LIGHT FIELD IMAGING

by

Zhan Yu

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer and Information Sciences

Fall 2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A light field captures a dense set of rays as scene descriptions in place of geometry. Recent advances on computational imaging have enabled novel and efficient light field acquisition devices. By mounting an microlense/lenslet array in front of an ultra-high resolution sensor (e.g., 11 megapixels), Lytro and Raytrix cameras are able to capture a light field in a single shot. However, the effective resolution is reduced by the number of microlenses. For example, the resulting image at a desired focal plane has only 1080×1080 pixels or roughly 1.2 megapixels, which is too low for photographic uses and computer vision tasks. Another drawback is low angular resolution on the captured light fields, usually less than 10×10 for each spatial sample, resulting in aliasing artifacts on the rendered image. Finally, the huge amount of the data in each captured light field (larger than 20 MB per frame) prohibits video capturing.

This dissertation focuses on exploring new image processing algorithms and camera designs to improve the spatial, angular, and temporal resolution of light field imaging.

Spatial Resolution: We develop a simple but effective technique for improving the image resolution of the plenoptic camera by maneuvering the demosaicing process. We first show that the traditional solution by demosaicing each individual microlense image and then blending them for view synthesis is suboptimal. We instead propose to demosaic the synthesized view at the rendering stage to obtain a higher resolution color result. We show that my solution can achieve visible resolution enhancement on dynamic refocusing and depth-assisted deep focus rendering.

Angular Resolution: We explore geometric structures of 3D lines in ray space for improving light field triangulation. The triangulation problem aims to fill in the ray space with continuous and non-overlapping simplices anchored at sampled points

(rays). Such a triangulation provides a piecewise-linear interpolant useful for light field super-resolution. We show that the light field space is largely bilinear due to 3D line segments in the scene, and direct triangulation of these bilinear subspaces leads to large errors. We instead present a simple but effective algorithm to first map bilinear subspaces to line and surface constraints and then apply Constrained Delaunay Triangulation (CDT).

The depth of each sample is required as a guidance to correctly conduct light field superresolution. To improve the current depth estimation algorithms, we propose two solutions: 1) We analyze the behavior of pixels under severe occlusion and show that it is possible to distinguish different depth layers based on statistics. Instead, we propose an iterative process to resolve occlusion. 2) We explore geometric structures of 3D lines in 4D ray space for improving light field stereo matching. We add the bilinear property of 3D lines as an additional constraint for the multi-view graph-cut framework.

Spatial-Angular Resolution: We present a unified framework to enhance spatial-angular resolution based on multiple light fields. Our solution first estimates the registration among the captured light fields, and then conducts projective warping to find the common subspaces among the light fields. Finally, our solution maps the inconsistency in the subspaces to the 4D graph-cut framework and finds the best seam to quilt those light fields.

Temporal Resolution: We construct a hybrid-resolution stereo camera system for producing the light field. Our system couples a high-res/low-res camera pair to replace the bulky camera array system. With the input stereo pair, we recover a low-resolution disparity map and upsample it via fast cross bilateral filters. We subsequently use the recovered high-resolution disparity map and its corresponding video frame to synthesize a light field using GPU-based disparity warping. We also use the image-space filtering technique to reduce aliasing. Finally, we generate racking focus and tracking focus effects using light field rendering. Compared with Lytro, our solution can produce images at the full resolution of the view camera.

Chapter 1

INTRODUCTION

Rays are directed lines in 3D space. They represent the visual information about a scene by their associated radiance function [3]. A light field [64, 42] captures a dense set of rays as scene descriptions in place of geometry. To represent each ray, a light field uses a two-plane parametrization (2PP). Every ray is parameterized by its intersections with two parallel planes: $[s, t]$ as the intersection with the first plane Π_{st} and $[u, v]$ as the second with Π_{uv}. Rays in a light field hence form a 4D space.

An important application of light fields is the light field rendering. Conceptually, one can obtain any view of the scene by extracting appropriate 2D slices from the 4D light field [64, 42]. Different parameterization of the light field and slices could result in perspective, orthographic, cross-slit [139], and multi-perspective [87] views. One can also synthesize dynamic depth of field (DoF) effects of a camera with finite aperture by integrating appropriate 4D subsets of a light field. In this case, different parameterizations of the light field correspond to views focusing on different fronto-parallel planes [48] or oblique planes [116] in the scene.

To capture the light field, numerous light field imaging systems have been built based on the idea of integral photography [68]. The Stanford light field camera array [122, 123, 114, 115] is a two dimensional grid composed of 128 1.3 megapixel firewire cameras which stream live video to a stripped disk array. The large volume of data generated by this array forces the DoF effect to be rendered in post processing rather than in real-time. Furthermore, the system infrastructure such as the camera grid, interconnects, and workstations are bulky, making it less suitable for on-site tasks. The MIT light field camera array [127] uses a smaller grid of 64 1.3 megapixel USB

webcams instead of firewire cameras and is capable of synthesizing real-time dynamic DoF with sacrificed image quality.

Recent realization of the hand held plenoptic/light field cameras [75, 70, 2, 39] replace the bulky camera array by coupling a microlens array with the mainlens to capture 4D radiance about a scene. Similar to the camera array, a light field camera, in essence, is a single-shot, multi-view acquisition device. Each captured microlens image maps to a perspective view from a different location in the scene. To densely sample the angular information, the commercial Lytro light field camera uses a 11 megapixel sensor to capture 0.11 million spatial samples and 100 angular samples on the plane of the microlens array. To achieve higher spatial resolution near the focal plane of the mainlens, Raytrix R11 camera uses a 10.7 megapixel sensor with increased spatial resolution at 0.47 million samples on the microlens array plane and reduced angular resolution at 23 samples per microlens.

Although impressive progresses on the sampling efficiency and imaging quality have been achieved based on recent light field imaging designs [75, 70, 2, 39, 118, 10], the problem of low spatial/angular resolution remains as an open but most challenging problem. The problem is inherent to the light field camera design – when using a 2D sensor to capture a 4D light field, we inevitably need to trade off between the angular (i.e., the number of views) and the spatial (the resolution of each view) resolutions [39]. A low angular resolution will lead to severe aliasing artifacts in refocusing and a low spatial resolution will produce images with low quality. Directly applying image superresolution techniques [17, 121] has limited capability of improving the image quality. Finally, nearly all light field cameras by far can only capture static images, i.e., they have extremely low temporal resolution. This is due to the huge amount of the data in each captured light field (larger than 20 MB per frame) which prohibits real-time video streaming and processing.

1.1 Dissertation Statement

This dissertation focuses on exploring new image processing algorithms and camera designs to improve the spatial, angular, and temporal resolution of light field imaging.

1.1.1 Spatial Resolution

To improve the image resolution of the light field camera, we develop a simple but effective technique by using a novel demosaicing process. A light field camera, same as traditional color cameras, captures color information with a Color Filter Array (CFA) masking the sensor pixels. We first show that the traditional solution [75, 70, 2, 39] that demosaics each individual microlens image and then blends them for rendering is suboptimal. In particular, this demosaicing process damages high frequency information recorded by each microlens image, hence greatly degrading the achievable resolution of the final photograph. We instead perform demosaicing on the synthesized color photograph at each refocusing plane. Specifically, we first reparameterize the light field to the desired focal plane and then apply frequency domain plenoptic resampling. A full resolution color filtered image is then created by performing a 2D integral projection from the reparameterized light field. Demosaicing is performed as a last step to obtain the final color result.

1.1.2 Angular Resolution

To increase the angular resolution, we present a new light field triangulation technique. The triangulation provides a natural anisotropic reconstruction kernel: any point in the space can be approximated using a convex combination of the enclosing simplexs vertices (samples). The simplest triangulation method is to apply high dimensional Delauney triangulation [31]. Such triangulations produce simplices (or pentatopes if in 4D) of "good shapes". However, triangulating the light field as such leads to severe aliasing. A better approach is to align simplices with ray geometry of

3D scene.

Line Assisted Light Field Triangulation We explore ray structures of a specific scene geometry, 3D line segments. We show that it is important to handle non-linear (bilinear) ray structures in order to properly triangulate the light field ray space and brute-force triangulating the light field causes large errors. In particular, we show that 3D lines in the light field follow bilinear constraints hence we propose new constrained triangulation methods to resolve this issue in 2D, 3D and 4D light fields. Specifically, we can first estimate the disparity (depth) of the feature pixels (rays), then map them to the edge constraints, and finally apply Constrained Delaunay Triangulation (CDT) [97]. We show this approach is still insufficient to produce high quality triangulations: the light field space contains a large amount of non-linear, or more precisely, bilinear substructures that correspond to 3D line segments. Brute-force triangulation of these bilinear structures leads to large errors and visual artifacts. We instead present a new solution that combines the bilinear and edge constraints for CDT.

Improved Light Field Stereo Matching To conduct light field triangulation, we need to first obtain a high quality disparity/correspondence map. We present two novel solutions.

Light Field Stereo In The Case Of Occlusions: One of the obstacles to recover a high-resolution depth map is occlusion. Traditional methods rely on either empirical or statistical solutions. We first analyze the behavior of pixels in such situations. We show that even under severe occlusion, one can still distinguish different depth layers based on statistics. To robustly resolve occlusion, we apply an iterative plane sweeping from the closest depth layer to the furthest, so that the occlusion pixels will be masked out when estimating local minima. However, pixels on constant color surfaces tend to choose small disparity since they will lead to small variance. To avoid these trivial solutions, we further propose a global optimization solution and an edge mask solution. Experimental results show that our algorithm is able to recover accurate depth map

from the light field images captured by plenoptic cameras.

Line Constrained Light Field Stereo Matching: Our ray geometry analysis of 3D lines also leads to a new light field stereo algorithm. We first introduce a new $\mathcal{F}^3$ energy term to preserve disparity consistency along line segments. We then modify the binocular stereo graph via the general purpose graph construction framework [53] and solve it using the extended Quadratic Pseudo-Boolean Optimization algorithm [91]. We validate our approach on Middlebury datasets, Stanford light field datasets [112] and real light field data acquired by the Lytro camera [71]. Experiments show that both our light field triangulation and stereo matching algorithms outperform state-of-the-art solutions in accuracy and visual quality.

1.1.3 A Unified Spatial-Angular Resolution

We further explore fusing multiple light fields under a common framework to simultaneously increase the spatial and angular resolution. We call this technique "Light Field Quilting". Given N captured low resolution 4D light fields L^1 to L^N with overlapped subspaces, our solution *quilts* them into a "super" light field with a higher resolution by finding smooth cuts in the overlapped subspaces. We start by modeling the registration between light fields as 5D homography matrices and then compute the homography by matching scale-invariant feature transform (SIFT) image features. Next, we iteratively warp the each light field towards its neighbor to find the overlapped subspaces. To find a cut in each overlapped subspace, we build a 4D light field graph and apply graph-cut optimization to find the optimal quilting paths. Since light fields are high dimensional, computing the cuts using graph-cut can be slow. We therefore employ a hierarchical approach [6] that uses the graph-cut result at a coarser resolution to prune the graph at a finer resolution in order to speed up the overall graph-cut speed of the high dimensional light field graph. Our approach can enhance light field resolutions in specific dimensions for various applications. For example, we can create a wide horizontal FoV light field from a series of light fields captured with a rotational light field camera. We can also create a megapixel (spatial resolution) light

field from a group of low spatial resolution light fields captured on a two dimensional angular grid. Finally, we can enhance the bokeh and parallax by either translating the light field camera or orbiting it around the object of interest.

1.1.4 Temporal Resolution

The high dimensionality of light fields prohibits continuous capturing by light field cameras. Alternatively, we construct a hybrid-resolution stereo camera system by coupling a high-res/low-res camera pair. We recover a low-res disparity map based on each pair of images and upsample it via fast cross bilateral filters. We then subsequently use the recovered high-resolution disparity map and its corresponding video frame to synthesize a light field. We implement a GPU-based disparity warping scheme and exploit atomic operations to resolve visibility. To reduce aliasing, we present an image-space filtering technique that compensates for spatial undersampling using mipmapping. Finally, we generate dynamic DoF effects using light field rendering. Our system can produce racking and tracking focus effects for at resolution of 640×480 at 15 frame per second.

1.2 Contributions

This dissertation makes the following contributions to the light field imaging.

1.2.1 Spatial Resolution

We develop a simple but effective technique for improving the image resolution of the plenoptic camera by maneuvering the demosaicing process. We first show that the traditional solution by demosaicing each individual microlens image and then blending them for view synthesis is suboptimal. We instead propose to demosaic the synthesized view at the rendering stage to obtain a higher resolution color result. We show that our solution can achieve visible resolution enhancement on dynamic refocusing and depth-assisted deep focus rendering.

1.2.2 Angular Resolution

We explore geometric structures of 3D lines in ray space for improving light field triangulation. We show that the light field space is largely bilinear due to 3D line segments in the scene, and direct triangulation of these bilinear subspaces leads to large errors. We instead present a simple but effective algorithm to first map bilinear subspaces to line and surface constraints and then apply Constrained Delaunay Triangulation (CDT).

The depth of each sample is required as a guidance to correctly conduct light field superresolution. To improve the current depth estimation algorithms, we propose two solutions: 1) We analyze the behavior of pixels under severe occlusion. We show that it is difficult to distinguish different depth layers based merely on statistics. Instead, we propose an iterative process to resolve occlusion. 2) We explore geometric structures of 3D lines in 4D ray space for improving light field stereo matching. We add the bilinear property of 3D lines as an additional constraint for the multi-view graph cuts framework.

1.2.3 Spatial-Angular Resolution

We present a unified framework to enhance spatial-angular resolution based on multiple light fields. Our solution first estimates the registration among the captured light fields, and then conducts projective warping to find the common subspaces among the light fields. Finally, our solution maps the inconsistency in the subspaces to the 4D graph cuts framework and finds the best seam to quilt those light fields.

1.2.4 Temporal Resolution

We construct a hybrid-resolution stereo camera system for synthesizing the light field in real time. Our system couples a high-res/low-res camera pair to replace the bulky camera array system. With the input stereo pair, we recover a low-resolution disparity map and upsample it via fast cross bilateral filters. We subsequently use the recovered high-resolution disparity map and its corresponding video frame to synthesize

a light field using GPU-based disparity warping. We also use the image-space filtering technique to reduce aliasing. Finally, we generate racking focus and tracking focus effects using light field rendering. Compared with Lytro, our solution can produce images at the full resolution of the view camera.

1.3 Blueprint of the Dissertation

This dissertation is organized as follows. Chapter 2 reviews the background and previous work on modeling the light field space, designing light field cameras and applying light field imaging on 3D reconstruction, dynamic depth of field rendering, and novel view synthesis. I also highlight their limitations caused by spatial, angular and temporal resolutions.

Chapter 3 discusses an approach for improving spatial resolution in light field photography by maneuvering the demosaicing stage of traditional light field rendering.

Chapter 4 describes a light field triangulation technique for enhancing the angular resolution of a captured light field.

Chapter 5 introduces two approaches that use light field stereo matching for depth/disparity estimation. The first approach iteratively resolve the occlusion problem during the disparity estimation. The second approach employ the 3D line constraints/priors to improve the stereo matching process.

Chapter 6 presents a high-dimensional image based rendering technique which takes multiple light fields as inputs and generates new light fields with higher spatial and angular resolution as outputs.

Chapter 7 develops a stereo based light field camera that can acquire dynamic light field videos and synthesize racking focus and tracking focus effect in real time.

Chapter 8 proposes an alternative light field imaging solution using a catadioptric mirror array and discuss its unique advantage on low light imaging.

Chapter 9 concludes the dissertation and discusses future extensions.

Chapter 2

PREVIOUS WORK

In this chapter, I briefly review the background and previous work on light field cameras. We first review the work on modeling the light field space. We then discuss existing work in light field cameras and their applications in computer vision and graphics. These include 3D reconstruction, dynamic depth of field rendering, and novel view synthesis.

2.1 Light Fields

The concept of light field can be traced back to 1936 by Arun Gershun in a classic paper on the radiometric properties of light in 3D space. The radiance along all rays in a region of 3D space illuminated by an unchanging arrangement of lights is called the plenoptic function [3]. The plenoptic function is the 5-dimensional function representing the intensity or chromacity of the light observed from every position and direction in 3-dimensional space.

In a plenoptic function, if the region of interest contains a concave object, then light leaving one point on the object may travel only a short distance before being blocked by another point on the object. There, it is difficult to measure the function in such a region. However, if we restrict ourselves to locations outside the convex hull of the object, then we can measure the plenoptic function by many ways, e.g., taking many photos using a camera. Moreover, since the radiance along a ray remains constant, this function contains one dimensional redundant information (along the ray). Therefore we can simplify the plenoptic function with a 4-dimensional function (that is, a function of points in a particular 4-dimensional manifold), as long as we ignore both rays flowing towards the object and rays emanating from the object on the opposite

side. Parry Moon dubbed this function the photic field (1981), while researchers in computer graphics call it the 4D light field [64] or Lumigraph [42]. Formally, the 4D light field is defined as radiance along rays in empty space.

In essence, light fields are simply data structures that support the efficient interpolation of the radiance estimates along specified rays. In conventional light fields, a 2PP is commonly used to represent rays, where each ray is parameterized in the coordinates of the camera plane (Π_{uv}) and a image plane (Π_{st}). Rays in the light field hence form a 4D space. A closely related representation to a light field is the lumigraph [42]. A lumigraph incorporates an approximate geometric model, or proxy, in the interpolation process, which significantly improves the quality of the reconstruction. To acquire better results with a severely undersampled light field, scam light field rendering [131] combined both parameterizations in their algorithm. More recently, surface light fields [124] suggested an alternative ray parameterization where rays are parameterized over the surface of a pre-scanned geometry model.

2.2 Acquisition of Light Fields

The light field acquisition devices range from a robotically controlled moving camera [113], a dense array of cameras [122, 127], to hand-held light field cameras [76, 70, 2, 39] and light field microscopes. We categorize them into three categories by their main optical components.

2.2.1 Light Field Camera Array

The most straightforward scheme to capture the light field is to move a camera along a 2D path to sample the 4D ray space [48, 64]. Although this method is simple and easy to implement, it is only suitable for acquiring static scenes. Wilburn et al. [122, 123] instead built a 2- dimensional grid composed of 128 1.3 megapixel firewire cameras which stream live video to a stripped disk array. The large volume of data generated by this array forces the DoF effect to be rendered in post processing rather than in real-time. The MIT light field camera array [127] uses a smaller grid of 64

1.3 megapixel USB webcams instead of firewire cameras and is capable of synthesizing real-time dynamic DoF effects. Both systems, however, still suffer from spatial aliasing because of the baseline between neighboring cameras. Moreover, constructing such a light field camera array is extremely time and effort consuming and requires substantial amount of engineering.

2.2.2 Hand-Held Light Field Camera

To replace the bulky system, various solutions have been proposed. A notable example is the realization of the hand-held plenoptic/light field camera [75, 70, 2, 39], a camera that uses a microlens array to capture 4D light field about a scene. In order to overcome the spatio-angular tradeoff, an ultra-high resolution sensor is commonly used. The resulting images, however, are still at a disappointingly low resolution. For example, Ng [75] improved the traditional plenoptic camera design and introduced new methods for computational refocusing. This plenoptic camera places the microlens array at plane Π in front of the camera sensor to separate converging rays (Fig. 2.1(a)). Specifically, the sensor is located at the focal plane of each microlens so that each microlens is focusing at its optical infinity (main lens principal plane). The F-numbers of the main lens and each microlens are matched to avoid "Cross-Talk" among microlens images. This design achieves high angular resolution by sacrificing spatial resolution near Π. Lumsdaine et al. [70] introduced another design by focusing the microlens array on Π and correspondingly adjusting the position of the microlens array and the sensor (Fig. 2.1(b)). In this case each microlens image will have samples with more spatial resolution and less angular resolution on Π. Therefore this design is capable of producing higher resolution results when focusing near the sampled image plane. However, the lower angular resolution may cause ringing artifacts in out of focus regions of the rendered image.

Most recently, based on Ng's design, the commercial Lytro light field camera uses an 11 mega pixel sensor to capture 0.11 million spatial samples and 100 angular samples on the plane of the lenslet array. Raytrix R11 camera follows the design of

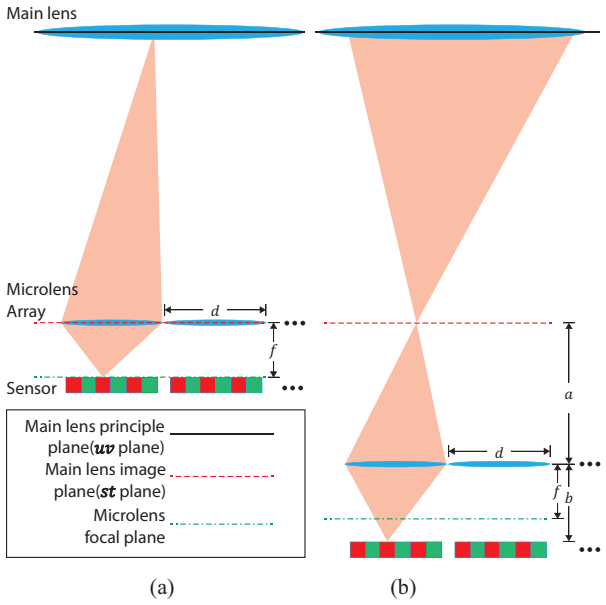


Figure 2.1: Plenotpic Camera Designs. (a) Ng. (b) Lumsdaine et al.

Lumsdaine and uses a 10.7 mega pixel sensor to capture 0.47 million spatial samples and 23 angular samples. However, these designs still suffer from the spatial angular tradeoff caused by multiplexing the 4D light field onto a 2D sensor: a low angular resolution will lead to severe aliasing artifacts in refocusing and a low spatial resolution will produce images with low quality.

2.2.3 Mask Based Light Field Camera

Instead of using a lenslet array to separate light arriving at the same pixel from different directions, Veeraraghavan et al. [Veeraraghavan07] proposed reversible modulation of 4D light field by inserting a patterned planar mask in the optical path of a lens based camera. The patterned mask attenuates light rays inside the camera instead of bending them, and the attenuation recoverably encodes the ray on the 2D sensor. This process can be viewed as heterodyning the incoming light field in the frequency domain. To recover the light field, they first transform the captured 2D

image into frequency domain and then rearrange the tiles of the 2D Fourier transform into 4D space. Finally, the light field of the scene is computed by taking the inverse 4D Fourier transform. Further, they can insert the mask at different location along the optical path of the camera to achieve dynamic frequency modulation. However, the mask partially blocks out the incoming light and greatly reduces light efficiency.

2.2.4 Mirror Based Light Field Camera

It is also possible to acquire the light field using a catadioptric mirror array. Unger et al. [111] combined a high resolution tele-lens camera and an array of spherical mirrors to capture the incident light field. The use of mirror arrays instead of lenslet arrays has its advantages: it avoids chromatic aberrations caused by refraction, it does not require elaborate calibration between the lenslet array and the sensor, it captures images at a wide FoV, and it is less expensive and reconfigurable. The disadvantages are three-fold: First, each mirror image is non-pinhole and therefore requires conducting forward projection for associating the reflection rays with 3D points. Second, the sampling of the light field is nonuniform. Third, a large F-number is required to avoid defocus blur on the mirror, hence reducing the light efficiency.

