CGI2013 manuscript No.
(will be inserted by the editor)

Interactive Chromaticity Mapping for Multispectral Images

Yanxiang Lan · Jiaping Wang · Stephen Lin · Minmin Gong · Xin Tong · Baining Guo

Abstract Multispectral images record detailed color spectra at each image pixel. To display a multispectral image on conventional output devices, a chromaticity mapping function is needed to map the spectral vector of each pixel to the displayable three dimensional color space. In this paper, we present an interactive method for locally adjusting the chromaticity mapping of a multispectral image. The user specifies edits to the chromaticity mapping via a sparse set of strokes at selected image locations and wavelengths, then our method automatically propagates the edits to the rest of the multispectral image. The key idea of our approach is to factorize the multispectral image into a component that indicates spatial coherence between different pixels, and one that describes spectral coherence between different wavelengths. Based on this factorized representation, a two-step algorithm is developed to efficiently propagate the edits in the spatial and spectral domains separately. The presented method provides photographers with efficient control over color appearance and scene details in a manner not possible with conventional color image editing. We demonstrate the use of interactive chromaticity mapping in the applications of color stylization to emulate the appearance of photographic films, enhancement of image details, and manipulation of different light transport effects.

Yanxiang Lan
Tsinghua University, Microsoft Research Asia
E-mail: layx03@gmail.com

Jiaping Wang
Microsoft Corporation
E-mail: jiapw@microsoft.com

Stephen Lin, Minmin Gong, Xin Tong
Microsoft Research Asia
E-mail: {stevelin,mgong,xtong}@microsoft.com

Baining Guo
Microsoft Research Asia, Tsinghua University
E-mail: bainguo@microsoft.com

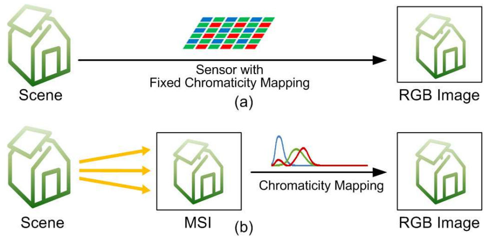

Fig. 1 Chromaticity mapping. (a) Conventional color imaging with fixed chromaticity mapping determined by color filters. (b) Multispectral imaging with chromaticity mapping that can be specified post-capture by the user.

1 Introduction

In this age of digital imagery, photographers seek ever more powerful tools for manipulating scene appearance attributes such as color, contrast and depth of field within a captured photo. The ability to perform such edits, however, is limited by the scene information that can be extracted from the image. For the case of scene radiance spectra, a conventional digital camera uses three color filters, each with a fixed spectral sensitivity function, to project the original scene spectra simply to 3D vectors of red, green and blue color components. This low-dimensional model of scene spectra generally lacks much of the original color content and associated surface details. To address this issue, multispectral imaging methods have been developed for recording dense samples of the color spectrum at image pixels. The rich spectral content in a multispectral image (MSI) enables high accuracy in color reproduction and stylization, the ability to reveal various surface details, and greater discrimination of pixel colors in comparison to conventional RGB images (as shown in Figure 2(a)(b)).

To display a multispectral image on conventional output devices, the scene spectrum of each pixel needs to be projected to RGB values. This projection, which we refer to as *chromaticity mapping*, is physically applied in a conventional digital camera during the imaging process by its fixed spectral sensitivity functions, as illustrated in Figure 1(a). With a multispectral image, the chromaticity mapping of each pixel can be specified and fine-tuned post-capture according to desired color characteristics of the final image, as shown in Figure 1(b). For full control over color reproduction of the scene or to flexibly tailor images for artistic purposes, a user needs to be able to specify different chromaticity mapping functions for different pixels. However, manipulating the chromaticity mapping for multispectral images is a nontrivial task. On one hand, the editing should allow the user to freely and efficiently manipulate the chromaticity mapping in both spatial and spectral domains. On the other hand, the intrinsic coherence of data samples in both domains should be maintained during editing (i.e., if the spectra of two pixels or the image intensities in two wavelengths are highly correlated, then these correlations should be maintained).

A naive solution is to apply traditional RGB image or appearance editing methods [2,22,30] for editing of chromaticity mapping, based on L2 distances of spectral vectors at pixels to propagate user edits at sparse pixels over the whole image. Unfortunately, none of these methods support editing in the spectral domain. Another possible solution is to regard the spectrum samples at different wavelengths as a stack of monochrome images and then apply video editing methods [15, 29] for this task. However, different from video data where the appearance distance of pixels at different time instances can be defined by color differences, the L2 distance of samples at different wavelengths cannot preserve coherence in the spectral domain and result in edit artifacts (shown in Figure 2(c)).

In this paper, we present an interactive method for manipulating the chromaticity mapping of multispectral images. The key idea of our approach is to represent the image at each wavelength, which we refer to as an *image slice*, as a linear combination of image slices at a sparse set of representative wavelengths. The image slices of representative wavelengths exhibit the different spatial details and thus record coherence between pixels, while linear combination weights for reconstructing other image slices from these representative slices reveal the spectral coherence between image slices of different wavelength. With this decomposition of an MSI into a spatial coherence component and a spectral coherence component, our method allows users to edit the chromaticity mapping with sparse user inputs and then propagates the edit throughout the MSI in the two domains separately. To this end, the users specify sparse input strokes on image slices at different representative wavelengths to indicate spatial and spectral regions that they want to emphasize or de-emphasize in the result. Then in a two-step propagation scheme, the edits are first propagated spatially, then spectrally, to efficiently obtain the spatially varying chromaticity mapping. Figure 2(d) illustrates the resulting RGB image of an MSI constructed with the chromaticity mapping function edited by our method.

The main technical contributions of our method are the MSI factorization method and the two-step edit propagation scheme for manipulating the chromaticity mapping. By decomposing the MSI into a spatial coherence component and a spectral coherence component, the factorization method not only preserves the spatial and spectral coherence of the input MSI in chromaticity mapping edit, but also provides an intuitive way for users to select different spatial and spectral features on the sparse set of representative image slices and edit their chromaticity mapping. Based on the MSI factorization, the two-step propagation scheme propagates the user edit in the two components separately. This not only reduces the computational cost in edit propagation, but also allows us to apply traditional image edit propagation methods in each step.

