Illumination Demultiplexing from a Single Image

Christine Chen
ETH Zürich
Zürich, Switzerland
christinechen@ethz.ch

Daniel Vaquero
UC Santa Barbara
Santa Barbara, CA, USA
daniel@cs.ucsb.edu

Matthew Turk
UC Santa Barbara
Santa Barbara, CA, USA
mturk@cs.ucsb.edu

Abstract

A class of techniques in computer vision and graphics is based on capturing multiple images of a scene under different illumination conditions. These techniques explore variations in illumination from image to image to extract interesting information about the scene. However, their applicability to dynamic environments is limited due to the need for robust motion compensation algorithms. To overcome this issue, we propose a method to separate multiple illuminants from a single image. Given an image of a scene simultaneously illuminated by multiple light sources, our method generates individual images as if they had been illuminated by each of the light sources separately. To facilitate the illumination separation process, we encode each light source with a distinct sinusoidal pattern, strategically selected given the relative position of each light with respect to the camera, such that the observed sinusoids become independent of the scene geometry. The individual illuminants are then demultiplexed by analyzing local frequencies. We show applications of our approach in image-based relighting, photometric stereo, and multiflash imaging.

1. Introduction

Illumination plays a central role in the appearance of a scene. Variations in the location, intensity, wavelength, and number of light sources directly impact the shading and shadows observed. Given the direct relationship between illumination and appearance, many techniques in computer vision apply controlled illumination to infer useful information about the scene. These approaches typically capture multiple images of the scene, varying illumination from image to image while the camera stays still. Examples of techniques in this category include photometric stereo [24], which explores differences in shading to estimate surface normals; image-based relighting [6, 13], which renders the scene as if it had been illuminated by a combination of captured illuminations; and multiflash imaging [18] and shape-from-shadows [3], which use detected shadows to compute depth discontinuities and 3D structure, respectively.

Figure 1: **Goal of illumination demultiplexing.** Given a single image of a scene illuminated by multiple light sources (left image), the objective is to obtain individual images as if they had been illuminated by each light source separately (images on the right).

The aforementioned techniques are more suitable for static rather than dynamic scenes, since the presence of moving elements requires compensation for motion from image to image. Due to the differences in illumination, it is challenging to provide a robust image alignment solution in this scenario. To overcome this limitation, we propose a method to separate (demultiplex) multiple illuminants from a single image, allowing for the capture of multiple illumination conditions in a single shot (Fig. 1). Given a single image of a scene illuminated by multiple light sources, our method generates individual images as if they had been illuminated by each of the light sources separately.

Demultiplexing the light contributions is an ill-posed problem in the absence of prior knowledge about the light sources. To make the problem more feasible, we encode each light source with a distinct sinusoidal pattern. The pattern for each light is strategically selected given the relative position of the light with respect to the camera, such that the observed sinusoids are independent of the scene geometry, building on previous work [22]. An advantage of our approach over [22] is that the projectors can be placed at arbitrary locations, as long as neither the projectors nor the camera occlude the fields of view or projection of others.

We first acquire a single image while the scene is simultaneously illuminated by all encoded lights, obtaining a multiplexed representation of the light sources. We then decode the contributions of individual light sources by locally analyzing the amplitude of sinusoids. The use of si-

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00

nusoids as codes results in a resolution trade-off; the output images will have lower resolution than the input image, but the loss in resolution is compensated by the ability to acquire all illumination conditions using a single image. We characterize the resolution trade-off in Section 4.2.

We present experiments comparing the demultiplexed images to ground truth images captured using only one illuminant at a time. Our method shows promising performance, despite the resolution trade-off. We also illustrate the applicability of our approach by using it to provide inputs to three different algorithms: image-based relighting, photometric stereo, and multiflash imaging. We apply the technique to multiplex illuminations into a single image and obtain the individual demultiplexed images, which are then provided as input to the algorithms. Our approach also enables the three applications to be performed on video on a frame-per-frame basis (see the supplementary material).

In summary, our main contributions are: (1) A method to generate sinusoidal coding patterns for arbitrarily placed projectors in projector-camera pairs such that the frequency of the observed sinusoid in the image does not depend on the scene geometry; (2) An approach to demultiplex the contributions of multiple light sources using a single image. The method is based on decoding geometry-invariant sinusoids, and sacrifices resolution for the ability to multiplex several illuminants into a single shot; (3) The application of the demultiplexing scheme to photometric stereo, multiflash imaging, and image-based relighting in dynamic scenes.

