2013-06-24

Interactive Depth-Aware Effects for Stereo Image Editing

Joshua E. Abbott
Brigham Young University - Provo

Interactive Depth-Aware Effects for Stereo Image Editing

Joshua E. Abbott

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Bryan S. Morse, Chair
William A. Barrett
Christophe Giraud-Carrier

Department of Computer Science

Brigham Young University

July 2013

ABSTRACT

Interactive Depth-Aware Effects for Stereo Image Editing

Joshua E. Abbott
Department of Computer Science, BYU
Master of Science

This thesis introduces methods for adding user-guided depth-aware effects to images captured with a consumer-grade stereo camera with minimal user interaction. In particular, we present methods for highlighted depth-of-field, haze, depth-of-field, and image relighting. Unlike many prior methods for adding such effects, we do not assume prior scene models or require extensive user guidance to create such models, nor do we assume multiple input images. We also do not require specialized camera rigs or other equipment such as light-field camera arrays, active lighting, etc. Instead, we use only an easily portable and affordable consumer-grade stereo camera. The depth is calculated from a stereo image pair using an extended version of PatchMatch Stereo designed to compute not only image disparities but also normals for visible surfaces. We also introduce a pipeline for rendering multiple effects in the order they would occur physically. Each can be added, removed, or adjusted in the pipeline without having to reapply subsequent effects. Individually or in combination, these effects can be used to enhance the sense of depth or structure in images and provide increased artistic control. Our interface also allows editing the stereo pair together in a fashion that preserves stereo consistency, or the effects can be applied to a single image only, thus leveraging the advantages of stereo acquisition even to produce a single photograph.

Keywords: depth-aware effects, stereo, surface normals, normal calculation, quotient image, image-based relighting, depth-of-field, highlighted depth-of-field, haze, disparity, depth pipeline, patchmatch stereo

ACKNOWLEDGMENTS

Thanks go to my adviser, Dr. Bryan Morse because of his patience, love for teaching, as well as a profound knowledge of the subject matter. Additionally, I would like to thank him for his faith in me. I never could have accomplished this without his help. I also thank others in the department that contributed to my success—specifically, Dr. Bill Barrett. He encouraged and inspired me to do something significant. Finally, I would like to thank my beautiful, wife, Brittany. She is the tender flower in my life that made the difficult parts of this project bearable. She was there for every step.

Table of Contents

List of Figures **v**

1 Introduction **1**

1.1 Related Work 4

1.2 Stereo Correspondence 8

2 Interactive Depth-Aware Effects for Stereo Image Editing **12**

2.1 Introduction 13

2.2 Depth Acquisition 15

2.3 Depth-Aware Effects 19

2.3.1 Highlighted Depth of Field 19

2.3.2 Haze 20

2.3.3 Relighting 22

2.3.4 Depth-of-Field 27

2.4 Depth-Aware Pipeline 32

2.5 Results 33

2.6 Conclusion 34

3 Conclusion **42**

References **43**

List of Figures

1.1 Example of haze 2
1.2 Depth-aware effects example 3
1.3 An example of parallax 9
1.4 Matching pixels 10
1.5 Inverse depth or disparity 11

2.1 Depth-aware effects 14
2.2 Highlighted depth-of-field and depth-of-field 15
2.3 Stereo hole filling 17
2.4 Stereo surface normal refinement 18
2.5 Highlighted depth-of-field 20
2.6 Haze 21
2.7 User-adjusted normals 25
2.8 Normal adjustment example 25
2.9 Relighting 26
2.10 Warped secondary view 29
2.11 Single and composite light fields 31
2.12 The depth-aware pipeline 32
2.13 An example of failed stereo correspondence 34
2.14 More highlighted depth-of-field 35
2.15 More haze 36
2.16 More relighting 37

2.17 Tilt-shift depth-of-field 38
2.18 More depth-of-field 39
2.19 Better depth-of-field 40
2.20 Another example of depth-aware pipeline 41

Chapter 1

Introduction

Given the popularity and availability of consumer-grade products associated with 3D capture of images and videos, image-editing tools are adapting traditional photo-editing methods in support of this new format. Because 3D capture is created by capturing two images of the same scene simultaneously, this offers information about the scene depth relative to the camera. With depth information, interesting effects can be introduced in an image that enhance the structure of the objects and draw the viewer's attention to certain areas. Such effects have been used by artists for centuries to enhance the feeling of depth and make objects in the scene appear to "pop" visually. For example, in *The Mona Lisa*, seen in Figure 1.1, a haze effect, commonly known to artists as *aerial perspective*, is used on the scenery behind the lady in order to enhance the feeling of depth in the scene. We refer to such effects as *depth-aware effects* and we focus on four specifically in this work: highlighted depth-of-field, haze, relighting, and depth-of-field. An example of these effects are shown in Figure 1.2.

While all of the aforementioned can be performed manually, the user interaction involved in doing so is more tedious without the depth information. For instance, in order to add depth-aware effects to an image, the user would have to estimate the depth at each pixel. For a rough approximation, large objects could be assumed to lie at the same depth. However, an image with differing levels of depth would require the user to segment the image at each level and then apply an intensity of each effect based on the level. Highlighted depth-of-field [14], in which each pixel in the image receives a particular amount of brightness

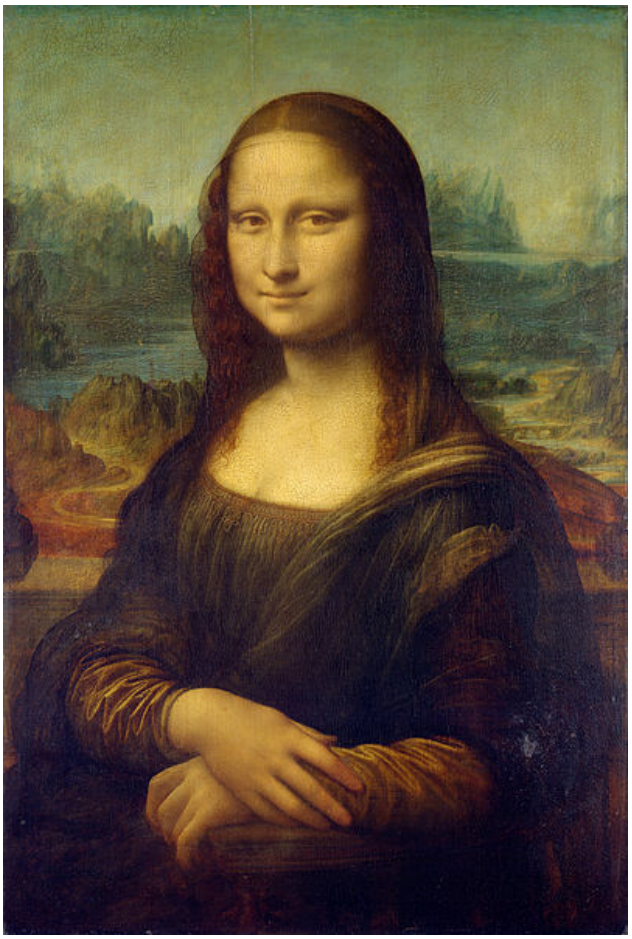


Figure 1.1: Notice the haze behind Mona Lisa to add a feeling of depth.

adjustment based on the distance to the focal point, provides a convincing example. Using traditional image-editing software, the user would need to adjust each segment the correct amount to create the same effect we can create using a click-and-wheel interface to place the focal point and adjust the amount of darkening. Haze and depth-of-field would require similar processes to produce manually. While a coarse segmentation enables the user to add similar effects as Reese and Barrett show [25], our work removes the requirement of segmenting the image into approximate depth layers.

Adding new light sources to an image would not only require segmenting the image into several layers, but these layers would need to be based on geometry rather than the depth alone. Then, each layer would have a varying amount of intensity and new color applied. If the user wanted to change the color or the intensity of the light, that would require changing

Figure 1.2: Depth-aware effects, such as depth-of-field, haze, highlighted depth-of-field, and relighting, are used to enhance the appearance of depth and structure. Left: Original images. Right: Depth-aware effects are added to enhance the structure and depth of the scene. Top-right: Highlighted depth-of-field. Second-right: Haze. Third-right: Depth-of-field. Bottom-right: A combination of all effects, including relighting.

the tint of all layers that the light affects. For multiple lights, the complexity of the task increases as the varying amounts of each light have to be combined properly for each pixel. Adding new lights to an image through our methods is done through a click-to-add interface. Once the lights are added, they can be moved to a new position via dragging. The properties of each light are also changeable and feedback is given to the user as the image updates in real-time with each change.

Our work is not only useful for reducing user tedium, but also for enhancing the three-dimensional structure or depth of objects in the image. Such effects allow artists fine-tuned control over the viewer's attention. Depth-of-field is a well-used example of this in cinematography. The viewer's attention is automatically drawn to objects or subjects that are in focus when this effect is applied. Other effects can have the same cause. Figure 2.2 shows an example of how highlighted depth-of-field can have a similar consequence to that of depth-of-field.

Depth-aware effects can also be useful for making objects appear to "pop" out of the image, as seen in Figure 1.2. This work provides a method for introducing such depth-aware effects in images by using the depth calculated from a stereo pair of images. Specifically, highlighted depth-of-field [14], haze, image relighting, and depth-of-field can be added to the image in a way that enhances the feeling of depth and structure. Additionally, each of these effects can be combined in a physically consistent manner to form a pipeline of effects, as shown in Figure 2.1.

