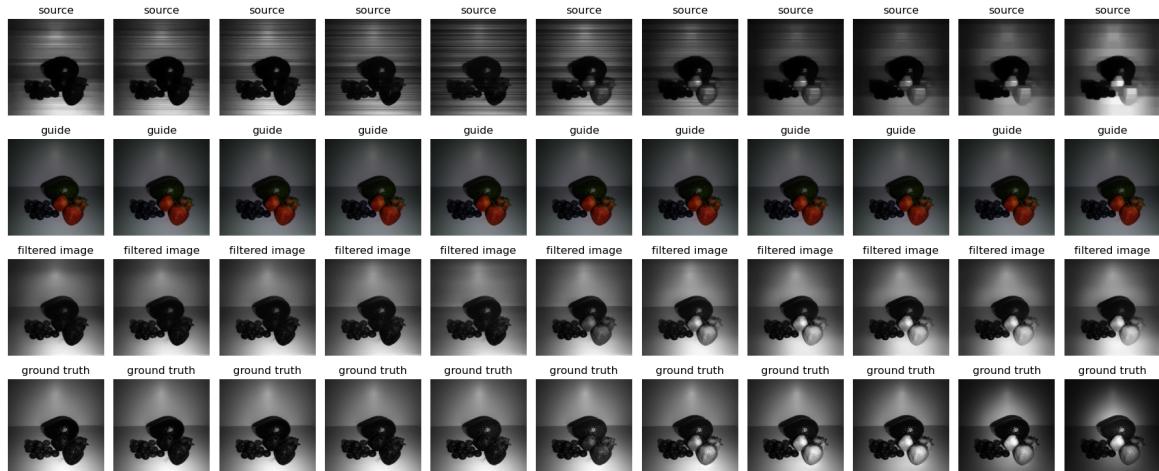
## A RUNTIME ANALYSIS

We implemented the reconstruction algorithm in Python using PyLops [45] for linear optimization and CuPy [42] for GPU-based acceleration. The code was run on a workstation equipped with an Intel Core i9 processor with 64 GB RAM and an NVIDIA GeForce RTX 3060 GPU with 12 GB of memory. The table below provides a detailed breakdown of the amount of time required to process a  $752 \times 752$  image. These times were benchmarked in Python using the `time` module across 10 runs with different images.

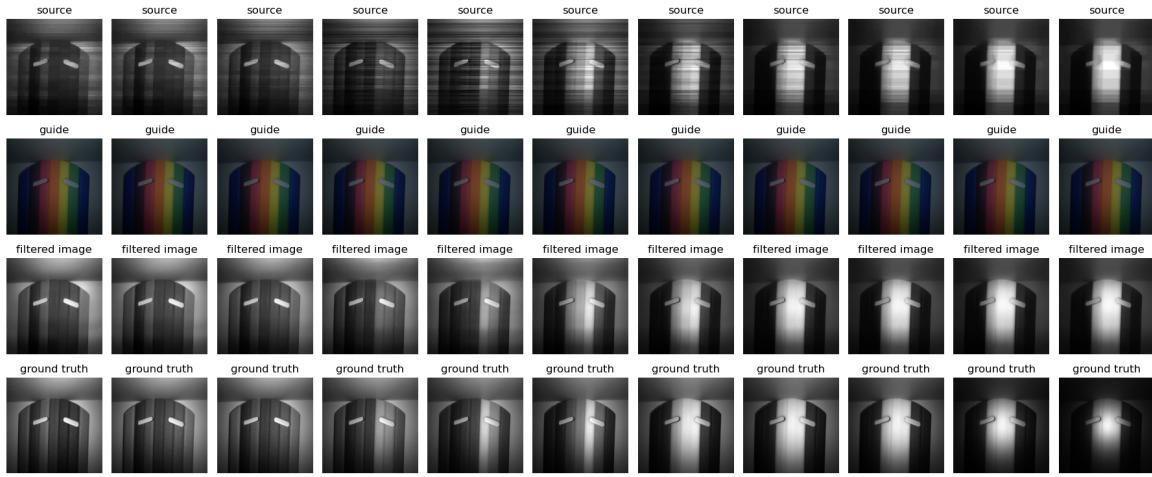
Component	Runtime (seconds)
Black level adjustment	$3.7 \times 10^{-2}$
RGB demosaicing	2.18
Normalization	$3.4 \times 10^{-4}$
Ambient light correction	$8.22 \times 10^{-5}$
Temporal alignment of rolling shutter and multispectral flash	3.06
ADMM	(3.66 per iteration) $\times$ 25 iterations = 91.39
Guided image filtering	( $3.5 \times 10^{-3}$ per channel) $\times$ 31 channels = 0.11
Total	96.77

## B RECONSTRUCTION ACROSS ALL SPECTRAL CHANNELS

The following figures provide additional examples of ChromaFlash’s reconstruction across all 11 spectral channels.