Two notable examples of these systems are the spherical mirror arrays by Ding et al. [34] and Taguchi et al. [108]. In [34], the authors applied the GLC-based forward projection on multi-view space carving for reconstructing the 3D scene. Taguchi et al. [108] developed both a mirror array and a refractive sphere array and applied the axial cone modeling for fast forward projection using GPU. They have shown various applications including distortion correction and light field rendering.

2.3 Light Field Rendering

An important application of light fields is the light field rendering. We briefly introduce three different trends of this application.

2.3.1 Spatial Domain Rendering

Isakesen et al. [48] first proposed to apply wide aperture filter to synthesize DoF from the light field. The pixel values on the image are proportional to the irradiance [103] received at the sensor, computed as a weighted integral of the incoming radiance through the lens:

$$I(s,t) \approx \iint L_{in}(u,v,s,t) cos^4 \Phi \mathrm{d}u\mathrm{d}v, \tag{2.1}$$

where $I(s,t)$ is the irradiance received at pixel (s,t), and Φ is the angle between a ray $L_{in}(u,v,s,t)$ and the sensor plane normal. This integral can be estimated as summations of the radiance along the sampled rays:

$$I(s,t) \approx \sum_{(u,v)} L_{in}(u,v,s,t) cos^4 \Phi. \tag{2.2}$$

Isaksen et al. [48] directly applied Eqn. 2.2 to render the DoF effects. Specifically, from each pixel $p(s,t)$ on the sensor, they first trace out a ray through lens center o to find its intersection Q with the focal plane. Then, Q is backprojected onto all light field cameras and blended with the corresponding pixels. Finally, the pixel value is computed by the weighted average of its corresponding pixels.

2.3.2 GPU Based Rendering

The spatial rendering lends itself well to parallel processing. Recently, Yu et al. [PG 10] proposed a new GPU based algorithm for efficient rendering of high-quality dynamic DoF effects from a single view and its depth information. Specifically, they first reconstruct the light field by warping the reference view to nearby views with the depth information, and exploit the atomic operations to resolve visibility when multiple pixels warp to the same image location. They then directly synthesize DoF effects from the sampled light field. To reduce aliasing artifacts, they rely on image-space filtering technique which compensates for spatial undersampling using mipmapping. More recently, Lumsdaine et al. [Plenoptic Rendering GPU] presented a progression of rendering approaches for focused plenoptic camera data and analyzed their performance

on popular GPU-based systems. They are able to render 39 mega-pixel light field data to 2 mega-pixel images at over 500 frame per second.

2.3.3 Frequency Domain Rendering

Inspired by the well-known Fourier Slice Theorem, Ng [75] presented a new Fourier Slice Photography algorithm for light field rendering in the frequency domain. The algorithm is based on the Fourier Photography theory which was derived from the geometrical optics of image formation. The theory states that in the frequency domain, a photograph formed with a full lens aperture is a 2D slice in the 4D light field. Photographs focused at different depths correspond to slices at different trajectories in the 4D space. This algorithm is significant faster than spatial-domain representation ($O(n^2 \log n)$ vs. $O(n^4)$). However, the preprocessing cost is relatively large ($O(n^4 \log n)$ for 4D fast Fourier transform) and the light field must be uniformly sampled.

2.4 Geometric Structures

The light field ray space is a vector space. Any linear combination of the $[s, t, u, v]$ coordinate of two rays is still a valid ray. To study ray geometry of local ray tangent plane, Yu and McMillan [130] developed a new camera model called the General Linear Camera (GLC). GLCs are 2D planar ray manifolds which can apparently describe the traditional pinhole, orthographic, pushbroom, and XSlit cameras. A GLC is defined as the affine combination of three generator rays $r_i = [u_i, v_i, s_i, t_i], i = 1, 2, 3$:

$$r = \alpha[u_1, v_1, s_1, t_1] + \beta[u_2, v_2, s_2, t_2] + (1 - \alpha - \beta)[u_3, v_3, s_3, t_3] \tag{2.3}$$

For example, in the ray tangent plane analysis, the three ray generators are chosen as $r, r + d_1$ and $r + d_2$. Similar to defining a 2D plane in 3D space, a GLC is the affine combination of three rays. Their studies have shown that the light field ray space is mostly linear: scene geometry such as 3D points or parallel directions maps to GLCs. There also exists non-linear structures in the light field. For example, 3D lines parallel to the 2PP maps to hyperplanes in the light field, while 3D lines not parallel to the light

field maps to bilinear surfaces. We will elaborate on 3D lines and explore the usage of bilinear structures in the context of light field triangulation and stereo matching in Chapter 4.

2.5 Frequency Structures

Recent studies have further characterized the frequency attributes of ray space.

Chai et al. [25] mathematically derived the analytical functions to determine the minimum sampling rate for light field rendering. They discovered that spectral support of a light field signal is bounded by the minimum and maximum depths only, not depth variations in the scene. They further obtained the minimum sampling rate for light field rendering by compacting the replicas of the spectral support of the sampled light field within the smallest interval. They also designed reconstruction filters based on an optimal and constant depth to reduce aliasing artifacts in light field rendering.

Also related to light field structures is light transport. Durand et al. [SIGGRAPH '05] presented a signal-processing framework for light transport. They studied the frequency content of radiance and how it is affected by phenomena such as shading, occlusion, and travel in free space. They characterized how the radiance signal is modified as light propagates and interacts with objects. In particular, they show that occlusion amounts in the frequency domain to a convolution by the frequency content of the blocker. Propagation in free space corresponds to a shear in the space-angle frequency domain, while reflection on curved objects performs a different shear along the angular frequency axis. Their extension shows how the spatial components of lighting are affected by this angular convolution. They also showed that their signal-processing framework predicts the characteristics of interactions such as caustics, and the disappearance of the shadows of small features. Predictions on the frequency spectrum of the radiance function can then be used to control sampling rates or the choice of reconstruction kernels for rendering. Other potential applications include pre-computed radiance transfer and inverse rendering.

Georgiev et al. [2007] presented a theory that encompasses both microlens based and mask based light field cameras into a single frequency domain mathematical formalism. In particular, inspired by the heterodyning concept, they derive a theory of recovering the 4D spatial and angular information from the multiplexed 2D frequency representation which applies for both microlens based and mask based light field cameras. Moreover, their theory also suggested new designs for light field cameras.

More recently, Levin et al. studied the designs of effective extended-DoF systems by analyzing defocus kernels in the 4D light field space in the frequency domain. Specifically, they showed that only a low-dimensional 3D manifold contributes to focus. Thus, imaging systems should concentrate their limited energy on this manifold in order to maximize the defocus spectrum. They also showed that conventional computational imaging systems either spend energy outside the focal manifold or do not achieve a high spectrum over the DoF. Guided by this analysis they further introduced the lattice-focal lens, which concentrates energy at the low-dimensional focal manifold and achieves a higher power spectrum than previous designs. They also built a prototype lattice-focal lens and presented extended depth of field results.

2.6 Improving Light Field Resolutions

Traditional light field parameterization denotes the samples on the uv/camera plane as spatial samples, and samples on st/sensor plane as angular samples. Moreover, each captured light field at a different time is a 4D sample along the temporal dimension. Hence light field images form snap shots of a 5D spatial-angular-temporal space.

Due to the high dimensionality of this space, most light field imaging devices focus on maximizing spatial-angular resolution to achieve good image quality. For example the light field camera arrays captures high spatial-angular resolution light fields. However the huge amount of data prohibits continuous capturing and real time rendering. To reduce the form factor, the commodity light field cameras either sacrifice

the spatial resolution or the angular resolution. Even so, the current data bandwidth still does not support light field video capturing.

In this section, we categorize the recent work by improving light field resolution on spatial, angular, or temporal dimensions.

2.6.1 Angular Resolution

Since representing all light rays present in a scene is usually impractical or impossible, a real light field generally contains only a finite sampling of the rays. Thus, as with any discrete sampling of a continuous signal, we are faced with the undersampling problem in signal reconstruction.

If an angularly undersampled light field is rendered using the common linear interpolation method, the result will exhibit an aliasing artifact called "ghosting", where multiple copies of a single feature appear in the interpolated light field views. To reduce aliasing artifacts during interpolation, various light field reconstruction/filtering approaches have been proposed. Levoy and Hanrahan [64] pointed out that light field aliasing can be eliminated with proper pre-filtering. Isaksen et al. [48] later showed that pre-filtering has the undesirable side effect of requiring the pre-decision of which part of the scene can be rendered in focus during reconstruction. Reconstruction of an undersampled light field can also benefit from the depth information. Analyses of Gortler et al. [42] and Chai et al. [25] both showed that the introduction of depth ameliorates the aliasing artifacts. However, as pointed by Steward et al. [102], it is usually inconvenient to acquire depth information from the real scenes. They subsequently proposed a hybrid reconstruction filter that combines a full aperture kernel with a band-limited kernel in the frequency domain. The filter can recover more useful information without introducing aliasing. However, their technique cannot handle non-linear phenomenon such as occlusions.

More recently, Wanner and Goldlüecke [121] formulate the problem of angular superresolution as a continuous inverse problem, which allows them to correctly take into account foreshortening effects caused by scene geometry transformations. They

employ state-of-the-art convex optimization algorithms to fast minimize the superresolution model energy. However, their method require accurate depth estimation as the prior knowledge. Levin and Durand [63] use the dimensionality gap prior to recover the 4D light field from a 3D focal stack using linear view synthesis. Their method does not require depth estimation but the assumption of most energy lying in the 3D manifold of the light field limits the scenes to be Lambertian.

In this dissertation, we propose two approaches for angular superresolution. First, we explore improving the angular resolution of the captured light field by conducting a constrained light field triangulation. Comparing with traditional light field superresolution, our triangulation does not require dense correspondence information, hence has the potential of light field compression. Next, we present a light field quilting framework by fusing multiple captured light fields into a single higher angular resolution light field.

2.6.2 Spatial Resolution

To improve the spatial resolution of the light field, Bishop et al. [17] reconstruct each view at higher resolution by explicitly modeling the image formation process and incorporating priors such as Lambertianity and texture statistics. They then map this modeling onto a variational Bayesian framework and perform the superresolution. Their performance, however, is prior dependent. Georgiev et al. [41] applied demosaicing after plenoptic rendering to improve plenoptic superresolution. Their approach used a straightforward demosaicing scheme on the refocusing plane, resulting in significant color artifacts in out-of-focus regions of the rendered images.

There is also an emerging trend of reconstructing sparsely sampled light field for light field compression. Lehtinen et al. [62] explored the anisotropy in the temporal domain and enhanced the reconstruction quality by a large factor. Marwah et al. used an overcomplete dictionary to reconstruct a sparse coded LF. However, the performance of their method is largely related to the relevance of the dictionary with the scene.

Heide et al. applied Markov Chain Monte Carlo sampling instead of uniform sampling on the target light field for better reconstruction.

In this thesis, we present a more robust light field demosaicing approach to increase the spatial resolution. We theoretically analyze the resolution enhancement and sampling pattern on the refocusing plane compared with each microlens, we then conduct light field demosaicing coupled with a light field resampling to reduce the color artifacts. We also use our light field quilting framework to acquire a higher angular resolution light field from several captured light fields.

2.6.3 Temporal Resolution

Due to the large amount of data in each captured light field image, currently there are few practical solution for continuously capturing the light fields. To enable the acquisition of light field videos, the straight forward approach is to build a camera array. However, the Stanford camera array mentioned in Sec. 2.2.1 generates high resolution images but does not support realtime processing, while the MIT camera array supports dynamic DoF in realtime but produces low quality results.

More recently, Agrawal et al. [10] proposed a mask based optical design to achieve spatial-angular-temporal tradeoffs using a time-varying aperture mask and a static mask close to the sensor. Their design allows variable resolution tradeoff depending on the scene with two novel outputs: 1) 1D refocusing on an object moving in depth and 2) single-shot video capture. However, their method trades spatial-angular resolution for temporal resolution, hence the output video sequence is at a much lower resolution than the captured image. Moreover, the dynamic components in the scene lose refocus capability. Lastly, the masks on the aperture and the sensor greatly reduce the light efficiency of the design.

In this thesis, we present a much simpler solution based on stereo matching. Our system is able to generate dynamic DoF effect at full camera resolution with interactive speed. Comparing with light field cameras, our system is restricted to Lambertian scenes but better preserves the spatial resolution. Comparing with Agrawal et al. [10],

our system captures the scene at a higher frame rate while providing dynamic DoF capabilities.

Chapter 3

ENHANCING SPATIAL RESOLUTION VIA EFFECTIVE DEMOSAICING

In this chapter, I discuss how to enhance the spatial resolution in light field photography by maneuvering the demosaicing process. Specifically, we first show that the traditional solution by demosaicing each individual microlens image and then blending them for view synthesis is suboptimal. In particular, this demosaicing process often suffers from aliasing artifacts, and it damages high frequency information recorded by each microlens image hence degrades the image quality. We instead propose to demosaic the synthesized view at the rendering stage. Specifically, we first reparameterize the captured light field to the desired focal plane and then apply frequency domain plenoptic resampling. A full resolution color filtered image is then created by performing a 2D integral projection from the reparameterized light field. Finally, we conduct demosaicing to obtain the color result.

3.1 Image Demosaicing

While a significant amount of work on plenoptic cameras has been focusing on improving the image resolution [102, 17, 40], demosaicing remains as an understudied problem. Demosaicing, in essence, converts single-CCD color representations of one color channel per-pixel into full per-pixel RGB. The most popular type of CFA in current use is the Bayer filter [14]. Demosaicing a raw Bayer image requires an underlying image model to guide decisions for reconstructing the missing color channels: at every pixel only one color channel is sampled and therefore we need to use its nearby samples to reconstruct the other two channels. Many sequential methods [66, 49, 1, 47, 74] have been introduced based on the assumption that green channel is less aliased than

the other two due to higher sampling frequency. More sophisticated methods impose local gradients [69] or frequency statistics [67, 11, 35, 73] as constraints to improve the performance.

However, by far nearly all demosaicing techniques aim to process images captured by commodity digital cameras and very little work has been focused on developing solutions specifically for plenoptic cameras. Existing plenoptic cameras typically demosaic each individual microlens image and treat the captured plenoptic function as a captured RGB image. One exception is the paper by Georgiev et al. [41] that applies demosaicing after plenoptic rendering to improve plenoptic superresolution. The approach presented in [41] used a straightforward demosaicing which does not resample the light field, resulting in significant color artifacts in out-of-focus regions of the rendered images. Other related work is the spatial domain multi-frame demosaicing and super-resolution technique reported in [37]. However, their focus is to combine multiple low resolution images whereas we aim to manipulate demosaicing to improve refocused images produced by plenoptic rendering.

3.2 Image Demosaicing in a Plenoptic Camera

We start by analyzing the traditional image demosaicing on plenoptic cameras. Before proceeding with our analysis, we introduce our notation. Let $I(\boldsymbol{s})$ represent the irradiance of pixel s on the image plane Π and r_i represent the RGB radiance of a single ray captured by microlens m_i. I_i is the *ideal* optical RGB image at m_i. In real cameras, we get a color filtered image I_{fi} instead of I_i due to color filtering. For each color channel, I_{fi} can be viewed as an undersampled version of I_i in that channel. The demosaicing operator D upsamples I_{fi} to recover I_i.

3.2.1 Classical Rendering

The classical plenoptic rendering approach first applies demosaicing to each individual microlens image and then applies integral projection for refocusing. Let b denote the distance from the sensor to the microlens array and $\boldsymbol{s}_i$ denote the location

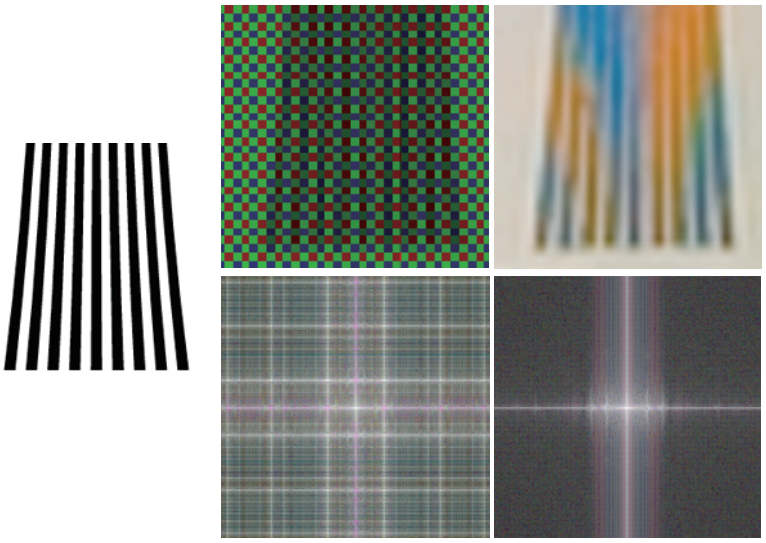


Figure 3.1: Artifacts on the captured light field introduced by classical demosaicing. (a) Ground Truth. (b) Raw microlens image and its frequency spectrum. (c) Demosaiced microlens image and its frequency spectrum.

of the optical center of m_i. In the discrete case, if we focus at Π with distance a to the microlens array, we can compute the irradiance $I'(\boldsymbol{s})$ by:

$$I'(\boldsymbol{s}) \approx \sum_i D(I_{fi}((\boldsymbol{s}_i - s)\frac{b}{a} + \boldsymbol{s}_i), \tag{3.1}$$

Let ω_i denote the highest frequency of I_i and ω denote the sampling frequency of I_{fi}. In the trivial case $(\forall i)[2\omega_i \leq \omega]$, we can completely recover the full frequency microlens images I_i and hence the refocused image. In the general case when $(\exists j)[2\omega_j > \omega]$, the spectrum of I_{fj} exhibits aliasing due to undersampling as shown in Fig. 3.1(b). In this case, the demosaic operator D is used to eliminate undersampling artifacts. However, D generally behaves as a low pass filter, indiscriminately removing high frequencies, thereby degrading the image sharpness of the final refocused image. Finally, if I_{fi} is severely undersampled, demosaicing (such as that performed by Adobe Photoshop Camera Raw) can introduce inconsistent color interpolation and cause color bleeding in the refocused image as shown in Fig. 3.1(c) (black and white patterns become

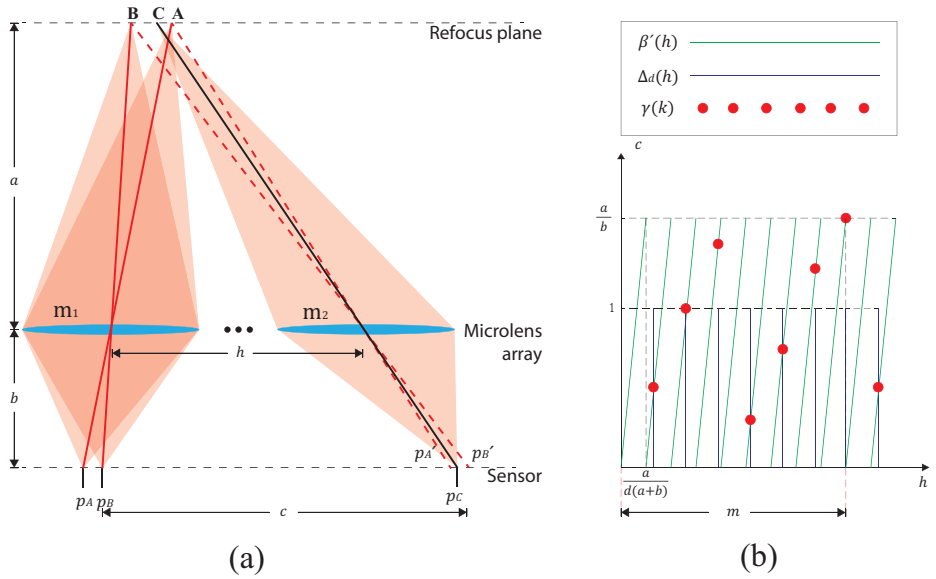


Figure 3.2: (a) Possible resolution enhancement on the refocus plane by projecting multiple microlens images. (b) Plots of function $\Delta_d(h)$, $\beta(h)$, and $\gamma(k)$.

colorful).

3.2.2 Resolution on the Refocus Plane

In this section, we provide a theoretical analysis to show that the projected image I_f on plane Π has a higher sampling frequency than any of the microlens images, hence performing demosaicing on I_f could greatly improve the image resolution. For simplicity, we model each microlens as a pinhole camera and only analyze rays passing through each optical center. Also for simplicity, we show only one spatial dimension s. Consider two adjacent pixels p_A and p_B ($p_A < p_B$) in a specific microlens m_1 that map to two points A and B on the target focal plane Π. Assume the distance between p_A and p_B is 1, the distance between two adjacent microlenses is d, Π lies at distance a to the microlens array, the sensor lies at distance b to the microlens array, and the spacing between m_1 and m_2 is h, as shown in Fig. 3.2(a). Note that since the pixel distance is vanishingly small compared with a, b, and d, we simply treat these latter quantities as integers.

Our goal is to study how many rays (pixels) from other microlenses would fall between A and B on Π. This number approximates the factor of resolution enhancement compared with the classical demosaicing followed by rendering approach. In order to estimate this number, we first introduce a function γ which maps the index of a given microlens to its sampling point between A and B. Since all the microlenses out of the minimum period T of γ are duplications of samples within T, we find out T of γ and use it as the upper bound of the resolution enhancement.

Note that for each microlens m_2 different from m_1, we can have at most 1 point between AB that maps to a pixel to m_2 as the length AB is preserved in all microlenses. Assume A and B map to points p'_A and p'_B in m_2, as shown in Fig. 3.2(a). Note that p'_A and p'_B may not be pixels. In the first case, p'_A and p'_B fall exactly on the pixels position. In that case, no additional rays (pixels) from m_2 would intersect the segment AB on plane Π. Therefore, m_2 would not contribute to enhancing the resolution between AB. Under similitude relationship, the conclusion holds for any pair of adjacent pixels in m_1 and m_2, i.e., m_2 would not contribute to enhancing the resolution to m_1's image.

In the second case, A and B do *not* coincide with pixels in m_k and there is exactly one point C between A and B that maps to a pixel in p_C in m_k. we call C a super-pixel as it will increase the resolution between AB. We can then compute $p_C = \frac{a+b}{a}h$ and the distance β between A and C on the focal plane as:

$$\beta(h) = (\frac{a+b}{a}h - \lfloor\frac{a+b}{a}h\rfloor)\frac{a}{b}. \tag{3.2}$$

Note that function $\beta(h)$ is a periodic function with a minimum period of $\frac{a}{a+b} < 1$. For each microlens, we can substitute its distance h into m_1 and compute the location of this super-pixel. If the super-pixels in some N microlenses have identical β values, then these microlenses only contribute 1 rather N super-pixels for enhancing the resolution between AB.