1.1 Related Work

Many have suggested ways for introducing depth-aware effects into photographs by first estimating depth from a single image, but this is a highly under-determined problem and each such approach makes various assumptions about the scene [2, 3, 6, 21, 29]. Some have used a geometric model for the depth obtained through extensive user interaction (still with

various scene assumptions) or a pre-existing model [10, 12, 15]. However, there are many scenes for which there exists no model, nor can such assumptions always be made.

Our methods do not require extensive user interaction to fit a model to the scene [12]. Other than controlling the type and amount of effect desired, the most extensive user interaction we require involves placing an estimated original light source when performing relighting (which is placed by simply clicking on the image or dragging to reposition it).

Others have used modified camera systems or rigs that enable depth-aware effects (e.g., light fields, active lighting approaches, etc.) [8, 14, 17–20, 22, 23, 30, 32]. However, other than the limited-resolution Lytro camera, these are not generally available to the consumer. Active lighting approaches (e.g., [14, 22]) are constrained to indoor scenes, and extensive lighting or camera rigs have limited portability. Our methods, on the other hand, make use of inexpensive consumer-grade stereo cameras. Such cameras are increasingly available and highly portable, handling both indoor and outdoor scenes. While stereo correspondence can have issues in areas that lack texture or in areas of occlusion, we demonstrate a variety of images for which our methods produce convincing results even with occasionally imperfect correspondence.

The work most similar to ours is that of Zhang et al. [33], who calculate stereo depth from a single input video. Just as we do not require a full three-dimensional model of the scene, they show that their effects can be applied with only the depth computed from the video sequence. However, their methods requires a moving camera (thus providing depth information similar to stereo images) while ours can be applied to a single (stereo) photograph.

Each of the effects performed is based on previous work, with modifications appropriately made to fit within our framework. A brief discussion of the background of each is given here.

Highlighted Depth-Of-Field Highlighted depth-of-field was introduced by Kim et al. [14] as a way to highlight desired areas of focus in the image. In their work, they use an active

light approach to approximate the depth of the scene by shining a dot pattern on the scene with a projector and calculating the depth in a post-processing step. Once the depth is calculated for the scene, darkening can be applied to areas intended to be out-of-focus. Thus, the viewer's attention is drawn to objects in the image that have not been darkened. Similar to their method of depth acquisition was that of Moreno-Noguer et al. [22]. However, these authors use the calculated depth to create traditional depth-of-field. As stated earlier, our method uses a stereo pair of images to calculate the depth for the scene and therefore, we are able to capture a wider variety of scenes (i.e., not limited to shallow, low-light scenes).

Haze Fattal uses a single image to either de-haze or increase haze in the image [6]. However, that work assumes haze is present in the image as it is used to estimate the depth. This assumption is not present for our work and thus we are not constrained to working only with hazy images.

Relighting Another technique that can enhance the feeling of depth or structure in the image is relighting [1]. In traditional graphics, a simple diffuse lighting model is useful for illustrating what information is needed to light a scene. First, the surface normals of the objects must be known. Second, the direction and strength of all lights must also be known. Using this information, a diffuse lighting can be applied to any graphics scene. Image-based relighting requires an additional knowledge of the original light source. In order to approach this problem, some have tried to estimate the reflectance of the objects in the scene, which is defined as the ratio of light reflected by the surface over the amount of light incident to the surface [4, 12, 28]. Trying to find surface reflectance independent of lighting in the image is a problem that is underdetermined. For instance, the color of a surface could be because of the original light color, the reflectance of another object, or the color of the object surface itself.

Other methods such as the classic shape-from-shading method [9] attempt to reconstruct the surface geometry of a scene (particularly the surface normals), though again they must make constraining assumptions [21, 29].

Given a set of images that have each been lit from a different lighting direction, it is possible to use the ratios of images to relight an image [24, 27]. We also use ratios to relight our images, but rather than using sets of images under varying lighting, we instead use the computed surface normals and assume the scene was lit by one or more user-specified light sources.

Depth-of-Field While relighting can enhance the structure of objects, depth-of-field can create an overall feeling of depth in an image. Most methods that offer a way to add depth-of-field after an image is captured either lack portability, are too expensive, or are simply unavailable. Bando et al. [3], for instance, suggest using a single image that already has depth-of-field in the image when captured to approximate correctly refocusing the image. While their results are promising, the constraint of existing depth-of-field is too restrictive. Alternatively, many have proposed modifying traditional cameras in order to add this effect [8, 17, 19, 20, 30]. However, these solutions are not typically viable for traditional consumers because they require the user to modify or add non-standard parts to a camera.

The work surrounding light fields [18] has demonstrated that given a light field encoding of an image, depth-of-field can be simulated in the finalized image. Light fields are an encoding of the rays of light in the scene rather than just the projection of those rays from a single vantage point (such as a traditional image). One way to capture a light field is to use a grid of cameras. Because only two points are needed to encode a ray, each camera coordinate on the grid represents one point of the ray while the actual pixel coordinate within that camera represents the other point. Light-field camera systems or cameras are typically hard to come by or suffer from capturing low-resolution images. Even the newly created Lytro consumer-grade light-field camera based on Ng's handheld light-field camera [23] has not yet escaped this problem, as its effective resolution is currently lower than most available point-and-shoot cameras or even cell phones.

Similar to our method, Yu et al. [32] also use a stereo system to capture the depth of the image in order to generate depth-of-field, however, theirs is a $1500 stereo rig rather than a consumer-grade system. The main difference between our work and theirs is the information used for creating the finalized image. We introduce a method for using information from both color images (where they only use one color image) to create the affected image. The second difference is our method of calculating depth in the images, as we take special care to calculate the normal map jointly with the disparity.

1.2 Stereo Correspondence

Because depth is necessary to performing our methods, we employ the work of Bleyer et al. to extend PatchMatch stereo—an algorithm which calculates the depth of every pixel in the image [5]. PatchMatch stereo inherently calculates surface normals at each pixel, yet they are inherently noisy. A modification of PatchMatch stereo, which is discussed in 2.2, enhances the surface normals, making them usable for the effects.

Because understanding how basic stereo correspondence algorithms work is fundamental to this work, we give a brief overview for the interested reader. Given two images captured simultaneously or of the same scene with some offset between them, it is possible to calculate the depth of the image [26]. Parallax, which is the difference of the apparent position of objects when viewed from two different vantage points along a line, makes this possible, as seen in Figure 1.3.

Assuming that the rows of the two images correspond to each other (which in a consumer-grade stereo camera, they do), the goal of stereo correspondence is to find the most likely match for each pixel on the row. For images that do not have matching rows, it is possible to rectify one to the other for stereo matching.

In order to find matching points, various methods have been constructed [26]. A common method is to perform support window matching. This is done by taking a support window of pixels and compare it to a support window of pixels in the other view. This

(a) Left image (b) Right image

Figure 1.3: Parallax: Notice that a pink tree is not seen in the left image, but in the right. Also, the left features a sidewalk light, which is not seen in the right image. This effect of objects in the scene moving more from one viewpoint to another the closer they are to the viewer is the effect of parallax.

is typically performed only along the scanlines as the stereo image pair is assumed to be rectified. The distance metric to minimize is typically the sum of the squared differences (SSD) or the sum of the absolute differences (SAD) of the pixels in the support window. An illustration of the matching process is shown in Figure 1.4.

Once the match is found, the offset between the two pixels is calculated, also known as *disparity*. Disparity is inversely related to depth as follows where d is disparity, D is depth, f is the focal length of the camera in pixels, and B is the baseline, or the distance between the two cameras, in millimeters:

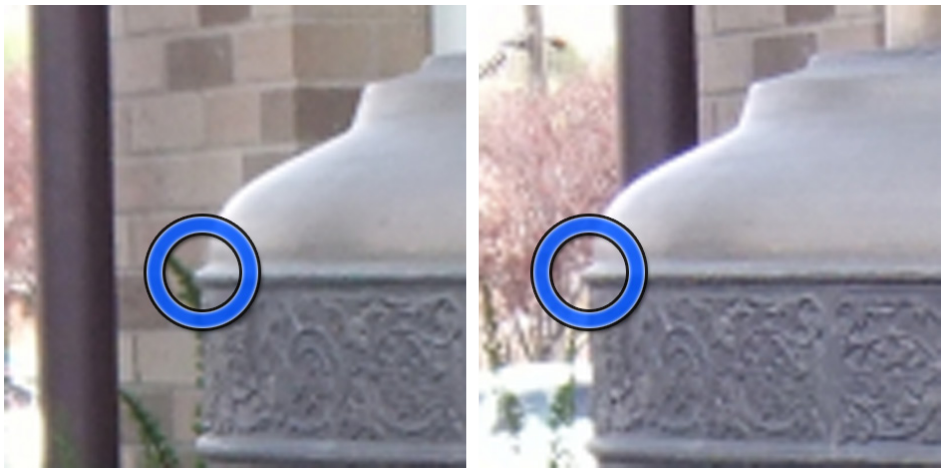
$$D = \frac{fB}{d} \tag{1.1}$$

An example of a disparity map is seen in Figure 1.5. All of the methods presented in this work rely on disparity, and thus, neither the baseline or the focal length needs to be calculated.

