Active Refocusing of Images and Videos

Francesc Moreno-Noguer*
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne

Peter N. Belhumeur†
Columbia University

Shree K. Nayar‡
Columbia University

(a) Acquired Image (b) Computed Depth (c) Refocused (Far) (d) Refocused (Near) (e) Alternate Lighting

Figure 1: *Active refocusing of images. (a) Image acquired by projecting a sparse set of illumination dots on the scene. (b) The dots are automatically removed from the acquired image, and the defocus of the dots and a color segmentation of the image are used to compute an approximate depth map of the scene with sharp boundaries. (c and d) The depth map and the dot-removed image are used to smoothly refocus the scene. (e) The refocusing can also be done for an image taken immediately before or after but illuminated as desired.*

Abstract

We present a system for refocusing images and videos of dynamic scenes using a novel, single-view depth estimation method. Our method for obtaining depth is based on the defocus of a sparse set of dots projected onto the scene. In contrast to other active illumination techniques, the projected pattern of dots can be removed from each captured image and its brightness easily controlled in order to avoid under- or over-exposure. The depths corresponding to the projected dots and a color segmentation of the image are used to compute an approximate depth map of the scene with clean region boundaries. The depth map is used to refocus the acquired image after the dots are removed, simulating realistic depth of field effects. Experiments on a wide variety of scenes, including close-ups and live action, demonstrate the effectiveness of our method.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, Shading, Shadowing, Texture; I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Imaging Geometry; I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms.

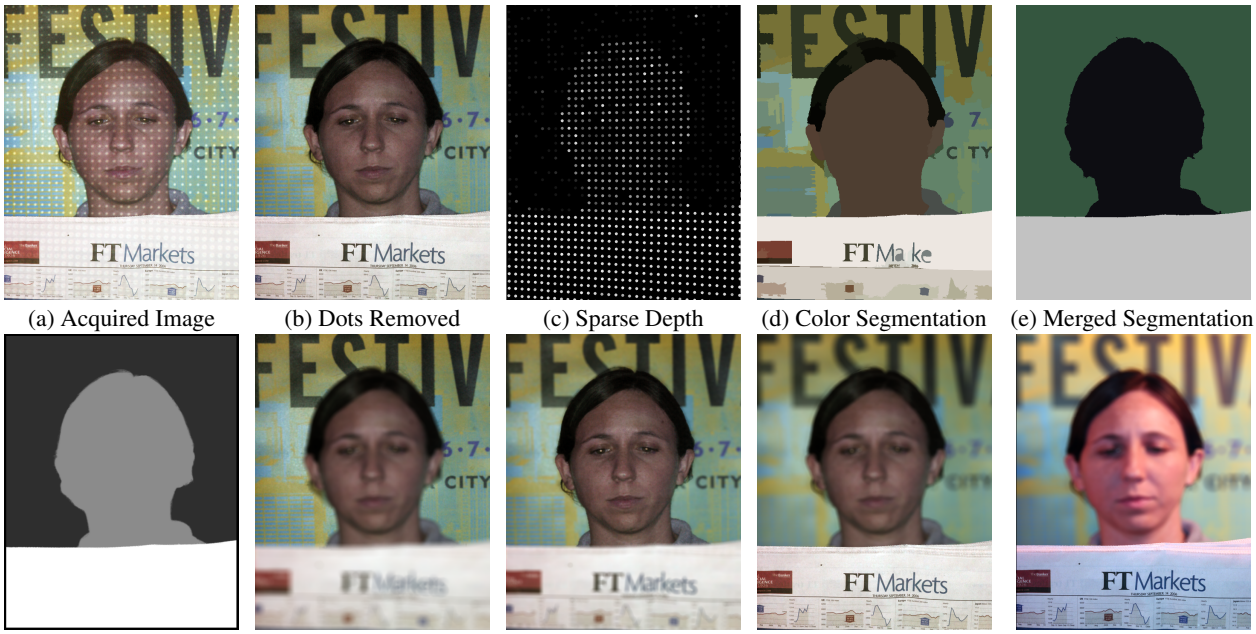
Keywords: active illumination, depth from defocus, image segmentation, depth of field, refocusing, computational photography.

*e-mail: francesc.moreno@epfl.ch
†e-mail: belhumeur@cs.columbia.edu
‡e-mail: nayar@cs.columbia.edu

1 Introduction

A method that allows for the refocusing of images and videos is a potentially powerful tool for digital photography and film editing. If one acquires an image with a wide depth of field, one can defocus the image by convolving it with a blur kernel whose size depends on the depth of each pixel [Potmesil and Chakravarty 1981]. However, to achieve this, one needs first to estimate the depth at each image pixel. While there exists a nearly endless literature on depth estimation from images, the requirements for the refocusing of a dynamic scene seem to preclude the use of most existing methods. First, because the scene is dynamic, the depth estimation needs to be done at a single moment in time – preventing the use of multi-frame active illumination depth estimation methods. Second, because we are refocusing the full image, we need depth estimates for every point in the image – preventing the use of multi-viewpoint depth estimation methods. Third, because our goal is to refocus the original image, we cannot use an active illumination method whose effects cannot be removed from the original image – preventing the use of existing single-frame active illumination methods.

In this paper, we present a simple single-frame active illumination method for depth estimation and incorporate it within a system for refocusing images and videos of dynamic scenes. Our method for estimating depth uses a single camera (with a wide depth of field) and is based on the defocus of a sparse set of dots projected onto the scene (using a narrow depth of field projector) (see Fig. 1(a)). A half-mirror is used to co-locate the dots' center of projection with the camera's focal point. In doing so, we ensure that all scene points illuminated by the projector are also seen by the camera and their locations in the acquired image are known. This avoids the correspondence and missing-part problems inherent to multi-viewpoint systems. The set of projected dots is distributed sparsely over the camera's field of view both to avoid overlap of the defocused dots and to simplify their removal from the image. While the sparsity of the dots limits the spatial resolution of the depth estimates, we couple the sparse depth estimates with a simple color segmentation algorithm to achieve a dense depth map with sharp object boundaries (see Fig. 1(b)). Such an approximate depth map is adequate since the refocusing of most scenes only requires the scene regions to be well segmented, with the proper ordering of depth.



(a) Acquired Image (b) Dots Removed (c) Sparse Depth (d) Color Segmentation (e) Merged Segmentation

(f) Dense Depth (g) Refocused (Far) (h) Refocused (Mid) (i) Refocused (Near) (j) Refocused (Near, Light 2)

Figure 2: *The steps involved in the refocusing method. (a) Acquired image; (b) image after removal of the projected dots; (c) sparse depth map estimated from the removed dots; (d) color over-segmentation of the dot-removed image in (b); (e) merging of segmented regions using the sparse depth map in (c); (f) depth map after boundary refinement using a matting algorithm; (g-i) refocused images with different depths of field; and (j) refocused image for an image taken with new lighting.*

We have also developed a refocusing algorithm which considers partial occlusions at object boundaries. In particular, our algorithm defocuses image points by respecting visibility changes for different points on a large aperture lens, and by more accurately mixing foreground and background pixels in the defocus computation. The algorithm is used with the computed depth map to refocus either the original image (see Figs. 1(c) and (d)), or an image taken immediately before or after under different lighting (see Fig. 1(e)).