We demonstrate various manipulations of color and image details that are possible with interactive feedback by using this technique. The application of color stylization is shown, in which an MSI is transformed into an RGB image with the color characteristics of a desired photographic imaging process. We also apply our technique to enhance various image components, including light transport, scattering effects, and surface details. In comparison to conventional software tools for picture editing, our method provides users with much greater versatility in manipulating and enhancing photo appearance.

2 Related works

Multispectral Imaging In recent years, multispectral images have been the focus of increasing research. Several works have been presented on MSI acquisition [7, 8, 11, 17, 18, 20, 23, 24, 31], analysis [5] and applications in various tasks. In applications, the additional color information in an MSI has led to improvements in image relighting [20], segmentation [8] and tracking [3]. To display an MSI on a conventional display device, the spectrum of each pixel needs to be mapped to RGB values. The rich spectral data in an MSI allows great flexibility in producing an RGB image by defining and fine-tuning the chromaticity mapping functions. However, to our knowledge, previous works have not addressed this issue. In this paper, we provide the user an intuitive tool for chromaticity mapping of MSIs to generate RGB images that are more artistic or informative than images obtained with a conventional RGB camera.

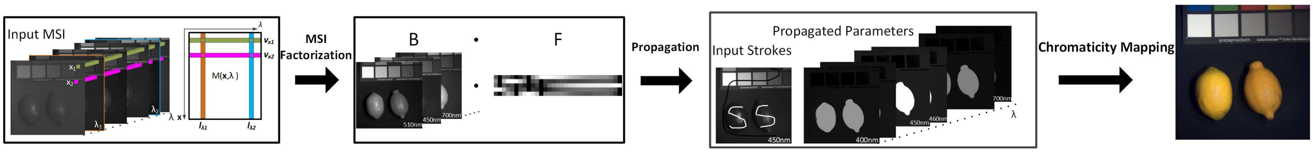

Fig. 3 Pipeline of chromaticity mapping edits. After loading the MSI M, it is factorized into representative slices **B** and a spectral weight matrix **F**. Then the user constraints specified on the representative slices are propagated to the entire MSI. Finally, a spatially varying chromaticity mapping function is generated to produce the final RGB image.

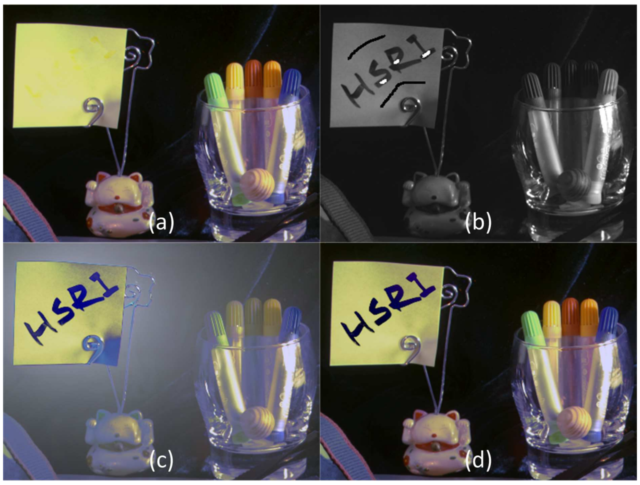

Fig. 2 Chromaticity mapping edits for multispectral images. (a) RGB image of an MSI generated by the default chromaticity mapping. (b) Multispectral image values at 436nm wavelength. Note that the text on the paper is invisible in the RGB image but clearly exhibited in this image. The black and white strokes are user input for editing the chromaticity mapping function. (c) The resulting RGB image of the MSI with the chromaticity mapping function edited by the method in [15]. (d) The resulting RGB image of the MSI constructed with the chromaticity mapping function edited by our method. The same set of user-specified strokes shown in (b) are used in both methods for edit propagation.

Color Reproduction and Simulation The color characteristics of an image may be altered to match a desired spectral sensitivity. Photoshop plug-ins such as Exposure4 allow users to give digital photographs the color appearance of a particular film [1]. In color reproduction, colors are modulated in images such that they perceptually match those of the original scene within the human visual system [26]. When these color transformations are applied to RGB images, accuracy is greatly limited by the low spectral resolution and loss of color content. In this paper, we address the color simulation problem as matching of color spectra in image irradiance, rather than matching of human retinal responses, since there exists wide variability in color perception among people. For example, the proportions of the three types of color-sensing cones on the human retina has been found to vary significantly among different individuals [4].

Image Enhancement Scene spectra often contains color features corresponding to rich surface details. In the field of image processing, there exists considerable work on image enhancement that aims to reveal such details in captured images. A survey of recent methods can be found in [10]. While these techniques can emphasize the appearance of surface details that exist in an image, details that do not register at the sensor cannot be enhanced in this way. In the imaging device of [17], specific wavelengths may be selected to recover particular surface details. These wavelength settings must be chosen at the time of image capture, and cannot be adjusted afterwards in an image editing process.

Edit Propagation Various edit propagation techniques have been proposed to efficiently propagate sparse user edits to other image regions of similar appearance in image colorization [14], tone mapping [16] and material editing [22, 30]. Farbman et al. [9] presented a new distance metric for computing the affinity between two image pixels. An et al. [2] accelerated the distance computation from sparsely sampled pixel pairs with the Nystrom method. All of these techniques propagate the edit spatially. It is not clear how to apply these methods to edit the chromaticity mapping in both the spatial and spectral domain. Recently, Xu et al. [29] used a K-D tree of appearance values to accelerate the affinity computation and edit propagation, and applied this technique for video editing. Li et al. [15] further accelerated the edit propagation with an interpolation scheme. However, these methods cannot be directly applied for editing chromaticity mapping of multispectral images because the sample values at different wavelengths cannot be directly compared. By contrast, our technique factorizes the MSI image and defines the affinity of sample values in the spectral as well as spatial domains, allowing for edit propagation in both domains. With our two-step propagation scheme, existing spatial edit propagation methods can be easily applied in each step.

