Projection Defocus Analysis for Scene Capture and Image Display *

Li Zhang Shree Nayar
Columbia University

Abstract

In order to produce bright images, projectors have large apertures and hence narrow depths of field. In this paper, we present methods for robust scene capture and enhanced image display based on projection defocus analysis. We model a projector's defocus using a linear system. This model is used to develop a novel temporal defocus analysis method to recover depth at each camera pixel by estimating the parameters of its projection defocus kernel in frequency domain. Compared to most depth recovery methods, our approach is more accurate near depth discontinuities. Furthermore, by using a coaxial projector-camera system, we ensure that depth is computed at all camera pixels, without any missing parts. We show that the recovered scene geometry can be used for refocus synthesis and for depth-based image composition. Using the same projector defocus model and estimation technique, we also propose a defocus compensation method that filters a projection image in a spatially-varying, depth-dependent manner to minimize its defocus blur after it is projected onto the scene. This method effectively increases the depth of field of a projector without modifying its optics. Finally, we present an algorithm that exploits projector defocus to reduce the strong pixelation artifacts produced by digital projectors, while preserving the quality of the projected image. We have experimentally verified each of our methods using real scenes.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning, display algorithm; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth cues, range data, shape.

Keywords: projector defocus, temporal defocus analysis, depth recovery, multi-focal projection, projector depixelation, refocus synthesis, image composition.

1 Introduction

Digital projection technologies, such as Digital Light Processing (DLP) and Liquid Crystal Displays (LCD), are increasingly used in consumer, commercial and scientific applications. In computer graphics and vision, video projectors have recently been used as per-pixel controllable light sources for real-time shape acquisition [Huang et al. 2003; Zhang et al. 2004; Davis et al. 2005; Koninckx et al. 2005], for complex appearance capture [Levoy et al. 2004; Sen et al. 2005] and control [Raskar et al. 2001; Grossberg et al. 2004; Bimber et al. 2005]. All these applications require the projectors to be focused for best performance. In practice, projectors are built with large apertures to maximize their brightness. As a result, virtually all projectors have very narrow depths of field; they are designed to produce focused images on a single fronto-parallel screen. An analysis of the defocus properties of projectors is therefore beneficial as it could lead to new methods that take advantage of, as well as compensate for, projection defocus.

In this paper, we provide the first systematic analysis of projector defocus and demonstrate its applications to robust scene capture and enhanced image display. We first present a simple linear model for projector defocus. Based on this model, we present a frequency-domain method for estimating the spatially-varying defocus kernel of a projector. The kernel estimated at each scene point is used to recover the 3D geometry of the scene. Based on the estimated kernel, we also present a technique that computationally manipulates an input image to minimize its defocus blur when it is projected onto a non-planar scene. Finally, we demonstrate that defocusing can be put to good use. A slight amount of defocusing, in conjunction with a compensation algorithm, can be used to reduce pixelation, a strong artifact produced by all digital projectors. Specifically, this paper makes the following three contributions:

Scene Geometry using Temporal Defocus Analysis: We propose a method called temporal defocus analysis that estimates depth at each camera pixel, independently, without using information from neighboring pixels (Section 3). Our method, compared to existing methods, such as stereo or even traditional depth from focus/defocus algorithms, is more accurate near depth discontinuities. Since the method is not based on triangulation, we can use a coaxial projector-camera system and compute depth at all camera pixels, without any missing parts. These advantages make the method uniquely suited to computer graphics applications like refocus synthesis and image composition, both of which we demonstrate.

Focused Projection at Multiple Depths: We present an iterative, spatially-varying filtering algorithm that compensates for defocus blur based on scene geometry (Section 4). This technique effectively increases the depth of field of a projector without modifying its optics. As a result, we are able to use a single projector to project well-focused images on multiple planes that are at different depths as well as on curved projection surfaces. We believe this capability addresses an important limitation of current projectors and widens their applicability in the real world.

Depixelation by Defocusing: Finally, we generalize our defocus compensation algorithm to reduce the strong pixelation artifacts produced by all digital projectors (Section 5). The key idea is to slightly defocus the projector so as to attenuate the high frequencies produced by pixelation and use the defocus compensation algorithm to make up for the induced projector defocus. This method is especially suited for projecting high resolution images using a low resolution projector, and we show several examples of results to illustrate its effects.

2 Previous Work

Many methods have been proposed in computer vision to recover 3D shape from images, including multi-view triangulation methods, single-view photometric methods, and camera focus/defocus methods. Triangulation-based methods [Faugeras 1993], e.g., structure from motion and stereo, require a point to be visible in at least two views to be reconstructed. Scenes with complex occlusions remain a challenging problem for these methods. Photometric methods [Horn and Brooks 1989], e.g., photometric stereo and shape from shading, estimate surface normals instead of surface depth. Converting normals to depths is an ill-posed problem for scenes with depth discontinuities.

*This work was conducted at the Computer Vision Laboratory at Columbia University. It was supported by an ITR grant from the National Science Foundation (No. IIS-00-85864). The authors thank Gurunandan Krishnan for his help with the experiments and the video and Anne Fleming for her help with the audio recording and the proofreading. The authors also thank `www.dpreview.com` for permitting us using their images as some of the examples shown in the paper.

Depth recovery methods based on camera focus and defocus, e.g. [Pentland 1987; Nayar and Nakagawa 1994; Nayar et al. 1996; Schechner et al. 2000; Favaro and Soatto 2005], have the potential to recover depth at every pixel, regardless of the scene complexity and occlusions. To resolve the focus ambiguity of textureless surfaces, patterns can be projected to force scene texture [Girod and Scherock 1989]. However, camera defocus kernels depend on local surface geometry. To simplify the kernel analysis, most previous works assume that, within a small spatial window, the surface depth is constant, the so-called *equalfocal assumption*. This assumption smears shape details and is invalid across depth discontinuities. To alleviate this problem, Jin et al. [2002] and Rajagopalan and Chaudhuri [1997] estimate depths for all pixels simultaneously via a large scale energy minimization, which is computationally expensive and prone to local minima. Our key observation is that, unlike camera defocus, the kernel for projector defocus is *scene independent*, for most scene surfaces. Specifically, when a 3D scene point sees the entire projector aperture, its defocus kernel depends only on its distance to the projector lens and not on its neighboring surface geometry. This difference arises from the fact that projector defocus convolution happens on the projector's image plane while camera defocus convolution happens on the scene surface. Exploiting this scene-independent property, we project a shifting pattern over the scene and compute depth at each pixel using just its intensity variation over time. Without using the equalfocal assumption, our method works well at depth discontinuities. Furthermore, it is simple and not subject to local minima.

Our temporal per-pixel defocus analysis is inspired by previous structured light range finding methods. In particular, Kanade et al. [1991] and Curless and Levoy [1995] use temporal intensity variation to resolve correspondences between camera pixels and a sweeping laser stripe. Huang et al. [2003] have developed a real-time range finder by sweeping periodic sinusoidal stripes using a DLP projector. Zhang et al. [2004] and Davis et al. [2005] have generalized these ideas to space-time stereo. All these range finding techniques are based on the principle of triangulation, which cannot estimate depth for points that are visible to only the camera or the source but not both. Since our method is not based on triangulation, we are able to use a coaxial configuration where the camera and projector share the same optical center, and compute depth at all camera pixels (no missing parts). To our knowledge, the only other technique that is able to estimate a complete depth map with clean discontinuity boundaries is time-of-flight, e.g., [Gonzales-Banos and Davis 2004], which requires expensive specialized hardware. Raskar et al. [2004] proposed a method to detect depth discontinuities using multiple flash lights. The detected discontinuities can be used to enhance stereo matching algorithms. Compared to this work, our method directly generates reliable depth estimation for all pixels, including the ones near discontinuities.