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, we give an overview of our system. In Section 4, we provide a radiometric and geometric analysis of defocused projected dots. In Sections 5 and 6, we present the algorithms for depth computation and refocusing, respectively. In Section 7, we present results for a variety of images and videos including human portraits and live action. Finally, in Section 8, we discuss the limitations of our method.

2 Related Work

We review the relevant related work, dividing it into two categories: prior work on depth estimation and prior work on image refocusing.

Depth Estimation: We use an active illumination method for depth estimation from a single image. Passive approaches for recovering depth from a single image, such as shape from shading and texture, cannot handle depth discontinuities, which play a crucial role in refocusing. Other passive methods such as stereo and structure from motion estimate depth from multiple views using triangulation. Apart from the inherent problem of establishing correspondence, these methods cannot guarantee depth estimates for all points in a single image because of partial occlusions.

Structured light methods (see [Salvi et al. 2004] for a review) solve the correspondence problem by projecting light patterns on the scene. These approaches also compute depth based on triangulation and hence cannot estimate depth within partially occluded regions. Furthermore, the projected light patterns are often too complex to remove from the acquired images [Proesmans and Van Gool 1997].

Methods based on camera focus and defocus avoid correspondence computations and are not as adversely affected by partial occlusions. Depth from focus techniques (e.g., [Nayar and Nakagawa 1994; Asada et al. 1998]) capture a set of images under different focus settings, and depth is estimated using a focus operator. These approaches cannot deal with dynamic scenes, as they need to acquire a sequence of images (about 10-12) while the scene remains stationary. In contrast, depth from defocus methods (e.g., [Pentland 1987; Subbarao and Surya 1994]) require processing only a few images (about 2-3) and depth is estimated by measuring relative blur.

Like stereo and structure from motion, depth from focus/defocus cannot produce depth estimates for textureless scene regions. To address this limitation, some methods [Girod and Scherock 1989; Girod and Adelson 1990; Nayar et al. 1996] use active illumination to project a texture onto the scene. We have adopted this approach in our system. Our depth estimation method is most closely related to [Girod and Adelson 1990], where a pattern is projected and its defocus is used to estimate scene depth from a single image, albeit with blurred boundaries. The primary objective of this previous work is to determine whether the computed depths lies in front of, or behind, the focal plane. This is done by projecting a pattern consisting of asymmetric shapes. The authors suggest that their patterns can be removed from the captured image using low-pass filtering. However, such an approach will not work for textured scenes as it will significantly degrade the quality of the image. In contrast, we show that by projecting dots on the scene and using ratios of the acquired image with a set of calibration images, the dots can be removed even for textured scenes, without any noticeable loss of image quality. We also show that the projection of sparse dots allows for control of the intensity falloff within the depth range of interest. By minimizing the intensity falloff, we avoid over- and under-exposure of the defocused dots and hence improve the robustness of depth estimation as well as dot removal. Furthermore, we show how a complete depth map with sharp boundaries can be obtained from the sparse dot depths by applying a depth-based segmentation algorithm to the dot-removed image.

Zhang and Nayar [2006] recently proposed a method that captures a set of images (around 20) of a still scene while it is lit by a shift-

ing light pattern. The depth of a pixel is computed by analyzing the temporal variation of its brightness due to defocus. The computed depth map is "image-complete" and can be used for refocusing. Our work is also closely related to this previous work, but we compute depth with a single image. Although our depth estimation is not as dense, it is applicable to images and videos of dynamic scenes. Our depth recovery is similar in spirit to the work of Hoiem et al. [2005] on automatically constructing rough scene structure from a single image. As in our method, an over-segmented image is computed which is subsequently merged into geometrically equivalent regions. However, their merging is based on a set of predefined appearance-based classes, while our merging uses sparse depth measurements.

Refocusing: A common approach to refocusing is to acquire a set of differently focused images. In [Rajagopalan and Chaudhuri 1999; Subbarao et al. 1995], depth from defocus is used to estimate the spatially varying blur of the scene and then compute an all-focused image. This image can be refocused using the computed blur. A similar approach is used in [McGuire et al. 2005] where two synchronized video sequences acquired under different focus settings are used to render a new video in which the focus setting can be controlled. These methods are passive but the range of refocusing effects that can be achieved is limited because of the small number of acquired images. Other methods compute an all-focused image from a larger set of acquired images [Burt and Kolczynski 1993; Nayar and Nakagawa 1994; Haeberli 1994; Krishnan and Ahuja 1996; Agarwala et al. 2004]. Due to the large number of images needed, these methods are difficult to use in the case of dynamic scenes.

A different approach to refocusing is to measure the light field associated with a scene. In this case, the measured rays can be combined to simulate new depth of field settings without explicitly computing depth. Levoy and Hanrahan [1996] compute a light field from a large number of images (between 256 and 4096) and use it to simulate synthetic camera apertures. This idea was further extended in [Isaksen et al. 2000] and [Levoy et al. 2004]. The drawback of this approach is that it either requires the sequential capture of a large number of images (which is not possible for dynamic scenes) or the use of a large camera array.

A novel approach to refocusing is to use integral photography, where the light field is measured using an array of lenses placed either behind the camera lens [Ng et al. 2005] or in front of it [Georgiev et al. 2006]. As with a camera array, the measured rays can be combined to achieve refocusing. The advantage of this approach over ours is that it is passive – no projected illumination is used. On the other hand, it comes with a significant reduction in image resolution as a single image detector is used to simultaneously capture a large number of images of the scene. For instance, with the system in [Ng et al. 2005] the final refocused image is 292×292 pixels when a detector with 4000×4000 pixels is used. In contrast, our active method produces a refocused image at the same resolution as the acquired image.

The problem of producing a limited depth of field image of a scene with known geometry has a long history [Cook et al. 1984; Potmesil and Chakravarty 1981; Rokita 1996]. However, most of these previous methods were designed to work on synthetic scenes with complete 3D models. In our case, we do not have a complete 3D model of the scene but rather an image-complete depth map. In the absence of a 3D model, the visibility effects at object boundaries are not well-defined. There are commercially available tools, such as Photoshop's lens blur feature and IrisFilter [Sakurai 2004], that can refocus an image with a user-provided depth map. As we will see in Section 6, these tools produce undesirable artifacts when the refocusing is done with a large aperture. We have developed an algorithm that uses a visibility change model for object boundaries to produce refocusing results of higher quality.

3 Overview of the Method

This section presents an overview of our refocusing method, which is illustrated in Fig. 2. The processing pipeline consists of the following main steps.

Calibration: Our depth estimation method is based on the defocus analysis of a grid of dots projected onto the scene. Before acquisition, the dots are projected onto a calibration board, which is swept through the working volume of the imaging system. The appearance of the board under uniform projected light is also recorded. This is a one-time calibration procedure – the calibration images are used to process all scene images taken with the same system parameters.