3 MSI Chromaticity Mapping Tool

A multispectral image is a 3D function $\mathbf{M}(\boldsymbol{x},\lambda)$ of image position $\boldsymbol{x}$ and spectral wavelength λ. We consider in this work the set of visible and near-infrared wavelengths. For an image resolution of $w \times h$ and a spectral resolution of n, a multispectral image is represented by a matrix $\mathbf{M}$ of n columns and $w \times h$ rows. Each column $l(\lambda)$ of $\mathbf{M}$ corresponds to an *image slice* that contains the light intensities of each image point at a certain wavelength. We refer to each row $v(\boldsymbol{x})$ of

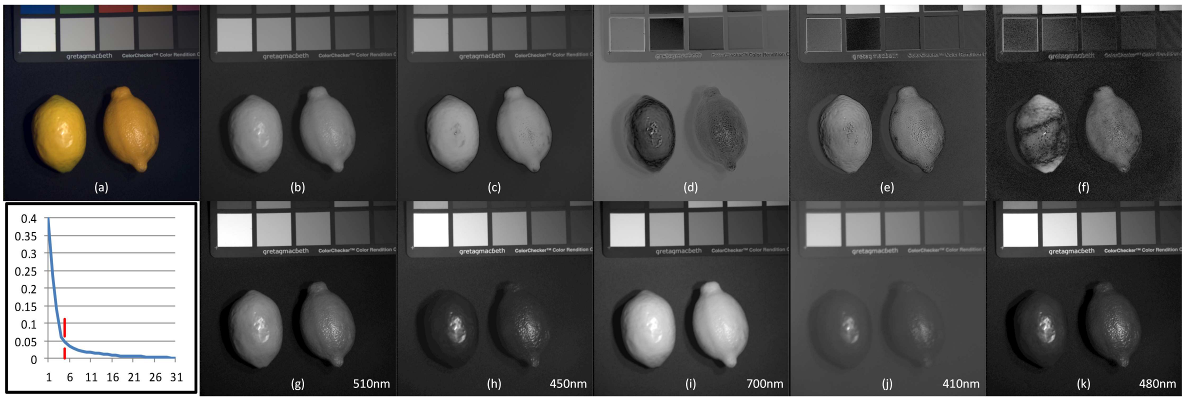

Fig. 4 Representative slices of a lemon MSI generated by PCA and selected by our algorithm. (a) Conventional RGB image. (b)-(f) First five PCA basis components of the lemon MSI. (g)-(k) Computed representative slices, showing different appearance features such as brightness and roughness. Left-bottom: Reconstruction errors of the factorized representation with different numbers of representative slices K. (Data from [32].)

$\mathbf{M}$ as a *spectrum vector*, which is composed of dense samples of the spectrum at a given pixel.

To display a multispectral image on conventional output devices, the scene spectrum of each pixel needs to be projected to RGB values by chromaticity mapping $\mathbf{s}_i(\boldsymbol{x},\lambda)$:

$$I_i(\boldsymbol{x}) = \int \mathbf{M}(\boldsymbol{x},\lambda)\mathbf{s}_i(\boldsymbol{x},\lambda)d\lambda, \tag{1}$$

where $i = R,G,B$ is one of three RGB channels, and $\mathbf{s}_i(\boldsymbol{x},\lambda)$ are the spatially varying chromaticity mapping functions that are manipulated post-capture according to the user's intentions. For simplicity, in the following text we only discuss how to edit the chromaticity mapping function $\mathbf{s}(\boldsymbol{x},\lambda)$ for one display channel. The same process can be applied to edit the chromaticity mapping functions of the other RGB channels.

The purpose of our work is to allow the user to easily adjust the chromaticity mapping function $\mathbf{s}(\boldsymbol{x},\lambda)$ in a spatially and spectrally varying manner. As illustrated in Figure 3, we first factorize the input MSI M into a set of representative image slices $\mathbf{B}$ and a set of weights for each wavelength $\mathbf{F}$ (Section 3.1). After that, the user can edit the chromaticity mapping on the representative slices $l(\lambda_b)$ of B by drawing strokes with specified constraint values $g(\boldsymbol{x}_b,\lambda_b)$, which can be positive or negative to emphasize or de-emphasize the edit region. The edit is then propagated throughout the MSI using a two-step propagation technique (Section 3.2). In particular, the propagation is executed first in the spatial domain $\boldsymbol{x}$ based on the appearance affinity $Z^v(B(\boldsymbol{x}_1),B(\boldsymbol{x}_2))$ of two pixels defined in $\mathbf{B}$, and then in the spectral domain λ based on the appearance affinity $Z^s(F(\lambda_1),F(\lambda_2))$ of two wavelengths defined in $\mathbf{F}$. Finally, we obtain the resulting chromaticity mapping $\mathbf{s}'(\boldsymbol{x},\lambda)$ from the propagated edits:

$$\mathbf{s}'(\boldsymbol{x},\lambda) = \mathbf{s}(\boldsymbol{x},\lambda) + \textstyle\sum_{j\in E} e_j(Z^v(B(\boldsymbol{x}),B(\boldsymbol{x}_j)),Z^s(F(\lambda),F(\lambda_j)),g(\boldsymbol{x}_j,\lambda_j)) \tag{2}$$

where $\mathbf{s}(\boldsymbol{x},\lambda)$ is the original mapping function, E is the set of user strokes, $e_j(Z^v(B(\boldsymbol{x}),B(\boldsymbol{x}_j)),Z^s(F(\lambda),F(\lambda_j)),g(\boldsymbol{x}_j,\lambda_j))$ is the edit of the chromaticity mapping function on MSI sample $M(\boldsymbol{x},\lambda)$ propagated from the user input $g(\boldsymbol{x}_j,\lambda_j)$ at the j-th stroke (Section 3.3).

3.1 MSI Factorization

An MSI contains a large number of samples in both the spatial and spectral domains. Recently, several techniques have taken advantage of coherence within an MSI and proposed sparse representations to reduce the data that needs to be captured in MSI acquisition [12,20,25]. However, these representations are targeted to acquisition and are nonintuitive for user-guided editing of chromaticity mapping. Here we propose a new MSI factorization for this purpose.