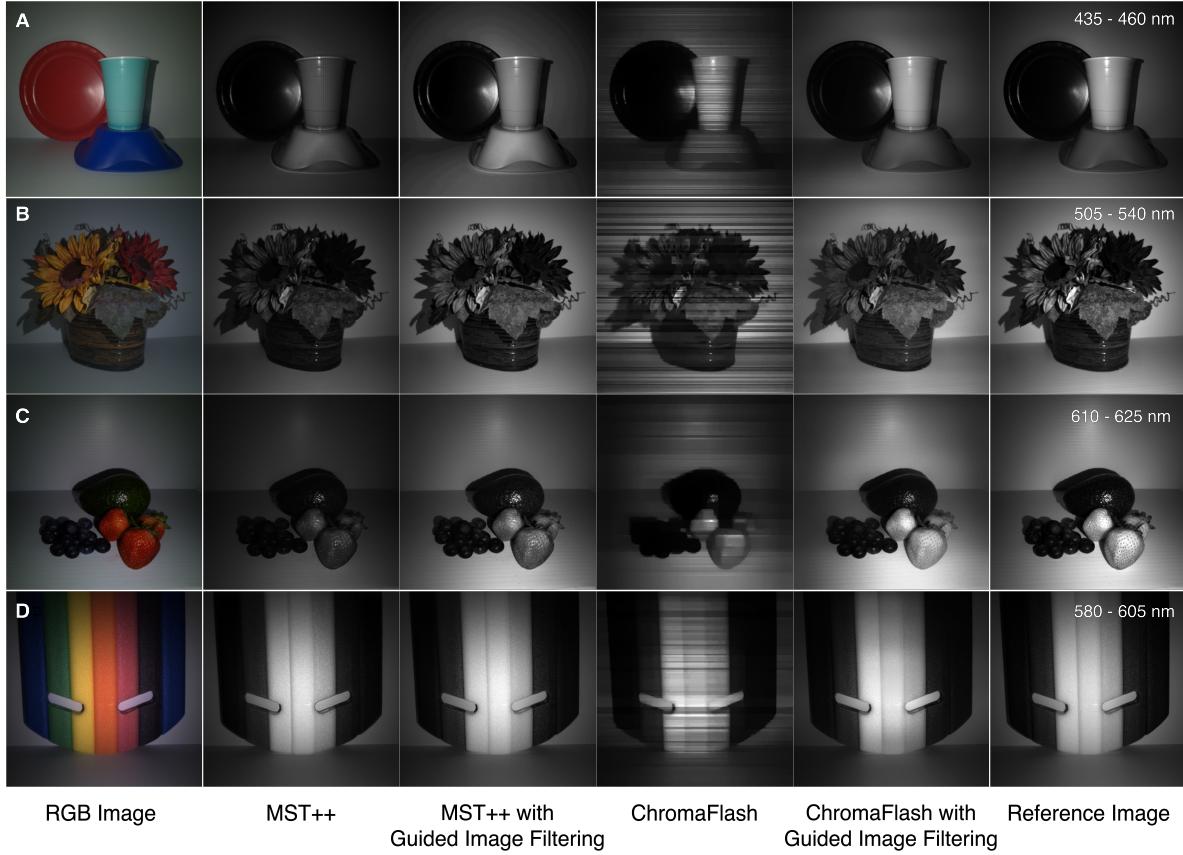
**Fig. 14. Reconstructions from real-world data for the fruits scene:** From left to right, each column represents a distinct spectral band within 400 nm to 700 nm. From top to bottom, rows correspond to the reconstruction output of ChromaFlash’s snapshot configuration, the guide image for post-processing, the reconstruction outputs after guided image filtering, and the ground-truth reference image for that given wavelength.



**Fig. 15. Reconstructions from real-world data on the pool toy scene:** From left to right, each column represents a distinct spectral band within 400 nm to 700 nm. From top to bottom, rows correspond to the reconstruction output of ChromaFlash’s snapshot configuration, the guide image for post-processing, the reconstruction outputs after guided image filtering, and the ground-truth reference image for that given wavelength.

### C COMPARISON WITH MST++ ON ADDITIONAL SCENES

The figure below visually compares the outputs of MST++ [13] and the snapshot configuration of ChromaFlash. While the reconstructed images from MST++ appear more spatially coherent than those from ChromaFlash without guided image filtering, we observe that they are comparable after this post-processing step. Furthermore, MST++ severely underperformed in spectral accuracy. For example, the strawberries in the fruits scene (row C) should be bright in the reconstructed images because they illustrate the amount of spectral information between 610 and 625 nm in grayscale. The strawberries are far brighter in the ChromaFlash images than they are in the MST++ images, showcasing ChromaFlash’s superior spectral accuracy.



**Fig. 16. Reconstructions from real-world data:** This is an extension to Figure 12, comparing ChromaFlash with MST++ [13] in reconstructing real-world scenes. From left to right, the columns show an RGB image of the scene captured under white light, the reconstructed image from MST++ at a specific wavelength, the reconstructed image from MST++ following guided image filtering, the reconstructed image from the snapshot configuration of ChromaFlash, the reconstructed image from ChromaFlash after guided image filtering, and the ground-truth reference image for that wavelength.