Sparse depth map from projection defocus: Given an image of the scene lit by the dots (Fig. 2(a)), the degree of defocus for each dot is estimated by comparing its blur to the dots in the calibration images. This comparison is done by taking the appropriate ratios of brightnesses in the acquired image with the calibration images. This results in the removal of dots from the acquired image (Fig. 2(b)) as well as the estimation of the dot depths (Fig. 2(c)).

Depth map completion using segmentation: The dot-removed image is segmented into a large number of small regions of nearly uniform color (Fig. 2(d)). Next, the sparse depth map previously computed is used to fit a surface to each one of the color segments, which are then merged according to depth similarity (Fig. 2(e)). Precise depth estimation near discontinuities is obtained using a matting technique (Fig. 2(f)).

Image refocusing: Finally, the image may be refocused with different focal plane and aperture settings, by convolving each pixel with a blur kernel whose size is proportional to the depth of the pixel. Realistic depth of field renderings are achieved by taking into account partial occlusions at object boundaries (Figs. 2(g-j)).

4 Projection Dot Defocus Analysis

We now describe the camera-projector system we have used to acquire scene images with projected dots and present an analysis of the defocus function associated with a projected dot. The results of our analysis are used to choose the system parameters so as to avoid under- and over- exposure of the projected dots.

4.1 System Design

Figure 3 shows our basic setup. We use a camera and projector that are co-located by means of a half-silvered mirror. Consequently, the scene is imaged onto the camera via the same optical path used to project the grid of dots onto the scene. This setup has the advantage of avoiding shadows, occlusions and foreshortening asymmetries between the camera's and projector's viewpoints. In addition, the locations of all the dots are known in the camera image, obviating the need to solve a correspondence problem.

The illumination pattern we use is composed of small square dots of brightness B_h regularly spaced over a background of brightness B_l, where $B_h > B_l$. Note that we need $B_l > 0$ in order to recover the appearance of the surface regions which are not illuminated by the dots. The separation between dots is such that it prevents overlapping of adjacent dots for the maximum defocus level.

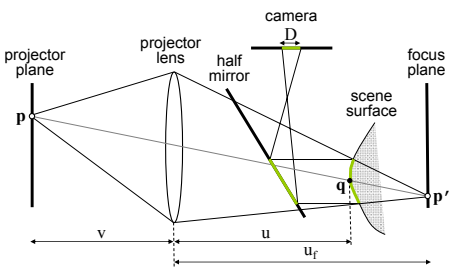


Figure 3: *System used to acquire images for refocusing.*

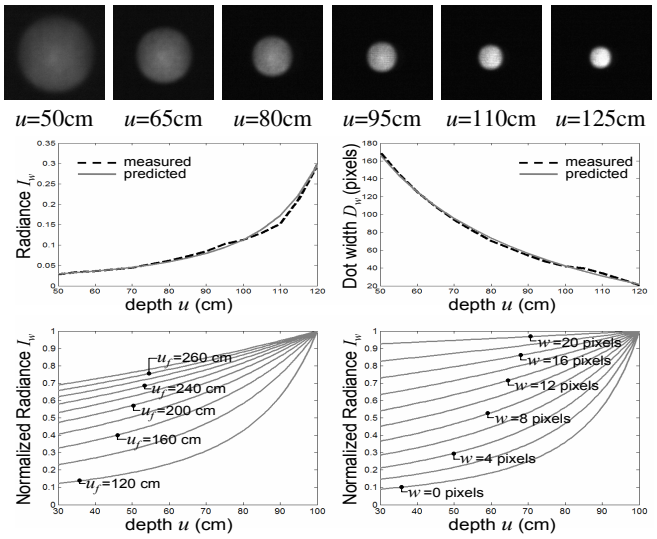


Figure 4: *Geometric and radiometric properties of projected dots. (top) Camera images of a square dot of* 3×3 *pixels projected onto different depths. (center) The dot width* D_w *and radiance* I_w *were measured from the images and compared to the values predicted by our models. (bottom) The radiance variation of a projected dot, within a chosen working range of the system, may be controlled by changing the parameters of the setup, such as the distance* u_f *of the focal plane or the width w of the dots.*

4.2 Defocus Geometry and Radiometry

Consider again the projector-camera system illustrated in Fig. 3. The projector is assumed to have a narrow depth of field (wide aperture) while the camera is assumed to have a wide depth of field (small aperture). A point light source at $\mathbf{p}$ on the projector plane is focused on a point $\mathbf{p}'$ in the scene. If $\mathbf{p}$ is projected onto a surface point $\mathbf{q}$ which lies in front of the focus plane, it produces a circular patch (*blur circle*) of uniform brightness in the camera image.[1] Although the shape of illuminated patch on the surface around $\mathbf{q}$ is a function of the local surface geometry, the shape of the patch as seen by the co-located camera remains circular. Using the lens law, the diameter D of the blur circle on the camera's image plane can be written as

$$D = \pm 2 f_c r \left(\frac{1}{u} - \frac{1}{u_f} \right), \tag{1}$$

where f_c is the camera focal length, r is the radius of the projector lens, u_f is the distance of the focal plane from the lens, and u is the distance of the surface from the lens. The "+" sign holds when the projector is focused behind the scene ($u \leq u_f$), and the "−" sign holds when the projector is focused in front of the scene ($u > u_f$).

The radiance I of the imaged blur circle is proportional to the irradiance E of the surface patch at $\mathbf{q}$. In Appendix A we show that E is proportional to the ratio between the light energy from the source at $\mathbf{p}$ that passes through the projector lens and the area of the surface patch that is illuminated by the source. Consequently, the radiance of the blur circle can be written as

$$I \propto \frac{\delta p}{\left(1 - u/u_f\right)^2}, \tag{2}$$

where δp is the area of the light source centered at $\mathbf{p}$. In practice, projectors cannot produce infinitesimally small light sources. If instead, we project a square dot of size $w \times w$ (in the projector plane), the dot width D_w and the radiance I_w of the blurred patch in the image plane are

$$D_w = D + w\frac{f_c}{v}, \qquad I_w \propto \frac{w^2}{\left(\pm\left(1 - \frac{u}{u_f}\right) + u\frac{w}{vr}\right)^2}, \tag{3}$$

where v is the distance of the projector plane from the lens.[2] Again, we refer the reader to the Appendix A for details.

The above models are approximations as they assume the pillbox blur function and hence do not account for the intensity falloff within the blur circle due to diffraction effects and lens aberrations. Nevertheless, we have experimentally verified that the models are adequate for selecting the parameters of imaging system. Using a high resolution, linear camera we acquired the appearance of a defocused 3×3 square patch projected onto a white board at different depths. A few of these images are shown in Figure 4(top). The width D_w and radiance I_w of these blur circles were manually measured from the images and used as input to the models in Eq. 3 to estimate the parameters u_f and w of the setup (r, f_c and v were estimated separately). We found the estimated values of u_f and w to be in good agreement with their known real values. Figure 4 (center) compares the measured values of D_w and I_w with ones obtained from the models in Eq. 3, using the estimated values for u_f and w.