Let us consider the MSI as a set of image slices with different wavelengths $\mathbf{M} = [l_1,l_2...l_n]$. With $\mathbf{M}$, we identify the least number of image slices with distinct features that can well represent all the other image slices. We refer to these image slices as *representative slices* of $\mathbf{M}$. Specifically, a set of K slices is automatically selected by minimizing the normalized reconstruction errors associated with the selected slices and the number of slices K:

$$\arg\min_K \sum_i \frac{||[l_{t_1}\dots l_{t_K}]\mathbf{w}_i - l_i||^2}{||l_i||^2}. \tag{3}$$

where l_i is the **i**-th column of $\mathbf{M}(\boldsymbol{x},\lambda)$, and $\mathbf{w}_i$ is a $K\times 1$ vector representing the reconstruction weights of slice l_i from representative slices $[l_{t_1}\dots l_{t_K}]$. This can be solved by a greedy solution. In particular, we start by initializing the bases number K to zero and then incrementally increase it one by one in a greedy way. In each step, we test all non-representative slices and add the one that can reduce the reconstruction error most in the representative slice set. We repeat this process until the relative reconstruction error falls below a threshold of $\varepsilon = 0.05$.

With the selected representative slices, we can decompose the MSI matrix $M = [l_1, l_2 \ldots l_n]$ as the product of the matrix of representative slices $\mathbf{B} = [l_{t_1} \ldots l_{t_K}]$ and the matrix of weights $\mathbf{F} = [\boldsymbol{w}_1, \boldsymbol{w}_2 \ldots \boldsymbol{w}_n]$:

$$\mathbf{M}_{p\times n} = \mathbf{B}_{p\times K} \bullet \mathbf{F}_{K\times n}. \tag{4}$$

Figure 4 (g)-(k) displays an example of representative image slices, which show different appearance features such as roughness and brightness. For comparison, we also display in Figure 4 (b)-(f) the first five basis slices determined by PCA [13]. As shown in this example, the representative slices generated by our factorization well exhibit the spatial and spectral features of the original MSI, while PCA does not guarantee a decomposition into components that are meaningful for editing.

From the MSI factorization, the representative slices in **B** contain the distinct spatial features of the MSI and are useful for both specifying the editing constraints and providing an indication of how the edits will be propagated.Spatial features in other slices are taken to be linearly correlated to those in **B**. The constraints within one slice are propagated to the whole image slice in the spatial domain. On the other hand, **F** encodes the spectral coherence between representative image slices in **B** and image slices at other wavelengths in **M**, and we use it for propagating the edits in one wavelength band to other wavelengths according to their coherence in the spectral domain. The final propagation result is computed as a combination of propagations in the spatial and spectral domains.

3.2 Two-Step Edit Propagation

User Inputs After factorization, the user selects the spatial and spectral features they want to manipulate by specifying a set of strokes $g(\boldsymbol{x}_b, l_{\lambda_b})$ at sparse pixels $\boldsymbol{x}_b$ on a representative slice l_{λ_b} of **B** and then applies the desired chromaticity mapping edit to the selected region. As shown in Figure 5(b), the white strokes refer to the pixels the user wants to apply the edit while the black strokes refer to the region that should be unchanged.

After this, we propagate the edit in the spatial domain and then propagate the edit in the spectral domain to obtain the final edit weight map for chromaticity mapping.

Spatial Propagation In the spatial propagation step, we propagate the sparse constraints $g_{\boldsymbol{x}_b,\lambda_b}$ defined at representative slices l_{λ_b} to all pixels in the same representative slice. To this end, we follow the method in [15] and formulate the spatial propagation process as a function interpolation that determines the propagated values:

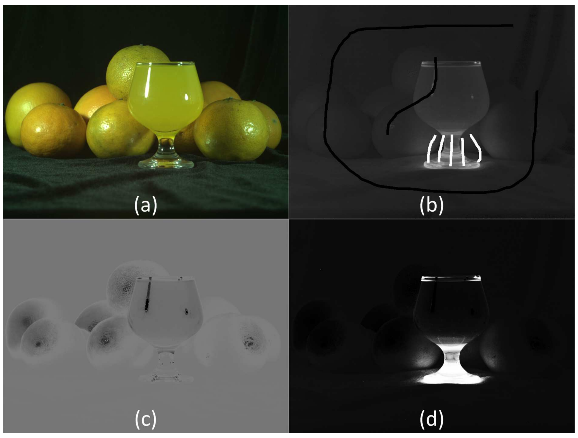

Fig. 5 Comparison of results with and without normalized spectrum distance in the spatial domain. The selected feature (with brush strokes) in the image slice at wavelength 880nm in (b) is subtle due to its low intensity in the original MSI. The propagation result computed with normalized spectrum distance in (d) is much better than the one computed with un-normalized spectrum distance in (c).

$$e^v(\boldsymbol{x}, \lambda_b) = \sum_{\boldsymbol{y}\in C} a_{\boldsymbol{y}} Z^v(B(\boldsymbol{x}), B(\boldsymbol{y})) \tag{5}$$

where C is a selected subset of the user-edited pixels on representative image slice $l(\lambda_b)$. $Z^v(B(\boldsymbol{x}), B(\boldsymbol{y}))$ is the appearance affinity in the spatial domain. $a_{\boldsymbol{y}}$ is determined through the following optimization [15]:

$$\arg\min_{a_y} \sum_{\boldsymbol{x}_b\in E} w_{\boldsymbol{x}_b,\lambda_b} (g_{\boldsymbol{x}_b,\lambda_b} - \sum_{\boldsymbol{y}\in C} (a_{\boldsymbol{y}} Z^v(B(\boldsymbol{x}_b), B(\boldsymbol{y}))))^2 \tag{6}$$

where E is the set of all user-edited pixels $\boldsymbol{x}_b$ on representative slice l_{λ_b}, and $w_{\boldsymbol{x}_b,\lambda_b}$ is the constraint strength on pixel $\boldsymbol{x}$ at image slice $l(\lambda_b)$ specified by the user. We set $w_{\boldsymbol{x}_b,\lambda_b} = 1.0$ in our implementation.