Stereo correspondence for depth reconstruction is not without problems, however. All issues stem from miscalculated matches, which can show up as inconsistencies in the final disparity map. Occluding objects causes failed matching in the disparity because things seen in one image are not seen in the other. Also, areas that are textureless can cause matching to fail. An example of this is seen in Figure 1.5. This can also happen at the borders of images, as the horizontal field of view shared by both images is less than each image's field of view

(a) Left image

(b) Right image



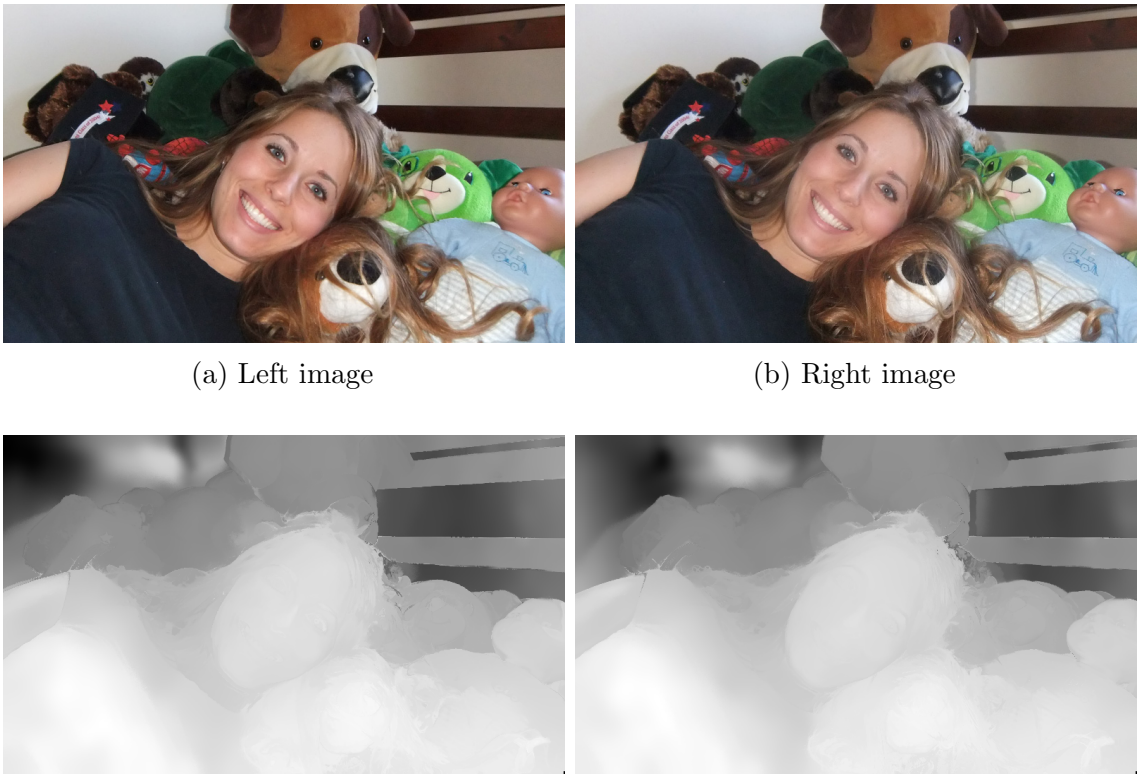
(c) Matching pixels

Figure 1.4: Matching Pixels: Every pixel is matched to a pixel in the corresponding view, producing a dense set of matches for the stereo images.

alone. Specularities or transparent objects are also problematic for matching, as the light effects are dependent on the viewpoint of the camera.

In order to accommodate these problems, first the inconsistencies must be detected and then a method for filling the areas must be devised. We discuss this in more detail in Section 2.2.

In summary, we use a stereo pair of images, captured with a consumer-grade stereo camera, to enable depth calculation of the scene. We then use the calculated depth and surface normals to add depth-aware effects to either the left or right image or both images simultaneously. Chapter 2 is an expanded form of a self-contained research paper recently accepted to 3DV, a conference on 3D vision topics.



(a) Left image (b) Right image

(c) Left disparity (d) Right disparity

Figure 1.5: Inverse depth or disparity: Disparity is derived from performing a dense pixel match in both the left and the right images. The grayscale values correspond to the distance relative to the camera, white being closest and black being farther away. Notice the mistake in the disparity in the textureless area of the wall at the top of the images.

Chapter 2

Interactive Depth-Aware Effects for Stereo Image Editing

Abstract

This paper introduces methods for adding user-guided depth-aware effects to images captured with a consumer-grade stereo camera with minimal user interaction. In particular, we present methods for highlighted depth-of-field [14], haze, depth-of-field, and image relighting. Unlike many prior methods for adding such effects, we do not assume prior scene models or require extensive user guidance to create such models, nor do we assume multiple input images. We also do not require specialized camera rigs or other equipment such as light-field camera arrays, active lighting, etc. Instead, we use only an easily portable and affordable consumer-grade stereo camera. The depth is calculated from a stereo image pair using an extended version of PatchMatch Stereo [5] designed to compute not only image disparities but also normals for visible surfaces. We also introduce a pipeline for rendering multiple effects in the order they would occur physically. Each can be added, removed, or adjusted in the pipeline without having to reapply subsequent effects. Individually or in combination, these effects can be used to enhance the sense of depth or structure in images and provide increased artistic control. Our interface also allows editing the stereo pair together in a fashion that preserves stereo consistency, or these effects can be applied to a single image only, thus leveraging the advantages of stereo acquisition even to produce a single photograph.

2.1 Introduction

This paper introduces a novel method for interactively adding depth-aware effects in real time to images captured with a consumer-grade stereo camera. We define "depth-aware" effects as those that enhance the perception of depth [11], allow the editor to draw natural attention to certain areas [14], or augment the three-dimensional structure of objects in the image [1], as seen in Figures 2.1 and 2.2. We provide four simple methods: highlighted depth-of-field (a term we adopt from Kim et al. [14]), haze, depth-of-field (DOF), and relighting. These allow a user to easily create effects in an image, such as those seen in Figure 2.1 through a simple click-to-add-here interface with minimal interaction. The simplicity of the interaction is best seen in the supplementary video.

Depth-aware effects are not new to graphics and are commonly employed in three-dimensional rendering. With geometric information of a scene one can create a variety of interesting effects in images, as many have shown [2, 3, 6, 8, 10, 12, 14, 15, 17–23, 29–33]. Given a full 3-D reconstruction of an image, one can use conventional rendering tools to re-render the associated scene with new lighting, depth-of-field, and an array of other effects. However, this requires the presence or creation of 3-D models from images, itself a difficult and sometimes manually intensive process. Similar to Zhang et al., we overcome this issue by allowing the user to edit and apply effects directly in image space and do not require a full 3-D reconstruction [33]. Unlike Zhang et al., our method does not require multiple input images or a moving camera to calculate depth accurate enough to perform realistic effects.

This work offers four novel contributions. First, we extend the PatchMatch Stereo algorithm [5] to calculate both disparity and improved normal maps, as the former only produces noisy normals as a by-product of the depth calculation. Second, we modify four depth-aware effects to fit within the framework of stereo-image editing, without a 3-D reconstruction. Third, we use GPGPU technology to apply the depth-aware effects in real-time, allowing the user to receive immediate WYSIWYG feedback rather than having to construct intermediate layers or geometry. Finally, we introduce a pipeline for the effects