4.3 Controlling Dynamic Range of Projected Dots

A key problem with using active illumination is that scene irradiance falls off with the inverse square of the distance. As a result, the operable range of the imaging system tends to be very limited. For example, flash images often suffer from saturation of nearby objects and weak illumination of distant ones. In our system, we avoid this by selecting appropriate values for the system parameters. From Eq. 3 we see that both the width D_w of the blur circle and its radiance I_w can be controlled through the distance u_f of the focal plane from the lens, the radius r of the lens and size w of the projected dots. Fig. 4(bottom) shows the effects of changing u_f and w within physically feasible ranges. Larger values of u_f tend to decrease the falloff of the dot brightness within the working distance. Similar effects may be observed by increasing the projected dot size w. However, note that increasing u_f and w results in larger values of D_w, requiring the spacing between neighboring dots to be increased to avoid overlap. Therefore, in practice, there exists a tradeoff between the spatial resolution and the dynamic range of the projected dot pattern.

As mentioned earlier, our illumination pattern is composed of square dots of brightness B_h, regularly spaced over a background of brightness B_l, where $B_h > B_l$. From Eq. 3 it can be seen that the irradiance of the dot decreases with the depth u when the projector is focused in front of the scene. In contrast, the irradiance of the background surrounding a dot always decreases with the depth, independent of where the projector is focused. As a consequence, when the projector is focused behind the scene, the contrast between the projected dots and the background is greatest. Therefore, in all our experiments, the projector was focused behind the scene.

5 Dot Removal and Depth Estimation

We now present the details of our algorithm for removing the dots, measuring the depths of the dots, and estimating a complete depth map from a single acquired image. For clarity we break the algorithm down into a number of simple steps.

5.1 Calibration

Assume that we are given a desired working range. Using the models presented in the previous section, we select the depth u_f of the

[1] For our analysis here, we assume the blur function to be a pillbox. This analysis is only used to select system parameters and hence a precise blur model is not required. Our depth estimation is done using a set of calibration images that accurately capture the blur function of the projector used in our system.

[2] The square patch is assumed to be small. Hence, the defocused patch measured by the camera remains a circular one with more or less uniform brightness (see Fig. 4(top)).

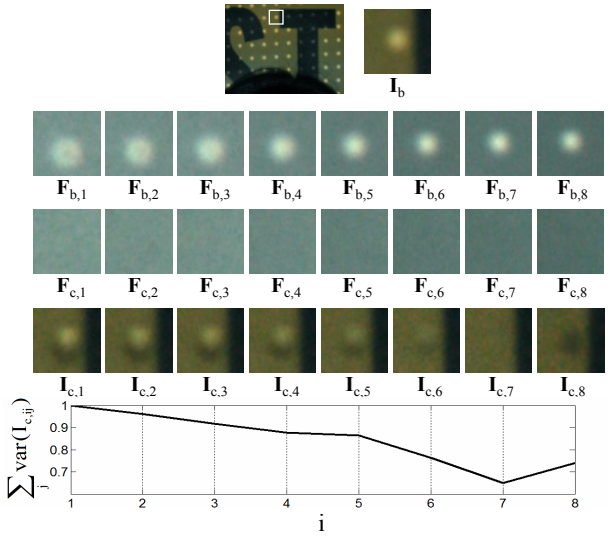


Figure 5: *Depth from defocus of projected dots. (top) Magnified region of the scene in Fig. 2, and the image patch $\mathbf{I}_b$ for which we wish to compute depth. (center) The calibration images. A white board is moved through the working range and its image (only the patch corresponding to a single dot is shown here) is acquired when it is lit by the dot ($\mathbf{F}_{b,i}$) and by uniform illumination ($\mathbf{F}_{c,i}$). Computed image patches $\mathbf{I}_{c,i}$ that represent how $\mathbf{I}_b$ would appear under uniform illumination if it had the depth corresponding to $\{\mathbf{F}_{b,i}, \mathbf{F}_{c,i}\}$. (bottom) The correspondence between the calibration patches and the acquired patch can be determined by finding the i that minimizes the variance of $\mathbf{I}_{c,i}$. This plot of the variance shows that the depth of the patch $\mathbf{I}_b$ is approximately the same as the depth of the calibration pair $\{\mathbf{F}_{b,7}, \mathbf{F}_{c,7}\}$.*

projector focal plane, the spacing between the dots, the dot size w, the brightness B_h of the projected dots, and the background brightness B_l.

With these parameters fixed, we acquire a series of calibration images in which the grid of dots of brightness B_h over a background of brightness B_l is projected onto a white board perpendicular to the camera's optical axis. The board is placed at the back of the working range and then stepped forward, with one calibration image acquired at each step. We then acquire a second series of calibration images by repeating this process, where the grid of projected dots is replaced by light from the projector of uniform brightness B_l.

5.2 Dot Removal and Dot Depth Estimation

Let $\mathbf{I}_b$ be an image patch of size $p \times p$ pixels containing one of the projected dots. The image patch is such that the blurred dot lies at its center and its width p completely contains the blurred dot, i.e., $p > D_w$. For each patch $\mathbf{I}_b$, there are N image patches $\mathbf{F}_{b,1}, \ldots, \mathbf{F}_{b,N}$ of the blurred dots acquired from the calibration board images as mentioned above. In addition, there are N image patches $\mathbf{F}_{c,1}, \ldots, \mathbf{F}_{c,N}$ of the board lit by uniform illumination (see Fig. 5). The subscript i on both $\mathbf{F}_{b,i}$ and $\mathbf{F}_{c,i}$ indicates that the corresponding images have been acquired when the calibration board was placed at a distance u_i from the projector. Our goal is to estimate the depth of the scene point imaged in $\mathbf{I}_b$ by comparing it with the corresponding patches captured in the calibration images.

Consider a patch $\mathbf{I}_b$ that corresponds to a scene region that is textureless. Let us assume that the actual depth u_x of the patch is known. Then, the following relation holds true for each and every point (pixel) in the image patch:

$$\frac{\mathbf{F}_{b,x}}{\mathbf{F}_{c,x}} = \frac{\mathbf{I}_b}{\mathbf{I}_{c,x}}, \tag{4}$$

Figure 6: *Depth map completion. Starting with an over-segmented image (left), the segments are iteratively merged based on color, texture and depth using a greedy algorithm. Note how the number of segments N_s decreases with the iterations (left to right).*

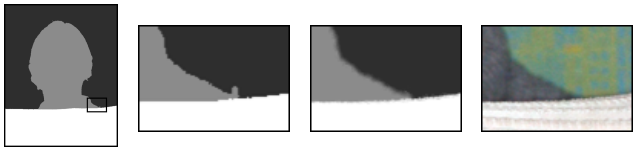


Figure 7: *Refinement of depth discontinuities. (a) Complete depth map obtained after segmentation and merging. The depth discontinuities have noisy artifacts due to limitations of the segmentation. (b) Magnified region of the depth map. (c) Refined depth map obtained by using matting. (d) Acquired image with dots removed.*

where $\mathbf{I}_{c,x}$ is the scene image (for the patch under consideration) one would obtain if the scene were lit by the projector with uniform illumination of brightness B_l.