Given the matrix B of representative slices, the appearance affinity in the spatial domain $Z^v(B(\boldsymbol{x}), B(\boldsymbol{y}))$ is defined as

$$Z^v(B(\boldsymbol{x}), B(\boldsymbol{y})) = \exp\left(-\frac{\sum_\lambda (B'(\boldsymbol{x},\lambda) - B'(\boldsymbol{y},\lambda))^2}{\sigma_v^2} - \frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma_p^2}\right), \tag{7}$$

where $\sum_\lambda (B'(\boldsymbol{x},\lambda) - B'(\boldsymbol{y},\lambda))^2$ and $\|\boldsymbol{x}-\boldsymbol{y}\|^2$ are the appearance and spatial distances of two pixels respectively, while σ_v and σ_p control the relative weights of appearance distance and spatial distance of two pixels. $B'(\boldsymbol{x},\lambda) = \frac{B(\boldsymbol{x},\lambda)}{\sum_x B(\boldsymbol{x},\lambda)}$ is the normalized representative slice value of B. Note that if we directly use the representative slice values B in the distance computation, the differences within a low-intensity representative slice may be washed-out by the differences in a high-intensity representative image slice, which leads to an undesirable appearance affinity map as shown in Figure 5(c). By using the distance computed by the normalized

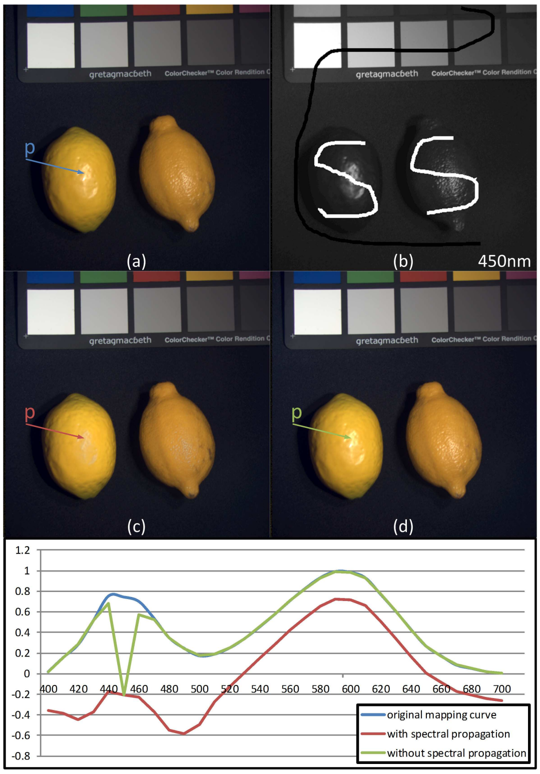

Fig. 6 Results of chromaticity mapping tuning with/without the spectral propagation. (a) color image generated using the conventional chromaticity mapping function; (b) constraints on the 450nm slice; (c) result with spectral propagation; (d) result without the spectral propagation. Bottom: the luminance curves of point P in (a)(c)(d) respectively. (Data from [32].)

representative slice value B', our method yields a reasonable appearance affinity map in the spatial domain (shown in Figure 5(d)).

Spectral Propagation The spatial propagation in the previous step propagates the user edit to all pixels within the given representative slices. In the spectral propagation step, we further propagate the edit to all wavelengths. Given the spatial propagation result $e^v(\mathbf{x},\lambda_b)$ defined at representative wavelength λ_b, we simply compute the propagated edit value $e^s(\mathbf{x},\lambda)$ at each wavelength λ as

$$e^s(\mathbf{x},\lambda) = \sum_{\lambda_b \in G} e^v(\mathbf{x},\lambda_b) Z^s(F(\lambda),F(\lambda_b)), \quad (8)$$

where G is the set of representative image slices with user strokes, and $Z^s(F(\lambda),F(\lambda_b))$ is the affinity function defined in the spectral domain, which is defined by

$$Z^s(F(\lambda),F(\lambda_b)) = \exp(-\frac{\sum_{k=0}^{K}(F(k,\lambda)-F(k,\lambda_b))^2}{\sigma_s^2}), \quad (9)$$

where σ_s is a parameter that controls the amount of propagation in the spectral domain.

Since the image feature of interest to the user may exist in a set of wavelengths, propagating the chromaticity adjustments to wavelengths with similar image slice appearance (i.e. the coherent wavelengths) would help to achieve their intentions. This is illustrated in Figure 6, where we would like to reduce the specular reflection on the lemon skin in (a) to make it more diffuse. To do this, the color image is pre-synthesized in HSI color space [28] with H and S fixed and the mapping curve applied to I. We draw strokes on the 450nm slice in (b) which exhibits strong specularity to reduce its contribution to the final color image. Without spectral propagation, the adjustment only affects wavelength 450nm and not other image slices that share this strong specular effect, which would lead to the result in (d) that still exhibits much specular reflection. With the spectral propagation, the adjustment is spread to wavelengths that similarly contain strong specularity, which allows easy achievement of our goal in (c).

3.3 Chromaticity Mapping

After the two-step propagation, we obtain the final edit $e^s(\mathbf{x},\lambda)$ for each spectrum sample. The final spatially-varying chromaticity mapping function is then computed by

$$s'(\mathbf{x},\lambda) = s(\mathbf{x},\lambda) + e^s(\mathbf{x},\lambda). \quad (10)$$

Finally, we apply the chromaticity mapping function $s'(\mathbf{x},\lambda)$ to the spectral values of the pixels $\mathbf{x}$ to compute the RGB color image via Equation 1.

Computational Complexity Our two-step propagation technique brings a significant reduction in computational cost. Let us consider an MSI $\mathbf{M}$ with $p = w \times h$ pixels and n wavelength bands. If we directly propagate from the constraint samples to the entire MSI with the affinity defined by

$$Z(M(\mathbf{x}_1,\lambda),M(\mathbf{x}_2,\gamma)) = Z^v(B(\mathbf{x}_1),B(\mathbf{x}_2))Z^s(F(\lambda),F(\gamma)), \quad (11)$$

the computational cost of this naive solution is $O(w \times h \times n)$. On the contrary, our method decomposes the propagation into two steps, which reduce the computational cost to $O(w \times h + n)$.

4 Experimental Results

We implemented our system on a PC with an Intel Xeon CPU E5440@2.83GHz with 4GB RAM. After loading an MSI, the pre-processing for MSI decomposition takes about one minute. Then our system provides interactive feedback,

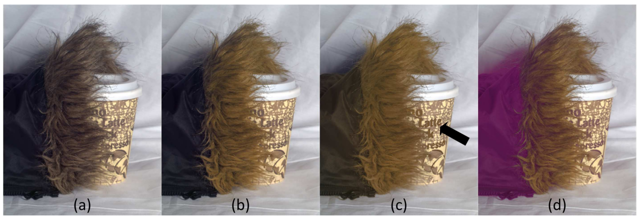

Fig. 7 Comparisons of different edit propagation methods for chromaticity mapping. (a) RGB image before editing. (b) Result computed with the chromaticity mapping generated by our method. (c) Result by directly editing the RGB image. (d) Result computed with the chromaticity mapping generated by the video editing method in [15].

with a response time for propagation of about 2 seconds. The MSIs used in our experiments have an image resolution of 800×600 and a spectral resolution of 50 slices from $400nm \sim 1000nm$. Some results in Figure 9 have been tone-mapped and cropped for display. For all results shown in the paper, the user inputs consist of fewer than six strokes. Please refer to the accompanying video for the input strokes of each result.