In addition to scene capture, several methods have been proposed that use cameras to enable projectors to display "better" images. These include compensating distortions due to surface geometry [Raskar et al. 2003] and correcting brightness variations due to surface color and texture [Grossberg et al. 2004; Bimber et al. 2005]. In these works, the projectors are often used to display images onto scenes with considerable depth variation, where defocus blur is inevitable and most often spatially-varying. Complementary to these previous methods, our work seeks to compensate for defocus blur, and hence could be beneficial to the previous methods. In [Bimber and Emmerling 2006], a system is proposed that can project focused images on multiple planes at different depths. This system uses multiple projectors where each projector is focused on a single plane. In our case, we use a single projector to simultaneously project well-focused images at multiple depths. Another interesting related work is by Majumder and Greg [2001], in which multiple projectors at different focal settings are combined to generate depth of field effects at an interactive rate.

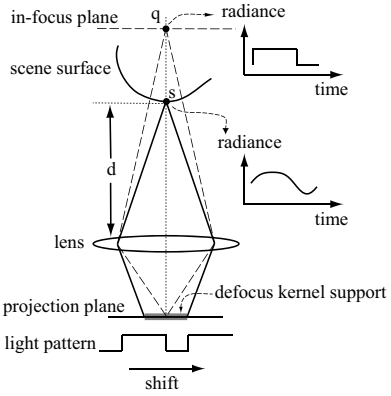
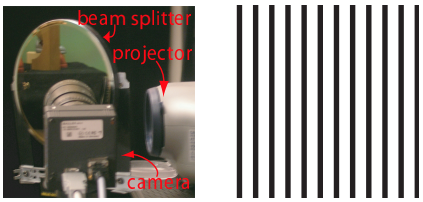


Figure 1: The principle of depth from projection defocus. Points at different distances to the projector lens exhibit different amounts of blur in their temporal radiance profile as a periodic illumination pattern is shifted across the scene.



(a) (b)

Figure 2: (a) A coaxial projector-camera system for depth from projection defocus. The system is made coaxial only to ensure that depth can be computed at all camera pixels. (b) The illumination pattern that is shifted across the scene to measure depth at each pixel, independently.

Deblurring is a well-studied topic in image processing where numerous techniques have been proposed, ranging from classical Wiener filtering and conjugate gradient optimization [Jain 1989] to more recent algorithms based on graph cuts [Raj and Zabih 2005] and belief propagation [Tappen et al. 2004]. Our technique is based on bound-constrained quadratic programming [Nocedal and Wright 1999], which is a variant of existing algorithms that best suits our formulation of the problem. As we show, our deblurring method can also reduce pixelation effects produced by digital projectors.

3 Depth from Projection Defocus

In this section, we formulate the problem of recovering 3D shape from projection defocus and present our solution.

3.1 Temporal Defocus Analysis

Consider a scene that is illuminated by structured light from a projector which is focused behind the scene, as shown in Figure 1. For a point **q** that is in focus, its irradiance comes from a single point on the projector's image plane. For a point **s** that is out of focus, its irradiance equals the convolution of its defocus kernel with the structured light pattern on the projector's image plane. Assuming that the surface is opaque, the radiance I of **s** along any given outgoing direction can be written as

$$I = \alpha f(\mathbf{x}; z) * P(\mathbf{x}) + \beta, \tag{1}$$

where $*$ denotes convolution, α is a factor depending on surface reflectance[1], β is the radiance due to the ambient light, $f(\mathbf{x}; z)$ is the

[1]To be precise, α takes into account all surface shading factors: BRDF, orientation with respect to the irradiance direction and the squared distance fall-off of the projector brightness.

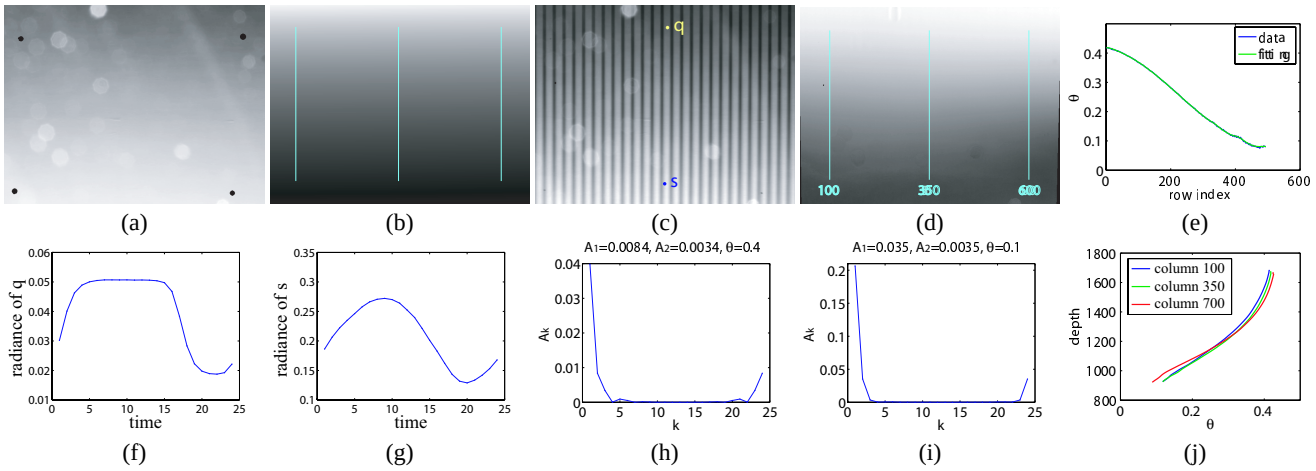


Figure 3: Illustration of the calibration procedure. (a) A white board with four markers tilted in front of the camera. (b) Depth map of the board computed using the markers. (c) The board under one of the stripe projection patterns. (d) θ map for the board. (e) θ values in column 350 of (d). (f,g) Temporal radiance profiles of the points **q** and **s**, respectively. (h,i) Discrete-time Fourier series (DFS) of (f) and (g). (j) Mappings from θ to depth z for columns 100, 350 and 600, respectively.

defocus kernel, and $P(\mathbf{x})$ is the illumination pattern. The defocus kernel f depends on the depth z.[2] We now describe how to recover the depth z by estimating the kernel f from radiance measurements.

3.2 Depth Estimation

To estimate the kernel f, we note that Eq. (1) defines a linear system in which the projection pattern and the scene radiance are the input and the output, respectively. Estimating the kernel of a linear system is a classical problem in system identification [Ljung 1998] and we take a frequency-domain approach to solve it in our setting. Our basic idea is to shift an illumination pattern with a wide range of frequencies within it across the scene. The radiance of a surface point over time is then the response of its defocus kernel to the excitation by the illumination pattern. As the pattern is shifted, points at different distances to the projector exhibit different amounts of blur in their *temporal* radiance profile, as illustrated in Figure 1. We use this temporal blur for depth recovery.