Eq. 4 can be used to compute the unknown depth u_x of each patch in the following manner. Given $\mathbf{I}_b$, we take the N pairs of calibration images $\{\mathbf{F}_{b,i}, \mathbf{F}_{c,i}\}$ and compute the corresponding image patches $\mathbf{I}_{c,i}$ (see Figure 5(center)). For the depth u_i that is closest to u_x, $\mathbf{I}_{c,i}$ should be an image of the scene lit by uniform illumination – it should not include the effects of the blurred dot. Therefore, we find i by simply finding the $\mathbf{I}_{c,i}$ that has lowest variance of brightness values (see Fig. 5(bottom)).

In order to deal with textured surfaces (texture by itself introduces brightness variation), each patch $\mathbf{I}_{c,i}$ is partitioned into subregions using the unsupervised algorithm described in [Figueiredo and Jain 2002], and a variance is computed for each subregion. Then, if $\mathbf{I}_{c,i} = \sum_j^{N_{ri}} \mathbf{I}_{c,ij}$, where N_{ri} is the number of subregions in $\mathbf{I}_{c,i}$, the depth u_x is determined as

$$u_x \approx u_i \;\; | \;\; \arg\min_i \left\{ \sum_{j=1}^{N_{ri}} \mathrm{var}(\mathbf{I}_{c,ij}) \right\}, \tag{5}$$

where $\mathrm{var}(\cdot)$ is the variance operator. By repeating the above process for all the projected dots, we obtain an image with all the dots removed like the one in Fig. 2(b) and a sparse depth map as in Fig. 2(c). The depth resolution for a dot depends on the number of depths used to acquire the calibration images. In our implementation, we perform a refinement of the computed dot depths. This is done by interpolating the calibration images closest to a computed dot depth and using the above variance test to find the final u_x, which may lie in between the discrete depths associated with the calibration images.

5.3 Depth Map Completion Using Segmentation

Thus far, we have estimated depths at a set of regularly spaced pixels in the acquired image. To achieve our goal of refocusing the image, we need to have depths at all pixels. To interpolate the dot depths and obtain a complete depth map we use a segmentation-based approach. First, we apply the Mean-Shift algorithm [Comaniciu and Meer 2002] to obtain an over-segmentation of the dot-removed image.

Each segment in the over-segmented image is characterized by three distinct features: color (**c**), texture (**t**) and depth (**d**). Color

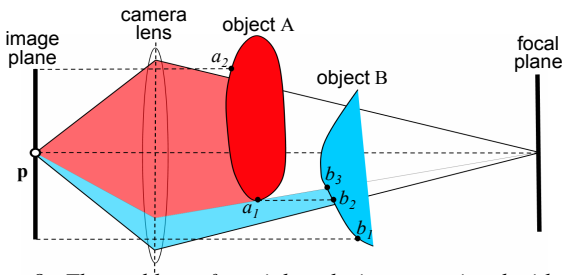


Figure 8: *The problem of partial occlusions associated with rendering a refocused image with a wide aperture, given a single all-focused image and a depth map.*

and texture are directly measurable from the dot-removed image, while depth is determined from the computed sparse depth map.[3] For segments containing several pixels with known depth, we assign the median of the depths to the segment. The use of the median enables us to remove outliers in the sparse depth map. On the other hand, a segment that does not contain any pixel with known depth is described by just its color and texture. Next, we use a greedy algorithm to group the image segments. The algorithm iteratively merges the two most similar neighboring segments and re-computes the features of the new merged segment.

To measure similarity between two segments S_i and S_j in iteration k, we use the following metric:[4]

$$\mathrm{sim}(S_i,S_j) = \lambda_c(k)\mathrm{d}(\mathbf{c}_i,\mathbf{c}_j) + \lambda_t(k)\mathrm{d}(\mathbf{t}_i,\mathbf{t}_j) + \lambda_d(k)\mathrm{d}(\mathbf{d}_i,\mathbf{d}_j), \quad (6)$$

where $d(\cdot,\cdot)$ is the Euclidean distance, and the parameters $\lambda_c(k)$, $\lambda_t(k)$ and $\lambda_d(k)$ determine the relative contributions of the three features. To discourage the merging of large regions with different depths, $\lambda_d(k)$ is set to a straight line function with positive slope, while $\lambda_t(k)$ and $\lambda_c(k)$ are set to straight line functions with negative slopes. In all the cases, the value of the slope is inversely proportional to the number of initial segments that need to be merged. Fig. 6 illustrates the evolution of the segmentation process for the acquired image in Fig. 2(a).

As can be seen in Fig. 7(b), the merged image includes noise around the depth discontinuities. To reduce these artifacts we automatically extract a trimap which separates the region around a depth discontinuity into a foreground F, a background G and an unknown layer U. Using the matting algorithm proposed by Wang and Cohen [2005] we compute an *alpha-map*, which assigns a probability p_F of belonging to the foreground to each of the pixels in U. The probability p_F is then used to estimate the depth of the pixels in U as a linear combination of the depth of the closest pixel in F and the depth of the closest pixel in G. The result of this refinement is shown in Fig. 7(c). By comparing with the original image in Fig. 7(d), we see that the edge artifacts are removed and the transition between the different depths is smooth.

6 Algorithm for Realistic Refocusing

In this section we present a refocusing algorithm that uses an image taken with a wide depth of field camera and its depth map to simulate novel images of the scene with different depths of field. The simulated depth of field may be controlled in terms of size of the lens aperture and the location of the focal plane of the lens. To render realistic depth of field effects it is important to consider the following two issues. First, for an object boundary, different parts of the lens may "see" different views due to partial occlusions. Second, in real images, pixels at depth discontinuities may receive contributions from the foreground and the background. Our refocusing algorithm addresses both these issues.

[3] Texture is represented by derivatives of oriented Gaussian filters.

[4] In the case of videos, the similarity measure includes a temporal constraint – the difference between the indices of the frames in which the segments appear.

Figure 9: *Realistic refocusing. (a) Original all-focused image. (b) Refocusing result obtained using the proposed algorithm. The virtual focal plane is placed on the background of the scene. (c-e) Magnified regions showing refocusing results for (c) the proposed algorithm, (d) Photoshop's lens blur tool, (e) the IrisFilter tool. (f) Real image taken with a Canon camera and a wide aperture.*

Partial occlusions: Consider the scenario illustrated in Fig. 8; we want to compute the irradiance of an image pixel **p** which receives light from a lens with a large aperture, focused behind the scene. Two objects A and B are in the field of view of **p**, where A is located in front of B. The total light energy received by **p** is the sum of the contributions of all the light rays from the lens. The contributions of these rays can be determined by tracing the rays from the lens to points on the surfaces of A and B. This computation is simple and can be done when the complete geometry of the scene is given.