Figure 2 compares the images generated by our method and results generated by the video editing method in [15] that is based on the simple L2 distance of spectrum sample values. Figure 7 compares the results generated by our method, the RGB based image edit method, and the video editing method in [15]. Note that the RGB based image editing methods cannot distinguish pixels with similar RGB colors but different spectral details, and thus generate undesirable editing results. Also, the simple L2 distance metric used in the video based method cannot measure the affinity of spectral samples at different wavelengths and thus generates artifacts in the result. By contrast, our method well preserves the spatial and spectral coherence of the input multispectral image.

4.1 Color Stylization

Different photographic imaging processes vary broadly in their mechanisms for producing color, which results in unique color styles that emphasize or suppress certain ranges of wavelengths. There exist some methods and tools [1] for color stylization, but they operate on the limited RGB input recorded on the camera CCD. With an MSI, we can manually tune or use the spectral sensitivity functions of a specific imaging process to generate a photo with a specific color style.

Similar to [6], we model a film-based imaging process by considering the film, development technique and digital film scanning as a whole. For digital cameras, the process similarly covers the full imaging pipeline, from the incoming scene radiance to the output image, while possibly being influenced by camera settings. Multiple images with different exposures are captured to calibrate the response function of the imaging process [6], and spectral sensitivity is assumed not to be affected by exposure time. With this calibration, color processing is modeled simply as an inner product of each spectral sensitivity function s of the overall system and the scene radiance spectrum $v_{\mathbf{x}}(\lambda)$ to give pixel values $p_{\mathbf{x}} = \int s(\lambda) v_{\mathbf{x}}(\lambda) d\lambda$.

We thus recover the spectral sensitivity functions of a given camera by imaging a dispersed spectrum with known image irradiance values, produced by passing a light beam with a known spectral power distribution through an optically calibrated prism onto a white matte screen with known reflectance. For each color channel of the system, its spectral sensitivity function is calculated by dividing the digitized values of the spectrum by their corresponding incoming radiance values.

Figure 8(b) shows color stylization results based on spectral sensitivities measured from a Minolta Dynax 500si film camera with Kodak Ektachrome 100VS film and standard E-6 development. Our stylization result in Figure 8(b) from a multispectral image well simulates the cold and vivid color style of this film, and closely matches the scanned film shown in Figure 8(a). The result shown in Figure 8(d) is generated from a conventional RGB image using an MSI with CIE 1931 RGB color matching functions by film emulation software (Exposure 4 [1]), which differs noticeably from the ground truth due to the limited color information in a conventional color image. We also show a local color stylization result in Figure 8(c). In this local stylization, the spectral sensitivities for the lens and pens is from Kodak Gold 200 film with a Minolta Dynax 500si film camera and C-41 development while those of other parts of the image are the same as in Figure 8(b). Note the difference in hue for the lens and pens in Figure 8(c) compared to that in Figure 8(b).

4.2 Detail Preservation

Another significant component of photo appearance is surface detail, which is preserved in the multispectral image data. Details at particular wavelengths or wavelength bands can be emphasized by manipulating chromaticity mapping functions. From a multispectral image, we synthesize a color image in the HSI color space [28] with H and S fixed according to a conversion from standard RGB colors [CIE 1986], and with the chromaticity mapping function of I determined for detail preservation.

We demonstrate surface detail preservation results in Figure 9 by tuning the chromaticity mapping function of I using our

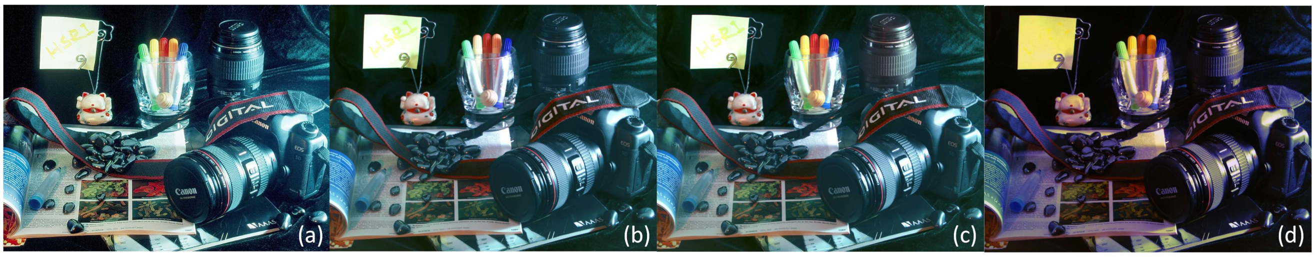

Fig. 8 Color stylization results. (a) Ground truth (scanned film); (b) Our stylized result with an MSI; (c) Our local color style editing results. Note the color style differences on the lens and pens compared to (b) and other parts of (c). (d) Software simulation with Alien Skin Exposure 4 using the conventional color image generated from an MSI with CIE 1931 RGB color matching functions.

tools. The strokes used for tuning are shown in the accompanying video. With different tunings, we obtain images that enhance different aspects of the scene, including light transport, scattering effects, and surface details.

Light Transport Enhancement The scene radiance that forms an image is the result of complex light transport among the scene objects. Extended from [19,21,27], multispectral light transport could be formulated as:

$$\mathbf{b} = \mathbf{T} \cdot \mathbf{i} \tag{12}$$

where $\mathbf{T}$ is an $(m \cdot t) \times (n \cdot t)$ light transport matrix from n light sources to m camera pixels, $\mathbf{i}$ is the illumination condition represented by an $n \times t$ vector from n light sources, and $\mathbf{b}$ is an $m \times t$ vector that represents the scene radiance observed by a camera with m pixels. $\mathbf{T}$, $\mathbf{i}$ and $\mathbf{b}$ all have a multispectral representation with t wavelength bands.