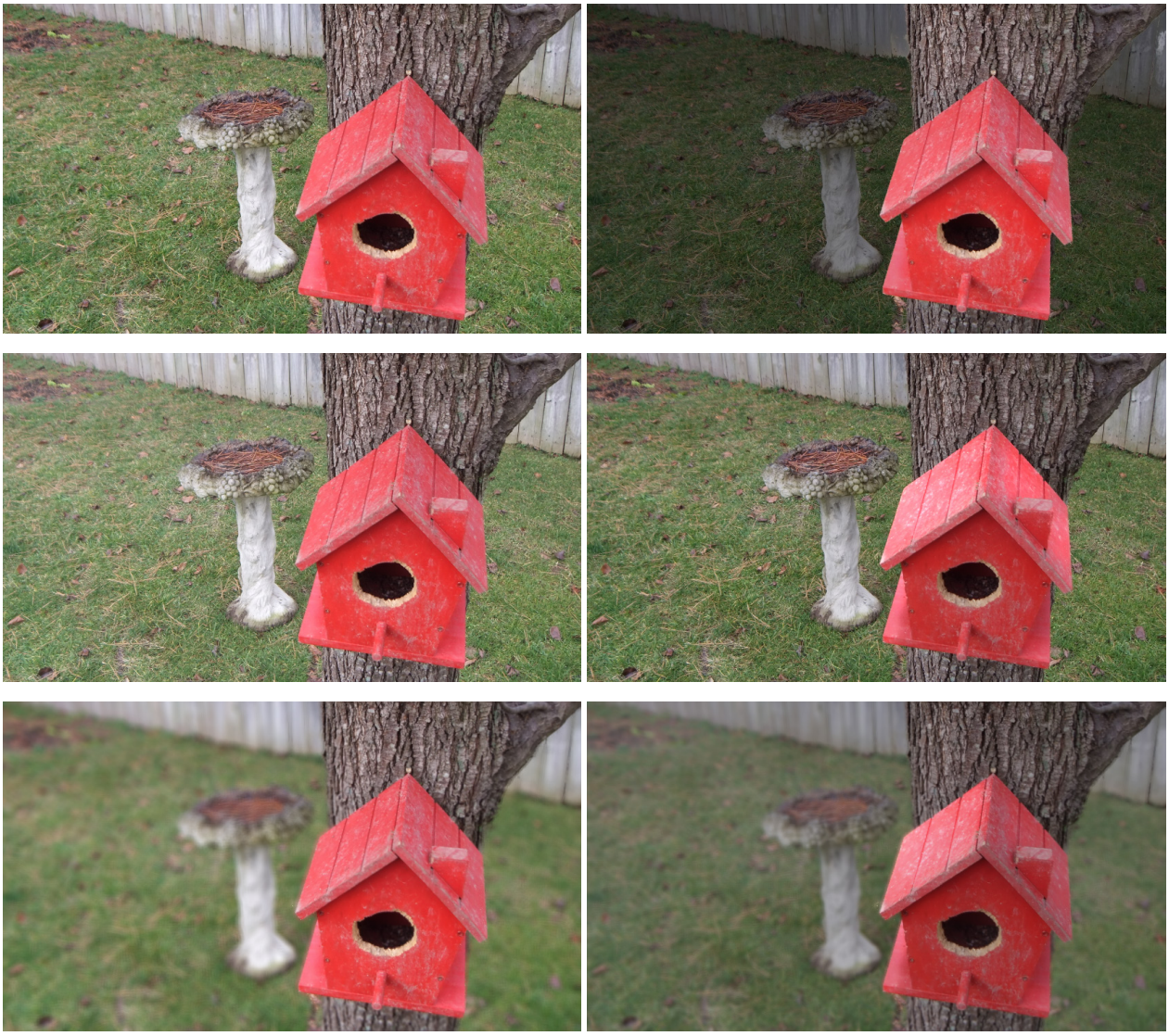


Figure 2.1: Depth-aware effects: Depth-aware effects used to enhance the appearance of depth and structure. Top-left: Original image. Top-right: Highlighted depth-of-field focuses attention on the birdhouse. Middle-left: A small amount of haze offers a feeling of more depth in the scene. Middle-right: Image relighting used to strengthen the direct lighting from the upper left of the scene enhances the 3-D structure. Bottom-left: Depth-of-field again focuses attention on the birdhouse. Bottom-right: Multiple effects (highlighted depth-of-field, relighting, haze, and depth-of-field) are combined in a physically realistic pipeline.

(a) Original (b) Foreground (c) Background

Figure 2.2: Depth-aware effects offer increased artistic control over the viewer's attention. a) original image; b) attention is drawn to the front using highlighted depth-of-field; c) attention is drawn to the background doll with DOF and highlighted DOF.

to combine them in a physically consistent order, allowing individual effects to be enabled, removed, or adjusted.

2.2 Depth Acquisition

We use off-the-shelf 3D stereo cameras to capture images that are fed to a stereo correspondence algorithm to calculate the necessary structural information. A full discussion of this well-studied area is beyond the scope of this paper, but we refer the interested reader to [26] for a broad overview.

We create a modified version of the PatchMatch Stereo approach proposed by Bleyer et al. [5]. This method uses slanted support windows to determine stereo correspondence, thus offering a way to calculate accurate disparity values at each pixel, even for slanted surfaces. Because PatchMatch Stereo estimates a slant plane per pixel, each has an associated normal. While normals could be calculated from the gradients of the disparity values with any stereo correspondence algorithm, PatchMatch Stereo offers better stereo correspondence in areas where many other stereo correspondence algorithms fail—specifically where the object is not frontoparallel with the imaging plane. In this method, the disparities around each pixel are fitted with a local slant plane f_p, and neighboring pixels are allowed to iteratively propagate their plane to other pixels until every unoccluded pixel has a proper plane. However, the normals produced by the original PatchMatch Stereo algorithm are noisy. We thus develop an enhanced version to produce smoother normals, which are critical for image relighting.

Like most stereo correspondence algorithms, PatchMatch Stereo works best in areas that are visible in both images. In areas where holes exist due to partial occlusion, Bleyer et al. suggest scanline filling the holes with the plane that assigns the lowest disparity from the valid left and right neighbors [5]. Because this can lead to streaking in the disparity map, the authors additionally propose a weighted median filter to smooth the filled-in values. Based on our experiments, this is not suited well to most natural images, as it creates artifacts that do not respect object boundaries. Also, the final median filter operates only on the disparity values and not the planes themselves.

In order to overcome both problems, we introduce a joint bilateral filter using only valid planes (determined through a validity test explained later) to fill in missing disparities and normals in the final output based on spatial proximity and color similarity . Because we have color information from the original image where holes exist in the disparities, we assume that pixels with similar colors will have similar plane information. Admittedly, this assumption fails when objects of the same color have different depths. We have not, however, found this extremely common in practice when considering only hole filling. The joint bilateral filter is based on spatial proximity $\|p - q\|$ and color similarity $rgb(p, q)$ as follows:

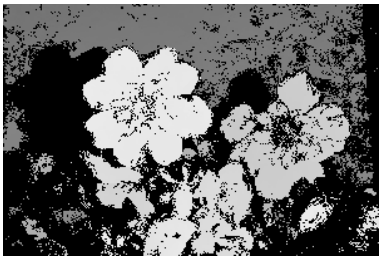
$$f_p = \sum_{q \in W_p} w(q) \frac{G_{\sigma_1}(\|p - q\|)\ G_{\sigma_2}(rgb(p, q))\ f_q}{G_{\sigma_1}(\|p - q\|)\ G_{\sigma_2}(rgb(p, q))} \tag{2.1}$$

where W_p denotes a neighborhood of pixels around pixel p, f_p and f_q denote the disparity planes of p and neighboring pixel q respectively, and

$$w(q) = \begin{cases} 1 & \text{if } d(q) \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

We find it useful to reset the origin for filtered planes to the pixel position, with the z-component as the filtered disparity value. In practice, we set the size of W_p to be one-third the width of the image, as many holes can be quite large. For the values σ_1 and σ_2, we use


(a) Original


(b) Before filling

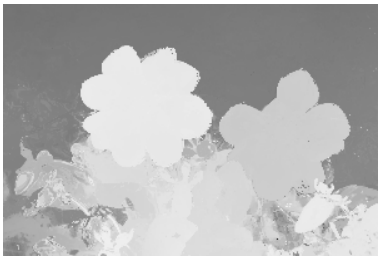

(c) After filling

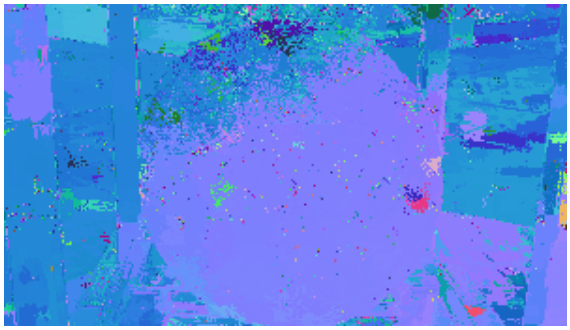
Figure 2.3: Hole filling disparities (dark regions)

the values $\frac{W_p}{5}$ and 7, respectively. We found that in practice, these values work best for most images. The joint bilateral filter has the effect of propagating valid plane information to neighboring pixels of similar color as seen in Figure 2.3.

Determining where holes exist in the disparity maps is performed with a simple validity test [5]. If we let $d(p)$ denote the disparity at point p, and p' the corresponding point in the other opposite view with disparity $d'(p')$, we can write this test as $|d(p) - d'(p')| \leq 1$. This acts as a consistency check between the two views to make sure corresponding disparities match to within a small threshold. The value of 1 is what the original authors used and can be tuned. However, we have found that this value works well on all the images in this work.

The normal maps extracted from each pixel's plane f_p are not accurate enough to perform certain depth-aware effects, such as relighting. This is because the planes in PatchMatch Stereo are used only as slanted support windows and do not necessarily reflect scene geometry when there are multiple slanted support windows that could lead to valid correspondence. Thus, the normal maps produced by the original PatchMatch algorithm do not always reflect the accurate geometry of the scene. The original normal maps can be seen in Figure 2.4b. In order to accommodate for the inaccurate normals, we perform a normal refinement step before hole filling. We use the method of Gopi et al. to perform SVD to reconstruct the normals from the disparities [7]. As can be seen in Figure 2.4c, the results of the normal refinement step improve the overall accuracy of the normals and better

(a) Original image



(b) Normal map from PatchMatch Stereo

(c) Normal map after hole filling and normal refinement

(d) Normal map after hole filling, refinement, smoothing, and user-defined constrast enhancement (discussed in Eq. 2.15)

Figure 2.4: Normal refinement. Because the original PatchMatch Stereo algorithm does not focus on the actual outcome of the normals, they are mostly incorrect. After hole-filling, normal refinement, and plane smoothing the normals are highly smoothed, but more accurately reflect scene geometry and allow for image-based relighting.

represent the true geometry in the scene compared to those produced by the PatchMatch Stereo algorithm without refinement (2.4b).