To finally compute the exact resolution enhancement, recall that in the microlens array setting, $h = kd$ from m_1, where k is some positive integer and $d > 1$. We can then concatenate the microlens sampling function (a Dirac comb) $\Delta_d(h)$ with the distance

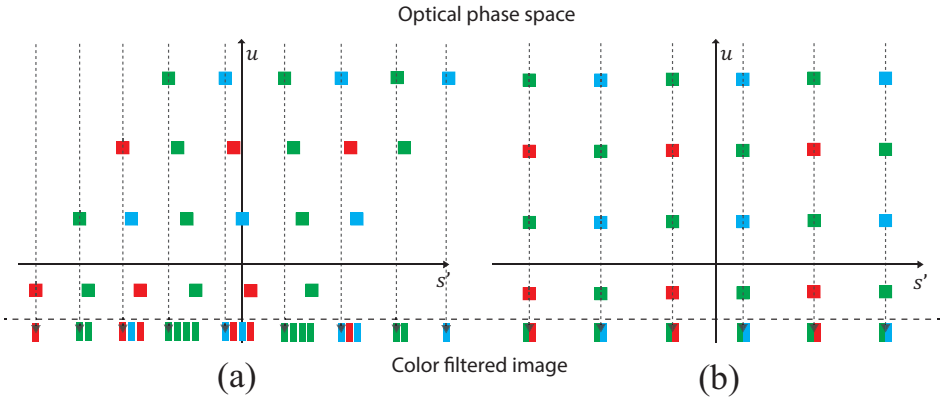


Figure 3.3: Optical phase space illustration of resampling the captured radiance. (a) Directly projecting the captured light field onto the refocus plane. (b) Projecting the resampled light field onto the refocus plane.

function $\beta(h)$ as: $\Delta_d(h) \cdot \beta(h)$. To further simplify, we can factor d into $\gamma(h)$ so that $\gamma(k) = \Delta(k) \cdot \beta'(k)$. where $\beta'(k)$ has period $\frac{a}{d(a+b)}$ and $\Delta(k)$ has period 1.

Clearly $\gamma(k)$ has minimum integer period equal to the least common integer multiple of $\frac{a}{d(a+b)}$ and 1. We rewrite $\frac{a}{d(a+b)}$ as an irreducible fraction two integers $\frac{m}{n}$. Thus, $S'(k)$ has minimum integer period $m = \frac{a}{gcd(a,d(a+b))}$, where gcd denotes the greatest common divisor operator (Fig. 3.2(b)).

Note that the number of microlenses sharing a field of view also constrains the number of distinct samples between p_A and p_B. Since the shift from one microlens image to another for any point p on Π is $\Delta = d\frac{b}{a}$, we can compute the number of microlens covering p as:

$$n_p = \lfloor \frac{d}{\Delta} \rfloor = \lfloor \frac{a-f}{f} \rfloor. \tag{3.3}$$

Combining with Equation 3.2 we obtain that the resolution enhancement factor from microlens image m_i to Π is equal to $min(m, n_p)$. Since $b = \frac{af}{a-f}$, this factor is controlled only by a and f, namely, the depth of the scene and the camera optics.

3.3 Demosaicing and Rendering on the Refocus Plane

Projecting samples of each microlens to the refocus plane Π gives us a higher resolution image I_f. However, as shown in Fig. 3.3(a), when the captured light field is transformed to Π for projection, as proposed by Georgiev et al. [41], the spacing of each color component is not uniform on I_f, resulting in random RGB patterns (Fig. 3.4(a)). This issue creates trouble for demosaicing I_f. Therefore a crucial step of our approach is to resample the light field with the parameterization of Π_{st} to achieve constant spacing on each dimension (Fig. 3.3(b)).

3.3.1 Resampling

We adopt a similar approach to that in [109], which was originally developed for multi-frame single channel image restoration. We use a frequency-domain approach to resample the 4D color filtered radiance. This simplifies to reconstructing a higher resolution color image by perfect registration with an array of low resolution color images taken at the same time in a 2D image restoration case.

Here we only consider the green rays. The other two channels can be computed in a similar manner. Suppose we have q microlenses. Each microlens captures a low resolution light field with $N_{\boldsymbol{s}}$ and $N_{\boldsymbol{u}}$ samples on each dimension. Let $r_o(\boldsymbol{s}, \boldsymbol{u})$ be the original green rays parameterized by Π_{st} and Π_{uv}. Given the distance a from Π_{st} to the microlens array, the registration of a recorded sub-light field r_i can be computed accurately as offsets $\sigma_{\boldsymbol{s}}, \sigma_{\boldsymbol{u}}$ on each dimension respectively. Therefore, the sampled rays by microlens m_i is $r_i(\boldsymbol{s}, \boldsymbol{u}) = r_o(\boldsymbol{s} + \sigma_{\boldsymbol{s}i}, \boldsymbol{u} + \sigma_{\boldsymbol{u}i})$. In frequency domain, this yields:

$$R_i(\boldsymbol{S}, \boldsymbol{U}) = e^{j2\pi(\sigma_{\boldsymbol{s}i}\boldsymbol{S} + \sigma_{\boldsymbol{u}i}\boldsymbol{U})} R_o(\boldsymbol{S}, \boldsymbol{U}) \tag{3.4}$$

where $R_o(\boldsymbol{S}, \boldsymbol{U})$ and $R_i(\boldsymbol{S}, \boldsymbol{U})$ are CFT of $r_o(\boldsymbol{s}, \boldsymbol{u})$ and $r_i(\boldsymbol{s}, \boldsymbol{u})$ respectively. Let pixels under m_i capture r_i with a uniform spacing $(T_{\boldsymbol{s}}, T_{\boldsymbol{u}})$, and $R_{d_i}(\Omega)$ be the discrete Fourier transform (DFT) of the rays recorded by i^{th} microlens at frequency $\Omega = (\omega_{\boldsymbol{s}}, \omega_{\boldsymbol{u}})$. From the aliasing relationship between CFT and DFT, $R_{d_i}(\Omega)$ satisfies the following equation:

$$R_{d_i}(\Omega) = K \sum_{m_{\boldsymbol{s}}} \sum_{m_{\boldsymbol{u}}} \left(R_i\left(\frac{\omega_{\boldsymbol{s}}}{N_{\boldsymbol{s}} T_{\boldsymbol{s}}} + m_{\boldsymbol{s}} f_{\boldsymbol{s}}, \frac{\omega_{\boldsymbol{u}}}{N_{\boldsymbol{u}} T_{\boldsymbol{u}}} + m_{\boldsymbol{u}} f_{\boldsymbol{u}}\right)\right), \tag{3.5}$$

(a) (b)

Figure 3.4: Rendered results using (a) the approach proposed by Georgiev et al. [41] and (b) our approach. The out of focus foreground objects exhibit RGB patterns in (a) due to non-uniform spacing of color components after integral projection.

where $K = \frac{1}{T_s T_u}$, and f_s, f_u are sampling frequencies on each dimension of all micro images. All $\sum$ operators range from $-\infty$ to ∞ and m_s, m_u are integers. Substituting R_i from Equation 3.4 to Equation 3.5 yields:

$$V_\Omega = M_\Omega R_\Omega, \tag{3.6}$$

where V_Ω is a q dimensional column vector with i^{th} element equal to $R_{d_i}(\Omega)$; Let B_S, B_U be periodic boundaries of R_o such that $R_o(\boldsymbol{S}, \boldsymbol{U}) = 0$ for any condition of $|\boldsymbol{S}| > B_S f_s, |\boldsymbol{U}| > B_U f_u$ satisfies; R_Ω is a $4B_S B_U$ dimensional column vector with the k^{th} element $R_o(\frac{\omega_s}{N_s T_s} + \gamma_s f_s, \frac{\omega_u}{N_u T_u} + \gamma_u f_u)$, and $\gamma_s = k \bmod (2B_S) - B_S$, $\gamma_u = \lfloor \frac{k}{2B_S} \rfloor - B_U$, and M_Ω is a $q \times 4B_S B_U$ matrix with $(i,k)^{th}$ element

$$\frac{1}{T_s T_u} \exp\{j2\pi[\sigma_{si}(\frac{\omega_s}{N_s T_s} + \gamma_s f_s) + \sigma_{ui}(\frac{\omega_u}{N_u T_u} + \gamma_u f_u)]\}.$$

Since we know the locations of Π_{st} and of each microlens m_i, σ_{si} and σ_{ui} can be accurately computed. $R_{d_i}(\Omega)$ can be acquired by performing the 4D DFT on the sampled light field by each microlens. Therefore Equation 3.6 is solvable for unknown R_Ω, which contains $2B_S$ and $2B_U$ frequency samples of $R_o(\Omega)$ on each dimension

respectively. Combining all R_Ω provides an estimate of R_o with $2N_{\boldsymbol{s}}B_{\boldsymbol{S}}, 2N_{\boldsymbol{u}}B_{\boldsymbol{U}}$ samples ranging from $(-B_{\boldsymbol{S}}f_{\boldsymbol{s}}, -B_{\boldsymbol{U}}f_{\boldsymbol{u}})$ to $(B_{\boldsymbol{S}}f_{\boldsymbol{s}}$ and $B_{\boldsymbol{U}}f_{\boldsymbol{u}})$ with spacing $(\frac{1}{N_{\boldsymbol{s}}T_{\boldsymbol{s}}}, \frac{1}{N_{\boldsymbol{u}}T_{\boldsymbol{u}}})$ on each dimension respectively. We then use it to estimate $r_o(\boldsymbol{s}, \boldsymbol{u})$ from $(0, 0)$ to $((N_{\boldsymbol{s}} - 1)T_{\boldsymbol{s}}, (N_{\boldsymbol{u}} - 1)T_{\boldsymbol{u}})$, with spacing $(\frac{T_{\boldsymbol{s}}}{2B_{\boldsymbol{S}}}, \frac{T_{\boldsymbol{u}}}{2B_{\boldsymbol{U}}})$. Hence the resolution of the resampled light field is increased by $2B_{\boldsymbol{S}}, 2B_{\boldsymbol{U}}$ on $\boldsymbol{s}$ and $\boldsymbol{u}$ compared with that of each original microlens image.

3.3.2 Integral Projection and Demosaicing

As shown in Fig. 3.3 (b), with the previous resampling process, we can achieve an evenly-sampled light field on the target focal plane Π. The integral projection is immediately applied to get I_f. An example of the green channel of I_f is shown by Fig. 3.5(c). However, due to the higher sampling rate of the green channel, a demosaicing process is still needed for red and blue channels of I_f to render a full RGB image with the resolution of the green channel.

Traditional sequential demosaicing frameworks first recover a full resolution green channel and subsequently use that green channel to facilitate the recovery of red and blue channels. In our case, the full resolution green channel is already known after the integral projection. Based on this green channel, the red and blue channels are reconstructed by applying the state-of-the-art anisotropic adaptive filtering [67] in the frequency domain. Fig. 3.4(b) shows that by employing the resampling scheme, the demosaicing can be performed on the integral projection result and the final image is free of RGB patterns.

Suppose the resampled light field has highest frequency ω'. The most common situation is $(\exists i)[\omega' > 2\omega_i > \omega]$. In this case the new demosaicing process preserves more high frequency information of the radiance, hence producing a higher resolution image (Fig. 3.5). In other cases such as $(\forall i)[\omega' > \omega > 2\omega_i]$ (very smooth regions such as places with constant color), both processes recover the full light field and the resolution of the resultant images are the same. If $(\forall i)[2\omega_i > \omega' > \omega]$ (texture rich regions or sharp edges), the final images are both over-smoothed.

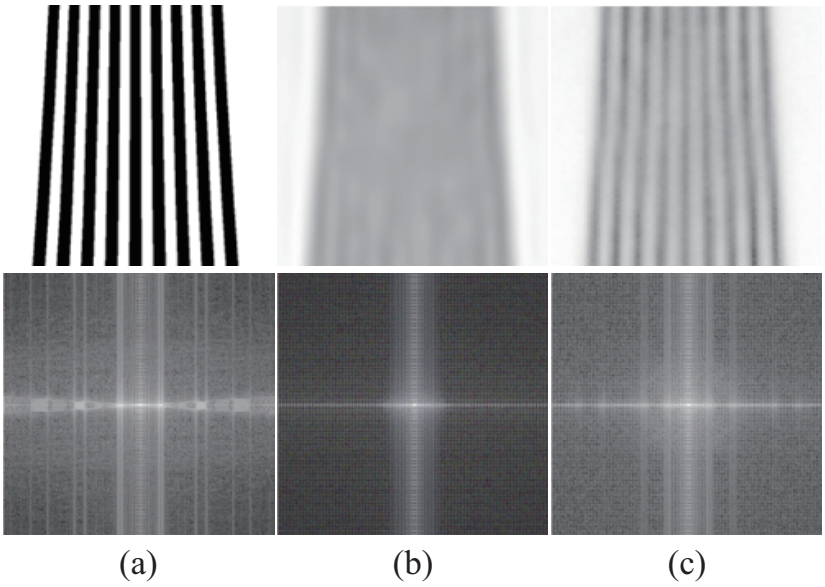


Figure 3.5: From (a)-(c), we compare the ground truth, the result using classical approach, and the result using our approach. The frequency spectrums are shown in the bottom row.

As illustrated by column (a) and (b) of Fig. 3.5, with the classical approach, significant losses in high frequency components occur in texture-rich regions and the rendered result suffers from over-smoothing compared with the ground truth. Column (c) shows our method preserves much more high frequency information of the ground truth, therefore capable of producing a higher resolution image.

3.4 Implementation and Applications

Fig. 8.1 shows the pipeline for implementing our proposed plenoptic demosaicing and rendering scheme. We first resample the radiance, then integral project it onto the spatial domain, and finally demosaic the color filtered result.

Our experimental data is captured by a plenoptic camera similar to that described in [70]. We use a 39-megapixel sensor with pixel size 6.8 μm. The main lens is mounted on the camera with a 13mm extension tube, which provides the needed spacing to establish an appropriate distance from the main lens focal plane to the microlens

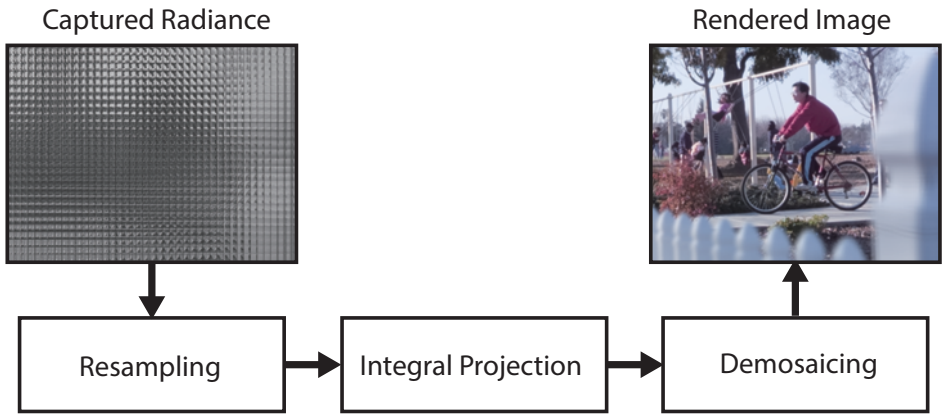


Figure 3.6: Our plenoptic demosaicing and rending pipeline.

array. The focal length of the main lens and of each microlens are 80mm and 1500μm respectively. The microlens pitch is 500 μm, which makes it work with the F-number of the main lens. The distances between microlenses are 74 pixels.

3.4.1 Enhanced Dynamic Refocusing

We first test our resolution enhancement performance by synthesizing photographs with a shallow DoF. Fig. 3.7 shows the comparison of our approach (b) and classical rendering (a) on a resolution chart scene. The bottom rows of (a) and (b) compare the demosaiced and raw microlens images of three highlighted regions. Note that severe aliasing effects appear on each raw microlens image and the structure of the resolution chart is not visible. If demosaicing is performed directly on each microlens image, colorful artifacts are introduced, damaging the high frequency information and over-smoothing microlens images. As a result, these regions could not be successfully reconstructed in the final image, as shown in (a). On the contrary, our approach utilizes each aliased microlens image to resample a high resolution light field before demosaicing is performed. Thus preserving a larger portion of high frequency information and producing a higher resolution image, as shown in (b). Also note that low frequency regions such as the left bottom part of the chart are equally clear in

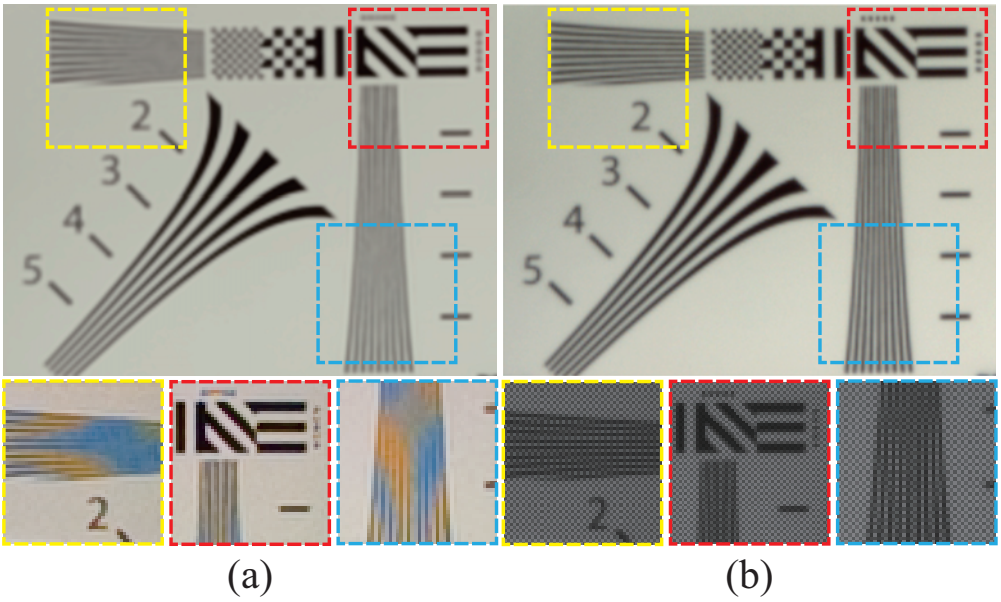


Figure 3.7: Comparison of rendered image employing classical approach and our approach. (a) Classical approach. Top row: Rendered image. Bottom Row: Demosaiced microlens image. (b) Our approach. Top row: Rendered image. Bottom row: Raw microlens image.

both cases, and very high frequency regions such as the bottom of the red highlighted region are both blurry.

The top row of Fig. 3.8 shows an outdoor scene. Apparently, the numbers on the licence plate in (b) are not visible but readable in (c). Another visible artifact of the classical framework here is that small regions of specular highlight appear less shiny due to over-smoothing on each microlens image.

Another real scene is shown in the second row of Fig. 3.8. In column (b), the first line of characters are barely readable using the classical rendering. Nevertheless, they are clearly rendered with our approach. Note that colorful artifacts introduced by demosaicing each microlens image remain on positions of "nf" and "ffi" in (b) and ringing artifacts also appear around the edges of the characters. Furthermore, the lower characters are totally blurry in (b) while still readable in (c).

Figure 3.8: Comparison of three results with classical approach and our approach. First and second row show shallow DoF rendering. The third row shows extended DoF rendering. (a) Our rendered result. (b) and (c) are enlarged highlighted regions in (a) with classical approach and our approach respectively.

3.4.2 Extended Depth of Field

Another popular application of our method is the extended DoF photography. Our approach pre-computes the depth of the sampled light field and renders each pixel by choosing its own depth among samples automatically.

The third row of Fig. 3.8 shows our extended DoF application on the same data as the second row. Note that the original out of focus regions such as the face and hair of the person are brought into focus, as if the photograph is captured by a pinhole aperture camera. However, with our framework, shown in (c), the rendered result preserves more high frequency information than the classical approach shown in

(b), therefore produces a much more detailed look.

3.5 Discussions and Limitations

We have presented a well-principled plenoptic demosaicing and rendering framework, which preserves more high frequency information from the captured light field and generates less aliasing artifacts compared with the classical approach.

Our framework does not apply demosaicing directly to the image captured by the plenoptic camera. Instead, with a resampling scheme which helps achieve constant spacing on each dimension, it dynamically performs demosacing after integral projection. Extensive experiments show that this framework could produce photographs with commercially acceptable resolution.

As analyzed in Section 3.2.2, the resolution enhancement of each plane in the scene achieved by our algorithm varies according to the depth of the plane. This could cause unpleasant results if the resolution enhancements are low on planes of interests. In the extreme case, the resolution could be as low as the classical framework. Like classical plenoptic photography, our approach assumes the captured light field contains thin rays in order to reconstruct a refocused image. This is also our assumption for theoretical resolution enhancement analysis. i

Chapter 4

ENHANCING THE ANGULAR RESOLUTION: LIGHT FIELD TRIANGULATION

In this chapter, I present a light field triangulation technique for enhancing the angular resolution of a captured light field. We first discuss the triangulation concept and present why it is useful for angular super-resolution of light fields. We then discuss the cons and pros of different light field triangulation schemes including edge-constrained and surface-constrained Delaunay triangulations. In particular, we study the geometric structure of 3D lines in the light field space and show that the light field space is largely bilinear due to 3D line segments in the scene. As a result, directly triangulating these bilinear subspaces leads to significant errors and visual artifacts. We instead present a simple but effective algorithm to map bilinear subspaces as surface constraints and apply Constrained Delaunay Triangulation(CDT).

4.1 Light Field Triangulation

4.1.1 Triangulation

In geometry, a triangulation is a subdivision of a geometric object into simplices. In particular, in the plane it is a subdivision into triangles, hence the name. Triangulation of a three-dimensional volume would involve subdividing it into tetrahedra ("pyramids" of various shapes and sizes) packed together. In most instances, the triangles of a triangulation are required to meet edge-to-edge and vertex-to-vertex.

A triangulation of a set of points P in the plane is a triangulation of the convex hull of P, with all points from P being among the vertices of the triangulation. Triangulations are special cases of planar straight-line graphs.

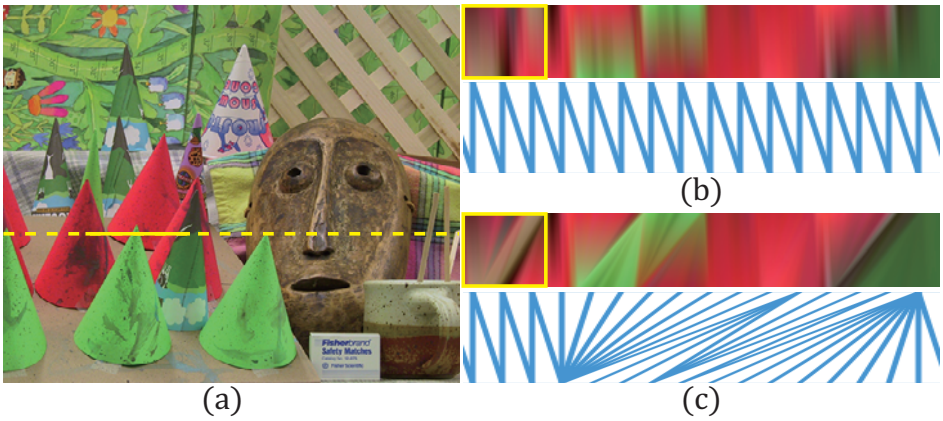


Figure 4.1: Triangulating a 2D light field (an EPI). (a) A scanline from a stereo pair; (b) RG Delaunay triangulation (bottom) performs poorly on light field super-resolution (top); (c) Using disparity as additional edge constraints, Constrained Delaunay triangulation significantly improves light field super-resolution.