In our case, however, we are given a single, narrow-aperture view of the scene and the corresponding depth map. There could be regions of the objects A and B that contribute to the irradiance of pixel **p** in the refocused image, that are not captured in the acquired image. This is illustrated in Fig. 8, where the acquired image is assumed to be an orthographic view of the scene (dotted horizontal lines). In this case, although we need the radiances of the points on object B that lie between b_2 and b_3, they are not included in the acquired image. We recreate such missing regions by detecting discontinuities in our depth map and extending the occluded surface using texture synthesis.

Foreground/background transitions: Note that the ray-tracing based method we use to consider the partial visibility assumes that each image pixel belongs either to the background or to the foreground, i.e., it assumes abrupt depth maps. However, since we have used matting to refine the depth estimation at object boundaries, our depth map is not abrupt and changes smoothly from foreground to background at depth discontinuities. To handle these smooth depth changes, we blend a foreground focused image with a background focused image within the boundary region.

In particular, let us say we wish to refocus an image with three types of regions: a region F (foreground) with depth d_F, a region G (background) with depth d_G, and a region C (boundary) with a depth that smoothly changes from d_F to d_G. Our matting step gives us the corresponding alpha-map **A**, which represents the probability of each pixel of belonging to the foreground. Given this input data, we then compute two different refocused images using the technique described to model the partial occlusions. First we compute $\mathbf{R}_{C\in F}$ where we have assigned a depth d_F to all the points in C. The second refocused image, $\mathbf{R}_{C\in G}$, is computed by assigning a depth d_G to the pixels in C. The final refocused image is computed as

$$\mathbf{R} = \mathbf{R}_{C\in F} * \mathbf{A} + \mathbf{R}_{C\in G} * (\mathbf{1} - \mathbf{A}) \quad (7)$$

where **1** is a matrix of ones of the same size as **A**, and $*$ denotes element-wise multiplication.

The proposed refocusing algorithm produces better results than existing approaches, especially when the virtual focal plane is located

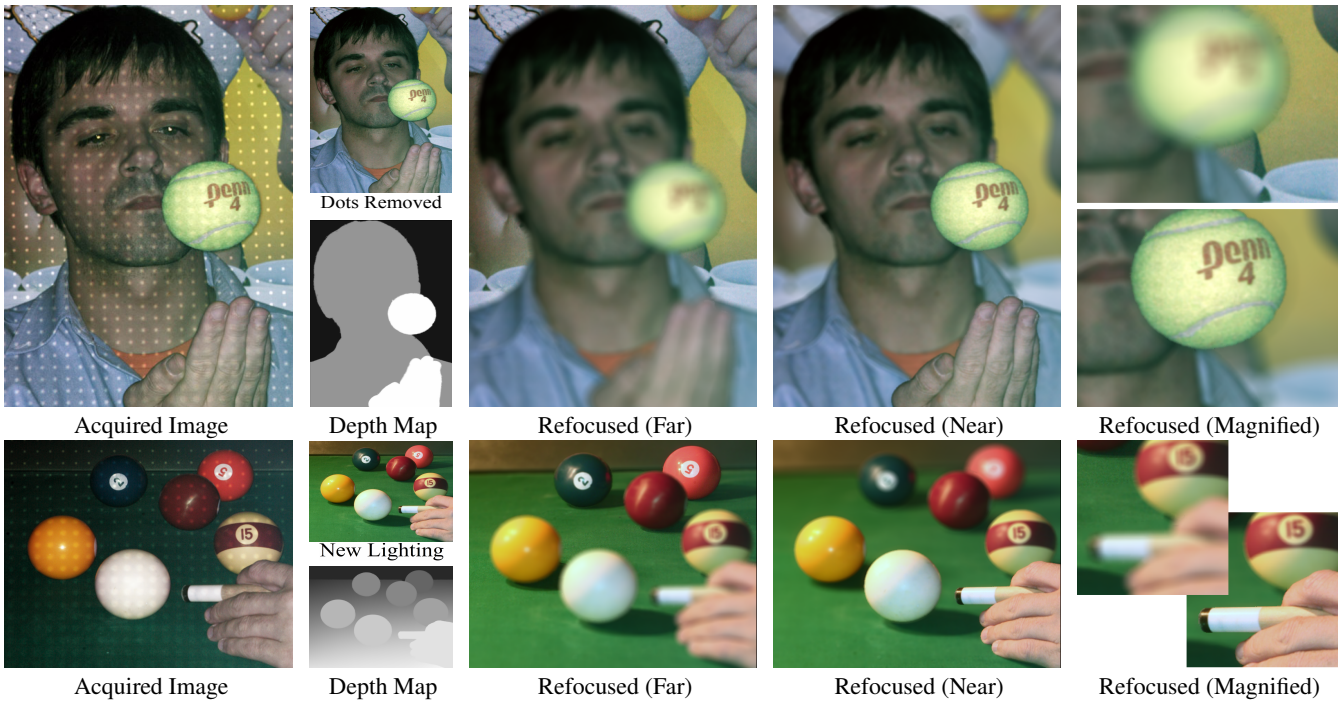


Figure 10: *Refocusing results for two different scenes. In each case, we show (from left to right), the image acquired by illuminating the scene with the dot pattern; the image obtained after dot removal or taken under new illumination; the computed depth map; two refocusing results where the focal plane is place at the back and in the front of the scene; and two magnified regions of the refocused images. In the case of the pool table (as well as the examples shown in Figs. 1 and 2), the sparse depth map computed from the acquired image is used to compute a complete depth map corresponding to a second image taken with different lighting. In this case, refocusing is applied to the second image.*

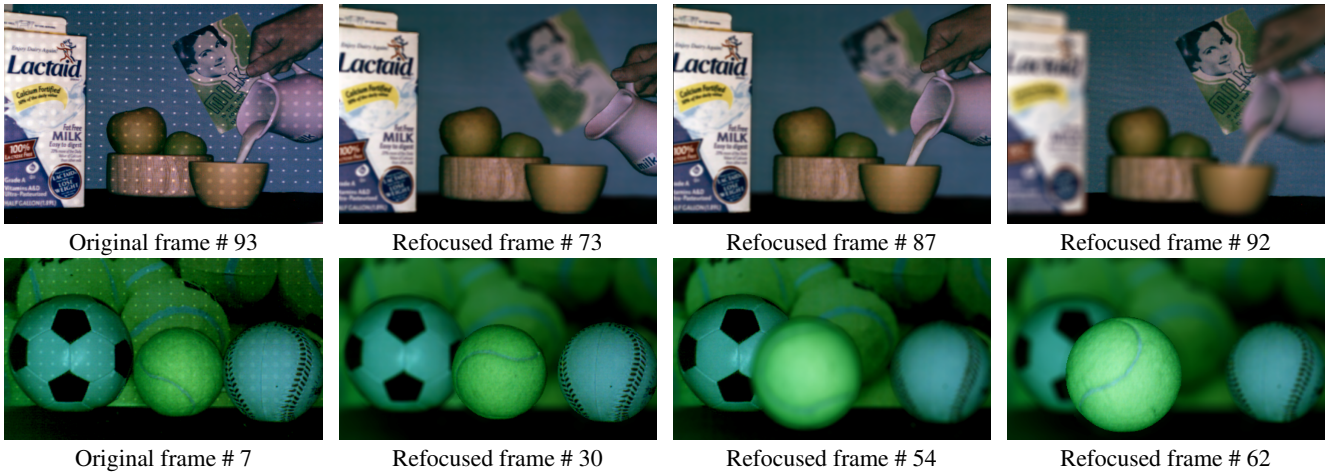


Figure 11: *Refocusing of videos of dynamic scenes. In each case, one of the acquired frames is shown on the left and three differently refocused frames are shown on the right.*