2. Related Work

Structured light refers to the technique of projecting known patterns of light onto a scene in a projector-camera system. By analyzing the reflected patterns observed by the camera, 3D scene reconstruction can be achieved. See [19] for a recent survey of the extensive literature in this area. Structured light was also used to construct optical super-resolution images [16]. Recently, structured light was applied to deal with a different problem [22]: keeping the frequency of projected sinusoids invariant regardless of the scene geometry, in order to facilitate frequency demultiplexing when multiple projectors illuminate the scene. The authors showed how to construct sinusoid patterns when the projectors and the camera are coplanar. We extend their results to projectors and camera placed at arbitrary locations.

Image separation has also been an active research area. A major interest is to recover intrinsic characteristics such as reflectance and illumination from images [8, 21, 23]. Another direction is to separate two superimposed images, such as reflections from a window [1, 14]. High-frequency structured light was used to separate the global and direct illumination components of a scene [15]. Researchers have also applied composite structured lighting patterns and image separation techniques [12] to reconstruct 3D geometry of a moving scene. However, their technique requires careful camera-projector alignment and does not compute individual illuminated images.

We address the problem of separating different illuminations simultaneously cast onto a scene. Different colors were used to encode illumination [7, 9] using a single image. However, color coding is not suitable for scenes containing colors similar to the light sources. Another single-shot color multiplexing method for photometric stereo was proposed in [11], but it required a special multi-channel capture setup with two cameras. The use of polarization to encode illumination conditions was also proposed [2], but the method requires a polarizing screen and needs to capture more than one image if more than three light sources are used. In [5], a multiplexing scheme based on time and color was introduced, but it requires the capture of at least $\lceil \frac{n+2}{3} \rceil$ images to multiplex n light sources. A theory for illumination multiplexing [20] was applied to improve the signal-to-noise ratio of captured images. However, multiple images of the scene are needed. Recently, a geometry-invariant scheme for coding lights using sinusoids [22] was introduced, and applied to demultiplex shadows from a single image. Our method extends this approach, by outputting general color or intensity images, instead of binary shadow/no-shadow maps.

3. Computational Illumination Multiplexing

In this section, we present our method to computationally encode light sources and multiplex lighting conditions into a single image. In our system, we have several projectors acting as illumination sources and a camera to record the scene. The light sources are spatially modulated by sinusoidal codes; consequently, the captured images are also encoded with the reflected codes. A challenging issue when demultiplexing illumination based on sinusoids is that the scene geometry can introduce variations in the projected patterns (Fig. 2(b-c)), making demultiplexing difficult.

However, by analyzing the epipolar geometry of a camera pair, and observing that a projector is the dual of a camera [17], we show that there is a family of codes such that their reflections in the captured image do not depend on scene geometry. In the dual perspective imaging pair formed by a camera and a projector, a projected epipolar line will be recorded in the camera as the corresponding epipolar line, regardless of the scene depth (Fig. 2(a)). Therefore, an illumination coding pattern which has a constant value along an epipolar line will be perspectively transformed into a coding pattern with the same value along the corresponding epipolar line in the image space. In this case, the reflected codes in the image are fully computable with epipolar geometry and camera calibration. The illumination code and the reflected code can be seen as fan-like patterns centered at epipoles in respective imaging planes.

(a) (b) (c) (d) (e)

Figure 2: **Computational illumination coding.** (a) In an imaging pair, an epipolar line projected by one device will be recorded as the corresponding epipolar line by the other device, regardless of the scene; (b-c) In settings where the camera and the projectors are in general positions, the frequency of a reflected conventional vertical structured light pattern may vary according to the scene geometry; (d-e) Projecting fan-like sinusoidal patterns centered at the epipole guarantees that the captured patterns are the corresponding epipolar lines, are independent of the scene geometry, and are fully computable.

Within this family of geometry invariant codes, we seek a code whose reflection is easy to demultiplex. We therefore design the reflected code to be a fan-like sinusoidal pattern centered at the camera epipole (Fig. 2(d-e)). With rectification, the reflected code can be transformed into a horizontal sinusoidal pattern when epipolar lines are rectified to be horizontal. To make decoding easier, we constrain the rectified code to have a constant spatial frequency. Given the desired reflected code in the captured image, it is then possible to compute the illumination code for each light source.

In the following, we first introduce our imaging and camera models. Then we discuss how to design the coding patterns observed in the image. Finally, we show how to compute the coding schemes for illumination sources.