There are special triangulations like the Delaunay triangulation which is the geometric dual of the Voronoi diagram. Subsets of the Delaunay triangulation are the Gabriel graph, nearest neighbor graph and the minimal spanning tree.

Previous studies show that the light field space is largely linear: a 3D scene point maps to a 2D ray hyperplane [130, 129]. This indicates that a light field can be "triangulated", i.e., the 4D light field can be partitioned into a set of space filling and non-overlapping simplices. Each simplex is a piecewise interpolant of the light field, hence the triangulation can be used for enhance the angular resolution. For a 2D epipolar plane Image (EPI), the simplices are 2D triangles; for a 3D light field, they are tetrahedra; and for the complete 4D light field, they are pentatopes. The triangulation provides a natural anisotropic reconstruction kernel: any point in the space can be approximated using a convex combination of the enclosing simplexs vertices (samples).

4.1.2 Simple Light Field Triangulation

The simplest light field triangulation is Delaunay triangulation without any constraints, or regular grid (RG) triangulation. Given a regularly sampled light field,

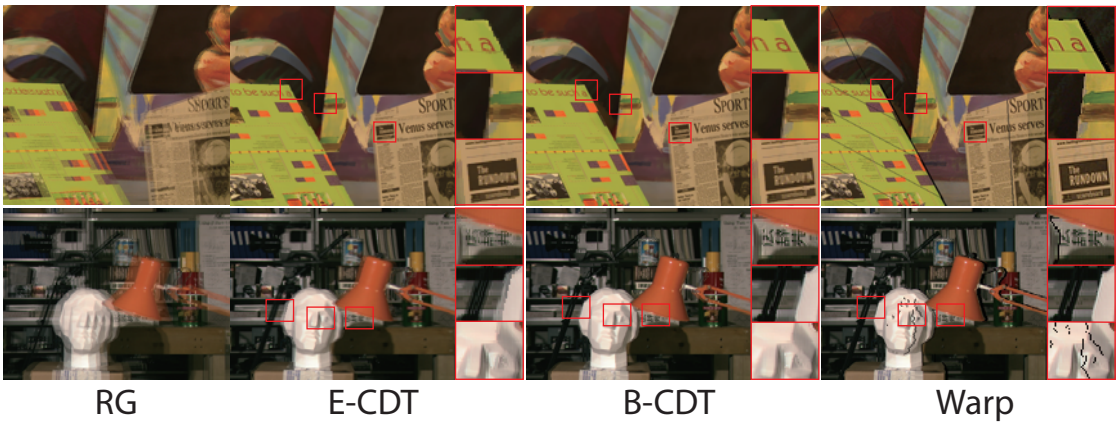


Figure 4.2: View interpolation using a triangulated 3D light field. We use the same set of feature points for RG, E-CDT, and B-CDT (ours). B-CDT produces comparable results to image warping but preserves continuity (no holes).

we can first build 4D hypercubes using two corners $[s, t, u, v]$ and $[s+1, t+1, u+1, v+1]$ and then triangulate each hypercube. Let us consider a 2D light field, an EPI formed by the same horizontal scanlines in a row of light field images. Fig. 4.1 (b) shows the RG triangulation of the EPI. If we use this triangulation to super-resolve the light field, i.e., by rasterizing the triangles, the result exhibits severe aliasing. This is because RG triangulation is analogous to bilinear interpolation and does not consider scene geometry (e.g., object depth or disparity).

4.1.3 Constrained Delaunay Triangulation

To improve RG triangulation, we can add epipolar constraints. Using stereo matching, we can first estimate every pixel's disparity and map it to a 2D hyperplane [129, 84] as a constraint. In the 2D EPI case, each pixel maps to an edge where the slope of the edge corresponds to its disparity (depth). We can then apply Constrained Delaunay Triangulation (CDT) [97]. We call this scheme EPI-CDT or E-CDT. Fig. 4.1 (c) shows an E-CDT triangulation and its super-resolution result. Specifically, we first detect 47 salient feature points along the scanline and add their corresponding EPI constraints. Our triangulation applies CDT to all pixels with these edge constraints.

Fig. 4.1 (c) show the closeup views of the triangulation. E-CDT greatly reduces aliasing while providing a continuous interpolant.

4.1.4 EPI Super-resolution

4.2 High-Dimensional Triangulation

An apparent question is whether we can directly apply E-CDT to triangulating higher dimensional light fields. The second column of Fig. 4.2 shows the E-CDT result of two images forming a 3D light field from the Middlebury Venus dataset. Specifically, we detect $10,132$ feature points (6% of total pixel) and use their disparities as edge constraints. We use the Tetgen [100] to conduct Constrained Delaunay Tetrahedralization. To illustrate its quality, we synthesize a new intermediate view between two source views by rasterizing the tetrahedralized 3D light field. The new view improves RG at non-occlusion regions but exhibits strong aliasing near linear occlusion boundaries.

4.2.1 Bilinear Ray Structures

We first briefly reiterate the ray geometry of 3D lines [129, 84]. If a 3D line l is parallel to Π_{uv} and Π_{st}, we can represent it with a point $\dot{P} = [P^x, P^y, P^z]$ on l and its direction $[\gamma^x, \gamma^y, 0]$. If a ray $r = [u, v, s, t]$ intersects l, there exist some λ_1 and λ_2 such that

$$\lambda_1[s, t, 0] + (1 - \lambda_1)[u, v, 1] = [P^x, P^y, P^z] + \lambda_2[\gamma^x, \gamma^y, 0]. \tag{4.1}$$

It is easy to see that $\lambda_1 = P^z$, and we can obtain that all rays passing through l satisfy the following linear constraint:

$$As + Bt + Cu + Dv + E = 0, \tag{4.2}$$

where $A = \gamma^y - \gamma^y P^z, B = \gamma^x P^z - \gamma^x, C = \gamma^y P^z, D = -\gamma^x P^z, E = \gamma^x P^y - \gamma^y P^x$. This reveals that lines in the 3D scene that are parallel to Π_{uv} will map to linear subspaces in the light field and hence can be triangulated.

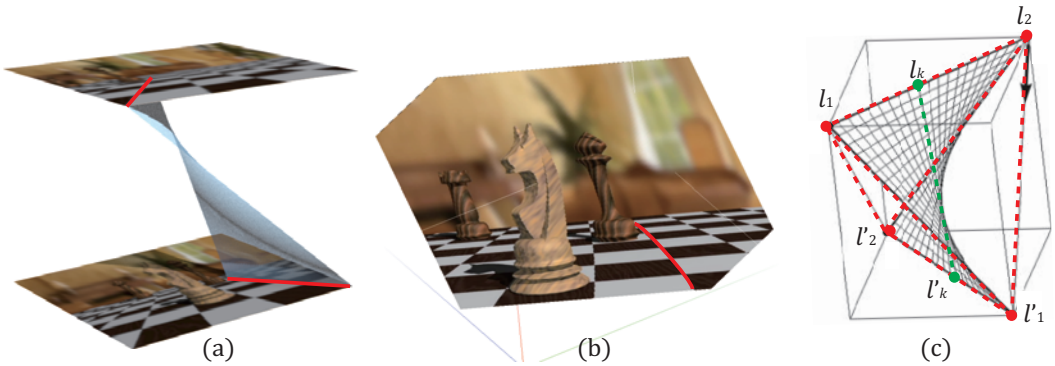


Figure 4.3: Bilinear ray structures. (a) A 3D line segment l maps to a bilinear subspace in a light field; (b) l maps to a curve on a diagonal cut; (c) Brute-force triangulation creates volume.

If l is not parallel to Π_{uv}, it then can be directly parameterized by a ray under 2PP as $[u_0, v_0, s_0, t_0]$.

All rays passing through l thus satisfy the following *bilinear* constraint:

$$\lambda_1[s, t, 0] + (1 - \lambda_1)[u, v, 1] = \lambda_2[s_0, t_0, 0] + (1 - \lambda_2)[u_0, v_0, 1]. \tag{4.3}$$

We have $\lambda_1 = \lambda_2$ and

$$\frac{s - s_0}{u - u_0} = \frac{t - t_0}{v - v_0}. \tag{4.4}$$

The bilinear ray geometry is particularly important since a real scene usually contains many linear structures unparallel to the image plane. This reveals that the light field ray space contains a large amount of bilinear structures. In Fig. 4.3 (a), we construct a 3D light field by stacking a row of light field images and cut it using the videocube tool [119]. Fig. 4.3 (b) shows a cut through a volume where 3D lines on the checkerboard appear curved due to their bilinearity.

To analyze the cause of aliasing, let us consider a 3D line segment l whose image is $(l_1^x, l_1^y) - (l_2^x, l_2^y)$ in light field view (u, v). Assume the disparity of l_1 and l_2 are d_1 and d_2 respectively. If $d_1 \neq d_2$, by Eqn. 4.4, l maps to a bilinear surface S formed by four corners (u, l_1^x, l_1^y), (u, l_2^x, l_2^y), $(u+1, l_1^x + d_1, l_1^y)$, and $(u+1, l_2^x + d_2, l_2^y)$ in 3D (u, s, t)

light field space. In geometric modeling, it is well known that any direct triangulation of S from the four vertices of S will introduce large error: S is a surface that does not occupy any volume. However, a triangulation of the four vertices will turn S into a tetrahedron which will occupy large volume when $|d_1 - d_2|$ is large, as shown in Fig. 4.3 (c). The tetrahedron will "erode" into neighboring space, i.e., nearby pixels will be forced to use this tetrahedron as the interpolant. Therefore it is important to add additional constraints onto the bilinear structure.

4.2.2 CDT with 3D Edge Constraints

We present a simple but effective scheme that directly maps bilinear ray structures of 3D lines into the CDT framework. Specifically, we apply a subdivision scheme [81] by discretizing the bilinear surface into slim bilinear patches and then triangulate each patch. Finally, we use edges of bilinear patches and disparity hyperplanes as constraints for CDT. We call this scheme Bilinear CDT or B-CDT.

4.2.2.0.1 3D Light Fields.

For a 3D light field, B-CDT can be effectively implemented using Tetgen [100]. In the Venus example (first row of Fig. 4.2), we detect additional 303 line segments in the reference view, subdivide their corresponding bilinear surfaces, and add them as constraints for conducting B-CDT. The new triangulation significantly improves the E-CDT result: it preserves most sharp edges and exhibits very little aliasing near occlusion boundaries. Compared with image warping that results in missing pixels or holes, the B-CDT provides a continuous representation of the light field where any new view corresponds to a valid 2D triangulated light field without holes. Notice that the texts on the newspaper are slightly blurred since they are not selected as constraints.

Fig. 4.2 row 2 shows our result on the Tsukuba dataset containing fewer linear structures. In this example, we select $15,748$ feature points (14% of total pixel) from the reference image and detect 120 line segments. Same as the Venus example, we compare E-CDT and B-CDT by rendering an new view between the two reference

views. E-CDT preserves non-boundary contents but exhibits strong aliasing near the boundary pixels such as the tripod, the light edges, and the bust. In contrast, B-CDT preserves both boundary and non-boundary contents. The Tsukuba scene has a relatively large disparity range and direct warping produces many holes. To patch these holes, one can use 2D interpolation schemes such as Delaunay triangulation. Such interpolation, however, is different from B-CDT: B-CDT provides a consistent triangulation throughout the light field volume while warping followed by hole patching produces an ad-hoc triangulation on each slice; Further, B-CDT only needs to be conducted once while hole patch needs to be conducted whenever rendering a new view.

4.2.3 4D Light Fields

Finally, we extend the B-CDT scheme to 4D light fields. In computational geometry, high dimensional CDTs [97] remains as an open problem for two reasons. First, a plausible solution may require inserting a large number of auxiliary vertices. This also occurs in 3D CDT although the number of inserted vertices is much smaller. Second, the computational complexity grows rapidly with respect to dimensionality [12]. To our knowledge, no practical 4-dimensional CDT is currently available to the public. Our solution is to convert the 4D problem to 3D. Specifically, to synthesize a new view V_{st} in the 4D light field with four sample views indexed as V_{00}, V_{01}, V_{10}, V_{11}, we first detect 3D line segments and apply 3D B-CDT to synthesize two new views V_{u0} and V_{u1} from 3D light fields $V_{00} - V_{10}$ and $V_{01} - V_{11}$, respectively. Next, we use the same 3D line constraints and B-CDT to triangulate a 3D light field $V_{u0} - V_{u1}$ for synthesizing V_{uv}. Fig. 4.4 shows an skyscraper light field with disparity range [0,300]. Results using RG exhibit severe aliasing where directly warping produces holes and discontinuity. Next, we select $90,269$ feature points (11% of total pixel) and 2092 line segments and apply the pseudo 4D CDT. Our results exhibits little aliasing while preserving smoothness.

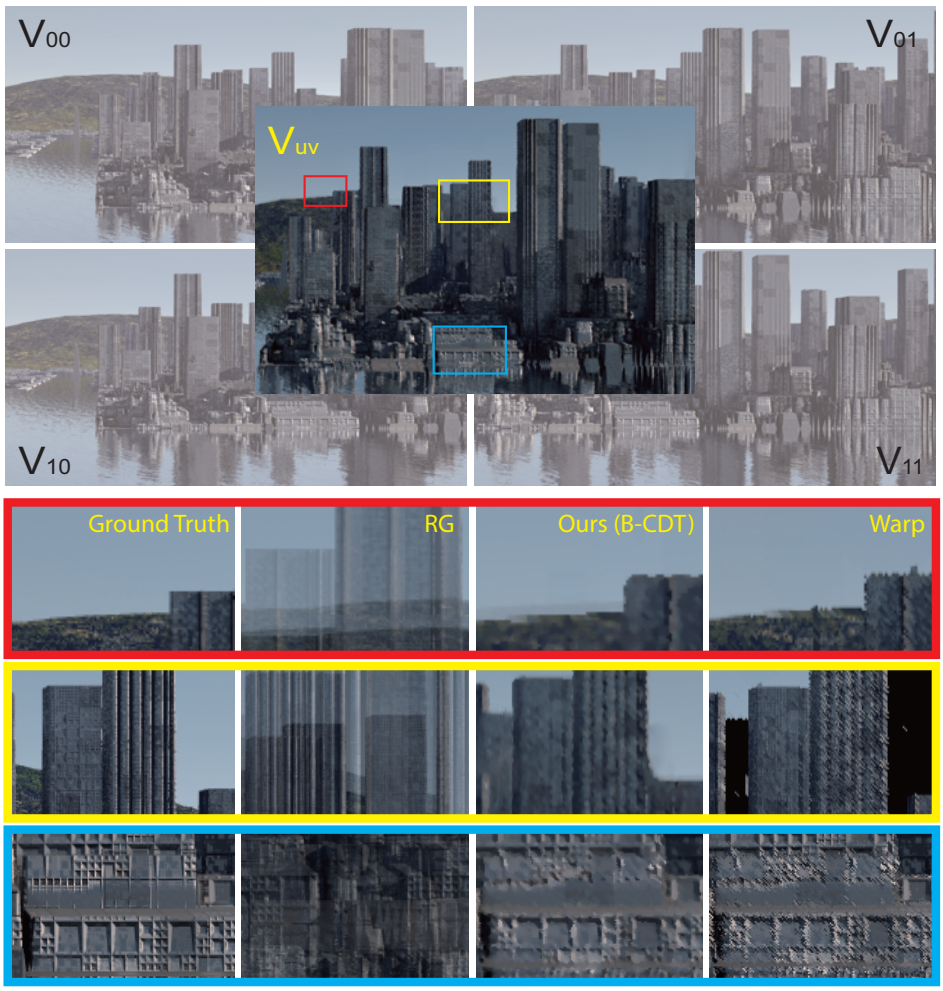


Figure 4.4: New view (central) synthesis from a 4D light field. Left: a light field of a skyscraper scene. Right: Closeup views of the synthesized results using different schemes.

4.3 Discussions

We have presented a light field triangulation approach by imposing ray geometry of 3D line segments as constraints. We utilize Constrained Delaunay Triangulation (CDT) and by far our solution is restricted to 3D and pseudo 4D light fields since 4D CDT is still an open problem in computational geometry.

The depth information of the feature points also plays a crucial rule in our triangulation. In fact, an accurate depth map is also important for most state-of-the-art light field reconstruction methods [17, 121]. To improve the current depth estimation methods, in Chapter 5, we present two approaches.

Chapter 5

LIGHT FIELD STEREO MATCHING

The core component in our light field angular resolution enhancement algorithm is the availability of high quality disparity maps. In this chapter, I discuss two approaches that use light field stereo matching for depth/disparity estimation.

5.1 Related Work

The availability of light field cameras has also renewed the interest on multi-view reconstruction. The seminal work by Kolmogorov and Zabih [54] extend the binocular graph-cut solution to multi-view stereo. In addition to the data and the smoothness terms, they add an occlusion term for handling complex occlusions. However, the smoothness term in their method restricts local disparity variation, hence is difficult to represent smooth disparity transition. To resolve this issue, Woodford et al. [125] further incorporate the second order smoothness priors and optimize the non-submodular objective function via Quadratic Pseudo-Boolean Optimization (QPBO) [91]. Recently, Bleyer et al. [18] impose soft segmentation and minimum description length as priors to solve for a non-submodular objective function. Georgiev et al. [40] apply a window based algorithm for producing coarse disparity maps to guide digital refocusing. More recently, Wanner and Goldlücke [92, 121] apply structure tensor to measure each pixel's direction in 2D EPI. They then encode the estimated edge directions into dense stereo matching with consistency check. However, their local EPI structure estimation is robust for disparities within a small range. Moreover, most previous algorithms do not explicitly consider or aim to preserve the 3D geometry such as 3D lines.

5.2 Occlusion Aware Disparity Estimation

One of the most challenging problems in light field stereo is the occlusion. To robustly resolve this, we first analyze the behavior of pixels in such situations. We show that even under severe occlusion, one can still distinguish different depth layers based on statistics. We estimate the disparity of each pixel by discretizing the space in the scene and conducting plane sweeping. Specifically, for each given disparity, we gather all corresponding pixels from other views and model the in-focus pixels as a Gaussian distribution. We show how it is possible to distinguish occlusion pixels, and in-focus pixels in order to find the disparities. To estimate the scene disparity based on the captured light field image, we consider the behavior of a pixel p_{00} in a view V_{00} of the light field image. This pixel can map to pixels in different views when assigned with different disparities.

5.2.1 No occlusion

Assuming all surfaces in the scene are Lambertian. As shown in Figure 5.1 (a), if p_{00} is assigned the correct depth d, it maps to a point P on a surface. All rays emitted from P have constant color. Therefore, rays captured by any other view V_{uv} at pixel p_{st} will have the same color as p_{00}. On the other hand, if incorrect depth d' is assigned to p_{00}, then the corresponding p_{st} will tend to have different colors than p_{00}.

With this observation, when assigning a disparity d to a pixel, we model the distribution of color over all pixels from different views as a unimodel Gaussian distribution to further compensate for the vignetting effect and camera noise. And the variance of the distribution defines the possibility of the p_0 actually lying on d. It is computed by:

$$V_{p_{00},d} = \frac{\sum (I_p - \bar{I})^2}{N}, \tag{5.1}$$

where I_p is the intensity of p, N is the number of pixels associated by mapping p to other views with disparity d, and $\bar{I}$ is the mean of intensities of all associated pixels.

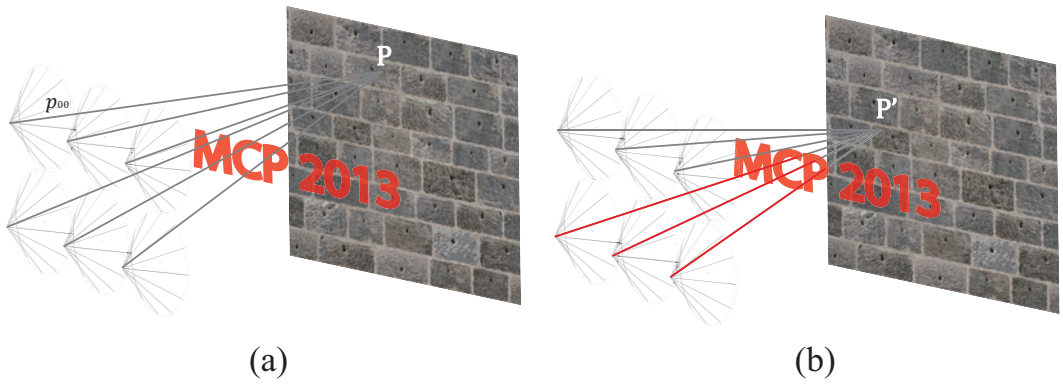


Figure 5.1: Color sampled by cameras without (a) or with (b) occlusion.

If $V_{p_{00},d}$ is small, meaning all the pixels have almost the same color, the probability of p_{00} having disparity d is high, and vice versa.

5.2.2 Disparity Estimation with Occlusion

Consider Figure 5.1 (b), where some of the views looking at P' are occluded. In this case, even with a correctly assigned disparity, due to occlusion, some rays emitted from the front surfaces replace the correct rays from the back surface, resulting in high variance in our Gaussian model.

To resolve this issue, Yu et al. [131] assume occlusion surfaces have similar color and model the problem with a bimodel Gaussian distribution. One can easily extend this approach to a N-model Gaussian distribution but deciding N is rather difficult. However, having similar color on all occlusion surfaces is a rather extreme assumption. Moreover, under a small number of views, sometimes there are not enough pixels to form Gaussian distribution. The state of the art globally consistent depth labeling method [92] proposed global labeling constraints on epipolar plane images (EPI). But it requires a a small disparity range (usually less than 3 pixels) in order to estimate local direction on the EPI. Therefore it does not fit our sparse sampling situation. However, to show the robustness of our algorithm, we still compare our result with this

algorithm by providing more views.

Next, we analyze the distribution of pixel intensities. In the regular case, images of P are still captured by some of the views. In this case, the Gaussian distribution still holds, but with noise around the region far from the mean. It is possible to explicitly separate out the occlusion samples or implicitly model this distribution as N-modal. However, in the extreme case where most samples are from occlusion surfaces, it is almost impossible to tell which samples are from the in-focus plane with a single observation.

Instead of trying to point out which samples are outliers directly from a single observation under a given disparity, we propose to iteratively mask out the layers that are in front of P. For each iteration it, we still loop over all the disparity values to check for the minimum variance for each pixel. The difference is that starting from the second interaction, we make use of the current disparity map and when testing disparity d_{it} on P, we ignore pixels that have larger disparity than d_{it}.