behind the scene. In Fig. 9 we compare refocusing results obtained using our algorithm to results generated using Photoshop's lens blur tool and the IrisFilter tool [Sakurai 2004]. The magnified images show that our result in Fig. 9(c) is close in appearance to the real image in Fig. 9(f), while the previous methods produce artifacts at the boundary of the foreground object.

7 Results

The proposed method has been used to refocus both single images as well as videos of dynamic scenes. The single images were captured with a Canon EOS 20D camera (with 1728×1152 pixels) and the videos were captured with a Prosilica CV640 camera (with 659×493 pixels). The dot illumination pattern was generated using a Sanyo PLC XT11 digital projector (with 1024×768 pixels) that is co-located with the camera using a half-mirror. Dot patterns with resolution (number of dots) ranging from 500 to 1000 dots were used. The working range for our experiments varied from 0.5 meters to 3 meters, although larger ranges can be handled by using a more powerful projector.

Fig. 10 shows single-image refocusing results for two scenes. In the first example, we see that the dot-removed image is of high quality and the depth map has four distinct depth layers – the ball, the hand, the face and body of the person, and the background. The refocused images reveal the quality of the computed results. In the first refocused image the background is in focus, while in the second image the tennis ball and the hand are brought into focus. The

second example in Fig. 10 shows a pool table. In this case, different (but constant) depths are assigned to the objects (balls, hand and pool cue) on the table. However, the table itself lies on an inclined plane. Since the color of the table is uniform, it is determined to be a single region. The sparse depths within this region are interpolated to obtain an inclined surface with the proper depth gradient.[5]

The pool-table scene, as well as the ones in Figs. 1 and 2, can be viewed as quasi-static scenes. They include humans in them and humans find it hard to remain perfectly still – it is difficult to capture two consecutive images without the scene changing. When the scene changes are small, our approach can be used to capture a second image of the scene with a different illumination (say, studio lighting) and use the sparse depth map computed from the first image to segment and compute a complete depth map corresponding to the second image. Then, the second image can be refocused as desired. This approach was taken to produce the refocused images of the pool table on the right of Fig. 10 as well as the refocused images shown in Figs. 1(e) and 2(j).

Fig. 11 shows refocusing results for two videos of dynamic scenes. In the first example, the video (including 150 frames captured at 24 fps) is of milk being poured from a jar into a cup. Although milk exhibits subsurface scattering effects, we see that the projected dots are clearly visible in the acquired frame (#93) shown on the left. As a result, even for this complex scene, we are able to recover a depth map that is of adequate quality to realistically refocus the sequence. In the refocused video, the depth of field is continuously varied while the scene changes. In our last example, we show the refocusing of the video (with 100 frames) of a scene that includes a soccer ball, a tennis ball and a baseball. The tennis balls in the background are actually a part of a picture on a flat poster. The real tennis ball rolls towards the camera and refocusing is used to vary the distance of the simulated focal plane as the ball approaches the camera. In this case, to reduce motion blur produced by the rolling ball (which can lead to erroneous depth estimates), the camera was operated at a higher speed of 66 fps.

8 Limitations of the Method

Although the proposed method works well for a wide variety of scenes, it suffers from the following limitations. (a) It uses active illumination and hence is more appropriate for indoor scenes (or a studio) rather than outdoor scenes with strong sunlight. (b) The method requires a reasonable over-segmentation of the image to start with, where scene regions with distinct depth are assigned to different segments. (c) Since the projected light pattern is sparse, fine depth details in the scene cannot be captured. (d) Translucent objects that exhibit subsurface scattering can cause the projected pattern to appear defocused even when it is not. For such objects, the estimated dot depths can have large errors. (e) When the dots are projected onto very dark and/or highly inclined surfaces (in our experience, greater than 70° with respect to the optical axis) the blurred dots can be too weak to detect.

Fig. 12 shows magnified regions from two scenes shown in Figs. 2 and Fig. 10 that highlight the limitations of the method. In the case of the pool table, the points a and b shown in the depth map should have the same depth, but have different depths due to errors in the depth estimation. This leads to subtle refocusing errors (the ball is infocus, while the table is not). In the second example, the holes between the hairs of the person are not precisely segmented and are assigned inaccurate depth estimates due to the sparsity of the projected dots. Again, one can see errors in the refocusing (hair and holes are refocused as if both regions had equal depth). It is worth mentioning that these errors are not easily perceived from the refocused images unless one carefully examines them.

[5] When dealing with inclined surfaces, the algorithm described in section 5.3 is modified slightly. In this case, the corresponding interpolated depth gradient is assigned to the surface as a depth attribute. When comparing a new region to the inclined surface, the similarity metric in Eq. 6 is computed with respect to the depth of the inclined surface closest to the new region.

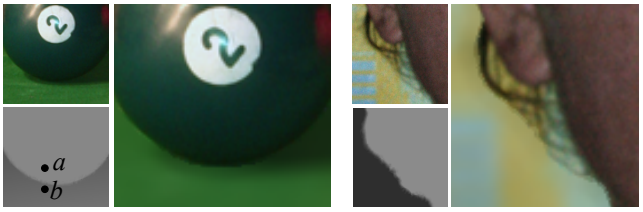


Figure 12: *Examples that show the limitations of the method. In each example, we show a region of the original image on the top left, the depth map on the bottom left and the refocusing on the right. In the case of the pool table, the ball and the table are assigned different depths due to errors in depth estimation. In the second case, the holes between the hairs of the person are assigned incorrect depths due to segmentation errors.*

9 Conclusions

We have developed a simple technique for refocusing a scene with the acquisition of a single image. The method can be used to refocus images as well as videos of dynamic scenes. The main limitations of the method arise from the sparsity of the depth estimation and errors in the initial segmentation of the image. Despite these limitations, the method is applicable to a wide variety of scenes as evidence by our experimental results. We are currently exploring ways to incorporate the method into digital cameras. This requires the design of new optical elements that can convert the light generated by a camera flash into the dot illumination pattern we use. Since some digital cameras recently introduced in the marketplace have infra-red filters in their color mosaics, we are also exploring the use of an infra-red source for projecting the dot pattern. The use of such a source and camera would obviate the dot removal step of our algorithm and make the depth estimation more robust in the case of highly textured scenes.