3.1. Imaging Model

In our image model, we assume that the camera is photometrically calibrated and that the relative position of the light sources and the camera remains fixed throughout the capture process. Thus, when multiple light sources are present and illuminate the scene simultaneously, the image I comprises the ambient light A and the sum of the reflected intensity from each light source I_i. Our goal is to separate each I_i from I. Given only $I(x, y)$, however, computing the exact values of each $I_i(x, y)$ is an ill-posed problem. In order to distinguish different illumination sources, we need to encode each I_i with a coding pattern X_i. In fact, we have

$$I(x, y) = A(x, y) + \sum_i I_i(x, y)X_i(x, y) \qquad (1)$$

where I_i is pixel-wise multiplied with X_i. $I_i(x, y)$ can be approximated by $L'_i(x, y)r(x, y)\cos\alpha_i(x, y)$, where $L'_i(x, y)$ is the illumination from the i-th projector observed from pixel (x, y), $r(x, y)$ is the reflectance component, and $\alpha_i(x, y)$ is the angle between the incident illumination flux and the surface normal. We can code the intensity of illumination L'_i in the camera space with X_i to achieve coding on I_i. This is equivalent to pixel-wise multiplying an illumination code $P_i(a, b)$ with illumination intensity $L_i(a, b)$ in the projector space (Fig. 3(a-c)). In fact, X_i is the reflected image of P_i in the image pair.

We choose P_i such that X_i is unique (Fig. 3(d)) for each illumination and can be computed independently of the scene. In the next section, we introduce our camera model, which will be used to derive P_i.

3.2. Camera Model and Epipolar Geometry

In order to find the projection patterns P_i, we need to calibrate all imaging devices in our system to learn their intrinsic and extrinsic parameters. We use a printed checkerboard pattern on a planar surface to calibrate the camera and projected checkerboard patterns on the same surface to calibrate the projectors. Let the intrinsics for the camera and the projectors be K_c and K_{p_i}, the extrinsics be $[R_c|t_c]$ and $[R_{p_i}|t_{p_i}]$, and the optical centers of the camera and the projectors be C_c and C_{p_i}.

To process the recorded image more efficiently, we rectify the image so that all epipolar lines are horizontal and parallel. We use the method proposed by [10] to apply a homography to the recorded images. Define the rotation matrix R'_i for a new coordinate system as

$$R'_i = \begin{bmatrix} \frac{r_1}{||r_1||} & \frac{r_2}{||r_2||} & \frac{r_3}{||r_3||} \end{bmatrix} \qquad (2)$$

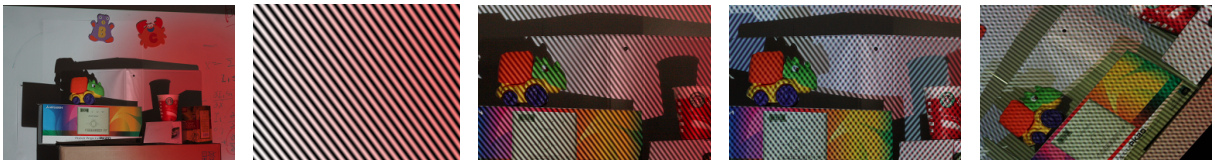
where $r_1 = (C_c - C_{p_i})$, $r_2 = k \times r_1$, $r_3 = r_1 \times r_2$, and k can be an arbitrary vector. The homography

$$H_i = K_c\, R'_i (K_c\, R_c)^{-1} \qquad (3)$$

will then rectify the captured photo into an image as if it had been taken by a camera with intrinsics K_c and extrinsics $[R'_i, t_c]$, where the epipolar lines are all horizontal.

3.3. Illumination Coding and Multiplexing

The authors of [22] have observed that the frequency of a projected sinusoidal pattern parallel to the baseline projected in a projector-camera system in canonical configuration remains invariant to the scene geometry when observed by the camera. Inspired by this result, we further notice that a fan-like sinusoid pattern P_i centered at the epipole of the projector will be perspectively transformed into a fan-like sinusoid pattern X_i centered at the camera epipole in the image (Fig. 2(a)). There exists a line-to-line correspondence between P_i and X_i, which makes either of them computable when the other is known.



(a) (b) (c) (d) (e)

Figure 3: **Illumination coding and multiplexing.** We code the illumination by pixel-wise multiplying it with the coding image. The coded illuminations are superimposed on the scene for each capture. (a) Scene lit by an uncoded colored illuminant; (b) the coded illumination we project onto the scene; (c) the scene lit by a coded source; (d) superimposition of two differently coded illuminations; (e) the scene with three superimposed coded illuminations and rectified with respect to one of the projectors.