Now we analyze this idea in detail. Assuming that the depths in the scene are corresponding disparities ranging from d_{max} to d_{min}, and that there are sufficient number of views to form different distributions when assigned with different disparities. It is also reasonable to assume that if all the occlusion pixels are masked out, the intensity distribution will achieve minimum variance at the correct disparity value. In the first iteration, we can successfully find the local minimum for the closest depth since no occlusion will occur on those pixels.

In the next iteration, we mask out those pixels when computing the depth for all pixels since they are considered as occlusions. Note that pixels not at d_{max} may also be assigned disparity d_{max} during the first iteration due to occlusion problems. However, by masking out all the pixels assigned with d_{max}, our algorithm guarantees that no pixels from d_{max} will affect the converges of pixels at d_{max-1}. Therefore during the second iteration, all pixels on d_{max-1} will be computed under with no occlusion involved.

Now we prove that in each iteration, our estimation is occlusion free.

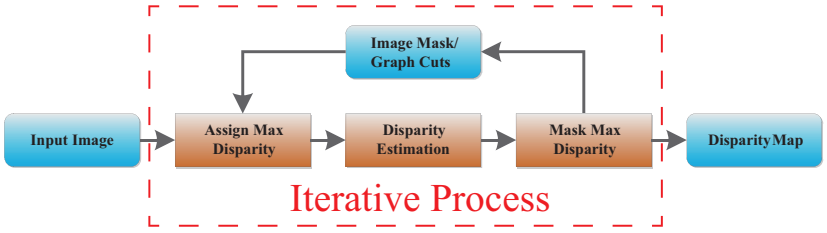


Figure 5.2: Our disparity estimation pipeline.

Base case. In iteration 0, all the depth are computed directly using the unimodel Gaussian distribution. In this case, all the pixels on d_max will be marked out correctly.

Induction. Suppose in iteration n, disparities larger than $d_{max} - n$ are all computed correctly, in iteration $n+1$, we ignore pixels with disparities larger than $d_{max} - n$. So that pixels with $d_{max} - (n + 1)$ can be computed with no occlusion involved.

5.2.3 Avoiding the trivial solution

However, as mentioned above, at each iteration it, pixels lying further than d_{it} could be incorrectly assigned with d_{it} due to unresolved occlusion. In this case, unnecessary pixels may get masked out, so that with by assigning a small disparity, trivial solutions with small variance could be produced for pixels on textureless surfaces. We propose two solutions to resolve the trivial solution: 1) using the boundary pixels to regulate the textureless pixels (global optimization); 2) using a edge mask to ignore the pixels on the surfaces in a later iteration (edge mask).

5.2.3.1 Edge Mask

The edge mask aim to mark the edge region. To resolve the issue of trivial solution, in each iteration, we only recomputed the disparity of the edge region and do not touch the regions which have been masked. Figure 7.9 illustrates our processing pipeline using edge mask approach. To compute the edge mask, consider one pixel in a given view. For each given disparity assumption, we gather all corresponding

pixels from other views and model the in-focus pixel value as a Gaussian distribution. We compute the variance for each trial disparity. This gives us the maximal variance V_{max} and minimal variance V_{min} at that particular pixel. We choose the disparity for a given pixel to be the one with minimal variance. We also compute the quantity $M = V_{min}/V_{max}$. The value of M at each pixel gives us a chance to estimate depth edges. We then sort the pixels by M and select top 30% of them as the edge pixels for further computation.

5.2.3.2 Global Optimization

Constant color surfaces in the scene are always a problem since it is difficult to estimate disparities directly from them. Traditional global optimization methods such as graph cuts or belief propagation use a smoothness constraint to compensate for this issue. We embed our algorithm into the graph cuts framework and let the smoothness constraint to resolve the trivial solution issue. Specifically, in each iteration, we minimize the energy function by constructing a graph with data term (variance of pixel intensities) as the links to source/target and smoothness term (disparity difference between neighboring pixels) as links to neighboring pixels. In this case, even though the textureless pixels may tend to choose small disparity, the pixels on the edge will force them to choose the correct disparity since the variance of trivial solution and of the correct disparity are similar. We reuse min-cut/max-flow algorithm [54, 52] to minimize the energy function. Note that the data term in our case is occlusion free because we do not consider pixels with depth lower than the current depth.

5.2.4 Experiments

All experiments were conducted on a PC with Intel Core i7 3.2GHz CPU and 8GB memory. The second row of Fig. 5.11 and Fig. 5.4 show the disparity maps of the captured light fields of a camera scene and a hand scene using our method with edge mask (EM), global optimization (GO) and brute force graph cuts (GC). On the third row, we render the disparity map using the light field rendering. GC has very

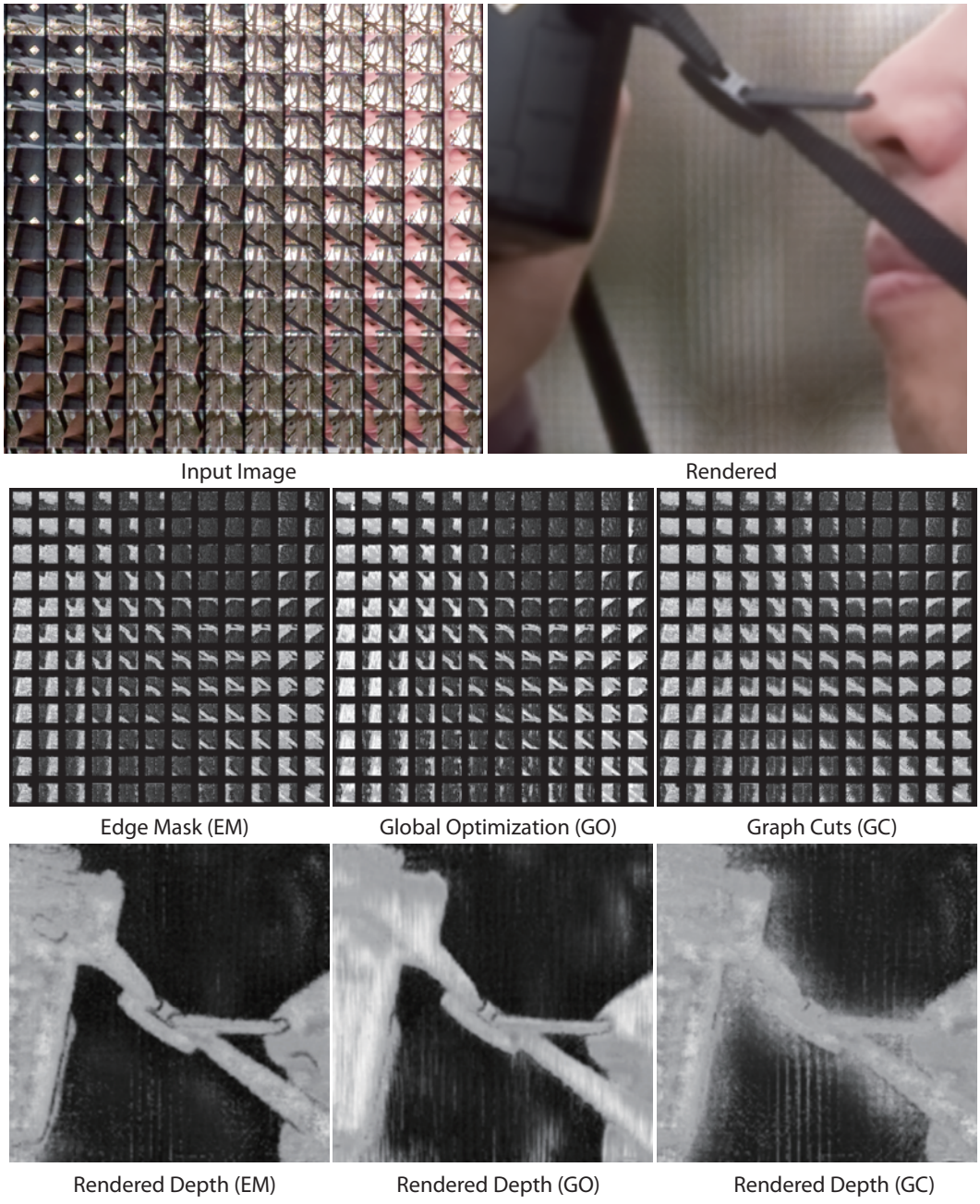


Figure 5.3: Estimated disparity map using different methods based on the input integral image of the camera scene.

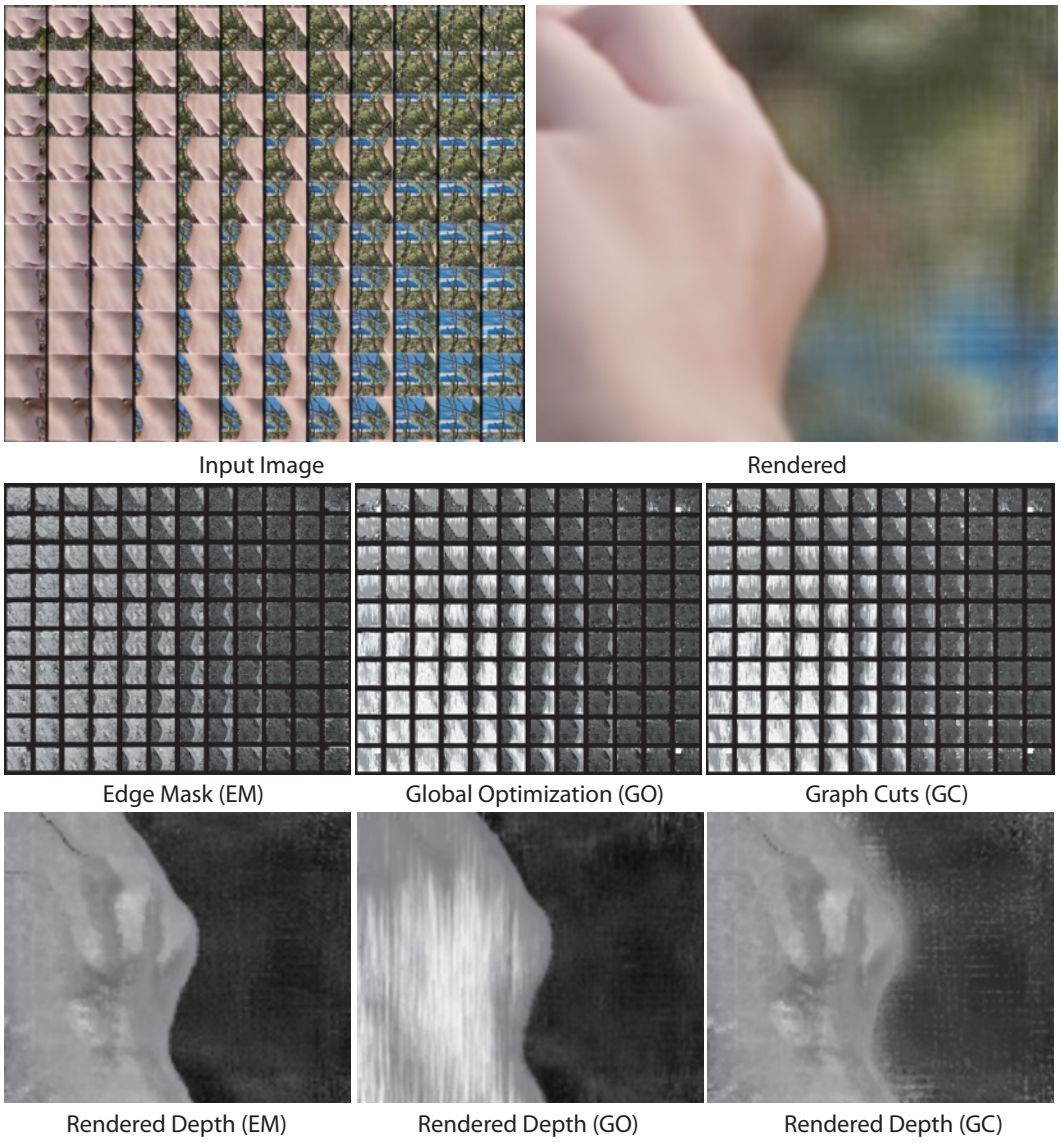


Figure 5.4: Estimated disparity map using different methods based on the input integral image of the hand scene.

noisy occlusion boundaries such as edges of the belt in Fig. 5.11 and the edges of the hand in Fig. 5.4 due to the the severe occlusion conditions. In contrast, GO and GC both accurately recover fine details and robustly handle the occlusion boundaries. However, the result of EM appears a little bit more variant on surfaces with constant disparity but GC better preserves the smoothness of surfaces.

5.2.4.1 Synthesizing novel views

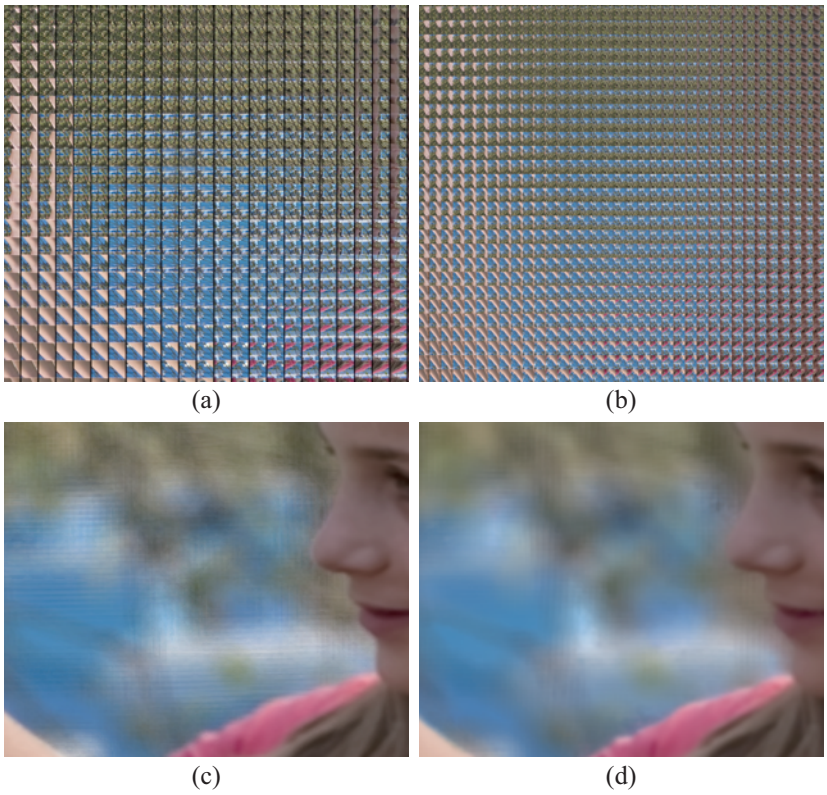


Figure 5.5: Different applications using the estimated disparity. (a) Input views (captured integral image). (b) Synthesized views. (c) Rendering using input views. (d) Rendering using synthesized views.

Figure 5.5 (b) shows our result of using disparity estimated from the input integral image to synthesize arbitrary views representing a new, denser integral image with more views. Given the input image of 25×25 views of a girl scene, we synthesize a new integral image with 25×25 views that are concentrated in the central area. With

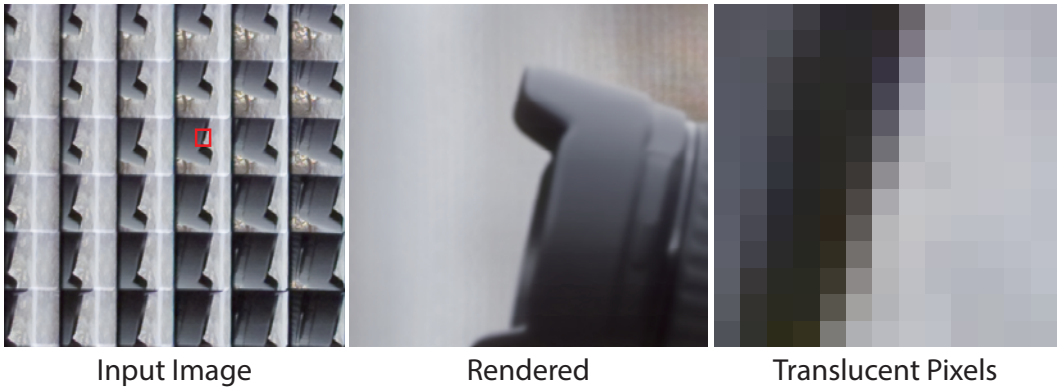


Figure 5.6: Translucent pixels appear near occlusion boundaries in the captured image.

the correctly estimated occlusion boundaries, we are able to faithfully recover the edges of the arm, wrinkles on the shirt on the foreground and thin branches and leaves of the trees, cover of the bee hives in the background. Note that our boundaries sometimes appears to be noisy. This is because of our algorithm assigns a single disparity value for each pixel and is not capable of handling translucent pixels on the edges. We will discuss this issue in Section 5.2.5.

5.2.4.2 Rendering aliasing reduced images

Aliasing in the rendered image is usually caused by under-sampling of the light field. To conduct anti-aliasing, we use our estimated disparity for the light field to synthesize a densely sampled light field of 100×100 views. We then render the dynamic depth of field effect using the new light field. As shown in Figure 5.5 (d), compared with the result using the original captured light field, when focusing on the foreground, we are able to greatly reduce the aliasing artifacts on the background and simulating a D-SLR quality image.

5.2.5 Discussions

It is known that boundary pixels require matting to resolve the translucency. Since our algorithm explicitly defines one disparity for each pixel, the disparities for the translucent pixels could not be correctly computed. As shown in Figure 5.6. It is our immediate future work to explore a model of multiple disparities per pixel in our algorithms.

In our edge map algorithm, the threshold for the edge map is empirically defined. In the future, we plan to analyze the statistics of the image and automatically choose the thresholds.

5.3 Line Assisted Light Field Stereo Matching

Our second algorithm is based on the observation that man-made scenes contains large number of linear structures that can be used as useful constraints/priors in the stereo matching process.

5.3.1 Disparity Interpolant

We first prove the linearity of disparity along a line segment, i.e., given two endpoints l_1 and l_2 of a 3D line segment l with disparity d_1 and d_2, the disparity d_k of any intermediate point $l_k = \lambda_k l_1 + (1 - \lambda_k) l_2$ is $\lambda_k d_1 + (1 - \lambda_k) d_2$. This property is well known, e.g., in perspective geometry in computer vision and in projective texture mapping in computer graphics. We present a different proof based on bilinear ray geometry of line l.

If l is parallel to Π_{uv} and Π_{st}, then the proof is trivial since $d_1 = d_2 = d_k$.

If l is not parallel to Π_{uv} and Π_{st}, l can be represented as a ray (s_0, t_0, u_0, v_0). Consider a specific pixel (s, t) in camera (u, v) that observes a point P on line l and pixel $s + \Delta s$ in a neighbor camera $(u + \Delta u, v)$ that also observes P. Both ray (s, t, u, v) and $(s + \Delta s, t, u + \Delta u, v)$ satisfy the bilinear ray constraint (Eqn. 4.4):

$$\frac{s + \Delta s - s_0}{u + \Delta u - u_0} = \frac{s - s_0}{u - u_0} = \frac{t - t_0}{v - v_0}. \tag{5.2}$$

Therefore, $\frac{\Delta s}{\Delta u} = \frac{t-t_0}{v-v_0}$. This reveals that disparity $\frac{\Delta s}{\Delta u}$ is a linear function in t along l, i.e., we can linearly interpolate the disparity along l.

5.3.2 Line-Assisted Graph Cut (LAGC)

To incorporate the linear disparity constraint into multi-view stereo matching, the most direct approach is to first detect line segments in the captured light field, then estimate their disparities, and use them as hard constraints in the graph-cut algorithm. The top row of Fig. 5.7 shows the result of this brute-force approach on a city scene. We render a light field of the scene (17×17 views at 1024×768 resolution). We detect line segments using the state-of-the-art line segment detector (LSD) [120] for each view (around $1100 \times 17 \times 17$ line segments). For the endpoints of each line segment l, we iterate over all possible disparities and interpolate the disparity for all intermediate points. Finally, we find the optimal disparity assignments to the endpoints that yield to highest consistency of all intermediate points. The results are then used as hard constraints for the multi-view graph-cut (MVGC) [54]. Fig. 5.7 shows improvements near edges and rich texture regions compared with MVGC. However, if the disparity of the line segment is incorrectly assigned, it will lead to large errors, e.g., on one of the chimneys on the building, as shown in Fig. 5.7.

Next, we study how to explicitly encode the disparity constraint of line segments into MVGC. MVGC aims to find the optimal disparity label that minimizes the energy function $E_{conventional} = E_{data} + E_{smooth} + E_{occ}$,where

$$E_{data} = \sum_{P,Q} E_d(P,Q), \quad E_d(P,Q) = ||I(P) - I'(Q)||_2 - K,$$
$$E_{smooth} = \sum_{P,P_N \in \mathcal{N}} E_s(P,P_N), \quad E_s(P,P_N) = min(||d_P - d_{P_N}||, T_c) \tag{5.3}$$

where P and Q correspond to the same 3D point given a disparity, $\mathcal{N}$ is the neighborhood of P, T_c is the truncation threshold, and K is a constant. The occlusion term E_{occ} measures if occlusion is correctly preserved when warping the disparity from I to I' [54].

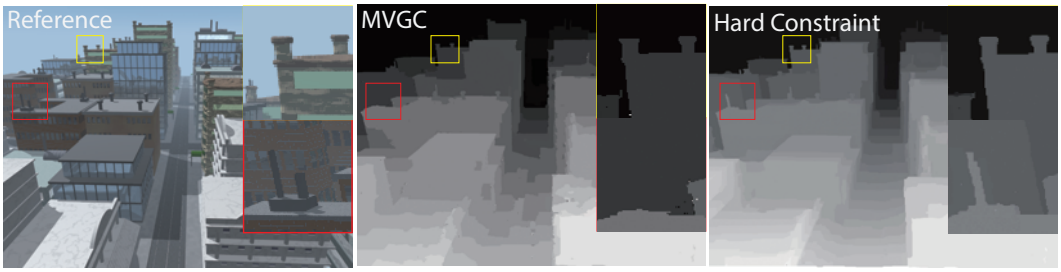


Figure 5.7: Encoding 3D line segments as hard constraints improves MVGC but misses important details, e.g. the chimney on the building.

We add the fourth *line constraint* term. Our key observation is that when assigning disparity labels to the two endpoints, every intermediate point along the line should check occlusion consistency. Specifically, given the two endpoints (pixels) l_i and l_j of line segment l and an intermediate pixel $l_k = \lambda_k l_i + (1 - \lambda_k) l_j$, we define

$$\begin{aligned} E_{line} &= \sum_l \sum_{l_k \in [l_i, l_j]} E_l(l_i, l_j, l_k), \\ E_l(l_i, l_j, l_k) &= ||\lambda_k d_{l_i} - d_{l_k} + (1 - \lambda_k) d_{l_j}||. \end{aligned} \tag{5.4}$$

Our goal is to minimize the new energy function $E_{conventional} + E_{line}$.

The work by Boykov et al. [20] and Kolmogorov and Zabih[52, 54] show that one can minimize $E_{conventional}$ by consecutively solving the two-label problem: at each iteration, a new disparity label is added and the algorithm decides whether each pixel should keep the old disparity or switch to the new disparity. We follow their convention to use *0* for keeping the old label and *1* for using the new label. To solve for the two label problem with *alpha*-expansion [20], the energy function needs to be regular. For example, E_{data}, E_{smooth} and $E_{occlusion}$ (the two-variable functions) are all regular.