The multispectral light transport matrix $\mathbf{T}$ has many details that cannot be captured by conventional RGB imaging. With an MSI, our method can reveal and enhance subtle details in the light transport matrix to generate new effects. For example, Figure 9(b,h) exhibits refraction caustics with significantly more detail than Figure 9(a,g) recorded by a conventional RGB image.

Scattering Effect Enhancement The appearance of an object arises from the wavelength-dependent interaction of light with the object surface. An object surface may thus have different scattering effects and appearance under different wavelengths. In Figure 9(f), the wavelength dependent subsurface scattering is enhanced, making the hand look soft and translucent with most of the wrinkles faded out. By contrast, Figure 9(e) shows the hand with surface scattering emphasized, making the hand appear hard and aged with evident wrinkles. Similar effects are seen in the pomelo of Figure 9(l), which exhibits smoother and softer skin than in the original RGB image of Figure 9(j). Also, the red silk ribbon on the box in Figure 9(i) appears more translucent than in Figure 9(g).

Surface Detail Enhancement Many natural objects exhibit rich details on their surface within narrow wavelength bands, but which are lost by the broad pass-bands of RGB filters in conventional cameras. With our tool, these details can be revealed and enhanced by tuning the chromaticity mapping functions. Figure 9(c) displays the green pattern on the orange skin which is lost by RGB filters in Figure 9(a). The details inside a yellow watermelon are enhanced in Figure 9(k) in comparison to Figure 9(j). Also, greater detail is visible in the millet of Figure 9(n) than in Figure 9(m).

5 Conclusion

We proposed an interactive framework for manipulating the chromaticity mapping functions for multispectral images based on an edit propagation scheme. In our method, a factorized representation of the MSI is proposed for decomposing the propagation into a two-step scheme, in which edits are propagated in the spatial and spectral domains separately, leading to a general and efficient scheme for chromaticity mapping manipulation. With our method, we show several results generated from MSIs for color stylization and enhancement of different image details which cannot be achieved by conventional color image editing. In future work, we would like to extend our method to edit the chromaticity mapping functions of multispectral video data. Another interesting topic we would like to investigate is to exploit the spectral coherence of the scene to facilitate MSI acquisition.

References

1. Alien Skin Software, LLC.: Exposure4. URL http://www.alienskin.com/exposure/index.aspx. [online] http://www.alienskin.com/exposure/index.aspx
2. An, X., Pellacini, F.: Appprop: all-pairs appearance-space edit propagation. In: SIGGRAPH '08: ACM SIGGRAPH 2008 papers, pp. 1–9. ACM, New York, NY, USA (2008). DOI http://doi.acm.org/10.1145/1399504.1360639
3. Cao, X., Tong, X., Dai, Q., Lin, S.: High resolution multispectral video capture with a hybrid camera system. Proc. of Comp. Vis. and Pattern Rec. (CVPR) (2009)
4. Carroll, J., Neitz, J., Neitz, M.: Estimates of l:m cone ratio from erg flicker photometry and genetics. Journal of Vision **2**((8):1), 531–542 (2002)
5. Chakrabarti, A., Zickler, T.: Statistics of real-world hyperspectral images. In: In IEEE Int. Conf. Comp (2011)