We have the freedom to choose X_i in a way that decoding can be facilitated. After applying H_i (Eq. 3) to the captured image, the observed coding pattern X_i will be transformed into Y_i, which has the same value along a horizontal line. Y_i also has a line-to-line correspondence with P_i. For easier processing, we choose Y_i to be a sinusoid pattern with constant spatial frequency. To be more precise,

$$Y_i(\cdot, y) = \frac{\sin(\omega_i y + \phi_i) + 1}{2} \tag{4}$$

where ω_i is the spatial frequency and ϕ_i is the phase. To find P_i corresponding to Y_i, we first compute the fundamental matrix that maps the rectified image into the projected image. Note that the rectified image is identical to an image taken by a camera with intrinsics K_c and extrinsics $[R_i', t_c]$. Thus the fundamental matrix from Y_i to P_i is

$$G_i = [e_i]_\times K_{p_i} \left[R_{p_i} | t_{p_i}\right] \left(K_c \left[R_i' | t_c\right]\right)^+ \tag{5}$$

where $[e_i]_\times$ is the skew-symmetric matrix of the epipole e_i and $\left(K_c \left[R_i' | t_c\right]\right)^+$ is the pseudo-inverse of $\left(K_c \left[R_i' | t_c\right]\right)$. Therefore, a pixel (a, b) in P_i will correspond to the epipolar line in Y_i

$$l = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T = G_i^{-1} \begin{bmatrix} a & b & 1 \end{bmatrix}^T \tag{6}$$

and $P_i(a, b) = Y_i(\cdot, l_3/l_2)$. P_i is a fan-like sinusoidal pattern centered at e_i with varying angular frequency. Projecting P_i onto the scene, we will observe a fan-like sinusoidal pattern X_i centered at the epipole E_i. And after rectifying the recorded image, the coding pattern will be transformed into Y_i, which has a constant spatial frequency ω_i, regardless of the scene geometry.

We code the illumination L_i by pixel-wise multiplying it with P_i. The scene under coded illumination will be I_i pixel-wise coded with X_i, where I_i is the uncoded scene. This coding process is done for every illumination with unique P_i. The multiplexed image is then captured as the scene is illuminated by all coded light sources. The image is in fact the superimposition of all I_i coded with X_i.

3.4. Choosing Coding Frequencies

In the sections above, we described how to encode each illumination source such that the image $I_i X_i$ corresponding to the i-th illumination has a scene-invariant code. The image captured by the camera will contain the coding information X_i from all illumination sources. We need to carefully choose each ω_i in Y_i so that the captured coding patterns have no aliasing and are well separable.

Aliasing may be in either projected patterns P_i or observed patterns X_i. On the projector side, we require P_i to have horizontal and vertical cycle lengths larger than 2 pixels across the whole image. Using Eq. 6, we can rewrite $P_i(a, b)$ as

$$P_i(a, b) = \frac{\sin(\omega_i g_i(a, b) + \phi_i) + 1}{2} \tag{7}$$

where

$$g_i(a, b) = \frac{G_{i_{(3,1)}}^{-1} a + G_{i_{(3,2)}}^{-1} b + G_{i_{(3,3)}}^{-1}}{G_{i_{(2,1)}}^{-1} a + G_{i_{(2,2)}}^{-1} b + G_{i_{(2,3)}}^{-1}} \tag{8}$$

and $G_{i_{(m,n)}}^{-1}$ is the (m, n)-th entry of G_i^{-1}. We require

$$\frac{\partial(\omega_i g_i(a, b))}{\partial a} < \pi \text{ and } \frac{\partial(\omega_i g_i(a, b))}{\partial b} < \pi \tag{9}$$

for all a, b in the image space. Therefore, we need

$$\omega_i < \pi / \max_{(a,b)} \left(\frac{\partial g_i(a, b)}{\partial a}, \frac{\partial g_i(a, b)}{\partial b} \right) \tag{10}$$

Similarly on the camera side, we require X_i to have horizontal and vertical cycle lengths larger than 2 pixels. Denote

$$h_i(x, y) = \frac{H_{i_{(2,1)}} x + H_{i_{(2,2)}} y + H_{i_{(2,3)}}}{H_{i_{(3,1)}} x + H_{i_{(3,2)}} y + H_{i_{(3,3)}}} \tag{11}$$

where $H_{i_{(m,n)}}$ is the (m, n)-th item of the matrix H_i. It follows that

$$\omega_i < \pi / \max_{(x,y)} \left(\frac{\partial h_i(x, y)}{\partial x}, \frac{\partial h_i(x, y)}{\partial y} \right) \tag{12}$$

for x, y in the camera image space.

To ensure that the final captured image is well separable, we require local neighborhoods to have distinct frequency spectra with minimum overlap. In neighborhoods where X_i's have very different directions this can be easily satisfied with little restriction on ω_i. However, in neighborhoods where X_i have similar directions, ω_i may need to be decreased to avoid overlapping in the frequency space.