Given the temporal radiance sequence, I_l, $l = 0,\ldots,L-1$, for a point, we quantify its blur by decomposing it into a discrete-time Fourier series (DFS) [Oppenheim and Willsky 1997] as

$$I_l = A_0 + \sum_{k=1}^{L-1} A_k \cos(\omega_k l - \phi_k), \tag{2}$$

where $\omega_k = \frac{2k\pi}{L}$, $A_k = (B_k^2 + C_k^2)^{\frac{1}{2}}$, $\phi_k = \arctan(B_k, C_k)$, $B_k = \frac{1}{L}\sum_{k=0}^{L-1} I_l \sin(\omega_k l)$ and $C_k = \frac{1}{L}\sum_{k=0}^{L-1} I_l \cos(\omega_k l)$. Since the kernel f is a low-pass filter, how quickly the coefficients A_k diminish with k is a measure of the amount of defocus, which in turn yields depth. Note that A_0 cannot be used to estimate depth because it depends on the ambient light β. Although both A_1 and A_2 are scaled by the albedo α, their ratio can be used to determine how severely the defocus kernel attenuates the second-order harmonic with respect to the first-order one. Therefore, we use the following ratio

$$\theta = \frac{A_2}{A_1} \tag{3}$$

as a measure of depth. In Eq. (3), $A_1 > A_2 > 0$ and $\theta \in [0,1]$ because f is a low-pass filter. Deriving the analytic mapping between θ and z is tedious but possible if we know precisely the optical design of the specific projector. However, analytically deriving a projector-specific mapping is not worth the effort as it would not apply to other projectors — each projector tends to have a unique optical design. Therefore, in the next section, we present a general data-driven approach that calibrates the $\theta - z$ mapping once and for all, for any given optical setting of any given projector-camera system.

[2] Due to lens aberration, f generally also varies across different pixels.

3.3 System Setup and Illumination Pattern

We have built a prototype camera-projector system to implement our depth from projection defocus method. Our system consists of an NEC LT260K DLP projector and a Basler A311f monochrome camera. To avoid shadows and occlusions, we approximately align the optical centers of the projector and the camera with a beam splitter (Edmund Optics stock #NT39-493), as shown in Figure 2(a)[3]. The projector is always focused on a plane behind the working volume to avoid a two-way defocus ambiguity. As the projector is quite bright[4], we stop-down the aperture of the camera to F11 so that it works approximately as a pinhole camera – any defocus introduced by the camera is negligible compared to that of the projector.

There are many choices of input sequences (excitation signals) in system identification theory. We have chosen to use a simple one – a binary periodic sequence 011011011011... with period 3, which is one type of M-sequence [Ljung 1998]. We encode this sequence as a stripe pattern in which each bit corresponds to an 8-pixel wide stripe, as shown in Figure 2(b). We shift this pattern, one pixel at a time, and take a total of $L = 24$ images for each experiment[5].

3.4 Calibration

We calibrate the mapping from the θ in Eq. (3) to the depth z in three steps. **Step 1**: We compute the correspondence between projector and camera pixels. This is achieved by projecting shifted sinusoids in both horizontal and vertical directions. The details of this procedure can be found in [Scharstein and Szeliski 2003]. **Step 2**:

[3] We align the camera and the projector by projecting an image onto a white board with a fence in front of it and adjusting the camera position until it does not see any shadows cast by the projector.

[4] During depth estimation, we have used the projector as a grayscale one by removing its color wheel from the light path to boost its brightness.

[5] In our experiments, we have found that the projector produces several undesirable effects: it vibrates at a high frequency, possibly due to the rotation of its fan; its brightness is not stable over time; the imprecise synchronization between the camera and projector causes measured radiances to fluctuate slightly over time. To resolve these issues, we repeat the shifting of the pattern about 85 times and compute a mean image for each shift. The camera runs at 60Hz and the total acquisition time is within a minute.

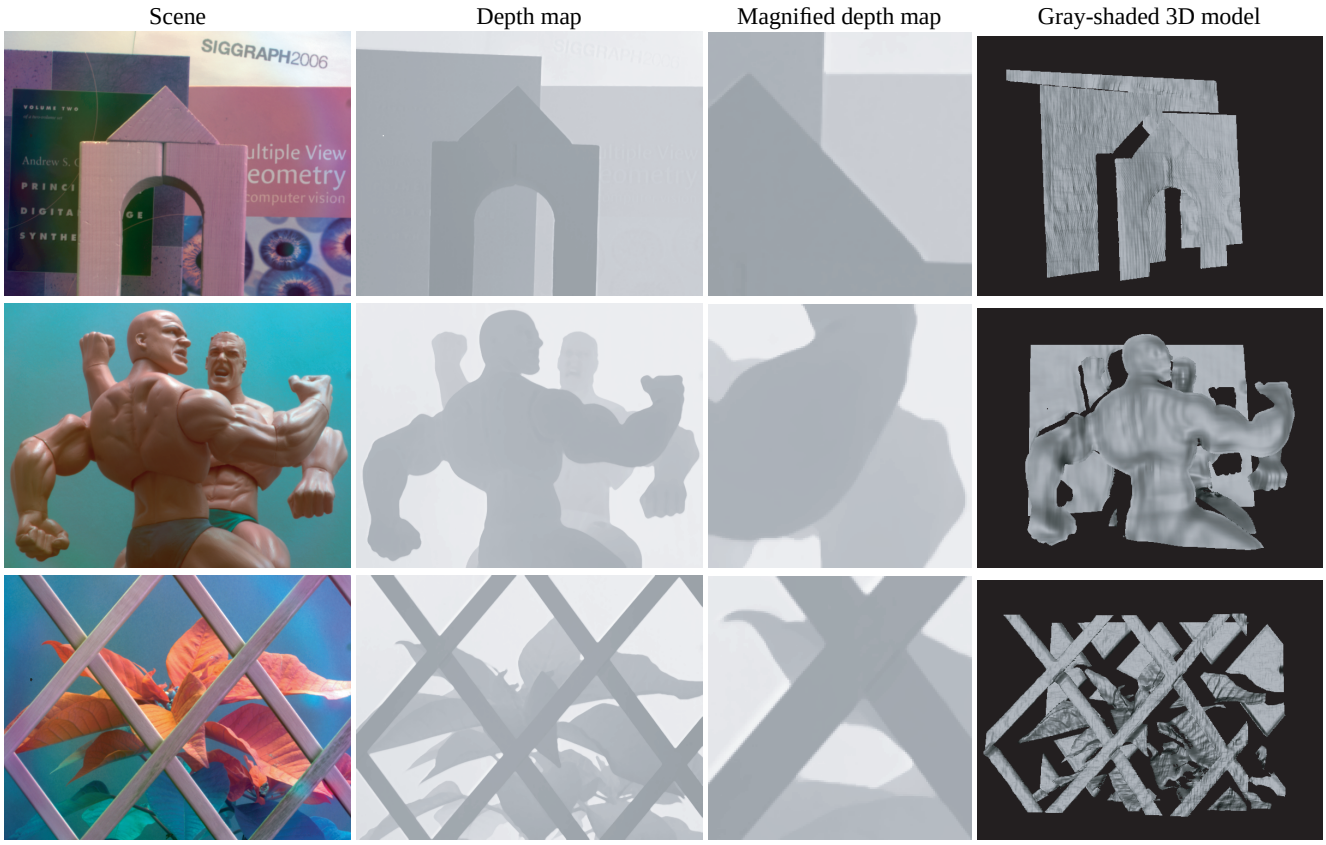


Figure 4: Depth recovery results for three scenes. From left to right: An image of the scene, the computed intensity-coded depth map, a close-up view of the depth map, a gray-shaded 3D model of the scene, as seen from a novel viewpoint. Notice the quality of the computed depth around scene discontinuities. **(Please see the companion video.)**

We tilt a foam board in front of the system and compute depth for each of its points. This is done by taking an image of the board with a few reference markers, as shown in Figure 3(a),[6] and computing the homography from the board to the projection plane. As detailed in [Zhang 2000], this homography allows us to estimate the position and orientation of the board[7], from which the depth of every point of the board can be easily computed. Figure 3(b) shows the intensity-coded depth map of the board. **Step 3**: We compute the θ values for all points on the board by shifting the stripe pattern and computing A_1 and A_2 for each pixel.