Notice that E_{line} (Eqn. 5.4) is a three-variable ($\mathcal{F}^3$) term, i.e., the endpoints and any intermediate point individually choose to relabel or not. E_{line} can be viewed as a general second order smoothness prior and is generally non-submodular. Therefore, *alpha*-expansion is not directly applicable to minimize E_{line}. We instead adopt the extended QPBO approaches proposed by Rother et al. [91]. To briefly reiterate, the QPBO algorithm [51] splits each node in the graph into two subnodes; when

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

Figure 5.8: Comparison using different optimization schemes. (a) *alpha*-expasion.(b) QPBO-I. (c) QPBO-P (d) Reference Image.

both subnodes are assigned to the source or sink after min-cut, they will be assigned the corresponding label. Otherwise, they will be treated as unlabeled. Theoretically, QPBO can potentially result in a large number pixels assigned unlabeled. Extensions of QPBO such as QPBO-P and QPBO-I [19, 91] as well as the more complex QPBO-R [125] can be further used to reduce the unlabeled pixels. For example, QPBO-I uses additional geometry priors (called the proposals) to improve optimization.

Recall in our problem, only pixels on 3D line segments (edges) can be potentially assigned unlabeled. Since they are generally sparse for natural scenes (for example, the number of non-submodular terms in all our experiments are around or under 10% of the total terms), we find that QPBO-P and QPBO-I are generally sufficient. For example, in QPBO-I, we use fronto-parallel surface priors as proposals. Fig. 5.8 shows the results on the city scene using QPBO-P, QPBO-I (with fronto-parallel surfaces as proposals), and standard *alpha*-expansion. QPBO-P and QPBO-I produce comparable results while *alpha*-expansion produces noticeable artifacts such as inaccurate edges.

5.3.3 Graph Construction

Next, we construct the graph so that we can reuse min-cut/max-flow algorithm to minimize our $\mathcal{F}^3$ energy function. We follow the general-purpose graph construction framework by Kolmogorov and Zabih [53]: each pixel corresponds to a graph node. We then add the source s node for label *0*, the sink node t for label *1*, the t-links

from the graph nodes to s or t, and the n-links between the graph nodes using 4-connectivity. We decompose the two variable term E_d and E_s to the corresponding t-links and n-links. For example, for two neighboring nodes n_i and n_{i+1}, we assign weights $E_s(1,0)-E_s(0,0)$ and $E_s(1,0)-E_s(1,1)$ to t-links (s,n_i) and (n_{i+1},t) respectively, and weight $E_s(0,1)+E_s(1,0)-E_s(1,1)-E_s(0,0)$ to n-link (n_i,n_{i+1}). The similar scheme can be applied for handling E_{occ}.

Different from E_s and E_{occ}, E_l is $\mathcal{F}^3$ and auxiliary nodes and links need to be added to the graph [53]. Specifically, for each edge tuple (l_i,l_j,l_k) (i, j the endpoints and k the intermediate point), we add three auxiliary n-links (an-links) l_i-l_j, l_i-l_k and l_k-l_j, as shown in Fig. 5.9 (b). We also add an auxiliary sink/source node n_k^* and the corresponding auxiliary t-links (at-links) using one of the two possible assignments: either (l_i,n_k^*), (l_j,n_k^*), (l_k,n_k^*), and (n_k^*,t) (Group at_1) or (n_k^*,l_i), (n_k^*,l_j), (n_k^*,l_k), and (s,n_k^*) (Group at_2), as shown in Fig. 5.9 (c). The selection of the group depends on the weight decomposition of E_l. Specifically, we follow the decomposition in [53] and edge assignment schemes for $\mathcal{F}^3$. Here we briefly reiterate this process. There are two possible decompositions of E_l. We first compute $p=(a+d+f+g)-(b+c+e+h)$. If $p\geq 0$, E_l can be decomposed using the upper branch of Table 5.1. We then assign the weights to edges as follows: 1) assign p to all four at-links in group at_1; 2) Assign p_1,p_2,p_3 to t-links for l_i, l_j, and l_k respectively, and finally assign p_{12},p_{23},p_{31} to an-links (n_i,n_j), (n_j,n_k), and (n_k,n_i). If $p>0$, we can decompose the table in a similar fashion as shown in the lower branch of Table 5.1 and assign the weights to edges accordingly.

We call our solution the line-assisted graph-cut (LAGC).

5.3.4 Evaluation

All experiments were conducted on a PC with Intel Core i7 3.2GHz CPU and 8GB memory. We first evaluate our algorithm on binocular stereo using the Tsukuba dataset. Fig. 5.10 compares the ground truth, global stereo reconstruction under second order smoothness priors (SOSP) [125], adaptive ground control point (GCP) [98] (rank 1 for all four Middlebury datasets [29]), MVGC [54], and our LAGC. Table

$$E_l = \begin{array}{|c|c|}\hline E(0,0,0) & E(0,0,1) \\ \hline E(0,1,0) & E(0,1,1) \\ \hline E(1,0,0) & E(1,0,1) \\ \hline E(1,1,0) & E(1,1,1) \\ \hline \end{array} = \begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline e & f \\ \hline g & h \\ \hline \end{array} =$$

$$A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline p_1 & p_1 \\ \hline p_1 & p_1 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline p_2 & p_2 \\ \hline 0 & 0 \\ \hline p_2 & p_2 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & p_3 \\ \hline 0 & p_3 \\ \hline 0 & p_3 \\ \hline 0 & p_{32} \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & p_{23} \\ \hline 0 & 0 \\ \hline 0 & p_{23} \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline p_{31} & 0 \\ \hline p_{31} & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline p_{12} & p_{12} \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & -p \\ \hline \end{array}$$

$$p_1 = f - b \quad p_{23} = b + c - a - d$$
$$p_2 = g - e \quad p_{31} = b + e - a - f$$
$$p_3 = d - c \quad p_{12} = c + e - a - g$$
$$p = (a + d + f + g) - (b + c + e + h) \geq 0$$

$$H + \begin{array}{|c|c|}\hline p_1 & p_1 \\ \hline p_1 & p_1 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline p_2 & p_2 \\ \hline 0 & 0 \\ \hline p_2 & p_2 \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline p_3 & 0 \\ \hline p_3 & 0 \\ \hline p_3 & 0 \\ \hline p_3 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline p_{32} & 0 \\ \hline 0 & 0 \\ \hline p_{32} & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & p_{13} \\ \hline 0 & p_{13} \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline p_{21} & p_{21} \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline p & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array}$$

$$p_1 = c - g \quad p_{32} = f + g - e - h$$
$$p_2 = b - d \quad p_{13} = d + g - c - h$$
$$p_3 = e - f \quad p_{21} = d + f - b - h$$
$$p = (a + d + f + g) - (b + c + e + h) < 0$$

Table 5.1: We follow the $\mathcal{F}^3$ decomposition scheme from [53](Table 7 and 9) for E_l

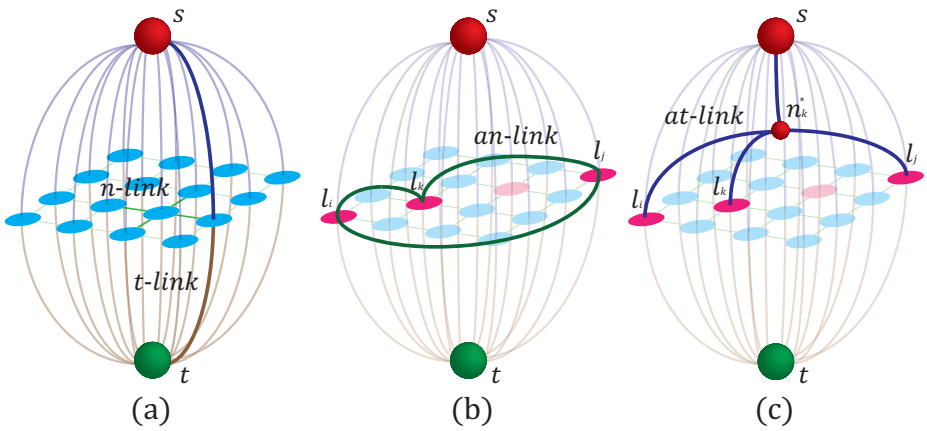


Figure 5.9: Graph construction for our LAGC algorithm. (a) The conventional graph for two-view stereo matching.(b) For a line segment (pink), we add auxiliary n-links (green). (c) We also add an auxiliary node n_k^* and auxiliary t-links (dark blue).

5.2 lists the percentage of bad pixels and the ranking (in subscripts) for each method. Compared with MVGC, LAGC effectively reduces errors and outranks MVGC (13 vs. 35 for non-occlusion, 14 vs. 38 for boundary, and 14 vs. 50 for all pixels). LAGC also preserves edges such as the feet of the table and the tripod and the arm of the lamp.

Next, we apply LAGC to the light field datasets. We compare our scheme with the recent globally consistent depth labeling (GCDL) scheme using the source code posted by the author [92]. We first test on a synthetic light field of a city scene composed of one million triangles. We render the scene using the POV-Ray ray-tracer

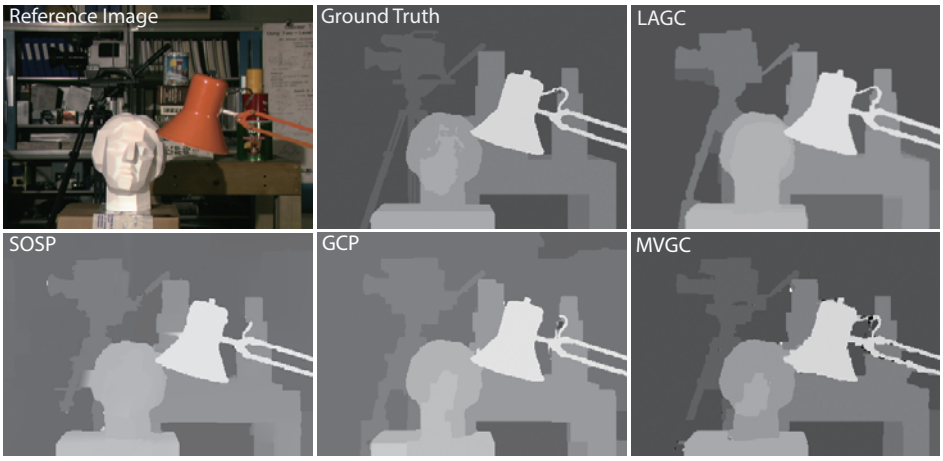


Figure 5.10: Stereo matching on the Tsukuba dataset. Our LAGC outperforms MVGC [54] and SOSP [125] but is slightly worse than GCP [98]. However, it better preserves edges, e.g., the left foot of the tripod. See Table 5.2 for numerical comparison.

[86] to generate an array of 17×17 images, each with a resolution of 1024×768. The scene has a disparity range from 0 - 16 pixels. Notice that the city scene exhibits repeated line patterns. Certain regions lack textures while the others contain complex textures. The scene hence is challenging for classical stereo matching. For this and the following examples, we fine-tune the parameters for both algorithms and compare only their best results.

The top row of Fig. 5.11 compares the disparity maps computed by GCDL and LAGC. GCDL misses fine details such as the contours of the chimneys and is highly

Algorithm	non-occlusion	all	discontinuity
LAGC	1.00_{13}	1.41_{14}	5.39_{14}
MVGC	1.27_{35}	1.99_{50}	6.48_{38}
SOSP	2.91_{103}	3.56_{92}	7.33_{57}
GCP	1.03_{14}	1.29_{5}	5.60_{16}

Table 5.2: Stereo matching using LAGC, MVGC [54], SOSP [125], and GCP [98] on Tsukuba. We show both the percentage of bad pixels and the algorithm's ranking (in subscripts)

noisy on surfaces with rich textures. Its error is also larger on distant buildings. In contrast, LAGC accurately preserves most fine details and robustly handles both distant and close objects. Although a real scene may not contain as many linear structures, our result demonstrates that LAGC is robust enough to handle such complex scenes.

We then experiment on real light field data. Fig. 5.11 row 2 shows the comparison on the Stanford Lego Gantry dataset [112] composed of 17×17 views at a resolution of 1280×960 of a Lego gantry crane model. The disparity range is between -3 to 5 pixels and we discretize it into 16 labels. On continuous regions such as the ground, LAGC produces much smoother disparity transitions whereas the result from GCDL contains large discontinuities. LAGC is particularly good at preserving edges, as shown on the hoist rope from the crane, the contours of the headlights and windows, etc. Fig. 5.11 row 3 shows the amethyst dataset which is expected to be challenging to LAGC: it lacks long linear structures but contains strong view-dependent features. The disparity range is small, from -3 to 3 pixels. We consider subpixel disparity with step size 0.2. LAGC can robustly handle this challenging scene and slightly outperforms GCDL, e.g., by better preserving the facets of the amethyst.

Finally, we test on a real light field acquired by the Lytro camera [71]. Lytro uses an array of 328×328 microlenses, each with 10×10 pixel resolution. We first resample the light field to a 17×17 light field at 800×800. The disparity range is ultra small (between -1 to 1 pixels). We discretize the disparity range using 0.2 step (10 disparity label). Notice that GCDL is originally designed to process the Raytrix data [89] that usually have a much larger disparity range. Although it uses subpixel for evaluating the structure tensor, directly applying GCDL on the Lytro data still results in poor disparity estimation (Fig. 5.11 row 4). LAGC, however, produces a better quality disparity map despite structure and texture similarities across the scene.

5.3.5 Discussions

We have presented a stereo matching framework by imposing ray geometry of 3D line segments as constraints. We have experimented on synthetic, pre-acquired

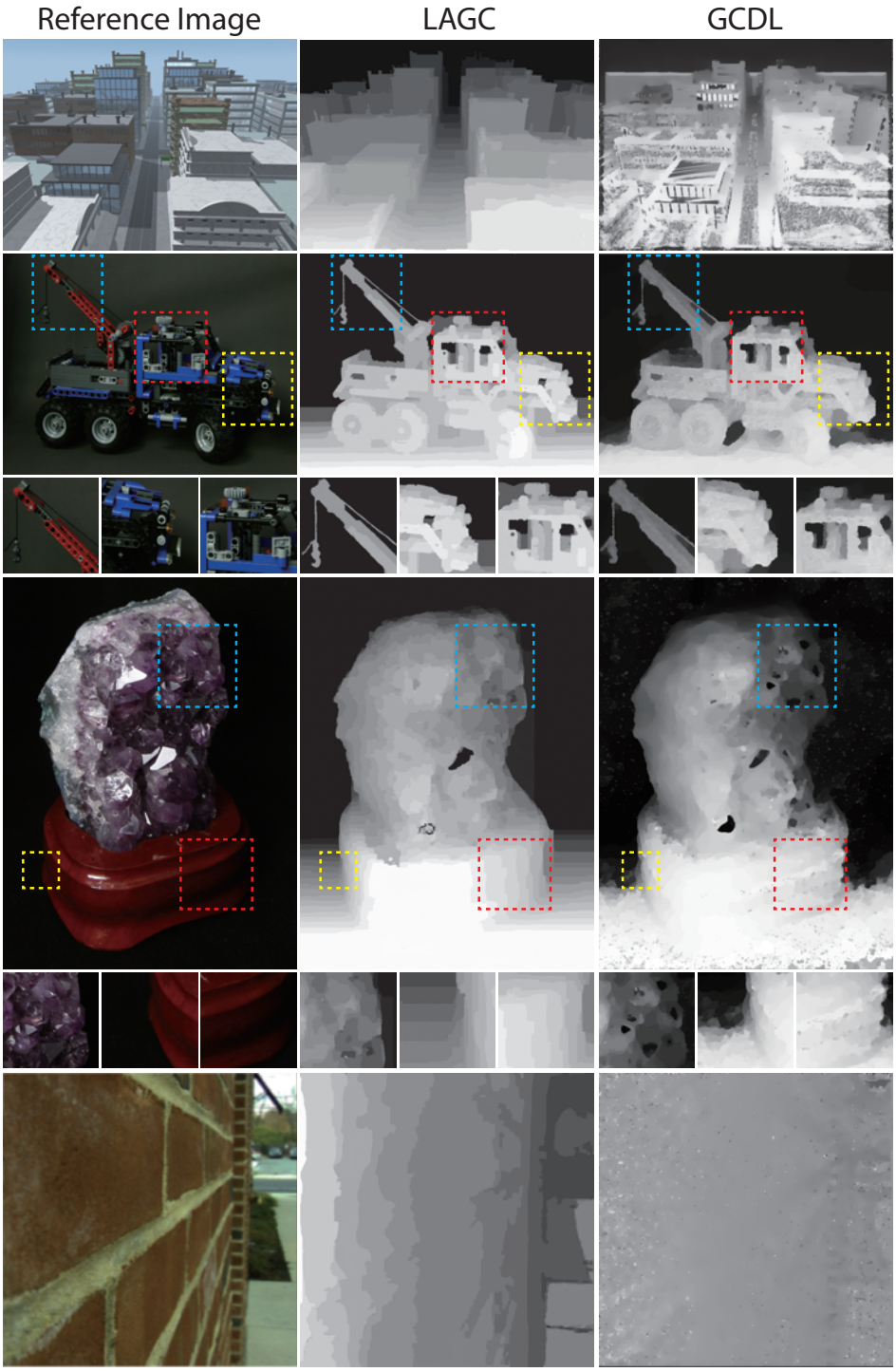


Figure 5.11: LAGC vs. GCDL [92] in light field. From top to bottom: a city scene light field ($17 \times 17 \times 1024 \times 768$) rendered using POV-Ray, the Stanford Gantry light field ($17 \times 17 \times 1280 \times 960$) and Amethyst light field ($17 \times 17 \times 768 \times 1024$), and a real light field captured by Lytro [71]

.

light field, and Lytro acquired light fields. An important following step is test our scheme on the Raytrix data which have a larger disparity range. In addition, given the increasing interest in light field imaging and the availability of commercial light field cameras, we also plan to build a light field stereo benchmark analogous to the Middlebury Stereo Portal [29], for evaluating light field stereo matching algorithms. Finally, it remains an open problem on how to handle view-dependent objects in both binocular and multi-view stereo. In the future, we will investigate robust algorithms for detecting and reconstructing these objects via light field analysis.

Chapter 6

UNIFIED SPATIAL ANGULAR ENHANCEMENT VIA LIGHT FIELD QUILTING

In this chapter, we present a high-dimensional image based rendering technique which takes multiple light fields as inputs and generates new light fields as outputs. We call our technique the "Light Field Quilting".

Our technique can be regarded as a general case of traditional image based rendering which aims to use a dense set of 2D images in place of 3D geometry to render a novel 2D view of the scene. To draw analogy from light field quilting to image based rendering, first, we interpret the image based rendering in the context of light field ray space.

In the ray space, a 2D image can be treated as a light field with only one angular sample for each spatial sample. For clarity, we call this kind of light field the 2D spatial light field. Therefore, traditional image based renderings aim to construct novel 2D spatial light fields by fusing multiple 2D spatial light fields using certain geometry proxies. The geometry proxy is crucial to the ray structure of the final 2D spatial light field. For example, a planar geometry leads to perspective panoramic views of the scene [80, 99, 4], a spherical geometry leads to a multi-perspective panoramic views of the scene [4], and the geometry of scene itself leads to novel views of the scene [64, 42]. Despite the exiting results, the final 2D spatial light field is limited by its sparsity on the angular sampling hence is always a static 2D image.

The recent advances of light field cameras allow us to easily take multiple 4D light fields towards the scene. With such a huge advantage, our technique takes a set of 4D light fields as inputs and produces novel 4D light fields depending on user defined hyper-geometry proxies in the light field space. Since the input light fields have denser

Figure 6.1: Our spatial quilting stitches 4 light fields (top row) captured by a rotating Lytro camera into a single wide FoV light field. The white circles show the enlarged red highlighted region of the light field images. The second row shows the EPIs (u, x slices) of each individual light field. The third row shows the shallow DoF renderings focusing at background sculpture (left) and foreground plants (right). The bottom row shows the quilted EPI based on the 4 EPIs on the second row.

angular samples, the result light field is no longer restricted by the angular limitation. Therefore, with this new technique, we can produce novel effects such as panorama with dynamic DoF, panorama with parallax, and novel views with larger parallax and shallower DoF.

Our technique is consisted of two key components: light field registration and light field stitching. To register two light fields L and $\tilde{L}$, similar to 2D image registration, we assume that L lies on a 5D hyperplane and we can project $\tilde{L}$ onto L with a 5D homography matrix. To estimate this matrix, we conduct correspondence matching of the scale invariant feature transform (SIFT) image features and then compute the optimal 5D homography matrix that minimizes difference of rays in the overlapped light field. Next, we use the homography matrix to warp $\tilde{L}$ to the L and seek to quilt through the overlapping region. This is also analogous to image stitching in synthesizing 2D panoramas from images. The key difference though is that we are dealing with a higher dimensional light field space. Specifically, to compute a cut in the overlapped subspaces, we build a 4D light field graph and apply graph cut optimization to locate the optimal quilting paths. Since light fields are high dimensional, brute-force implementation for computing the cuts can be extremely slow. We therefore employ a coarse-to-fine scheme [6]: after we compute the cut of the graph at a coarse level, we upsample the graph but prune unnecessary nodes by using the estimated coarse cut.

We demonstrate the approach for enhancing the light field resolution at different dimensions. For example, we can create a wide horizontal FoV light field from a series of light fields captured by rotating the Lytro camera on a tripod. We can also create an ultra-high spatial resolution light field using an array of Lytro cameras. The same structure allows us to increase the size of the virtual aperture and hence the bokeh. Finally, we can increase the parallax between light field views by orbiting the Lytro camera around the object of interest.

6.1 Related work

Our light field quilting work is related to earlier work in image-based modeling and rendering, classical image stitching, and the emerging light field superresolution research and classical image stitching.

6.1.1 Image-based Modeling and Rendering

Our technique can be viewed as image based rendering on a higher dimension. The major differences are: 1) We take 4D light fields as inputs. 2) We use 5-dimensional geometry proxies for light field projection instead of 2D ray projection. 3) We get novel 4D light fields as outputs. Similar to the image based rendering, our geometry proxies can be predefined or estimated, depending on different scenarios. In the applications of this paper, we explore using a 5D hyper plane as the geometry proxy.

Light Field Rendering and Lumigraph In computer graphics and computer vision, image-based modeling and rendering aim to generate a 3-dimensional model and then render novel views of this scene based on a set of 2D images of a scene. Even though the idea is old, it well represents the concept of recording and manipulating rays flowing in the scene for purposes such as synthesizing new views. For example, the main use of light fields in the very beginning is to synthesize new perspective views by performing 2D slices in the recorded 4D ray space [64]. In the ray space, the slicing approach can be interpreted as projecting the captured rays onto a 3D planar surface. However, rays coming from scene points out of this plane will appear as aliasing artifacts on the final result. To ameliorate this effect, Levoy et al. [64] and Davis et al. [30] resorted to pre-filtering of the light field with different kernels. The lumigraph [42] also relies on a dense set of views of the scene coupled with the scene geometry for constructing novel views. Gortler et al. [42] and Buehler et al. [24] resolved this issue by projecting the rays onto the given 3D scene geometry. Overall, most image based rendering approaches require certain geometry proxies for image construction.