However, refining the normal maps may not totally remove incorrect normals, as shown in Figure 2.4c. To correct these we use a trilateral filter based on spatial proximity, color similarity, and disparity similarity. In practice, we reset W_p to five percent of the image width, as we no longer have large holes to fill, while σ_1 and σ_2 are $\frac{W_p}{5}$ and 15, respectively. For the third Gaussian that weights disparity similarity, we set σ_3 as 10% of the current disparity range. These values were tuned based on a variety of images and we have found that they work well for most images. This step is necessary to remove any noise in either the disparities or the normal map as the relighting effect is highly sensitive to miscalculated normals. The

results of this final step are shown in Figure 2.4d. We also allow the user to globally adjust the normal map contrast (via the mouse wheel), which we discuss in Section 2.3.

Once the entire process of hole-filling, normal refinement, and smoothing is finished, we repeat PatchMatch Stereo through its original iterations and our enhancement steps for a user-determined number of times (in practice, three has been sufficient). For large images, we use a hierarchical method with bilateral upsampling, similar to that proposed by Yu et al. [32], to increase calculation speed. We also find that this process can further smooth the normal map for relighting.

2.3 Depth-Aware Effects

We now introduce each of the effects in order of complexity, not necessarily according to the physically-realistic rendering pipeline described later in Section 2.4.

2.3.1 Highlighted Depth of Field

A method for highlighted depth of field and traditional (out of focus) depth of field was proposed by Kim et al. [14]. In their work, the authors use an active light approach by projecting dots onto the scene in order to estimate the depth. Afterwards, a certain depth plane is selected to remain undarkened, while objects that are not within that plane are darkened based on their distance to the selected depth. This allows the user to selectively draw attention to objects in an image while using depth to drive the result. Similar to the depth capture method of Kim et al. is that of Moreno-Noguer et al. [22]. While this work uses a projected dot pattern to capture the depth of the scene, the effects applied do not include highlighted depth of field but rather traditional depth-of-field. However, it suffers from lack of portability and is limited to certain indoor scenes, as is [14]. Depth capture through stereo handles both outdoor and indoor scenes. While stereo correspondence can have issues in areas that lack texture or in areas of occlusion, we show a variety of images for which our methods succeed.

(a) Before darkening (b) After darkening (c) Middle highlight

Figure 2.5: The results of highlighted depth of field.

To perform our highlighted depth of field, the user first selects a point in the image to establish a depth of interest. We denote the disparity of this point as $d(c)$ and then perform selective darkening of original image I to produce output image I' by modulating the intensity of each pixel p as follows:

$$I'(p) = \begin{cases} I(p) & \text{if } |d(p) - d(c)| < r \\ \alpha \; M(p) \; I(p) & \text{otherwise} \end{cases} \tag{2.3}$$

based on the difference between that pixel's disparity $d(p)$ and the plane of interest $d(c)$:

$$M(p) = \frac{|d(p) - d(c)|}{\max(d(c) - d_{\min}, d_{\max} - d(c))} \tag{2.4}$$

where $d_{\min}$ and $d_{\max}$ are the respective minimum and maximum values in the disparity map, and r is a range of disparities around the selected one that maintains full brightness. The strength of the effect is controlled by the parameter α, which the user adjusts with the mouse's scroll wheel. An example of this effect is shown in Figure 2.5.

2.3.2 Haze

Similar to highlighted depth-of-field, adding haze to a scene offers a way to be selective about what objects should receive the viewer's attention. It also provides a greater sense of depth to the viewer. An example of this is shown in Figure 2.6. *Aerial perspective* is a term used by artists to describe this technique. Fattal proposed a method for adjusting haze in an image

(a) Before haze (b) After haze (c) After more haze

Figure 2.6: The results of adding haze. Notice that a slight amount of haze can increase the sense of depth in the image. In both images, $d(s)$ is selected to be the front of the road (see Eq. 2.5). The difference between 2.6b and 2.6c is the amount of haze the user adds via the mouse scroll wheel.

but imposed the requirement that haze be present to both estimate the depth and adjust the effect [6].

As with most previous methods, we add haze to the image using an alpha blend between the original image $I(p)$ and a desired haze color (or pattern) $H(p)$:

$$I'(p) = \begin{cases} H(p)\ (1 - t(p)) + I(p)\ t(p) & \text{if } d(p) < d(s) \\ I(p) & \text{otherwise} \end{cases} \tag{2.5}$$

where the blend is determined by a transmission factor $t(p) = e^{-\alpha\ \frac{|d(s)-d(p)|}{d_{\max}-d_{\min}}}$ and α is the strength of the effect, which is again adjusted using mouse's scroll wheel.

We have found it useful to allow the user to define a near barrier or stopping plane beyond which the haze starts and in front of which the scene retains full contrast. The user selects this point with a simple mouse click, and we denote the disparity of this point as $d(s)$, as used in Equation 2.5.

The haze map $H(p)$ can be a constant (user-selected) color C_H: $H(p) = C_H$ or can be used to introduce a spatially-varying component to the haze. If the user chooses the latter option, we use a simple 2D midpoint-displacement generated cloud $m(p)$ that encompasses the image: $H(p) = m(p)\ C_H$. An example of this effect can be seen in Figures 2.6b and 2.6c.

2.3.3 Relighting

Lighting conveys a strong sense of shape, and relighting an image can enhance the 3-D structure of the objects in the scene as well as convey information about shape or features [1]. Many techniques have tried to estimate the inherent reflectance of objects in a scene independent of scene lighting—a so-called *intrinsic image* [4]—including for example the work of [12, 28]. This problem is inherently an underdetermined one, and to separate these conflated elements requires prior knowledge, which might not be available, or assumptions, which may not be correct. Other methods such as the classic shape-from-shading method [9] attempt to reconstruct the surface geometry of a scene (particularly the surface normals), though again they must make constraining assumptions [21, 29].

Given a set of images that have each been lit from a different lighting direction, it is possible to use ratios of images to relight an image [24, 27]. We also use ratios to relight our images, but rather than using sets of images under varying lighting, we instead use the computed surface normals and assume the scene was lit by one or more user-specified light sources. We use a diffuse lighting model that assumes no shadowing, a user-positioned approximate location for an original lighting source (or multiple sources), and a user-specified new light (or, again, multiple lights). Our method does not require explicitly solving for an intrinsic image, though Lee et al. [16] showed recently that having depth information can assist with its extraction, and one part of our process can be considered an approximation of it. While our lighting model does not handle complex lighting, as Karsch et al. [12] demonstrate, and the assumptions of the model limit the physical accuracy of the resulting image, our required user interaction is much simpler and can still lead to desirable artistic effects. An example of interaction is shown in the supplementary video.

A diffuse lighting model, found in any introductory graphics text, can be written as the sum of two lighting components, one for the diffused direct light and one for the ambient light in the scene:

$$I(p) = \mathit{diffuse}(p) + \mathit{ambient}(p) \tag{2.6}$$

The ambient light reflected off a point, *ambient*, is simply the product of the ambient light strength, S_a, its color, C_a, and the point's surface reflectance $R(p)$:

$$ambient = R(p)\ S_a\ C_a \tag{2.7}$$

The diffuse light reflected off a point, *diffuse*, is proportional to the cosine of the angle made between the surface normal at that point $n(p)$ and the direction to a point light source L, the strength S_d and the color C_d of the light source, and the point's surface reflectance:

$$diffuse(p) = R(p)\ S_d\ C_d\ \max(n(p) \cdot L, 0)) \tag{2.8}$$

Suppose that the same scene is lit by a different point light source with direction L' of the same strength and color to produce a new image I'. We assume for now that the ambient lighting is the same. The new image is thus

$$I'(p) = diffuse'(p) + ambient(p) \tag{2.9}$$

and the new diffuse component $diffuse'(p)$ is

$$diffuse'(p) = R(p)\ S_d\ C_d\ \max(n(p) \cdot L', 0)) \tag{2.10}$$

If we divide Equation 2.9 by Equation 2.6, we get

$$\frac{I'(p)}{I(p)} = \frac{diffuse'(p) + ambient(p)}{diffuse(p) + ambient(p)} \tag{2.11}$$

Substituting,

$$\frac{I'(p)}{I(p)} = \frac{R(p)\ S_d\ C_d\ \max(n(p) \cdot L', 0)) + R(p)\ S_a\ C_a}{R(p)\ S_d\ C_d\ \max(n(p) \cdot L, 0)) + R(p)\ S_a\ C_a} \tag{2.12}$$

Canceling out the unknown (but assumed non-zero) surface reflectance $R(p)$ and rearranging:

$$I'(p) = I(p) \; \frac{S_d \; C_d \; \max(n(p) \cdot L', 0)) + \; S_a \; C_a}{S_d \; C_d \; \max(n(p) \cdot L, 0)) + \; S_a \; C_a} \tag{2.13}$$

We may thus write the new image in terms of the original one given our calculated surface normals and the user-specified estimates of the lighting directions and parameters, independent of the original surface reflection, by modulating the original image by the fractional quantity in Equation 2.13. (If we replace the numerator in the fraction by 1, the righthand side is an approximation of the intrinsic image.)