Figure 3(c) shows an image of the board under one of the shifted patterns, and (f) and (g) show the temporal radiance profiles of two points, **q** and **s**, on the board. As **s** is closer to the projector and more defocused than **q**, its temporal radiance profile is more blurred. Figures 3(h) and (i) show plots of the DFS coefficients A_k for the profiles in (f) and (g), respectively. Figure 3(d) shows the intensity-coded θ map and (e) shows a plot of the θ values (blue curve) for the column 350 in (d). The raw θ values include a small mount of noise and hence we fit a smooth curve which is shown as the green curve in (e). Using the depth estimation in Step 2, we can tabulate the mapping from θ to z. In our current implementation, we build a lookup table for each column, assuming that the defocus kernel is vertically invariant but has some horizontal variation. Figure 3(j) shows three θ–z mapping curves for the columns 100, 350 and 600. An even more comprehensive calibration would involve translating the board across the whole working volume and generating a lookup table for each pixel. Such a calibration would account for higher-order projector lens aberrations as well.

3.5 Depth Recovery Results

Once we know the mapping between θ and z, the depth recovery is straightforward. We take 24 images while shifting the illumination pattern across the scene. From these images, we compute θ for each camera pixel using Eqs. (2,3) and then transform the θ image to a depth image using the pre-computed lookup table. Figure 4 shows several depth recovery results. In the first row, we show results for a scene with books and wooden blocks that has simple boundaries. Our method recovers the sharp depth discontinuities at the boundaries. In the second row, we show results for a scene with two toy wrestlers with similar skin color. Again, our method can separate the two figures with clean boundaries, even though the surfaces have a specular component. In the third row, we show results for a scene with leaves behind a fence. This result shows that the performance of our method is independent of occlusion complexity. In fact, it has been shown [Scharstein and Szeliski 2003] that it is very difficult to obtain a depth map that is complete in any camera view for a scene like the one with leaves using triangulation-based methods, even if multiple cameras and projectors are used. All the results are computed within a minute using Matlab.

We have also computed noise statistics for our depth estimation. Specifically, we chose 20 planar patches from the experimented scenes that are located at different depth within the working volume. We then computed the mean and standard deviation of the

[6] The glare is produced by a number of dust particles on the half-mirror. Since the half-mirror is very close to the projector, these particles are lit with roughly 100 times the intensity compared to the scene, thereby producing bright spots in the image that have the shape of the camera aperture.

[7] Zhang [2000] takes several images of a board at different orientations to estimate a camera's intrinsic parameters. In our work, we approximately estimate the projector's intrinsics from the frustum specification in its manual. Therefore, we only take one image of the board to compute its pose.

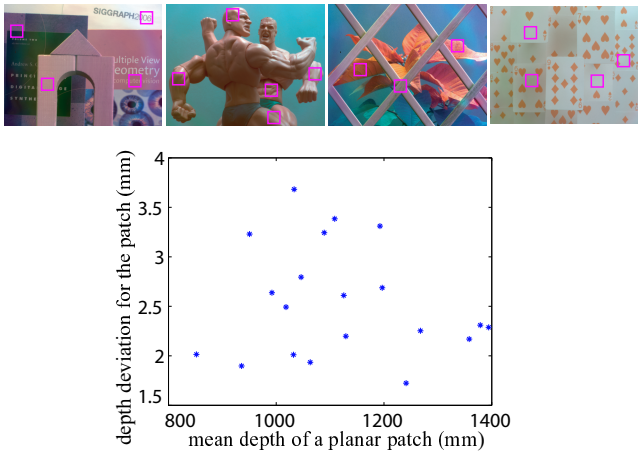


Figure 5: Noise statistics in the depth estimation. Twenty planar patches are chosen from the experimented scenes, shown on the top, and the mean and standard deviation of the depth for those patches are plotted at the bottom. Within a 600mm working volume, our depth estimation noise is around 4mm.

(a) (b) (c)

Figure 6: (a) Refocused images of the book scene in Figure 4. The defocus effects are synthesized using the recovered depth map. (b) A collection of playing cards placed at four depth layers and their estimated depth map. (c) A moving hand from another video is inserted into the card scene with all the desired occlusion effects. **(Please see the companion video.)**

depths inside these patches, as shown in Figure 5. From this figure, we can see that our depth estimation noise is about 4mm for a working volume of 600mm.

3.6 Applications: Refocusing & Video Composition

A distinctive feature of our depth recovery method is that it estimates depth at every pixel in the camera's view and the estimation is reliable near depth discontinuities. This makes our approach particularly well-suited for a variety of image-based applications in computer graphics. Here, we show two examples, namely, refocusing and object insertion to compose a new image or video.

Figure 6(a) shows two refocused images of the book scene in Figure 4: the top one is focused on the foreground wooden blocks and the bottom one is focused on the background letters. These two images were generated with the "lens blur" tool in Adobe PhotoShop.[8] This tool takes an image and its depth map as inputs to generate new images that are focused at any desired depth.

[8]In theory, without having the full light field available, refocusing can not be simulated exactly. The PhotoShop implementation of lens blur is proprietary. We think a practical way to implement it would be to heuristically impaint the occluded layers a little bit to obtain the full light field.

Figure 6(b) shows a collection of playing cards placed at four depth layers. The repetitive textures on the cards and the complex occlusions make it very hard to segment the scene into layers automatically, or even manually. Using projection defocus, our method produces a clean depth map that segments the scene into layers. This segmentation makes it easy to insert objects into the scene with correct occlusion effects. Figure 6(c) shows a real hand from another video inserted into the scene with all the desired occlusion effects. **Please see the video for more results.**

We believe that when a complete depth map is available for an image, a variety of image editing applications become possible or easier to implement. Examples other than the ones we have shown are creating layered representations of complex scenes for view morphing, segmentation for matting, object replacement, and shadow removal. In the past, such applications have not been easy to implement as "clean" depth maps have not been easy to obtain. For most image editing tasks, the depth maps need not be highly precise (as with time-of-flight sensors), but they need to be complete and they need to be reliable at discontinuities, as humans are particularly sensitive to edge artifacts. We hope our depth recovery method, although not as accurate as laser scanners or structured light sensors, will inspire new types of image editing applications.

4 Focused Projection at Multiple Depths

Due to their optical design, standard projectors can only be focused on a single fronto-parallel plane. In some applications, it is desirable to project images onto non-planar structures; for example, multiple planes or a dome to create a virtual environment [Raskar et al. 1998]. In such cases, most parts of the image are blurred due to defocus. One way to solve this problem is to design sophisticated optics. Even if this is possible, the optics cannot be modified to accommodate changes in the structure that is being projected onto. Another approach is to use multiple projectors [Bimber and Emmerling 2006], where the number of depths for which the projected image can be in focus equals the number of projectors. In this section, complementary to the multi-projector approach, we propose a computational method that processes an input image in a scene-dependent way to minimize the defocus blur at all points in the projected output image. Our method requires just a camera and can be applied to any off-the-shelf projector.