Our work in many ways resembles the image and video stitching techniques for synthesizing panoramic images or videos. In particular, our light field quilting can be treated as finding the best seam among the given 4-dimensional spaces, a general case of most, if not all, 2D and 3D problems.

1D and 2D Panorama 1D and 2D panoramas mainly aim to synthesize a wide FoV image of the scene based on multiple captured views. There exists many approaches for generating 2D panoramas. 2D rotational panorama [107, 126, 44, 105] and its variants such as strip panorama [4], pushbroom panorama [95], X-slit panorama [138], etc. have been extensively studied over the past few years. Recently, Sargent et al. [83] proposed an impressive system to deal with the challenges of accessing appropriate parts of the gigapixel video as one pans and zooms.

3D Panorama 3D panorama aims to increase the spatial resolution on a 3D data such as stereoscopic images and videos. On the angular domain: Peleg et al. [80] proposed to generate omnistereo panoramas by mounting the camera on a rotating arm. However, this strategy suffers from visible seams and vertical parallax. More recently, Richardt et al. [90] proposed a solution for generating high quality stereoscopic panoramas. They first described robust solutions to correct issues such as perspective distortion and vertical parallax on the input images. Next, they apply an optical-flow-based technique to reduce aliasing during the stereo panorama generation. While generating impressive results, their method requires accurate optical-flow estimation, which is inconvenient to acquire in many scenes. On the time domain: Kwatra et al. [58] applied graph cuts in 2D and 3D to perform video texture synthesis in addition to regular image synthesis. Agarwala et al. [6] showed how to amend panoramic imagery with video textures such as water waves and blowing leaves to enliven content. Rav-Acha et al. [88] used a sweeping video to create a panoramic video of, for example, a very wide waterfall, by allowing time to vary across the panorama. Couture [26] proposed to generate loopable panoramic stereo videos with a pair of commodity video

Figure 6.2: Top row: a region of the result by Panorama light-field imaging [16]. Bottom row: the enlarged highlighted regions. Note that there exists severe boundary bleedings of the defocus regions which makes the result look artificial.

cameras. Their method uses full frames rather than slits and uses blending rather than smoothing or matching based on graph cuts. However, their results suffer from ghosting artifacts.

Light Field Panorama Closest to ours is probably the work by Birklbauer and Bimber [16] that stitches multiple light fields into a panoramic light field. Their method first computes all in focus images from each light field by estimating per-pixel frequency. Next, they stitch the all in focus images into a 2D panorama. Finally, they generate a focal stack by blending views from the original light fields onto the all in focus panorama and rely on linear view synthesis to recover a panoramic light field.

The main advantage of our algorithm over [16] is that our approach directly stitches the 4D light fields while their approach is based on lower dimensional processes such as stitching 2D all in focus images and linear view synthesis based on 3D focal stacks. Moreover, the per-pixel frequency analysis in [16] is error-sensitive. Specifically, the highest frequency of each pixel on the focal stack does not necessarily correspond to

the depth of the pixel due to occlusion and aliasing in angular dimensions. Therefore, this scheme leaves artifacts on depth edges Fig. 6.2. Moreover, all in focus estimation and linear view synthesis restrict [16] to Lambertian scenes.

6.1.2 Light Field Superresolution

Our goal shares the same interests with light field reconstruction and super-resolution on finding a higher resolution light field. However, the major difference is that our solution leverages multiple captured light fields to stitch into a super light field with higher spatial and angular resolution, while the light field reconstruction and superresolution focus on a single light field. Here we briefly review the prior work in this area.

Based on a single light field, Bishop et al. [17] formulated the spatial light field superresolution in a variational Bayesian framework. Wanner et al. [121] presented a unified variational framework to spatial-angular superresolution. However, both methods require accurate depth estimation as the prior knowledge.

There is also an emerging trend of reconstructing sparsely sampled light field for light field compression. Lehtinen et al. [62] explored the anisotropy in the temporal domain and enhanced the reconstruction quality by a large factor. Marwah et al. [72] used an overcomplete dictionary to reconstruct a sparse coded light field. Heide et al. [45] applied Markov Chain Monte Carlo sampling instead of uniform sampling on the target light field for better reconstruction. These techniques also have their limitations. First, they do not have the complete light field data hence require schemes such as training or approximation to acquire the higher resolution light field. Second, the additional schemes are slow and error-prone. A key advantage of our technique is its simplicity.

6.2 Algorithm Overview

To quilt N light fields, we first divide the problem into $N-1$ pairwise light field quilting problems, we then iteratively go though each pair to quilt the "super"

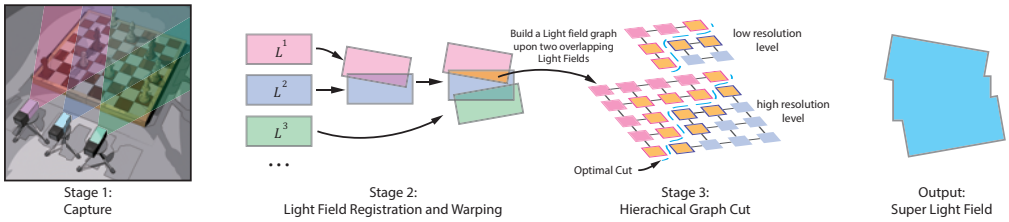


Figure 6.3: The pipeline of our proposed light field quilting algorithm. We represent the 4D light fields in 2D for simplicity.

light field. Similar to conventional multi-labeling problems such as disparity estimation [54, 20] and texture synthesis [58], our iterative process runs several rounds (each round with $N-1$ pairwise quilts) before finding the local minimum. In our experiments, commonly 1 round is enough to achieve reasonable results.

Fig. 7.9 shows our proposed processing pipeline. To quilt two light fields, our strategy is to model light field registration as a 5D homography matrix and then employ quilting to eliminate visual discontinuities/aliasing. It is worth noting that our technique is analogous to image stitching used for synthesizing panoramas: light field registration maps to image registration (Sec. 6.3) while light field stitching maps to image calibration and blending (Sec. 6.4).

Before proceeding, we explain the notations. To represent each ray in the light field, we follow the conventional two-plane parametrization (2PP). Every ray is parameterized by its intersection points with two planes: $[u, v]$ as the intersection with the camera plane Π_{uv} and $[s, t]$ as the second with sensor plane Π_{st}. The two planes are parallel to each other with distance f. Since we consider only pairwise quilting problems, we denote the first light field as L and the second as $\tilde{L}$.

To represent each captured light field image, we discretize Π_{st} and Π_{uv} (i.e., each discrete sample $[u, v]$ is the camera or microlens position and $[s, t]$ is the pixel location). Each light field is embedded in the 4D space where each point R in the space $[R_s, R_t, R_u, R_v]$ maps to a ray. We denote I_R as color (intensity) of R in L and $I_{\tilde{R}}$ as in $\tilde{L}$. Under this parametrization, each 2D slice (u = u0, v = v0) in the 4D space

corresponds to a view in the light field captured by a pinhole camera centered at [u0, v0].

Note that light field images captured by Lytro cameras do not directly resemble pinhole views in the scene. Therefore, we first map the in lens light field captured by the camera to out of lens camera. In this case, each sub-aperture image corresponds to a virtual pinhole view outside the mainlens. Note that the sensor of does not lie on the same plane as the microlens array, hence when we map the sensor to the virtual sensor of the virtual pinholes, there exists a relative movement between the virtual sensor and the virtual pinholes. We then parameterize this pinhole array for our light field quilting.

To fuse two light fields L and $\tilde{L}$, the brute-force approach would be to resample them onto a common 2PP. Specifically, we first define a common 2PP as $\Pi_{st} - \Pi_{uv}$ in the 3D space. Next, for each ray R in L and ray $\tilde{R}$ in $\tilde{L}$, we find the intersection points $[R_s, R_t, R_u, R_v]$ and $[\tilde{R}_s, \tilde{R}_t, \tilde{R}_u, \tilde{R}_v]$ on $\Pi_{st} - \Pi_{uv}$, hence we represent both L and $\tilde{L}$ with $\Pi_{st} - \Pi_{uv}$ and find the common subspace.

To use this approach, on one hand, we need to assign an extremely dense sampling frequency for the common 2PP in order to record all rays from L and $\tilde{L}$, otherwise, some rays will intersect at points in between our sampling locations, in that case, we will miss those rays. On the other, with the dense sampling, most points on the common 2PP will not have any information since no ray from L or $\tilde{L}$ will intersect at those points, hence making the process memory inefficient.

An alternative approach is to assign a fixed spatial-angular sampling frequency on the common 2PP and start intersecting rays from the 2PP to L and $\tilde{L}$ to fetch information. In this case, we avoid dense sampling. However, we face the problem of low sampling frequency of L and $\tilde{L}$, i.e. most rays intersecting L and $\tilde{L}$ will not coincide with any ray sampled by L or $\tilde{L}$. Again we need a dense sampling with low memory efficiency to capture all rays.

In Fig. 6.5, we show an example of brute-force resampling. First, we briefly explain our setup: light field L, $\tilde{L}$ and $\bar{L}$ (with common 2PP) are initialized with same

2PPs and sampling frequencies ($[u, v] = [10, 10]$, $[s, t] = [500, 500]$). We first rotate $\tilde{L}$ with 40 degrees clockwise around the v axis at the central view $[u = 5, t = 5]$, so that L and $\tilde{L}$ together capture a light field with large horizontal (s, u direction) FoV of the scene. Next, we rotate $\bar{L}$ by 20 degrees and resample all rays captured by L and $\tilde{L}$ by intersecting rays from $\bar{L}$ to L (in blue) and $\tilde{L}$ (in red) to fetch information. Note that in each $[u, v]$ view of $\bar{L}$, the $[s, t]$ samples are very sparse due to the undersampling of L and $\tilde{L}$. Moreover, $\bar{L}$ does not capture all information sampled by L and $\tilde{L}$.

To avoid the memory issue, an effective method is the unstructured lumigraph rendering [24]. However, without an accurate geometry proxy of the scene, such approach would introduce blurry and aliasing artifacts. Our high-dimensional geometry proxy assumes each light field as a 5D hyper plane. By representing the registration as the projection one plane to another, it better preserves the continuity on spatial and angular dimensions.

We present three applications using different capturing configurations: 1) Horizontal and vertical FoV enhancements. 2) Rotational parallax enhancement. 3) Translational parallax and bokeh enhancement.

6.3 light field Registration

As mentioned in the related work, traditional image-based rendering approaches use geometry proxies such as simple planes or cubes for synthesizing new views or stitching different views. The quality of the results depend on how well the proxy approximates the actual scene geometry and most previous techniques are designed for handling simple objects [64, 42] or scenes with few depth variations [25]. Conceptually, we can adopt the similar scheme for light field registration. For example, if we assume all captured rays are coming from planar surface in the scene, we could estimate a 3D homography based on feature correspondences of two views of two light fields images. However, the assumption of simple scene geometry contradicts what a light field camera aims to capture: for producing effective DoF, the scene should exhibit strong depth variations and cannot be approximated by simple planes. Therefore, if we use simple

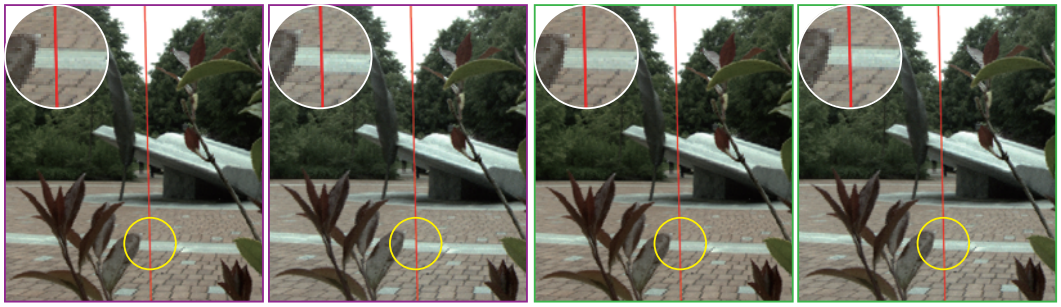


Figure 6.4: Comparison of 3D homography and light field homography on two views from L and $\tilde{L}$. The red line divides the view from L (left) and the view from $\tilde{L}$ (right). The white circle shows the enlarged yellow highlighted pavement. (a) Result of using 3D homography to warp view $[u = 5, v = 5]$ in L and $[u = 5, v = 5]$ in $\tilde{L}$. (b) Result of using the matrix in (a) to warp view $[u = 0, v = 0]$ in L and $[u = 0, v = 0]$ in $\tilde{L}$. (c) Result of using 5D light field homography to warp view $[u = 5, v = 5]$ in L and $[u = 5, v = 5]$ in $\tilde{L}$. (d) Result using the same 5D light field homography in (c) of view $[u = 0, v = 0]$ in L and $[u = 0, v = 0]$ in $\tilde{L}$.

geometry for quilting light fields, the results can exhibit strong discontinuity artifacts on different views. For example, by assuming the view $[u = 3, v = 3]$ in L and view $[u = 3, v = 3]$ in $\tilde{L}$ are related by a homography, we can compute a transformation matrix M for the warping of the two views. As shown in 6.4 (a), to the left red line shows the view in L and to the right shows the warped view in $\tilde{L}$. M successfully warped the two views. However, if we apply M for warping view $[u = 0, v = 0]$ in L and $\tilde{L}$, as shown in (b), the result exhibits severe discontinuity such as the pavement in front of the sculpture. A straight forward improvement is to find homography for each individual view pairs. However, since we do not have the correspondences views in the light fields, the search space will increase quadratically with the number of views.

In this dissertation, we present a novel light field registration technique that conducts pair-wise light field warping. Given two light fields L and $\tilde{L}$, we first assume that each 4D light field is lying on a different 5D hyperplane, similar to 2D images captured in 3D space. Therefore, warping from $\tilde{L}$ to L can be represented by a 5×5

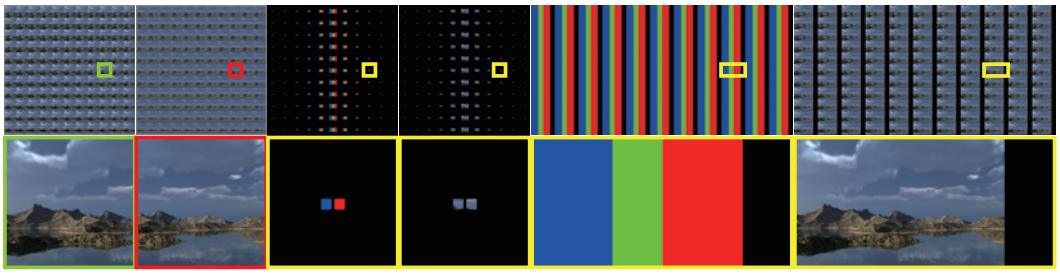


Figure 6.5: (a) and (b) columns: two synthetic light fields at $[u = 11, v = 11, s = 500, t = 500]$. (c): The resampling pattern of the new light field with brute-force light field resampling (blue from L and red from $\tilde{L}$). (d): The new light field at $[u = 11, v = 11, s = 500, t = 500]$. (e): resampling pattern of the new light field with our light field homography. (f): The new light field by our algorithm.

projective transform matrix M. We call it the light field homography. Specifically, we first embed each 4D ray into the 5D homogeneous space. M then maps each ray $\tilde{R} = [\tilde{R}_s, \tilde{R}_t, \tilde{R}_u, \tilde{R}_v, 1]$ in $\tilde{L}$ to $R = (R_s, R_t, R_u, R_v, w)$ in L as:

$$\begin{pmatrix} wR_s \\ wR_t \\ wR_u \\ wR_v \\ w \end{pmatrix} = \begin{pmatrix} M_{00} & M_{01} & M_{02} & M_{03} & M_{04} \\ M_{10} & M_{11} & M_{12} & M_{13} & M_{14} \\ M_{20} & M_{21} & M_{22} & M_{23} & M_{24} \\ M_{30} & M_{31} & M_{32} & M_{33} & M_{34} \\ M_{40} & M_{41} & M_{42} & M_{43} & M_{44} \end{pmatrix} \begin{pmatrix} \tilde{R}_s \\ \tilde{R}_t \\ \tilde{R}_u \\ \tilde{R}_v \\ 1 \end{pmatrix}. \tag{6.1}$$

Expanding Eqn. 6.1, we have

$$\begin{cases} R_s = (M_{00}\tilde{R}_s + M_{01}\tilde{R}_t + M_{02}\tilde{R}_u + M_{03}\tilde{R}_v + M_{04})/w \\ R_t = (M_{10}\tilde{R}_s + M_{11}\tilde{R}_t + M_{12}\tilde{R}_u + M_{13}\tilde{R}_v + M_{14})/w \\ R_u = (M_{20}\tilde{R}_s + M_{21}\tilde{R}_t + M_{22}\tilde{R}_u + M_{23}\tilde{R}_v + M_{24})/w \\ R_v = (M_{30}\tilde{R}_s + M_{31}\tilde{R}_t + M_{32}\tilde{R}_u + M_{33}\tilde{R}_v + M_{34})/w \end{cases}, \tag{6.2}$$

where $w = M_{40}P^1_s + M_{41}P^1_t + M_{42}P^1_u + M_{43}P^1_v + M_{44}$.

By Eqn. 6.2, each pair of corresponding rays in L and $\tilde{L}$ provides 4 equations, to estimate all 25 unknowns in M, we need to select at least 7 pairs of rays of the two light fields that minimize the difference in the overlapped subspace. Similar to 2D image registration, we use SIFT to find feature rays. We then perform global color matching to find the potential matching pairs. To remove outliers, our algorithm uses RANSAC where the 5D projective transformations are used as its precondition.

Conceptually, it is ideal to use a 4D SIFT feature detector. However, while the sampling in the spatial domain of an acquired light field is generally dense enough, the sampling of the captured light field on the angular domain [39] is much sparser (10×10 for Lytro cameras and 17×17 for Stanford light field data sets). Consequently, the ray samples can be highly discontinuous along angular dimensions, i.e., the disparity of its corresponding 3D point can be much larger than 1 pixel. Applying the gradient-based SIFT feature detector in the angular domain can lead to large errors. We therefore only use the 2D spatial SIFT feature detector. More sophisticated schemes based on depth estimation can be potentially used and is important future work.

Next, we apply the RANSAC algorithm (Alg. 1) to find the best m pairs of ray correspondences. The algorithm is a straightforward extension of the one used for the 2D homography estimation.

Finally, we use the SVD to estimate M by rewriting Eqn. 6.2 as:

Algorithm 1 Light Field Homography with RANSAC

Require: feature ray $\tilde{R}_i$ from $\tilde{L}$ and R_i from L ($i \in N$)

Minimum error $Err_{min} = \infty$.
Best homography matrix M_{min} = identity matrix.
Iteration $it = 0$.
Assign k, t and d empirically.
while $it < k$ **do**
  Initialize C with randomly selected m pairs of feature rays from L and $\tilde{L}$ with color difference smaller than t.
  Estimate homography matrix M via SVD from C.
  for every point $\tilde{R}_j$ in $\tilde{L}$ not selected in C **do**
    Warp $\tilde{R}_j$ onto L with M.
    if it corresponds with a ray R_j in L with color difference smaller than t **then**
      Add $\tilde{R}_j, R_j$ to C.
    end if
  end for
  if the number of pairs in C is larger than d **then**
    Recompute the homography matrix M via SVD from C.
    Measure the current error E by warping each ray $\tilde{R}_i$ onto R_i and measure the spatial distance.
    if $E < Err_{min}$ **then**
      $Err_{min} = E$.
      $M_{min} = M$.
    end if
  end if
end while

$$\begin{pmatrix} \tilde{R} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tilde{R}R_s \\ \mathbf{0} & \tilde{R} & \mathbf{0} & \mathbf{0} & -\tilde{R}R_t \\ \mathbf{0} & \mathbf{0} & \tilde{R} & \mathbf{0} & -\tilde{R}R_u \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{R} & -\tilde{R}R_v \\ & & ... & & \end{pmatrix} \begin{pmatrix} M_{00} \\ M_{01} \\ ... \\ ... \\ M_{24} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ ... \\ ... \\ 0 \end{pmatrix}, \tag{6.3}$$

where $\mathbf{0} = (0, 0, 0, 0, 0)$. In our experiments, setting $m > 8$ pairs of feature rays generates robust light field registration.

Fig. 6.5 demonstrates the result of our technique that warp $\tilde{L}$ to L. Yu [129] proved that the registration of two light fields with different 2PP is quadratic rational. As shown in Fig. 6.4, the light field homography successfully warped both views (shown in (b) and (d)) from $\tilde{L}$ to L. Even though our light field homography is only an approximation of the true registration, it better preserves the spatial-angular continuity of the warped light field, which is more crucial to human observers. As shown in Figure 6.5, the warped light field is more continuous both in spatial dimensions (in each $[u, v]$ view) and angular dimensions (across different $[u, v]$ views) compared with brute-force resampling.

6.4 Graph Cuts based Quilting Framework

Once we register the two light fields, we then set out to stitch them. The simplest case is to stitch two images (i.e., two 2D light fields) into a panorama where a graph-cut based solution is often adopted. We follow the same strategy for the 3D/4D case. Given two overlapping light fields, our goal is to assign a binary label l_R for each ray R in the overlapped region, to specify which one of the two light fields it should use. In the 2D case panorama stitching case, this can be done by first constructing a graph of pixels in the overlapped region and then search for a 2D cut through the region that

minimizes the overall differences. Although we adopt the same graph-cut approach for stitching higher dimensional light fields, the naive approach of separately stitching each 2D slice fails. This is because there lacks control over the consistencies of the cuts across different slices, e.g., two adjacent stitched slices may appear significantly different as shown in the second row of Fig. 6.7.

We instead directly conduct high-dimensional cuts. As shown in Fig 7.9, with estimated registration information, we first warp $\tilde{L}$ towards L to mark the overlapped subspace $\hat{L}$. In the quilting stage, we build a hierarchical 4D graph upon $\bar{L}$ and map the spatial angular coherence as edges in the graph. Finally, we conduct graph cuts to acquire the optimal seam and quilt L and $\tilde{L}$ into a super light field $\bar{L}$. Next, we find the optimal cut through the overlapped 4D space. We formulate the problem of finding the cut as energy minimization. Specifically, we define the energy function as:

$$E = \sum_{\bar{R} \in \bar{L}} E(l_{\bar{R}}) + \sum_{\bar{R}, \bar{R}' \in \mathcal{N}} E(l_{\bar{R}}, l_{\bar{R}'}) \tag{6.4}$$

where $E(l_{\bar{R}})$ denotes the cost of assigning $\bar{R}$ with label $l_{\bar{R}}$, $E(l_{\bar{R}}, l_{\bar{R}'})$ denotes the cost of assigning labels $l_{\bar{R}}$ and $l_{\bar{R}'}$ to $\bar{R}$ and its adjacent ray $\bar{R}'$, and $\mathcal{N}$ is the set of adjacent rays.