If we allow the user to also vary the strength and color of the new ambient light and point light sources (S'_a, C'_a, S'_d, and C'_d respectively),

$$I'(p) = I(p) \; \frac{S'_d \; C'_d \; \max(n(p) \cdot L', 0)) + \; S'_a \; C'_a}{S_d \; C_d \; \max(n(p) \cdot L, 0)) + \; S_a \; C_a} \tag{2.14}$$

This is done through adjusting the parameters for a given light source (either ambient or directional). Adjusting the new ambient light strength, for instance, can brighten or darken the image. Adjusting the new ambient color can tint the entire image to that color. On the other hand, adjusting strength and color of new directional lights has the effect of highlighting certain areas of objects with a certain color and a certain strength.

The normal maps produced during depth acquisition can often be flatter than desired, thus producing only subtle changes when relighting. In order to increase the contrast of the normal map, a global adjustment of the normals is produced by increasing the distance of each normal to the front-facing normal, $n_0 = (0, 0, 1)$:

$$n'(p) = \frac{n(p) + \alpha \; (n(p) - n_0))}{\|n(p) + \alpha \; (n(p) - n_0))\|} \tag{2.15}$$

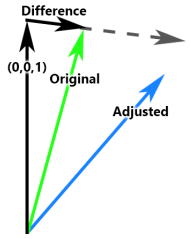


Figure 2.7: Normal adjustment: The original normal is subtracted from the front-facing normal. Then, that difference is increased via a user controlled parameter. The final adjusted vector is normalized. See Equation 2.15.

(a) Original image (b) Normal map (c) Adjusted using Eq. 2.15.

Figure 2.8: Normal adjustment example: The original normal map is allowed to be globally adjusted to increase contrast for relighting. See Equation 2.15. Notice that relighting can enhance the 3D structure of an object. Where an object was once flat-looking, it can appear to pop out.

α is tuned through a user interface, allowing the user to choose a desirable contrast for the normal map. An illustration of Equation 2.15 is given in Figure 2.7, while Figure 2.8 shows how α can be tuned to increase the normal map contrast.

(a) Original

(b) White light to the right

(c) Red and green on the sides

(d) Red and green vertically

(e) Blue light to the left

(f) Red, green, and blue lights

Figure 2.9: Relighting: The lighting direction can be adjusted using Equation 2.17. The relighting effect handles multiple lights, each with different strengths and colors.

Using these enhanced-contrast normals, our relighting thus becomes

$$I'(p) = I(p)\ \frac{S'_d\ C'_d\ \max(n'(p)\cdot L', 0)) + \ S'_a\ C'_a}{S_d\ C_d\ \max(n(p)\cdot L, 0)) + \ S_a\ C_a} \tag{2.16}$$

Finally, we generalize this relighting to multiple original light sources and multiple new light sources:

$$I'(p) = I(p)\ \frac{\sum_j\ S'_{d_j}\ C'_{d_j}\ \max(n'(p)\cdot L'_j, 0)) +\ S'_{a_j}\ C'_{a_j}}{\sum_i\ S_{d_i}\ C_{d_i}\ \max(n(p)\cdot L_i, 0)) +\ S_{a_i}\ C_{a_i}} \tag{2.17}$$

2.3.4 Depth-of-Field

Depth-of-field effects can be used to control which objects receive the viewer's attention. An example of this is given in both Figures 2.1 and 2.2c. While many have suggested adding depth-of-field to an image after it has been captured, most methods either lack portability, availability, or are too expensive for the average consumer [3, 8, 17–20, 22, 23, 30].

Using depth, it is possible to approximate the depth-of-field image if the circle of confusion is known for each depth in the image, as is done in [13]. In the following discussion, we show that the circle of confusion can be approximated with only disparity values calculated from stereo correspondence. For a camera with aperture size A and focal length F, given a plane-in-focus depth P the circle of confusion for each pixel p with depth $D(p)$ can be calculated as follows [31]:

$$CoC(p) = \left| A\ \frac{F\ (P - D(p))}{D(p)\ (P - F)} \right| \tag{2.18}$$

Using an interface and notation similar to Section 2.3.1, suppose that the user has specified a point c with disparity $d(c)$ as a point to be in focus. The circle of confusion for any other point p with disparity $d(p)$ can be solved for by substituting $P \propto \frac{1}{d(c)}$ and $D \propto \frac{1}{d(p)}$. If we simplify and assume that $P \gg F$, which is a reasonable assumption for stereo work, Equation 2.18 can be approximated by

$$CoC(p) \propto |d(p) - d(c)| \tag{2.19}$$

Thus, for stereo images the circle of confusion is approximately proportional to the difference in disparity between the point in question and that of points lying on the plane in focus. The

constant of proportionality here folds together the unknown focal length, aperture size (or equivalently F-stop), and (uncalibrated) stereo baseline. This avoids the need for a calibrated camera and leaves the user with a simple-to-understand single means for controlling the strength of the effect, which (as with our earlier methods) a user can control with the mouse's scroll wheel.

One could use the circle of confusion map $CoC(p)$ to create the depth-of-field image using methods such as [13], but we instead use dynamic light field generation similar to the methods of [31] and [32] because these works use disparity directly to compute the light fields, and thus it is well-suited to a stereo pair of images. Yu et al. [31] used depth to dynamically generate the light field using GPU architecture, yet their solution required the depth information to be known beforehand. Yu et al. [32] used disparity directly calculated from a stereo pair to dynamically generate the light field for the depth-of-field reconstruction. One disadvantage of [31] and [32] is that they use only a single image in the depth-of-field reconstruction. We differ in the fact that we generate two light fields for each image of the stereo pair in order to borrow rays from the other camera's view where one camera's view is lacking, as displayed in Figure 2.11a. The secondary light field comes from the opposite view's disparity map and is generated as seen from the primary reference view. An example of the secondary view seen from the primary image's point of view is shown in Figure 2.10. We offer more detail when discussing the depth-of-field reconstruction. We briefly reiterate the light field generation equations from the aforementioned works to assist in explaining our own methods.

Light Field Generation As noted in [32], any ray within a light field can be parameterized by (s, t, u, v), where s, t represents the image in the light field and u, v are the pixel coordinates within that light field image. The light field pixel coordinate for any pixel in the image is calculated as follows:

$$(u, v) = p - d(p)\,(s, t) \tag{2.20}$$

(a) The original primary view (b) The secondary seen from the primary.

Figure 2.10: Secondary view as seen from the primary view: The secondary view is used as supplemental information to fill in holes where the primary light field lacks information (see Figure 2.11). This will happen near occlusion boundaries.

In order to generate the secondary light field (the secondary view as seen from the primary's point of view) into the same coordinate space, $1 - s$ is used rather than s, thus yielding:

$$(u, v)' = p - d'(p)\,(1 - s, t) \tag{2.21}$$

where $d'(p)$ is the disparity at p in the secondary view. Assuming the primary image's coordinates in the light field are $(0, 0)$, then $1 - s$ effectively puts the secondary image as seen from the primary image (if the primary image is the left stereo image). When performing Equations 2.20 and 2.21 for the right view (as opposed to the left) we negate s and the quantity $(1 - s)$, respectively. Thus, for any given (s, t) we can generate two light fields per view according to their respective disparity maps.

As suggested by both [31] and [32], generating color light fields is problematic because the warping process of Equation 2.20 potentially warps more than one pixel to a light field image pixel coordinate, thus overwriting any previously written value. In order to write the correct pixel at a particular light field coordinate, the maximum disparity value associated with that pixel color is used to preserve proper occlusion [31, 32]. Similar to the methods of [31] and [32] we dynamically generate our light fields in parallel using atomic max operator on GPU architecture and only store the disparity values for both speed and memory efficiency.

Once the disparity values have been written, it is trivial to look up the original color pixel associated with that disparity using the following inverse lookup equation:

$$p = (u, v) + d(p)\,(s, t) \tag{2.22}$$

When performing the inverse lookup on the secondary view, $(1 - s)$ is again used:

$$p' = (u, v) + d'(p)\,(1 - s, t) \tag{2.23}$$

Again, if the right view is being used then we negate s and the quantity $(1-s)$ in Equations 2.22 and 2.23.

In order to reduce aliasing, we generate a jittered sampling pattern from a Halton sequence for the (s, t) values [31]. A significant improvement in our method over that of [31] and [32] is that the (s, t) values are generated on the fly within the size of the current aperture rather than over a predetermined area and then selected from within a changing aperture, producing fewer holes in the final light fields. Because forward warping in light field generation is prone to produce holes, [31] uses a splatting technique to fill in the missing information. We do not find that necessary as we have two light fields per view and our light field images are within the size of the aperture. An illustration of how this generates fewer holes in the light field is shown in Figure 2.11.

Depth-of-Field Generation Both [31] and [32] discuss the difference between using an in-lens camera and out-of-lens camera for generating the depth-of-field image. We focus, as do they, on using the out-of-lens camera:

$$I'(p) = \sum_{(s,t)} L_{out}(s, t, p - d(c)(s, t))\, cos^4\phi \tag{2.24}$$

(a) The primary light field without the secondary view.

(b) The composite light field with both primary and secondary views.

Figure 2.11: Light fields: Information is used from both views to generate a more complete light field.