4.1 Defocus Compensation Algorithm

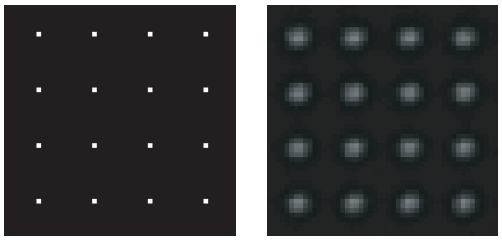
Consider the scenario where an image I needs to be projected onto a surface with given depth variation. The radiance of a point on the surface due to illumination by the projector is governed by the projection defocus equation Eq. (1). As in previous work, e.g. [Raskar et al. 2003; Fujii et al. 2005], we use a camera as a proxy for the human eye and seek to make the scene radiance captured by the camera be the same as the input image I. To do so, we can project a *compensation image*, P^*, by solving the projection defocus equation as

$$P^* = (\alpha f)^{-1} * (I - \beta), \tag{4}$$

where $(\alpha f)^{-1}$ is the inverse of the kernel αf. Theoretically, if P^* is projected, the scene brightness will be the same as the input image I. However, f is a low-pass filter and its inverse will have strong ringing effects. Therefore, Eq. (4) will not be feasible to implement unless the projector has an infinite dynamic range. Instead, we cast the problem of computing the compensation image as a constrained minimization problem, as follows:

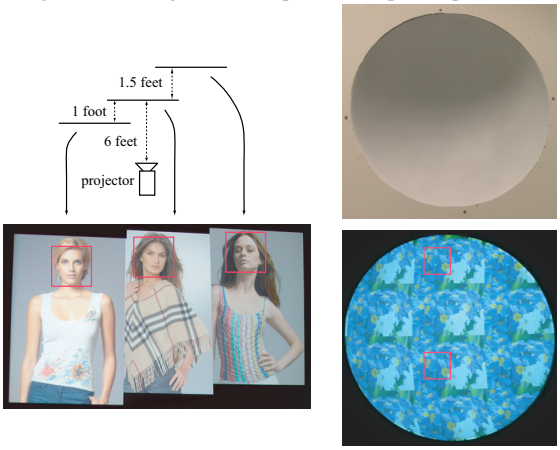
$$P^* = \arg\min_P \{d(\alpha f * P + \beta, I) \mid \forall \mathbf{x},\ 0 \le P(\mathbf{x}) \le 255\}, \tag{5}$$

where $\mathbf{x}$ is the projector pixel coordinate and $d(\cdot,\cdot)$ is an image distance metric. Eq. (5) finds the compensation image P^*, with all its brightness values within the projector's dynamic range, that after defocus blurring most closely matches the input image I.



(a) (b)

Figure 7: Measuring the spatially-varying defocus kernel. (a) A portion of the dot pattern projected onto the scene. (b) The corresponding camera image with samples of the point spread function.



(a) (b)

Figure 8: Two test scenes for focused projection at multiple depths. (a) Projection onto three planes at different depths from the projector. (b) Projection onto a hemispherical dome.

In our implementation, we have used the sum-of-squared pixel differences for the image distance metric $d(\cdot,\cdot)$. The compensation image P^* is found by applying an iterative, constrained, steepest-descent algorithm [Nocedal and Wright 1999]. We represent the defocus convolution, $\alpha f * P$, as a matrix multiplication, FP, where each row of F is the defocus kernel of the corresponding pixel modulated by its albedo. The algorithm starts with $P_0 = I$ and iterates the following two steps:

$$\tilde{P}_{i+1} = P_i + \eta_i G_i \quad (6)$$

$$P_{i+1} = \texttt{CLAMP}(\tilde{P}_{i+1};\, 0, 255) \quad (7)$$

where $G_i = F^{\mathrm{T}}(I - \beta - FP_i)$, $\eta_i = \frac{\|G_i\|^2}{\|FG_i\|^2}$, and CLAMP is a pixel-wise clamping operation. Notice that G_i is the gradient of the image distance $\|FP+\beta-I\|^2$ with respect to P. Evaluating G_i is straightforward – it involves two image filterings with the kernel matrices F and F^{T}, respectively. Note that these filterings are spatially-varying and scene-dependent, unlike the standard sharpening operations that are often built into projectors by their manufacturers. Without Eq. (7), iterating Eq. (6) alone is a standard steepest-descent algorithm, which converges to the solution of Eq. (4). Combining Eq. (7) and Eq. (6) minimizes the difference between the defocused compensation image and the original input image within the dynamic range of the projector.

4.2 Kernel Estimation

We now describe how we obtain β and αf for any given scene. We obtain the ambient term β by turning off the projector and taking an image. To obtain αf for each pixel, we could shift a dot pattern across the scene and the temporal radiance profile for each pixel is then its kernel. This method is effective for a scene with intricate geometry, such as the scenes shown in Section 3. However, the scenes used in our experiments here are at least piece-wise smooth. Therefore, we simply project a sparse binary dot pattern, like the one in Figure 7(a). In our experiments, the distance between neighboring dots is 12 pixels. The point spread patterns captured by the camera, shown in Figure 7(b), are approximately the kernels for the projector pixels that are '1'. From these kernels, we interpolate the kernels for other pixels that are '0' in a bilinear manner. In the end, we have a per-pixel kernel map that accounts for spatially-varying defocus effects. We repeat the same procedure to capture kernels for each of the three color channels of the projector. Calibrating such a kernel map also helps to compensate lens aberrations as well. For the above computations of β and αf, we always warp the image from the camera to the projector's coordinate frame, which makes it convenient to compare the defocused compensation image, $FP+\beta$, and the input image I. The correspondences between the camera's and the projector's pixels are determined by shifting sinusoidal patterns in both horizontal and vertical directions, as in Section 3.

4.3 Focused Projection on Multiple Planes

In our first experiment, our goal was to project an image of three fashion models onto three planes that are at different depths, as shown in Figure 8(a). The projector is focused on the middle plane. We show the focus compensation results for the three face regions in the left three columns of Figure 9. The first row of Figure 9 shows the original image regions for the three faces. The third row shows the defocused original image regions, captured by a camera, without any compensation. This row represents the best the projector (without compensation) can produce. Notice that even though the middle plane is in focus, the face still looks a little blurred for this plane due to projector artifacts.[9] As expected, the faces on the left and right planes are even more blurred due to defocus. The second row shows the compensation image regions resulting from Eqs. (6,7), which represent the new input to the projector. These image regions look like high-pass filtered versions of the original input ones. This is expected as our method boosts high-frequency components to compensate for the defocus induced by scene geometry. The fourth row shows the defocused compensation image regions captured by the same camera, which represent our results. Notice that these image regions are less blurry and more closely resemble the input image regions. Among these three defocused compensation image regions, the one on the middle plane (second column) is almost identical to the input. The ones on the left and right planes (first and third columns) are slightly blurred with respect to their inputs because defocus is more severe in these two cases, and certain high frequencies in the input image that are cut by the defocus cannot be fully compensated for due to the limited dynamic range of the projector.

4.4 Focused Projection on a Dome

In our second experiment, we project an image of a tiled flower texture onto the inside of a hemispherical dome, as shown in Figure 8(b). The projector is focused on the front plane of the dome. In the right two columns of Figure 9, we show the focus compensation results for two patches on the dome – one is from the top of the dome and the other is from the center. The top patch is less blurred than the center one because the former is closer to the focal plane of the projector. As with the three-plane experiment, the compensation results closely resemble the input images.