4. Illumination Demultiplexing

In the previous section, we discussed how to compute the coding pattern P_i for each projector and how to multiplex all illuminants into a single image. We now show how to separate different illuminations when they are all present in the scene. Recall that the image comprises an ambient component and the sum of the coded illuminants (Eq. 1). After rectifying with H_i, we have

$$I^i(x,y) = A^i(x,y) + \sum_j I^i_j(x,y)X^i_j(x,y) \qquad (13)$$

where I^i, A^i, I^i_j and X^i_j are the resulting images after applying homography H_i to I, A, I_j, and X_j, respectively. When $i = j$, X^i_j is precisely Y_i (Fig. 3(e)). In this section, we describe how to recover each I_i. Our approach is to apply the Short Time Fourier Transform (STFT) to each I^i to separate each I^i_i and then inverse rectify it to get I_i.

4.1. Demultiplexing with STFT

After rectification, from Eqs. (4) and (13) we have

$$I^i(x,y) = A^i(x,y) + I^i_i(x,y)\frac{\sin(\omega_i y + \phi_i) + 1}{2} + \sum_{j\neq i} I^i_j(x,y)X^i_j(x,y) \qquad (14)$$

Assuming that each X^i_j where $j \neq i$ has a different spectrum from Y_i at all (x,y), we can run a local Fourier Transform in a small neighborhood to extract the amplitude of the sinusoid Y_i alone. With a Hann window function W of size $M\times N$ on the local neighborhood, the STFT of a $M\times N$ neighborhood surrounding pixel $I^i(x,y)$ is $F^i_{(x,y)}$, where

$$F^i_{(x,y)}(\alpha,\beta) = \sum_{(a,b)} I^i(a,b)W(a-x,b-y)e^{-j(\alpha a+\beta b)} \qquad (15)$$

Denote the Fourier transform of W as w, and the Fourier transform of the $M\times N$ neighborhood around $I^i(x,y)$ as $f^i_{(x,y)}$. Then $F^i_{(x,y)}$ is the convolution of w and $f^i_{(x,y)}$:

$$F^i_{(x,y)} = f^i_{(x,y)} \star w \qquad (16)$$

Assuming the pixel values in this neighborhood are constant and equal to $I^i_i(x,y)$, the amplitude $|f^i_{(x,y)}(0,\omega_i)|$ of the horizontal sine wave of frequency ω_i is $I^i_i(x,y)/2$.

In summary, with known $F^i_{(x,y)}$ and w, we can approximate $f^i_{(x,y)}(0,\omega_i)$ and then recover $I^i_i(x,y)$. We only need to apply the inverse of rectifying homography H_i to I^i_i to retrieve I_i. We execute the calculations above for each color channel in I_i and repeat for all i until all I_i are computed.

4.2. Resolution Tradeoff

In the previous section, we discussed how to recover each pixel of I^i_i by computing STFT in a local window of size $M\times N$. We assume the amplitude of sine wave is constant locally and therefore we can keep $1/(M\cdot N)$ resolution in the result. The choice of window size is essential for the quality of the results. A larger window can reduce ringing artifacts, but can increase the loss of high frequency information. A smaller window can have greater response where amplitude changes happen but will be less precise in calculating local amplitude and may cause ringing artifacts in certain regions.

Since our code is rectified to be parallel, the choice of M determines whether the amplitude of I^i_i can be recovered correctly. Denote $M = m \times \frac{2\pi}{\omega_i}$. In our application we choose m to be at least 3, which results in a window height of at least three times the vertical cycle length.

The window width N determines whether other superimposed $I^i_j(x,y)X^i_j(x,y)$ can be eliminated. We notice that N in fact depends on the largest horizontal cycle length of $X^i_j(x,y)$ $(j \neq i)$ in the local neighborhood. Denote $N = n \times \frac{2\pi}{\Omega^i(x,y)}$ where $\Omega^i(x,y)$ is the minimum horizontal frequency of all $X^i_j(x,y)$ when $i \neq j$. Assuming that all $X^i_j(x,y)$ have constant frequencies in the local neighborhood, we choose n to be at least 2. To simplify, assume that all ω_i have already been chosen to avoid aliasing (Sec. 3.4). The final resolution loss rate of the i-th image in the local neighborhood (x,y) can be expressed as

$$\left(m \times \frac{2\pi}{\omega_i}\right) \cdot \max_{j\neq i}\left(n \times \frac{2\pi}{\hat{h}_i(x,y)\cdot\nabla(\omega_j h_j(x,y))}\right) \qquad (17)$$