The attenuation term, $cos^4\phi$, is calculated as $\frac{1}{s^2+t^2+1}$[31]. In the previous works, when constructing the depth-of-field image, the authors note that any holes in the light field are ignored and just not accounted for in the final image [31]. We use the added contextual information of the secondary view in the stereo pair to fill in holes where the primary light field is lacking (mainly at occlusion boundaries). An illustration of the difference is shown in Figure 2.11.

Thus, Equation 2.24 is augmented to use the secondary light field, L'_{out}.:

$$I'(p) = \sum_{(s,t)} \begin{cases} L_{out}(s,t,p-d(c)(s,t))\, cos^4\phi & \text{if no hole in } L_{out} \\ L'_{out}(s,t,p-d(c)(1-s,t))\, cos^4\phi & \text{otherwise} \end{cases} \tag{2.25}$$

When using Equation 2.25 on the right image we negate s and $(1-s)$ in the latter parts of both L_{out} and L'_{out}. In the rare case that both L_{out} and L'_{out} have a hole, we use the minimum disparity value for $d(p)$ in Equation 2.22 when calculating p and then use it to look up a color value in the primary view as part of the summation in Equation 2.25.

(a) Original (b) Relighting (c) Highlighted

(d) Haze (e) DOF

Figure 2.12: The depth-aware pipeline

2.4 Depth-Aware Pipeline

To allow users to combine multiple effects, we use an image rendering pipeline that models the order in which these effects would occur physically during light transport and acquisition. Specifically, we apply the effects in the following order: image relighting (based on position and properties of the light source(s)), highlighted depth of field (based on the amount of light falling on various depths in the scene), haze (introduced as the light travels to the camera), and depth of field (as the camera acquires the image).

Our system simplifies the process of generating any combination of effects by allowing out-of-order edits to be rendered in the correct order. Once users apply the desired effects, they can selectively turn off effects, turn on effects, or adjust the parameters of an effect without having to reapply the others—in this way the pipeline serves the same purpose as effects layers in common image-editing software.

An example of this pipeline can be seen in Figure 2.12. We allow a user to modify any effect and give real-time visual feedback as rendered in the depth-aware pipeline.

2.5 Results

All effects are performed in real-time on an Intel i7 host CPU and an NVidia GTX 480. We reiterate that the depth-aware effects do not require any three-dimensional reconstruction or scene re-rendering in order to apply them to an image.

Our implementation of PatchMatch stereo is performed using GPU computing in order to speed up the stereo correspondence. We can acquire reasonable disparity and normal maps for our effects in both the left and right images on 1080p (2MP) images in two minutes, using the bilateral-upsampling technique discussed in [32]. All effects are performed using fragment shaders in order to achieve real-time WYSIWYG feedback The exception to this is the depth-of-field generation, which is also real-time but implemented using general-purpose GPU computing technology to speed up the light field generation and depth-of-field calculation. The results of each effect are shown in Figures 2.14 through 2.19, while Figure 2.20 shows more results for the depth-aware pipeline.

Admittedly, if our depth acquisition has any mistakes those will propagate to the effects, as shown in Figure 2.13. For the depth-of-field effect, mistakes in the secondary view's disparity are magnified when it is warped into the primary view. Thus, forward warping the secondary image to generate an additional light field for the primary view can propagate minor artifacts when generating depth-of-field around the border of objects out-of-focus. Such failures often occur in featureless or ambiguously matching areas of the image. Some stereo algorithms use global alignment methods to fill in textureless areas [5, 26], which could be used in combination with our incremental solution to PatchMatch stereo. Additionally, a method that allows the user to easily and interactively correct disparity mistakes (without requiring them to understand the underlying complexities) could help with this and similar stereo-image-editing problems.

Our relighting system is entirely dependent on the quality of the calculated normals, which if incorrect, can create unwanted artifacts. Ideally, a more precise method of calculating normals to avoid over-smoothing is desirable. An additional limitation with our current

(a) The original image includes part of sky, which is textureless

(b) Highlighted depth-of-field with incorrect disparities

Figure 2.13: An example of failed stereo correspondence: In places where stereo correspondence miscalculates disparities over large areas, our effects perform poorly. Notice the dark area in the featureless sky is incorrectly darkened when performing highlighted depth of field.

relighting method is the reliance on a diffuse, shadow-free model. For future work it would be beneficial to include a full lighting model complete with de-shadowing, shadowing, and other complex lighting effects.

2.6 Conclusion

Using consumer-grade cameras to capture stereo images, we offer a method of editing images to enhance 3-D structure, making objects appear to visually "pop", and allow control over the viewer's attention. We have introduced four novel contributions. First, we provide a solution for improving the PatchMatch Stereo algorithm to extract smooth normal fields and perform better hole-filling. Second, we introduce the methods for generating the depth-aware effects presented here: highlighted depth-of-field, haze, depth of field, and relighting. Third, all of the effects are applied in image space, as we require no 3-D reconstruction or re-rendering of a scene model for any of the effects. Finally, we present a method for combining multiple effects in a physically realistic pipeline, which allows users to add, remove, or edit effects without having to reapply others.

(a) The original image

(b) The original image

(c) Highlighting the background

(d) Highlighting the background

(e) Highlighting the foreground

(f) Highlighting the foreground

(g) Highlighting the foreground and relighting with red and cyan lights

(h) Highlighting the foreground and relighting from the right side

Figure 2.14: Highlighted depth-of-field on outdoor images.

(a) The original image

(b) Morning mist

(c) The original image

(d) Ground fog

Figure 2.15: Haze added to ground images to show illusion of both mist and fog. This effect not only brings out foreground objects, but adds to the feeling of depth.

(a) Left image

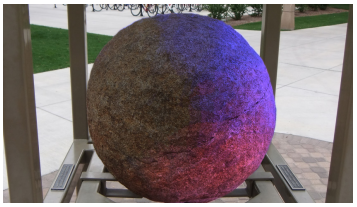
(b) Right image

(c) Relit from top left

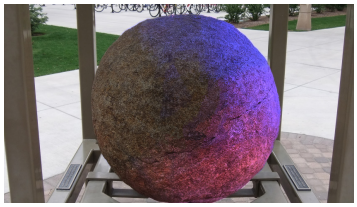
(d) Relit from top left

(e) Relit from bottom

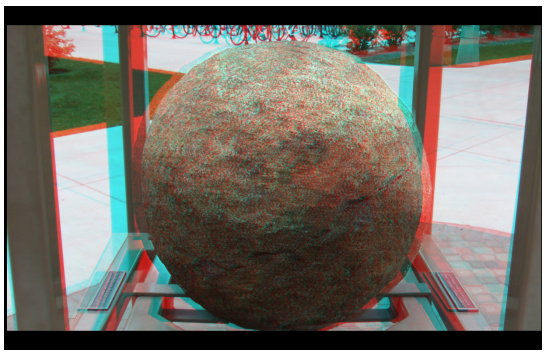
(f) Relit from bottom



(g) Pink and blue lighting



(h) Pink and blue lighting



(i) Relit from top in 3D.

Figure 2.16: Our relighting tool allows the user to add multiple lights as well as masking, which is useful for creating images where certain objects should be the only parts of the image that are affected.We also offer an interface to edit both the left and right stereo pair simultaneously

(a) Original image

(b) Back focus

(c) Front focus

Figure 2.17: Using depth-of-field to simulate an effect akin to that of tilt-shift cameras, as seen in the skyline.

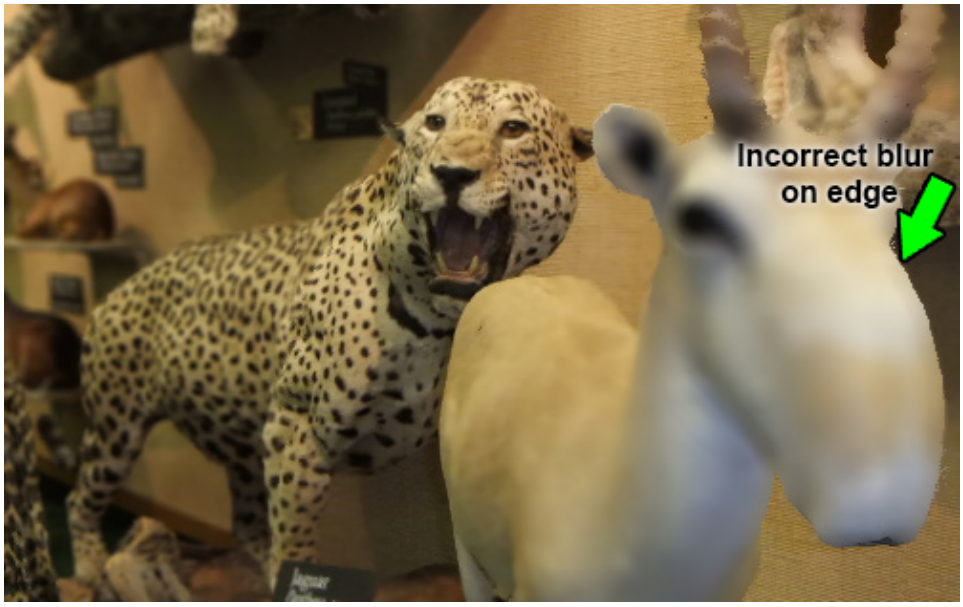
(a) The original image

(b) Back focus

(c) Mid focus

(d) Front focus

Figure 2.18: Our depth of field can mimic DSLR cameras.