5 Depixelation

In this section, we show how the defocus compensation method discussed in Section 4 can be used to reduce projector pixelation.

[9] We found that even if we focus a projector on a fronto-parallel plane, when projecting a binary dot pattern, we see "light leakage" from one pixel into its neighbors. We observed this effect on different projector models.

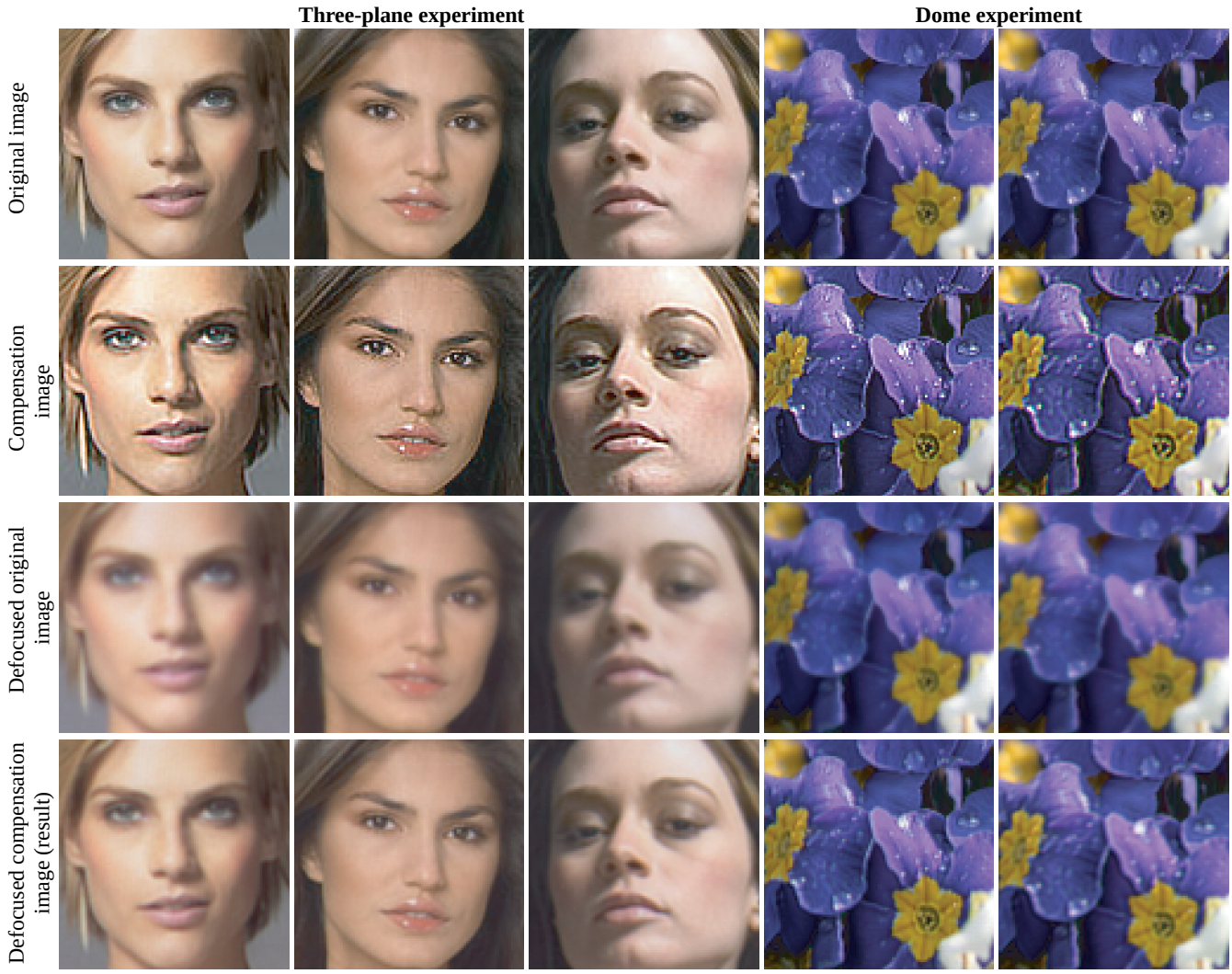


Figure 9: Results for defocus compensation. The three faces on the left were projected onto the three planar surfaces in Figure 8(a). The two flower textures on the right were projected onto the hemispherical dome in Figure 8(b). The defocus compensation method results in less blurry image regions (fourth row) than the uncompensated image regions (third row). The original image regions are shown in the first row and the compensation image regions computed by the method and used as input to the projector are shown in the second row. The compensation method is scene independent as it can handle a spatially-varying defocus kernel. **(Please see the companion video.)**

Pixelation is a clearly noticeable artifact produced by all digital projectors. It is caused by two factors. The first is the spatial digitization due to the finite resolution of the projector. The second is the gap (dead-zone) between adjacent pixels on the projector's physical image plane that arises because the pixel fill-factor is never 100 %. The digitization is known to create jaggy boundaries when the resolution of the projector is not high enough for the given application. This effect is observed particularly when high quality captured images are projected, as the resolution of most LCD and DLP projectors has remained at 1024x768 or less for the last seven years, while digital cameras have tripled in their resolution during the same time period. The dead-zone between pixels does not generate light and produces thin black lines on the projection screen, known as the *screen-door effect*. This effect makes pixelation more pronounced as it clearly marks out pixel boundaries on the screen.

In our implementation of depixelation, we assume that the images are projected onto a single fronto-parallel screen, i.e., the traditional projection scenario. Our basic approach is to focus the projector slightly in front of (or behind) the projection screen so that the image on the screen is slightly blurred. In this case, a slight amount of light is leaked into the black gaps and neighboring pixels, which reduces the screen-door effect as well as smoothes out the jaggy pixel boundaries. However, doing so also creates blurry images. Since this purposely induced blur is very slight, we can use the compensation method in Section 4 to process the input image so that the defocused projection looks very similar to the original input image. Therefore, the compensation method can be used in conjunction with the minor induced defocus to achieve depixelation.

Specifically, we use the compensation method to compute an optimal image P^* at the projector resolution whose defocused projection most closely resembles the input image, which may have higher resolution. Note that if the input image resolution is higher than the projector's native resolution, the number of rows in F is larger than the number of columns. Each row of F is a defocus kernel modulated by both surface albedo and the screen-door effect.

We have tested the depixelation method on several high resolution input images (each one with 3 times the projector resolution). The capture of β and F is done using the method in Section 4, except that we capture a higher resolution kernel map by zooming-in our camera so that 3×3 camera pixels see about one projector pixel.