A. Radiometry of a Projected Dot

Consider the dot projection system illustrated in Fig. 3. Light energy from a light source of area δp centered at $\mathbf{p}$ is projected by a thin lens of radius r onto a scene patch of area δq centered at $\mathbf{q}$. The projector lens is focused at a point $\mathbf{p}'$ behind the scene. Hence, the patch δq represents a defocused projection of δp. Our goal here is to determine the irradiance of the patch δq. Based on the image irradiance equation derived in [Horn 1986], it can be shown that the power δP emitted from the source δp and falling on the lens is related to the brightness B of the projector as:

$$\delta P = v^{-2} B \pi r^2 \cos\alpha^4 \delta p\,, \tag{8}$$

where α is the angle that the line from $\mathbf{p}$ to $\mathbf{q}$ makes with the optical axis of the projector. The foreshortened area of the patch δq, considered from the viewpoint of the projector, is a circular patch of radius r_q, where $r_q = r(1-u/u_f)$. Therefore, the irradiance of the surface patch δq is

$$E = \frac{\delta P}{\delta q} = \frac{B\cos\beta\cos\alpha^4}{v^2}\,\frac{\delta p}{\left(1-u/u_f\right)^2}\,, \tag{9}$$

where β is the angle that the surface normal at $\mathbf{q}$ makes with the optical axis of the projector.

Let us now consider the scene illuminated by a small squared patch of size $\omega \times \omega$ as depicted in Fig. 13. From simple planar geometry, it can be shown that the foreshortened area $\delta\omega$, considered again

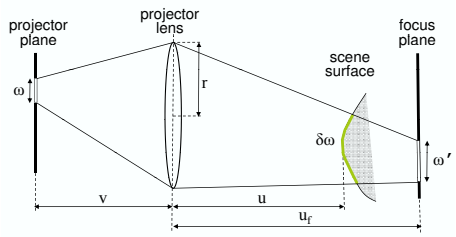


Figure 13: *Geometry of a projected patch.*

from the viewpoint of the projector, is a circular patch of radius r_ω, where $r_\omega = r(1-u/u_f)+\omega u/u_f$. Consequently, the irradiance of the surface patch $\delta\omega$ will be now:

$$E_w = \frac{\delta P}{\delta \omega} = \frac{B\cos\beta\cos\alpha^4}{v^2}\frac{w^2}{\left(1-\frac{u}{u_f}+\frac{uw}{vr}\right)^2}, \qquad (10)$$

where $w = \omega v/u_f$ is the size of the patch expressed in projector pixels.

Acknowledgements

This research was conducted at the Computer Vision Laboratory in the Department of Computer Science at Columbia University. It was funded by the NSF Grants IIS-03-08185 and ITR-03-25867. We thank Irene Plana for posing in the experiments and Anne Fleming for her help in the video narration. We also thank the anonymous reviewers for constructive suggestions on the paper.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. In *Proc. SIGGRAPH*, 294–302.

ASADA, N., FUJIWARA, H., AND MATSUYAMA, T. 1998. Edge and depth from focus. *Int. J. Comput. Vision 26*, 2, 153–163.

BURT, P. J., AND KOLCZYNSKI, R. J. 1993. Enhanced image capture through fusion. In *Proc. ICCV*, 173–182.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5, 603–619.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. In *Proc. SIGGRAPH*, 137–145.

FIGUEIREDO, M. A., AND JAIN, A. K. 2002. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 3, 381–396.

GEORGIEV, T., ZHENG, K. C., CURLESS, B., SALESIN, D., NAYAR, S. K., AND INTWALA, C. 2006. Spatio-angular resolution tradeoff in integral photography. In *Proc. Eurographics Symposium on Rendering*.

GIROD, B., AND ADELSON, E. 1990. System for ascertaining direction of blur in a range-from-defocus camera. In *US Patent No. 4,965,422*.

GIROD, B., AND SCHEROCK, S. 1989. Depth from focus of structured light. In *Proc. SPIE*, vol. 1194, Optics, Illumination, and Image Sensing for Machine Vision.

HAEBERLI, P. 1994. *A multifocus method for controlling depth of field*. Graphica obscura web site. http://www.sgi.com/grafica/.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. In *Proc. SIGGRAPH*, 577–584.

HORN, B. 1986. *Robot Vision*. MIT Press.

ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *Proc. SIGGRAPH*, 297–306.

KRISHNAN, A., AND AHUJA, N. 1996. Panoramic image acquisition. In *Proc. CVPR*, 379.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *Proc. SIGGRAPH*, 31–42.

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

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

NAYAR, S. K., AND NAKAGAWA, Y. 1994. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell. 16*, 8, 824–831.

NAYAR, S. K., WATANABE, M., AND NOGUCHI, M. 1996. Real-time focus range sensor. *IEEE Trans. Pattern Anal. Mach. Intell. 18*, 12, 1186–1198.

NG, R., LEVOY, M., BRDIF, M., DUVAL, G., HOROWITZ, M., AND HANRAHAN, P. 2005. Light field photography with a hand-held plenoptic camera. In *Tech Report CSTR 2005-02, Computer Science, Stanford University*.

PENTLAND, A. P. 1987. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell. 9*, 4, 523–531.

POTMESIL, M., AND CHAKRAVARTY, I. 1981. A lens and aperture camera model for synthetic image generation. In *Proc. SIGGRAPH*, 297–305.

PROESMANS, M., AND VAN GOOL, L. 1997. One-shot active 3d image capture. In *Proceedings SPIE*, vol. 3023, Three-Dimensional Image Capture, 50–61.

RAJAGOPALAN, A. N., AND CHAUDHURI, S. 1999. An mrf model-based approach to simultaneous recovery of depth and restoration from defocused images. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 7, 577–589.

ROKITA, P. 1996. Generating depth-of-field effects in virtual reality applications. *IEEE Computer Graphics and Applications 16*, 2, 18–21.

SAKURAI, R. 2004. *IrisFilter: http://www.reiji.net/*.

SALVI, J., PAGÈS, J., AND BATLLE, J. 2004. Pattern codification strategies in structured light systems. *Pattern Recognition 37*, 4, 827–849.

SUBBARAO, M., AND SURYA, G. 1994. Depth from defocus: A spatial domain approach. *Int. J. Comput. Vision 13*, 271–294.

SUBBARAO, M., WEI, T., AND SURYA, G. 1995. Focused image recovery from two defocused images recorded with different camera settings. *IEEE Trans. Image Processing 4*, 12, 1613–1628.

WANG, J., AND COHEN, M. 2005. An iterative optimization approach for unified image segmentation and matting. In *Proc. ICCV*, 936–943.

ZHANG, L., AND NAYAR, S. 2006. Projection defocus analysis for scene capture and image display. In *Proc. SIGGRAPH*, 907–915.