(a) Depth-of-field with an image-editing tool using depth as input

(b) Our implementation of depth-of-field from generated light fields

Figure 2.19: Better depth-of-field: As noted earlier, some photo editing solutions do allow the user to use the depth as an offline input channel for depth of field. Yet even with depth as input, the effects produced by such software are often incorrect as shown above. Notice that in our implementation, the depth-of-field has blurred edges on the gazelle that should be out-of-focus (foreground), while image editing software, even with a knowledge of the depth, incorrectly blurs out-of-focus gazelle, assigning its border with a sharp edge rather than blurred.

(a) Original image

(b) Relight face

(c) Darkening in pipeline

(d) Haze in pipeline

(e) Depth of field in pipeline

Figure 2.20: Depth-aware pipeline: Note the use of relighting here to brighten the strongly backlit face and front of the boy.

Chapter 3

Conclusion

Using consumer-grade cameras to capture stereo images, we offer a method of editing images to enhance 3-D structure, making objects appear to visually "pop", and allow control over the viewer's attention. We have introduced four novel contributions. First, we provide a solution for extending the PatchMatch Stereo algorithm to extract smooth normal fields and perform better hole-filling. Second, we introduce the methods for generating the depth-aware effects presented here: highlighted depth-of-field, haze, depth-of-field, and relighting. Third, all of the effects are applied in image space, as we require no 3-D reconstruction or re-rendering of a scene model for any of the effects. Finally, we present a method for combining multiple effects in a physically realistic pipeline, which allows users to add, remove, or edit effects without having to reapply others.

A natural extension to these methods would be twofold: one, increase the accuracy of disparity and normal maps; and two, enhance the complexity of the effects. Examples of the latter include, creating a way to dehaze images rather than only allowing haze to be introduced, offering a more complex relighting system (e.g., ability to place lights in the scene, shadowing, modeling translucent objects, refocusing images, etc.). Increasing the accuracy of the depth acquisition could be devising a better way to fill in occluding object holes and using a more precise method to capture the surface normals in the scene. Using methods to calculate the depth of a single image to perform these effects would also be an interesting progression of this work.

References

[1] David Akers, Frank Losasso, Jeff Klingner, Maneesh Agrawala, John Rick, and Pat Hanrahan. Conveying shape and features with image-based relighting. In *IEEE Visualization (VIS'03)*, pages 349–354, 2003. ISBN 0-7695-2030-8. doi: 10.1109/VISUAL.2003.1250392. URL `http://dx.doi.org/10.1109/VISUAL.2003.1250392`.

[2] Soonmin Bae and Frédo Durand. Defocus Magnification. *Computer Graphics Forum*, 26 (3):571–579, September 2007. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2007.01080.x. URL `http://doi.wiley.com/10.1111/j.1467-8659.2007.01080.x`.

[3] Yosuke Bando and Tomoyuki Nishita. Towards digital refocusing from a single photograph. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 363–372, 2007. ISBN 0-7695-3009-5. doi: 10.1109/PG.2007.64. URL `http://dx.doi.org/10.1109/PG.2007.64`.

[4] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26. Academic Press, 1978.

[5] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference*, pages 14.1–14.11, 2011.

[6] Raanan Fattal. Single image dehazing. *ACM Transactions on Graphics*, 27(3):72:1–72:9, 2008. doi: 10.1145/1399504.1360671. URL `http://doi.acm.org/10.1145/1399504.1360671`.

[7] M. Gopi, Shankar Krishnan, and Cláudio T. Silva. Surface Reconstruction Based on Lower Dimensional Localized Delaunay Triangulation. *Computer Graphics Forum*, 19: 467–478, 2000.

[8] Paul Green, Wenyang Sun, Wojciech Matusik, and Frédo Durand. Multi-aperture photography. *ACM Transactions on Graphics*, 26(3):68:1–68:7, 2007. doi: 10.1145/1275808.1276462. URL `http://doi.acm.org/10.1145/1275808.1276462`.

[9] B. K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, MIT, 1970.

[10] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proceedings of SIGGRAPH 97*, Annual Conference Series, pages 225–232, 1997. ISBN 0-89791-896-7. doi: 10.1145/258734.258854. URL `http://dx.doi.org/10.1145/258734.258854`.

[11] Ian P. Howard. *Perceiving in Depth*. Oxford University Press, 2012.

[12] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):157:1–157:12, 2011. doi: http://doi.acm.org/10.1145/2024156.2024191. URL `http://doi.acm.org/10.1145/2024156.2024191`.

[13] Michael Kass, Pixar A. Studios, Aaron Lefohn, U. C. Davis, and John Owens. Interactive depth of field using simulated diffusion on a GPU. *Computing*, (#06-01), 2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.162.3047&rep=rep1&type=pdf`.

[14] Jaewon Kim, Roarke Horstmeyer, Ig-Jae Kim, and Ramesh Raskar. Highlighted depth-of-field photography: Shining light on focus. *ACM Transactions on Graphics*, 30(3):24:1–24:9, May 2011. ISSN 0730-0301. doi: 10.1145/1966394.1966403. URL `http://doi.acm.org/10.1145/1966394.1966403`.

[15] Johannes Kopf, Boris Neubert, Billy Chen, Michael F. Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and Dani Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics*, 27(5):116:1–116:10, 2008.

[16] Kyong Joon Lee, Qi Zhao, Xin Tong, Minmin Gong, Shahram Izadi, Sang Uk Lee, Ping Tan, and Stephen Lin. Estimation of intrinsic image sequences from image+depth video. In *European Conference on Computer Vision (ECCV)*, 2012.

[17] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3):70:1–70:9, 2007. doi: 10.1145/1275808.1276464. URL `http://doi.acm.org/10.1145/1275808.1276464`.

[18] M Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of SIGGRAPH 96*, Annual Conference Series, pages 31–42, 1996. ISBN 0897917464. URL `http://dl.acm.org/citation.cfm?id=237199`.

[19] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H. Chen. Programmable aperture photography: Multiplexed light field acquisition. *ACM Transactions on Graphics*, 27(3):55:1–55:10, 2008. doi: 10.1145/1399504.1360654. URL `http://doi.acm.org/10.1145/1399504.1360654`.

[20] Stephen Lin and Shree Nayar. Coded aperture pairs for depth from defocus. *IEEE 12th International Conference on Computer Vision (ICCV)*, pages 325–332, September 2009. doi: 10.1109/ICCV.2009.5459268. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5459268`.

[21] Jorge Lopez-Moreno, Jorge Jimenez, Sunil Hadap, Erik Reinhard, Ken Anjyo, and Diego Gutierrez. Non-photorealistic, depth-based image editing. *Computers & Graphics*, 35(1): 99–111, February 2011.

[22] Francesc Moreno-Noguer, Peter N. Belhumeur, and Shree K. Nayar. Active refocusing of images and videos. *ACM Transactions on Graphics*, 26(3):67:1–67:9, 2007. doi: 10.1145/1275808.1276461. URL `http://portal.acm.org/citation.cfm?doid=1275808.1276461`.

[23] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02, Stanford CS, April 2005. URL `http://graphics.stanford.edu/papers/lfcamera/`.

[24] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. *ACM Transactions on Graphics*, pages 52:1–52:10, 2007. doi: 10.1145/1275808.1276442. URL `http://doi.acm.org/10.1145/1275808.1276442`.

[25] LJ Reese and WA Barrett. Image editing with intelligent paint. In *Proceedings of Eurographics*, volume 21, pages 714–724, 2002.

[26] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002. ISSN 0920-5691. doi: 10.1023/A:1014573219977. URL `http://dx.doi.org/10.1023/A:1014573219977`.

[27] Amnon Shashua and Tammy Riklin-Raviv. The quotient image: Class-based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 23(2):129–139, February 2001. ISSN 0162-8828. doi: 10.1109/34.908964. URL http://dx.doi.org/10.1109/34.908964.

[28] Marshall F. Tappen, William T. Freeman, and Edward H. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, September 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.185. URL http://dx.doi.org/10.1109/TPAMI.2005.185.

[29] Corey Toler-Franklin, Adam Finkelstein, and Szymon Rusinkiewicz. Illustration of complex real-world objects using images with normals. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, August 2007.

[30] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics*, 26(3):69:1–69:12, 2007. doi: 10.1145/1275808.1276463. URL http://doi.acm.org/10.1145/1275808.1276463.

[31] Xuan Yu, Rui Wang, and Jingyi Yu. Real-time depth of field rendering via dynamic light field generation and filtering. *Computer Graphics Forum*, 29(7):2099–2107, September 2010. ISSN 01677055. doi: 10.1111/j.1467-8659.2010.01797.x. URL http://doi.wiley.com/10.1111/j.1467-8659.2010.01797.x.

[32] Z Yu, Christopher Thorpe, Xuan Yu, S Grauer-Gray, F Li, and J Yu. Dynamic depth of field on live video streams: A stereo solution. In *Computer Graphics International*, 2011. URL http://www.eecis.udel.edu/~jye/lab_research/11/cgi11.pdf.

[33] Guofeng Zhang, Zilong Dong, Jiaya Jia, Liang Wan, Tien-Tsin Wong, and Hujun Bao. Refilming with depth-inferred videos. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):828–840, 2009.