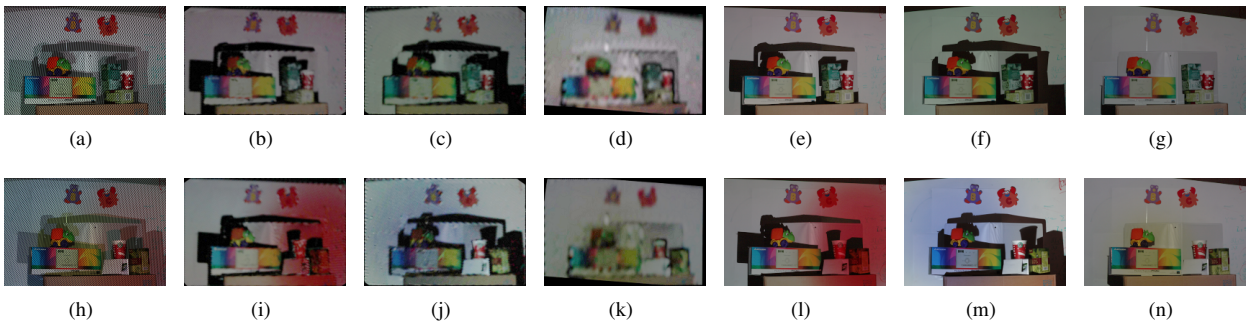
where $\hat{h}_i(x,y)$ is the normalized tangent directional vector of $h_i = 0$ at (x,y) and ∇ is the gradient sign. In the next section, we show results of our method using different window sizes.

5. Experimental Results

In the previous sections, we described how to multiplex coded illumination sources into a single image and how to demultiplex to recover the scene under individual uncoded illumination conditions. Now, we show the results from our experiments. In each experiment, we used our technique to multiplex three illumination conditions into one shot and demultiplex them into three differently lit images. For comparison, we took images of the scene lit by uncoded individual light sources to serve as ground truth.

The experiments were done with three Mitsubishi PK20 projectors and a Canon DSLR XSi camera. The projectors were placed at the top-right, lower-left and lower-right side of the camera. We experimented with two different lighting conditions, one with all uniform white lights and the other with three different diffuse colored lights. The images were taken with 1/30 second exposure, F/4 aperture, 50 mm focal length, and ISO value of 400. The camera was photometrically calibrated. For viewing purposes, we applied the same gamma function to result and ground truth images. The average sensor noise is equal to 1 per pixel in an 8-bit image.

We applied Hann window functions of different sizes



(a) (b) (c) (d) (e) (f) (g)

(h) (i) (j) (k) (l) (m) (n)

Figure 4: (a-g) **Demultiplexing uniform white illuminants.** (a) input image to our method; (b-d) individual demultiplexed lighting conditions; (e-g) ground truth acquired using uncoded individual lights. (h-n) **Demultiplexing diffuse colored illuminants.** (h) input image to our method; (i-k) individual demultiplexed lighting conditions; (l-n) ground truth acquired using uncoded individual lights.

	White lights			Colored lights		
# cycles	**2**	**3**	**4**	**2**	**3**	**4**
3	5.059	4.779	4.966	6.121	5.838	6.151
4	4.919	4.691	4.889	5.977	5.746	6.055
5	4.840	4.632	4.799	5.885	5.665	5.965
6	4.767	4.565	4.746	5.846	5.622	5.950

Table 1: Average pixel value differences between our demultiplexed results and ground truth images from experiments with uniform white lights and diffuse colored lights, measured in the RGB color space. We used different window sizes $M \times N$ in STFT to generate our results. Each row shows the results for the same m, while each column shows the results for the same n. The cycle length in our experiments is between 10 and 12 pixels.

$M \times N$ for demultiplexing. A set of results with $m = 3$ and $n = 2$ for both lighting conditions is shown in Fig. 4.

We analyzed the differences between the demultiplexed results and the ground truth images. The resulting images are of a smaller resolution than the originals (Section 4.2); hence, we downsampled the ground truth images to the same resolution. The comparison is shown in Table 1. Each entry shows the average per-pixel-per-channel difference between our results and the ground truth.

The comparison shows that our results are very close to the ground truth. The error decreases when the window height increases, because more horizontal cycles can attenuate information on the frequency of interest. The error also decreases when the window width increases, until a threshold is reached. This is mainly due to the ringing artifacts caused by applying STFT in a small window, which can be eliminated when the window is wide enough. However, as the window width keeps increasing, the error increases again. Here, the assumption that the frequencies of all X_i^j remain constant in local windows breaks, and STFT cannot categorize different signals.