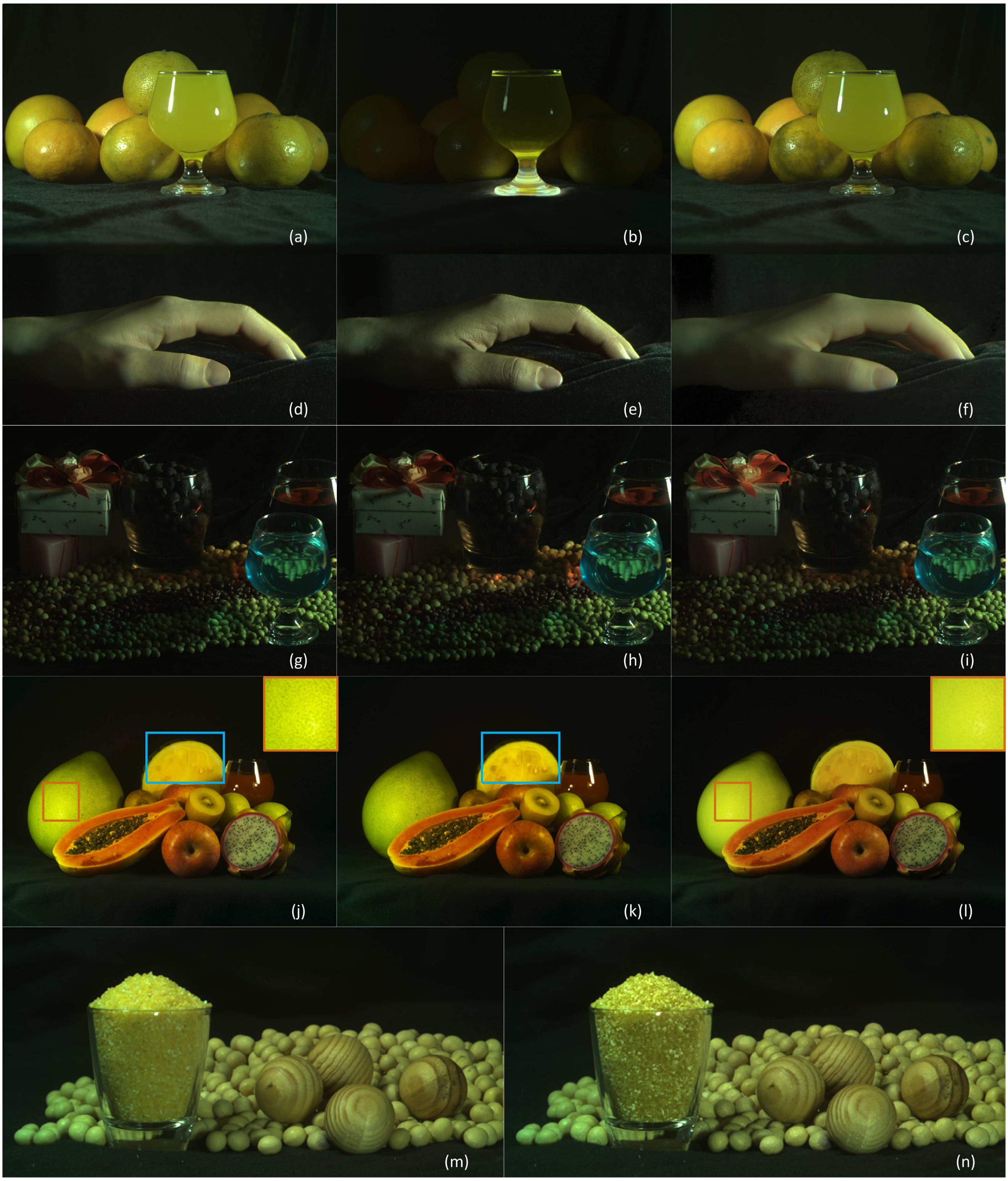

Fig. 9 Result of our method showing different detail preservation effects. The left column shows images generated with conventional chromaticity mapping functions of RGB cameras. The middle and right columns are results produced by our system to preserve and enhance different types of scene details. The parameters (σ_v,σ_p,σ_s) for each row are (0.002,0.001,0.03), (0.006,0.001,0.01), (0.004,0.0001,0.003), (0.002,0.0005,0.003), and (0.002,0.003,0.003), respectively.

6. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, pp. 369–378 (1997)
7. Descour, M.R., Dereniak, E.L.: Computed-tomography imaging spectrometer: experimental calibration and reconstruction results. Applied Optics **34**(22), 4817–4826 (1995)
8. Du, H., Tong, X., Cao, X., Lin, S.: A prism-based system for multispectral video acquisition. Proc. of Int'l Conf. on Comp. Vis. (ICCV) (2009)
9. Farbman, Z., Fattal, R., Lischinski, D.: Diffusion maps for edge-aware image editing. ACM Trans. Graph. **29**(6), 145:1–145:10 (2010). DOI 10.1145/1882261.1866171. URL http://doi.acm.org/10.1145/1882261.1866171
10. Galatsanos, N., Segall, A., Katsaggelos, A.: Digital image enhancement. Encycl. Optical Engineering (2005)
11. Gat, N.: Imaging spectroscopy using tunable filters: A review. In: SPIE Wavelet Appl. VII, vol. 4056, pp. 50–64 (2000)
12. Gehm, M.E., John, R., Brady, D.J., Willett, R., Schultz, T.: Single-shot compressive spectral imaging with a dual-disperser architecture. Optics Express **15**(21), 14,013–14,027 (2007)
13. Jolliffe, I.: Principal Component Analysis. Springer Series in Statistics. Springer-Verlag (2002). URL http://books.google.com/books?id=_olByCrhjwIC
14. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM Trans. Graph. (SIGGRAPH 2004) **23**(3), 689–694 (2004)
15. Li, Y., Ju, T., Hu, S.M.: Instant propagation of sparse edits on images and videos. Comput. Graph. Forum pp. 2049–2054 (2010)
16. Lischinski, D., Farbman, Z., Uyttendaele, M., Szeliski, R.: Interactive local adjustment of tonal values. ACM Trans. Graph. **25**(3), 646–653 (2006). DOI http://doi.acm.org/10.1145/1141911.1141936
17. Mohan, A., Raskar, R., Tumblin, J.: Agile spectrum imaging: Programmable wavelength modulation for cameras and projectors. Comput. Graph. Forum **27**(2), 709–717 (2008)
18. Mooney, J.M., Vickers, V.E., An, M., Brodzik, A.K.: High-throughput hyperspectral infrared camera. J. Opt. Soc. Am. A **14**(11), 2951–2961 (1997)
19. Ng, R., Ramamoorthi, R., Hanrahan, P.: All-frequency shadows using non-linear wavelet lighting approximation. ACM Transactions on Graphics **22**(3), 376–381 (2003)
20. Park, J., Lee, M., Grossberg, M.D., Nayar, S.K.: Multispectral Imaging Using Multiplexed Illumination. In: Proc. of Int'l Conf. on Comp. Vis. (ICCV) (2007)
21. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. ACM Transactions on Graphics **28**(1), 3:1–3:18 (2009)
22. Pellacini, F., Lawrence, J.: Appwand: editing measured materials using appearance-driven optimization. ACM Trans. Graph. **26**(3) (2007). DOI 10.1145/1276377.1276444. URL http://doi.acm.org/10.1145/1276377.1276444
23. Schechner, Y.Y., Nayar, S.K.: Generalized mosaicing: wide field of view multispectral imaging. IEEE Trans. Patt. Anal. Mach. Intel. **24**(10), 1334–1348 (2002)
24. Vandervlugt, C., Masterson, H., Hagen, N., Dereniak, E.L.: Reconfigurable liquid crystal dispersing element for a computed tomography imaging spectrometer. SPIE **6565** (2007)
25. Wagadarikar, A., John, R., Willett, R., Brady, D.J.: Single disperser design for coded aperture snapshot spectral imaging. Applied Optics **47**(10), B44–B51 (2008)
26. Wandell, B.A., Silverstein, L.D.: The Science of Color, 2nd ed., chap. Digital Color Reproduction. Opt. Soc. Am. (2003)
27. Wang, J., Dong, Y., Tong, X., Lin, Z., Guo, B.: Kernel nyström method for light transport. ACM Trans. Graph. (SIGGRAPH 2009) **28**(3), 29:1–29:10 (2009)
28. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formulae. Wiley-Interscience (2000)
29. Xu, K., Li, Y., Ju, T., Hu, S.M., Liu, T.Q.: Efficient affinity-based edit propagation using k-d tree. ACM Trans. Graph. **28**, 118:1–118:6 (2009). DOI http://doi.acm.org/10.1145/1618452.1618464. URL http://doi.acm.org/10.1145/1618452.1618464
30. Xu, K., Wang, J., Tong, X., Hu, S.M., Guo, B.: Edit propagation on bidirectional texture functions. Computer Graphics Forum **28**(7), 1871–1877 (2009)
31. Yamaguchi, M., Haneishi, H., Fukuda, H., Kishimoto, J., Kanazawa, H., Tsuchida, M., Iwama, R., Ohyama, N.: High-fidelity video and still-image communication based on spectral information: Natural vision system and its applications. In: SPIE/IS&T Electr. Imaging, vol. 6062 (2006)
32. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S.: Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum. Tech. rep. (2008)