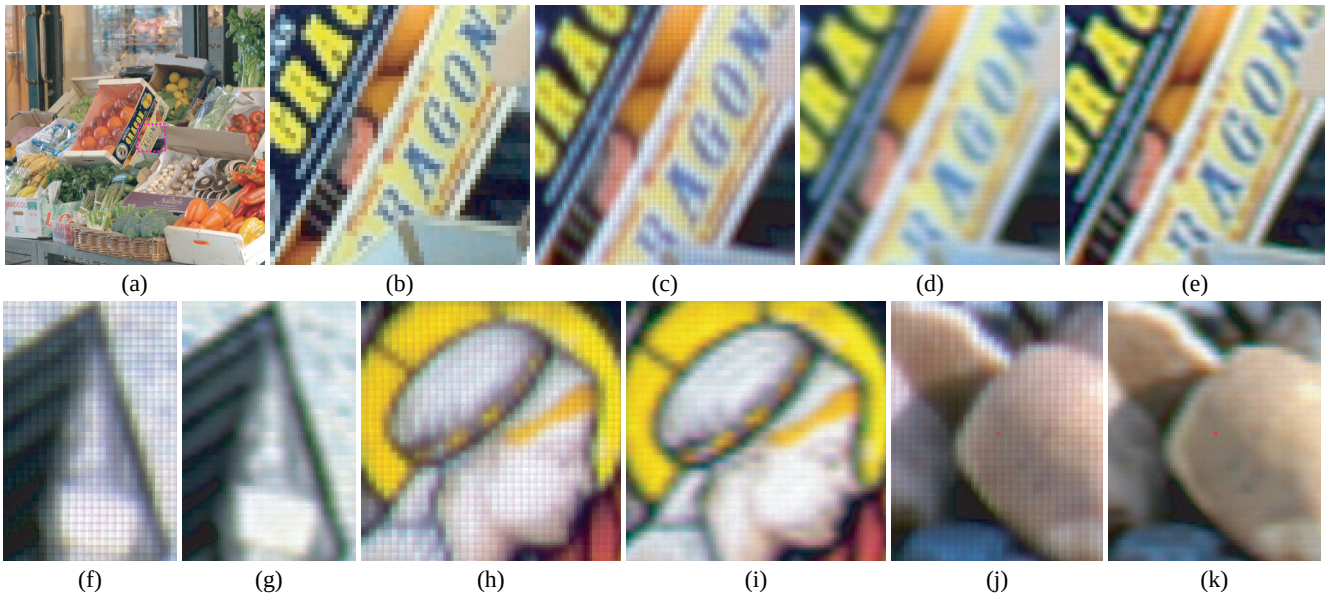


Figure 10: Examples of Projector Depixelation. (a) An original image of 3× projector resolution. (b) A patch in (a) of projector resolution. (c,d) Projection of (b) under focused and defocused settings, respectively. (e) Our depixelation result. (f-k) Three more examples of depixelation with (f,h,j) as focused projection and (g,i,k) as depixelated projection. In all examples, the compensation greatly reduces pixelation effects while preserving image quality. **(Please see the companion video.)**

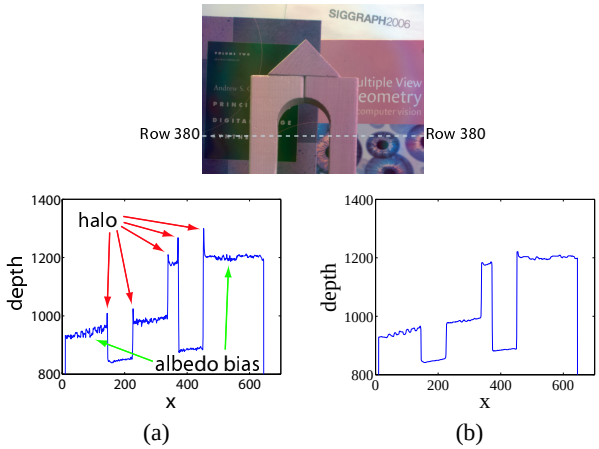


Figure 11: Albedo-correlated bias and halo effects in depth estimation. (a) The depth profile for row 380 of the scene. (b) The depth profile after removing noise and halo by a 7×7 median filter.

Figure 10 shows four examples of depixelation. In all the cases, spatial digitization and screen-door effects are greatly reduced and yet image sharpness is preserved. Our current implementation of the compensation algorithm uses Matlab. The algorithm takes about 10 iterations to converge and takes 3-5 minutes to produce the final compensation image. Since the compensation method only involves filtering with F and F^{T}, we believe it can be implemented on image processing chips or graphics hardware to achieve real-time (frame-rate) performance.

6 Discussion

In this paper, we have modelled projection defocus as a linear system and developed a temporal defocus method that recovers depth at all pixels in an image, regardless of the scene complexity. We also proposed a scene-dependent filtering algorithm that increases the depth of field of a projector without modifying its optics. Finally, we showed that defocus can be exploited to reduce pixelation artifacts produced by digital projectors. We now discuss the limitations of our work and open problems as topics for future research.

First, our temporal defocus method should, in principle, be invariant to surface albedo. In practice, we found that the surface albedo sometimes does affect the depth estimation, as shown in Figure 11. We believe that this is only an issue with our current implementation. Our beam-splitter not only reflects light from the scene to the camera, but also transmits light from a black backdrop placed behind it. Therefore, the defocus kernel f in Eq. (1) is actually a sum of the defocus kernels of both the scene and the backdrop. If the scene is much brighter than the black backdrop, Eq. (3) is dominated by the scene's defocus kernel. However, when the scene point is dark, due to low albedo or because its surface orientation is near-grazing for the projector, the light from the scene point to the camera is weak and comparable to the light from the backdrop. In this case, Eq. (3) will depend also on the defocus kernel of the backdrop, which causes errors in depth estimation. The backdrop kernel remains constant and can be calibrated to reduce depth estimation errors. Alternatively, the beam-splitter can be placed in a custom-designed chamber that absorbs all forms of stray light.

The projector defocus is for most cases independent of scene geometry, and this independence makes is superior to camera defocus for depth estimation. However, there is a special case that this independence does not hold. Specifically, when a closer object occludes a distant object, there will be a narrow band of points near the occluding contour on the distant object that "see" only part of the projector's aperture; so the projected pattern will be more infocus than it would be if the entire aperture were visible. This partial aperture effect results in a halo artifact: depth estimation biased toward the focal plane, as shown in Figure 11. However, even in this special case, projector defocus is still better than camera defocus. Camera defocus will cause estimation errors for both the foreground and background pixels near the occluding boundary. In contrast, projector defocus will only cause errors at the background pixels, while leaving the foreground pixels unaffected. Furthermore, in practice, because the solid angle subtended by the projector aperture is only about 1 degree with respect to a scene point, the number of affected background pixels is small. Therefore, as shown in Figure 11, the halo effect can be greatly reduced by a simple median filtering.

In practice, using projector defocus for depth estimation does not require moving detectors or changing optical settings as traditional camera defocus methods do [Pentland 1987; Nayar and Nakagawa 1994]. Therefore, our method can be more conveniently implemented. However, a coaxial implementation of depth from camera defocus, as in [Nayar et al. 1996], allows for capturing dynamic scenes, which our method currently is incapable of, because we use 24 images for depth estimation. We believe this number can be greatly reduced, which would enable us to develop a real-time sensor that can handle dynamic scenes. In Eq. (1), there are three unknowns: α, β, and z. If we can calibrate the defocus kernel f a priori, then, in theory, 3 images are enough to solve for these unknowns. Interestingly, this minimum number is the same for other shape acquisition methods, like the phase-shift method [Huang et al. 2003], the time-of-flight method [Gonzales-Banos and Davis 2004], and photometric stereo [Horn and Brooks 1989].

Our temporal defocus method can deal with specular surfaces, whenever the BRDF's are approximately constant within the solid angle subtended by the projector aperture from the scene point. Nevertheless, the extreme case of near-mirror reflection, as well as saturated highlight pixels, will cause errors. Although our method is developed under the direct illumination assumption, it's not very sensitive to the indirect illumination. This is because indirect illumination usually creates only very low frequency intensity fluctuation, which will mainly affect Fourier coefficient A_0. Our depth is however estimated via A_1 and A_2. Of course, for extremely specular reflection, the global illumination can change dramatically with the projected patterns and hence our method will break down.