6. Applications

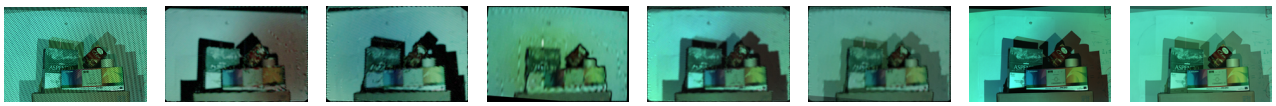
We present three applications of our proposed method. Image-based relighting, photometric stereo, and depth edge detection using multiflash imaging traditionally need multiple images of a scene under varying illumination conditions, requiring robust motion compensation solutions when used with dynamic scenes. In contrast, our approach captures multiple lighting conditions with a single image, avoiding the need for addressing motion-based issues.

6.1. Image-Based Relighting

Image-based relighting [6, 13] uses a collection of images of a scene captured under different illumination conditions to render the scene as if it had been illuminated by a combination of the lighting conditions. Our approach can provide input images for image-based relighting from a single shot of the scene, extending its application to scenes with moving objects. To illustrate, Fig. 5 shows examples of relighting a scene captured using our method. Fig. 5(a) shows the input image to our method, and Fig. 5(b-d) display the individual demultiplexed lighting conditions, obtained from Fig. 5(a); the images in Fig. 5(b-d) were then combined to render Fig. 5(e-f). For comparison, Fig. 5(g-h) show the results obtained by equivalently combining multiple images, individually acquired under different lighting conditions, as input to the relighting algorithm. The results are very similar, and the loss of resolution in our approach is compensated by the ability to perform a single image capture.

6.2. Photometric Stereo

Photometric stereo [24] exploits differences in shading under multiple illumination conditions to estimate surface normals. We applied our technique to obtain three illumination conditions from a single image, and used them as inputs for photometric stereo. Fig. 6 shows the input images and the resulting surface normals. Fig. 6(a) shows the input image to our method, and Fig. 6(b-d) display the individual lighting conditions demultiplexed from 6(a); Fig. 6(e) shows the surface normal map obtained using the images in Fig. 6(b-d) as inputs; and Fig. 6(f) shows the surface normal map obtained using multiple images instead of a single shot, for comparison. Our technique makes it possible to apply photometric stereo using a single image, effectively extending its applicability to dynamic scenes.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5: **Image-based relighting.** (a) input image to our method; (b-d) individual demultiplexed lighting conditions; (e-f) examples of scene relighting, obtained from demultiplexed images; (g-h) relighting results obtained using multiple images instead of a single shot.

(a) (b) (c) (d) (e) (f)

Figure 6: **Photometric stereo.** (a) input image to our method; (b-d) individual demultiplexed lighting conditions; (e) surface normal map obtained from the demultiplexed results – uncertainties and shadow regions are shown in gray; (f) surface normal map obtained using multiple images instead of a single shot. (Best seen in color.)

6.3. Depth Edge Detection

The authors of [18] proposed a multiflash approach for depth edge detection using images of the scene captured with different lighting. By detecting shadows, depth discontinuities can be reliably extracted by following the light epipolar rays. Our approach can provide multiple input images for this algorithm from a single multiplexed camera capture. Fig. 7 shows depth edge detection results obtained using our technique. We used three light sources for the experiment. Fig. 7(a) shows the input to our method, which is demultiplexed and provided as input for depth edge detection; Fig. 7(b-d) display the ratio images calculated using [18] for shadow detection; Fig. 7(e) is the extracted depth edge map; and Fig. 7(f) shows the depth edge map computed using individually lit images as inputs. Some edges are missed due to the resolution trade-off, but the result was obtained using a single image.

7. Discussion

As seen in the experiments and applications, our technique is able to perform illumination demultiplexing from a single image. This is very useful to extend the feasibility of applications requiring the capture of multiple illumination conditions to dynamic scenes. The coding scheme that results in constant frequency considerably facilitates the demultiplexing task. With calibration and rectification, there are no constraints on the positions of the light sources.

The proposed method also presents a few limitations. The algorithm computes demultiplexed images at a lower resolution (Eq. 17). As the projectors we used for our experiments were of resolution much lower than the camera's, ω_i was heavily restricted by Eq. 10 and the cycle length was of 10-12 pixels, leading to a resolution loss rate of 1000-2000x. However, ω_i can increase significantly when using higher resolution projectors, and the cycle length can be as low as 2-4 pixels, dropping the loss rate to 25-100x.

The assumption of sinusoids having constant amplitude in local patches breaks in the presence of high frequency variations in albedo (e.g., in textured areas), surface normals, ambient illumination, and depth discontinuities, which cause non-uniform modulation of the projected pattern within a local patch. This could be mitigated by increasing the resolution of the projected patterns. When global illumination components such as interreflections are present, our method is likely to fail because the coding pattern will depend on the unknown scene geometry.