One challenging open problem for defocus-based methods is handling translucent objects. A translucent material introduces additional blur due to subsurface scattering, which would cause a systematic bias in our methods. The amount of bias depends on the amount of translucency blur compared to defocus blur. If the defocus blur dominates the translucency blur, the error will be small. Otherwise, the error will be large. It would be interesting to extend our analysis to handle blur induced by translucency. In practice, the depth of mixture pixels at boundaries is often a blend of its neighboring depth values. It would be interesting to accurately model the mixture pixels, e.g., combining our approach with the defocus matting method [McGuire et al. 2005], to estimate mattes for scenes with continuous depth variations.

Our method for defocus compensation essentially extends the depth of field of the projector. However, when the blur is very severe, our method cannot fully compensate for the blur due to the limited dynamic range of the projector. In such a case, combining our method with multiple-projector solutions [Bimber and Emmerling 2006] could be beneficial. Finally, we plan to extend our work on depixelation to make it a practical and useful projector feature. How objectionable pixelation artifacts are, is really a subjective issue. For our technique to be incorporated into commercial projectors, a few preset options can be provided so that the user can decide how much focus quality they are willing to tradeoff for depixelation.

References

BIMBER, O., AND EMMERLING, A. 2006. Multi-focal projection. *IEEE Trans. on Visualization and Computer Graphics to appear*.

BIMBER, O., WETZSTEIN, G., EMMERLING, A., AND NITSCHKE, C. 2005. Enabling view-dependent stereoscopic projection in real environments. In *Proc. Int. Symp. on Mixed and Augmented Reality*, 14–23.

CURLESS, B., AND LEVOY, M. 1995. Better optical triangulation through spacetime analysis. In *Proc. Int. Conf. on Computer Vision*, 987–994.

DAVIS, J., NEHAB, D., RAMAMOOTHI, R., AND RUSINKIEWICZ, S. 2005. Spacetime stereo : A unifying framework for depth from triangulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 27*, 2, 296–302.

FAUGERAS, O. 1993. *Three-Dimensional Computer Vision*. MIT Press.

FAVARO, P., AND SOATTO, S. 2005. A geometric approach to shape from defocus. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (in press).

FUJII, K., GROSSBERG, M., AND NAYAR, S. 2005. A Projector-Camera System with Real-Time Photometric Adaptation for Dynamic Environments. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 814–821.

GIROD, B., AND SCHEROCK, S. 1989. Depth from defocus of structured light. In *Proc. SPIE Conf. on Optics, Illumination, and Image Sensing for Machine Vision*.

GONZALES-BANOS, H., AND DAVIS, J. 2004. A method for computing depth under ambient illumination using multi-shuttered light. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 234–241.

GROSSBERG, M., PERI, H., NAYAR, S., AND BELHUMEUR, P. 2004. Making One Object Look Like Another: Controlling Appearance using a Projector-Camera System. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. I, 452–459.

HORN, B., AND BROOKS, M. 1989. *Shape from Shading*. MIT Press.

HUANG, P. S., ZHANG, C. P., AND CHIANG, F. P. 2003. High speed 3-d shape measurement based on digital fringe projection. *Optical Engineering 42*, 1, 163–168.

JAIN, A. K. 1989. *Fundamentals of Digital Image Processing*. Prentice Hall.

JIN, H., AND FAVARO, P. 2002. A variational approach to shape from defocus. In *Proc. Eur. Conf. on Computer Vision*, 18–30.

KANADE, T., GRUSS, A., AND CARLEY, L. 1991. A very fast vlsi rangefinder. In *Proc. Int. Conf. on Robotics and Automation*, vol. 39, 1322–1329.

KONINCKX, T. P., PEERS, P., DUTR, P., AND GOOL, L. V. 2005. Scene-adapted structured light. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 611–619.

LEVOY, M., CHEN, B., VAISH, V., HOROWITZ, M., MCDOWALL, I., AND BOLAS, M. 2004. Synthetic aperture confocal imaging. In *SIGGRAPH Conference Proceedings*, 825–834.

LJUNG, L. 1998. *System Identification: A Theory for the User*, 2 ed. Prentice Hall.

MAJUMDER, A., AND WELCH, G. 2001. Computer graphics optique: Optical superposition of projected computer graphics. In *Proc. Eurographics Workshop on Virtual Enviroment/Immersive Projection Technology*.

MCGUIRE, M., MATUSIK, W., PFISTER, H., HUGHES, J. F., AND DURAND, F. 2005. Defocus video matting. In *SIGGRAPH Conference Proceedings*, 567–576.

NAYAR, S. K., AND NAKAGAWA, Y. 1994. Shape from focus. *IEEE Trans. on Pattern Analysis and Machine Intelligence 16*, 8, 824–831.

NAYAR, S. K., WATANABE, M., AND NOGUCHI, M. 1996. Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*, 12, 1186–1198.

NOCEDAL, J., AND WRIGHT, S. J. 1999. *Numerical Optimization*. Springer.

OPPENHEIM, A. V., AND WILLSKY, A. S. 1997. *Signals and Systems*, 2 ed. Prentice Hall.

PENTLAND, A. 1987. A new sense for depth of field. *IEEE Trans. on Pattern Analysis and Machine Intelligence 9*, 4, 523–531.

RAJ, A., AND ZABIH, R. 2005. A graph cut algorithm for generalized image deconvolution. In *Proc. Int. Conf. on Computer Vision*.

RAJAGOPALAN, A. N., AND CHAUDHURI, S. 1997. A variational approach to recovering depth from defocused images. *IEEE Trans. on Pattern Analysis and Machine Intelligence 19*, 10, 1158–1164.

RASKAR, R., WELCH, G., CUTTS, M., LAKE, A., STESIN, L., AND FUCHS, H. 1998. The office of the future : A unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH Conference Proceedings*, 179–188.

RASKAR, R., WELCH, G., LOW, K., AND BANDYOPADHYAY, D. 2001. Shader lamps. In *Proc. Eurographics Workshop on Rendering*.

RASKAR, R., VAN BAAR, J., BEARDSLEY, P., WILLWACHER, T., RAO, S., AND FORLINES, C. 2003. ilamps: geometrically aware and self-configuring projectors. In *SIGGRAPH Conference Proceedings*, 809–818.

RASKAR, R., HAN TAN, K., FERIS, R., YU, J., AND TURK, M. 2004. Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. In *SIGGRAPH Conference Proceedings*, 679–688.

SCHARSTEIN, D., AND SZELISKI, R. 2003. High-accuracy stereo depth maps using structured light. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 195–202.

SCHECHNER, Y. Y., KIRYATI, N., AND BASRI, R. 2000. Separation of transparent layers using focus. *Int. J. on Computer Vision 39*, 1, 25–39.

SEN, P., CHEN, B., GARG, G., MARSCHNER, S. R., HOROWITZ, M., LEVOY, M., AND LENSCH, H. P. A. 2005. Dual photography. In *SIGGRAPH Conference Proceedings*, 745–755.

TAPPEN, M. F., RUSSELL, B. C., AND FREEMAN, W. T. 2004. Efficient graphical models for processing images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 673–680.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, 548–558.

ZHANG, Z. 2000. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22*, 11, 1330–1334.