Coding several illuminants within a single image limits the effective dynamic range of individual images. Adding light sources while keeping the camera's dynamic range fixed makes saturated areas more likely to occur. Specular highlights may also introduce similar problems.

Using projectors as illumination sources makes implementation convenient but may introduce color artifacts and pose additional limits on final resolution. An alternative is to use slide masks in front of light sources if the illumination does not change throughout the capture process. Despite these limitations, we believe our approach provides new insights that may lead to practical applications, especially as the size and cost of structured illumination devices continue to decrease.

8. Conclusions and Further Research

We have proposed a technique for multiplexing and demultiplexing multiple illuminants from a single image. A key component of our approach is a sinusoidal coding scheme, which chooses codes to be independent of the scene geometry in the observed image. The decoding technique sacrifices resolution to enable illumination demultiplexing from a single image. This is useful in applications that require the capture of multiple illumination conditions, such as photometric stereo, image-based relighting, and depth edge detection using multiflash imaging.

This work opens up opportunities for further research. Using information present in the coded multiplexed im-

(a) (b) (c) (d) (e) (f)

Figure 7: **Depth edge detection.** (a) input image to our method; (b-d) the ratio images obtained from the demultiplexed images. The shadow regions are clearly visible; (e) resulting depth edge map; (f) depth edge map obtained using multiple images instead of a single-shot.

ages to increase the resolution of the results could be helpful. Applying texture segmentation techniques within local patches might be useful for improving the quality when amplitudes of sinusoids vary. Finally, for video data, combining information from multiple frames using space-time consistency [4, 25] might improve the results.

The proposed structured light scheme can be understood as a projection of a carrier signal that can be recovered with constant frequency after going through a medium (the scene). This property may find important applications beyond the scope of our work [16].

Acknowledgements

We thank Ramesh Raskar for the encouragement and stimulating insights that led to the development of this work.

References

[1] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2005.

[2] O. G. Cula, K. J. Dana, D. K. Pai, and D. Wang. Polarization multiplexing for bidirectional imaging. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2005.

[3] M. Daum and G. Dudek. On 3-D surface reconstruction using shape from shadows. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 1998.

[4] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: a unifying framework for depth from triangulation. *IEEE Trans. on PAMI*, 27(2), 2005.

[5] B. De Decker, J. Kautz, T. Mertens, and P. Bekaert. Capturing multiple illumination conditions using time and color multiplexing. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2009.

[6] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 2000.

[7] M. S. Drew. Photometric stereo without multiple images. *Proc. of SPIE*, 3016(1):369–380, 1997.

[8] H. Farid and E. Adelson. Separating reflections and lighting using independent components analysis. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 1999.

[9] R. Feris, R. Raskar, and M. Turk. Dealing with multi-scale depth changes and motion in depth edge detection. In *Brazilian Symp. on Comp. Graph. and Image Processing*, 2006.

[10] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, July 2000.

[11] G. Fyffe, X. Yu, and P. Debevec. Single-shot photometric stereo by spectral multiplexing. In *IEEE Conf. on Comp. Photography*, 2011.

[12] C. Guan, L. Hassebrook, and D. Lau. Composite structured light pattern for three-dimensional video. *Opt. Express*, 11(5):406–417, Mar 2003.

[13] P. Haeberli. Synthetic lighting for photography. Available at `http://www.sgi.com/grafica/synth/index.html`, January 1992.

[14] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2004.

[15] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):935, 2006.

[16] P. Rangarajan, V. Bhakta, M. Christensen, and P. Papamichalis. Perspective imaging under structured light. In *European Conf. on Comp. Vision*, 2010.

[17] R. Raskar and P. Beardsley. A self-correcting projector. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2001.

[18] R. Raskar, K. Tan, R. Feris, J. Yu, and M. Turk. A non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2004.

[19] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Patt. Recog.*, 43(8):2666–2680, 2010.

[20] Y. Schechner, S. Nayar, and P. Belhumeur. A theory of multiplexed illumination. In *IEEE Intl. Conf. on Comp. Vision*, 2003.

[21] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. on PAMI*, 27(9):1459–72, 2005.

[22] D. Vaquero, R. Raskar, R. Feris, and M. Turk. A projector-camera setup for geometry-invariant frequency demultiplexing. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2009.

[23] Y. Weiss. Deriving intrinsic images from image sequences. In *IEEE Intl. Conf. on Comp. Vision*, 2001.

[24] R. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[25] L. Zhang, B. Curless, and S. Seitz. Spacetime Stereo: Shape Recovery for Dynamic Scenes. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, 2003.