Recall that multiple light fields may overlap at the same region. This translates to an N-labeling problem, i.e., each ray can be assigned as one of the N labels. For high-dimensional graphs, it is well studied that computing the optimal cut is an NP-hard problem. We there adopt an approximation solution following the seminal work by Boykov et al. [20]. To reiterate, the approach first divides the problem into an iterative binary labeling problem. In each iteration, a new label is randomly selected and each ray R has to choose whether to stay as the original label or choose the new label to minimize energy function 6.4. Next, it maps the terms in the function onto a graph and conduct max-flow/min-cut algorithm to find the global minimum. To our knowledge, it is the first time that the graph-cut technique is used on light field.

6.4.1 Energy Formulation

Before defining the terms in the binary energy function 6.4, without loss of generality, we denote $l_{\bar{R}} = 0$ for assigning $\bar{R}$ to L and $l_{\bar{R}} = 1$ for assigning $\bar{R}$ to $\tilde{L}$. The first term $E(l_{\bar{R}})$ of Eqn. 6.4 guarantees that for any ray $\bar{R}$ in $\bar{L}$, if $\bar{R}$ does not lie in the overlapped space, it will be assigned with the label of the light field it comes from. Otherwise, we rely on $E(l_{\bar{R}}, l_{\bar{R}'})$ to determine its label. Specifically, we define $E(l_{\bar{R}})$ as:

$$E(l_{\bar{R}}) = \begin{cases} \infty & , \quad \bar{R} \notin \hat{L} \wedge ((\bar{R} \in L \wedge l_{\bar{R}} = 1) \vee (\bar{R} \in \tilde{L} \wedge l_{\bar{R}} = 0)) \\ 0 & , \quad otherwise \end{cases} \tag{6.5}$$

The second term $E(l_{\bar{R}}, l_{\bar{R}'})$ measures the spatial-angular coherence of the stitched light field. The key observation here is that to reliably stitch two light fields, we need to measure the differences of adjacent rays in both spatial and angular dimensions. Previous work [58, 5, 4] have used tailored energy functions in the 2D and 3D case:

$$E(l_{\bar{R}}, l_{\bar{R}'}) = |I_{\bar{R}} - I_{\bar{R}'}| + |I_{\bar{R}} - \tilde{I}_{\bar{R}'}|, \tag{6.6}$$

where $|\cdot|$ denotes the norm (e.g., L1 or L2). Eqn. 6.6 represents $E(1,0) + E(0,1)$, which corresponds to the cost of assigning the neighboring rays with different labels. However, this simple measurement does not well reflect the complexity of rays in the 4D light field space because it ignores cost of assigning the neighboring rays with the same labels. We instead use the following function:

$$E(l_{\bar{R}}, l_{\bar{R}'}) = |\tilde{I}_{\bar{R}} - I_{\bar{R}'}| + |I_{\bar{R}} - \tilde{I}_{\bar{R}'}| - |I_{\bar{R}} - I_{\bar{R}'}| - |\tilde{I}_{\bar{R}} - \tilde{I}_{\bar{R}'}|. \tag{6.7}$$

The new energy function represents $(E(1,0) + E(0,1)) - (E(0,0) + E(1,1))$, where $E(1,0)+E(0,1)$ corresponds to the cost of assigning the neighboring rays with different labels and $E(0,0) + E(1,1)$ correspond to the one that assigns the rays with the same label. The new term computes the difference of the two cases and use it as the first order smoothness term in global optimization.

Note that in order to run standard graph cuts (α-expansion), $E(l_{\bar{R}}, l_{\bar{R}'})$ must be regular (non-negative). To guarantee this property, we propose two schemes.

Truncation In this scheme, if $E(l_{\bar{R}}, l_{\bar{R}'})$ happens to be negative, we simply truncate it to zero, i.e., $E'(l_{\bar{R}}, l_{\bar{R}'}) = max(0, |\tilde{I}_{\bar{R}} - I_{\bar{R}'}| + |I_{\bar{R}} - \tilde{I}_{\bar{R}'}| - |I_{\bar{R}} - I_{\bar{R}'}| - |\tilde{I}_{\bar{R}} - \tilde{I}_{\bar{R}'}|)$.

Linear Mapping Note that the truncation may bias towards assigning the rays with the same label by chopping the negative values to zero and maintaining the positive values. To avoid it, we adopt a normalization step to map the range of $E(l_{\bar{R}}, l_{\bar{R}'})$ to $[0, 1]$. Specifically, we first compute the cost $E(l_{\bar{R}}, l_{\bar{R}'})$ for all neighboring rays $\bar{R}, \bar{R}'$ based on Eqn. 6.7 and find the min and max values. We then map $[E_{min}, E_{max}]$ to $[0, 1]$ through linear mapping as:

$$E'(l_{\bar{R}}, l_{\bar{R}'}) = \frac{E(l_{\bar{R}}, l_{\bar{R}'}) - E_{min}}{E_{max} - E_{min}}. \tag{6.8}$$

Gradient Compensation Since seam (cut) boundaries are more prominent in the low frequency regions than in the high frequency ones, previous approaches also incorporate gradient priors to better preserve smoothness. In our case, since the angular dimensions are generally sparse due to undersampling, directly measuring the gradient along these dimensions may introduce large errors. We therefore only measure the gradients in the spatial domain as:

$$E_G(l_{\bar{R}}, l_{\bar{R}'}) = \frac{E'(l_{\bar{R}}, l_{\bar{R}'})}{|\mathrm{G}_s(\bar{R})| + |\mathrm{G}_t(\bar{R}')|}, \tag{6.9}$$

where $\mathrm{G}_x(\bar{R})$ measure the gradient of $\bar{R}$ on dimension x.

6.4.2 Graph Construction

Next we construct the graph in a way that we can reuse the max-flow/min-cut algorithm to minimize our energy function. We follow the general purpose graph construction framework by Kolmogorov and Zabih [53]: each ray in the new light field L is a node in the graph, therefore the graph is 4 dimensional. We then add the source node S for label 0 and sink node T for label 1, the t-links from the graph nodes to S or

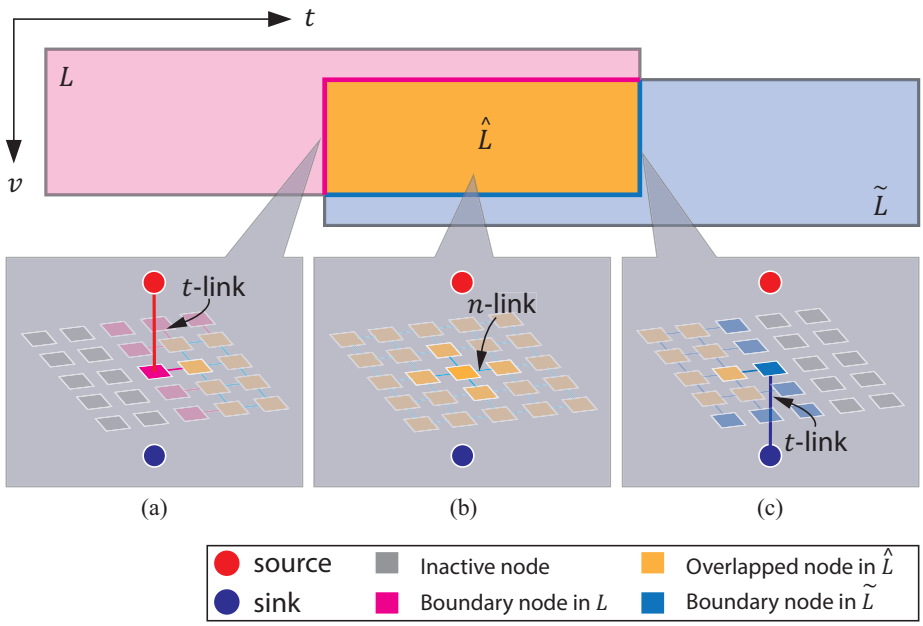


Figure 6.6: Graph construction for our light field quilting algorithm. Top row: Warped light field L and $\tilde{L}$ with overlapped subspace $\hat{L}$ (simplified in 2D). (a) and (c): The enlarged boundary regions of L and $\tilde{L}$. Boundaries nodes in L and $\tilde{L}$ are linked with source/target. They are also linked to nodes in $\hat{L}$ with ∞ capacity. (b): Nodes in $\hat{L}$ does not have t links but n-links.

T, and the n-links between the graph nodes using 8-connectivity (2 neighbors in each dimension).

Given two overlapped light fields L, $\tilde{L}$, and the overlapped subspace $\hat{L}$, we classify the nodes into three types:

1. Active node: the ones locates in $\hat{L}$.
2. Boundary node: locates in L or $\tilde{L}$ but at least one of its neighbor is an active node.
3. Inactive node: locates in L or $\tilde{L}$ and all neighbors are in L or $\tilde{L}$.

As shown in Fig. 6.6 (a), we add edges to the graph by first marking the inactive nodes in L and $\tilde{L}$. These nodes only have t-links connected to S if it is within L and to T if it is within $\tilde{L}$. They labels are fixed in prior and we can ignore them in the graph cut process. Next, we add edges to the boundary notes. Specifically, we add t-links to each boundary node to either S or T with ∞ capacity, depending on if it belongs to L or $\tilde{L}$. We also add n-links for connecting each boundary node to its neighboring active nodes with weight ∞ (Fig. 6.6 (a), (c)). This is because the boundary node must belong to either S or T. Finally, as shown in Fig. 6.6 (b) for each active node R_i, it is not connected to either S or T. Instead, it is only connected to its active node neighbor R_j, with capacity $E_G(l_{R_i}, l_{R_j})$.

To illustrate the importance of adding angular coherence, we conduct an an experiment on the Tsukuba dataset [29] from the Middlebury database as shown in Fig. 6.7. The Tsukuba dataset is a 5×5 light field. We select two columns (Column 1 and Column 3) and stitch them horizontally to increase the horizontal FoV. (row 1×5 views from the original 5×5 light field and spatially stitch them to increase the horizontal FoV.

We illustrate the case in 2D for clarity. The first row shows the two 1×5 light fields and the two slices on v and t dimensions that correspond to their epipolar plane images (EPI). A successful quilting should not only maintain the smoothness on the spatial dimension t, but also on angular dimension v. The second row shows the labeling on all views by adding only spatial constraints same as the traditional image

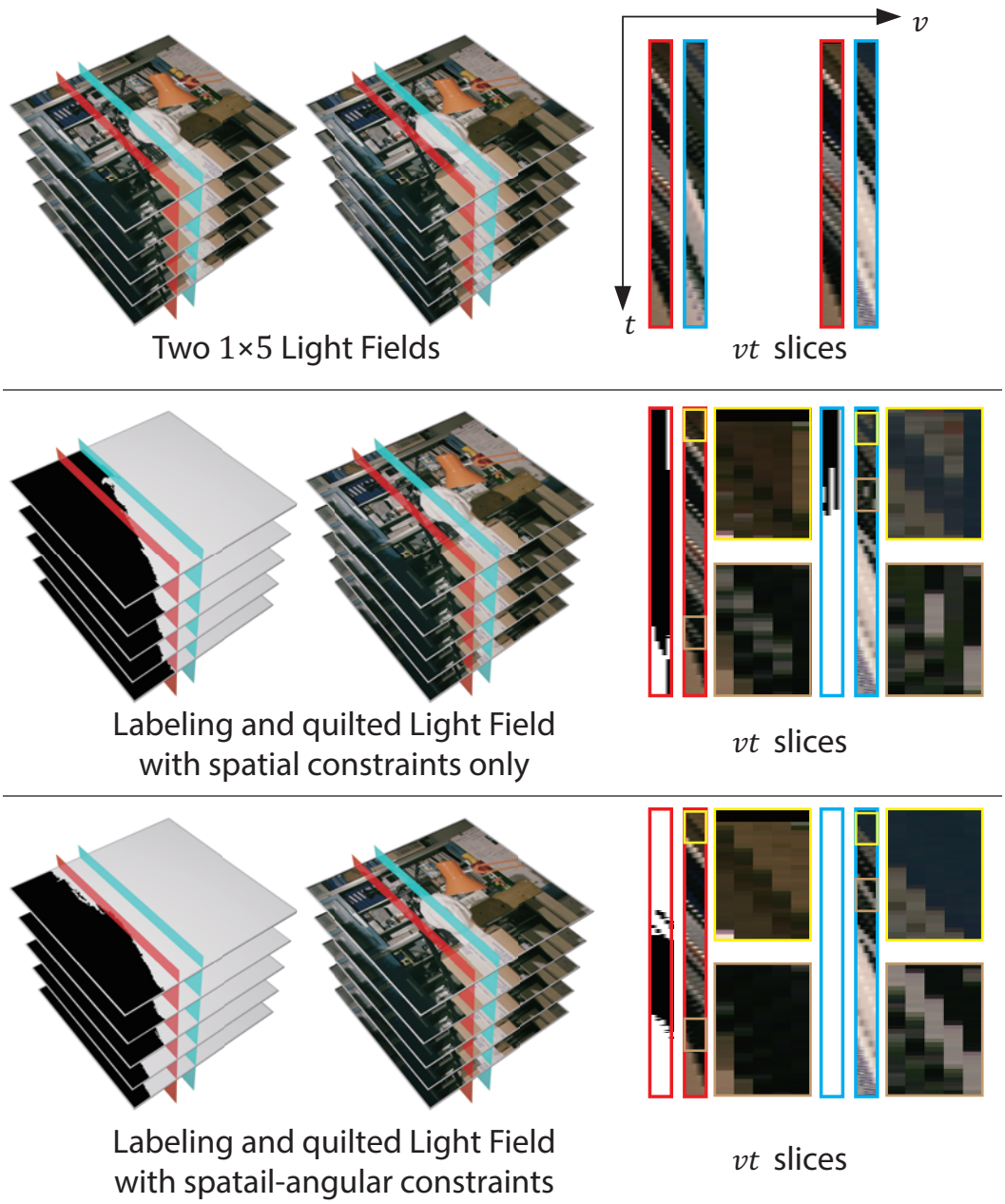


Figure 6.7: Top row: Two 3D light fields $(v, s, t$ dimensions) from Tsukuba dataset [29]. The red and blue slices are the EPIs $(v, t$ slices) of each light field. Second row: Labeling and quilted light field by warping and graph-cut with only spatial constraints. Notice the discontinuity on EPIs of new light field. Third row: Labeling and quilted light field by our light field quilting. Notices the consistency on the EPIs.

stitching techniques. As shown in the second row of Fig. 6.7, there exists noticeable discontinuities on the EPIs. This is because the seams only represent local minimum within each view, but not the cost cross different views. When using the stitched light field to synthesize new views, we observe strong inconsistency across the views in the t dimension. By enforcing angular coherence, our approach is able to both preserve the smoothness on the EPI and maintain coherence across views in the stitched LF.

Hierarchical Graph Cuts Recall that our brute-force approach add edges at all dimensions. As a result, the 4D graph is very large. For example, given two light fields at $[u = 11, v = 11, s = 800, t = 800]$, we may construct a graph with roughly 70 million nodes and 0.6 billion edges. Applying the max-flow/min-cut algorithm on such graph incurs significant memory and computational overhead. In fact, on a computer with Intel i7-3930 CPU and 64GB memory, stitching only one pair of light fields captured by Lytro using our approach demands tens of gigabytes of memory and takes hundreds of hours to compute.

We resort to the coarse-to-fine approach [6] for simultaneous speeding up the graph cuts process and reducing the memory requirements. Our strategy is to first conduct graph cuts on a lower-resolution graph to find an approximate cut. We then go one level finer and use the approximate cut to eliminate unnecessary nodes in the new graph and reapply the graph-cut. Specifically, we first build a low-resolution graph G' using the same graph construction process where each node now represents a patch of rays in the light field L. Next, we conduct graph cuts on G' and map each labeled nodes to a patch of nodes in the graph G the original resolution. The nodes that lie far away from the cut will keep their label obtained on G'. Only nodes near the cut will be deemed active or boundary. This significantly reduces the number of nodes and edges of the graph and greatly accelerates the graph-cut algorithm. Although more levels can be used in this coarse-to-fine approach, we find in most cases two levels are generally sufficient, with first level at $\frac{1}{4}$ resolution on each dimension and second level at $\frac{1}{2}$ resolution.

6.5 Results

We conduct our experiments on PC with Intel i7-3930 CPU and 64GB memory. For synthetic scenes, we use Bryce raytracer for simulating a set of light fields. For real scenes, we use the commodity Lytro light field camera to capture a set of light fields at spatial resolution 328×378 and angular resolution 10×10 on the plane of the microlens array. We demonstrate our approach to create four different types of light field effects: light field panorama, light field mosaic, rotational parallax enhancement, and translational parallax enhancement.

Note that the the in-lens light field changes with the main lens focal length. To maintain the light field during the capture process, we fix the mainlens focal length. We also leverage the creative mode provided by the camera to match the shutter speed and ISO to minimize the color difference among light field images. Since the Lytro SDK is not open source, we use the toolkit by Dansereau et al. [28] to extract the raw images and conduct calibration, anti-vignetting and demosaicing.

6.5.1 Light Field Panorama

Different from traditional 1D and 2D panorama, light field panorama not only gives a large horizontal FoV of the scene, it also introduces dynamic DoF effects and parallax effects.

Figure 6.8 illustrates our light field panorama result on a synthetic mountain scene. To render four light fields, we first synthetically build 11×11 camera array with each camera at a resolution of 500×500 covering 60 degree horizontal and vertical FoV. We subsequently rotate the the camera array 30 degree at its the central camera CoP around its central camera up direction (aligned with y axis) to capture each light field. The overall horizontal FoV covered by all four light fields is approximately 120 degrees.

The top row in Fig 6.8 shows the central views of the four light fields. The middle row shows the EPI image of the red highlighted slice in each view. Notice that there exists subtle difference on the common FoV on the EPIs, e.g. the disparity of

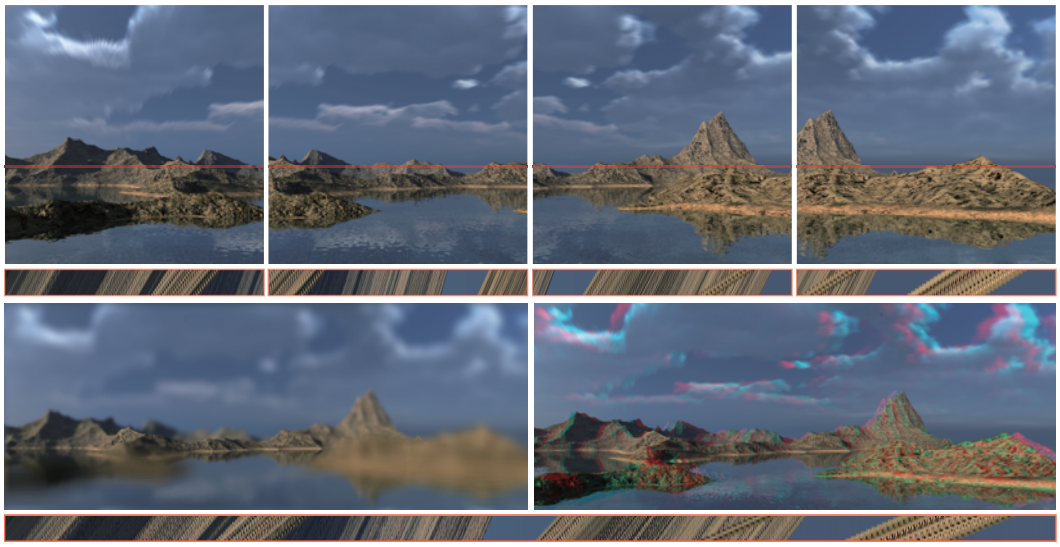
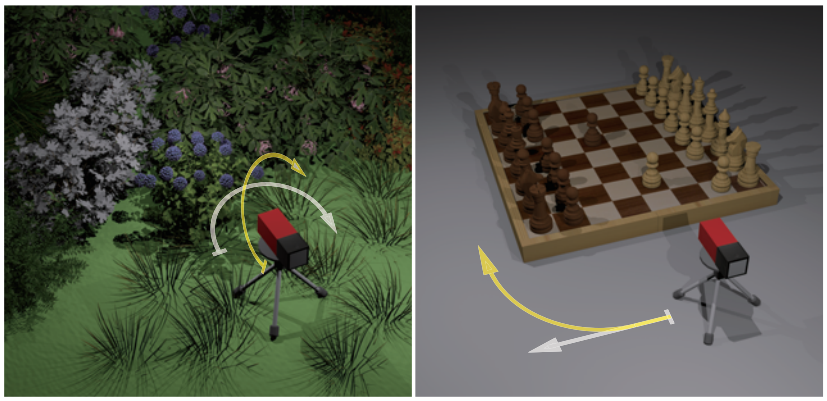


Figure 6.8: Light field quilting on a synthetic mountain scene. Top row: Central view of each light field. The red line highlighted the u, s slices of the EPIs. Second row: the EPIs (u, s slices) of each light field. Third row: The shallow DoF rendering (left) of the new light field, and the red-cyan anaglyph rendering of the new light field. Bottom row: The quilted EPI.

the mountain at the beginning of the third EPI is larger than the one at the end of the second EPI. This is because we only preserve the ray geometry of the central view while the rest views can exhibit inconsistency. We rely on the stitching framework to reduce inconsistency. The third row shows a synthetic shallow DoF image rendered using the stitched light field. Notice how the in focus regions appear as sharp as the original all-in-focus image whereas the defocus blurs vary smoothly with respect to scene depth. The bottom EPI image of red highlighted slice of the stitched light field demonstrates that our framework is able to maintain both spatial and angular consistency.

Figure 7.1 shows our light field panorama result on a real sculpture scene. To capture this scene, we mount a Lytro camera on a tripod and horizontally rotate it 4 consecutive times, each about 30 degree away from the center (Fig. 6.10 (a)). The first row shows the raw light field images of the captured 4 light fields. Note that we do not and cannot guarantee that the rotation is exactly around the central view.

Figure 6.9: The light field quilting on a real garden scene. Top row and second row: Shallow DoF rendering focusing at the background fountain (top) and foreground flowers (second). Third row: The EPI (u, s slice) of the red highlighted line. Bottom row: red-cyan anaglyph rendering of the new light field.



(a) Spatial Quilting (b) Angular Quilting

Figure 6.10: The light field capturing processes for the applications in this dissertation. (a) Capture process for spatial quilting: The white arrow shows the process of horizontally rotating the Lytro camera on a tripod for capturing a 1D light field panorama.. The yellow and white arrows together show the process of capturing a 2D light field panorama. E.g. 4 steps on each arrow will build a 4×4 light field array. (b) Capture process for angular quilting: The yellow arrow shows the process of capturing a rotational light field array for orbiting parallax enhancement. The white arrow shows the process of capturing a translational light field array for translating parallax enhancement.

As a result, we observe large differences on the EPIs of across light fields. The third row shows our synthetic shallow DoF results focusing at the background sculpture and foreground plants. And the bottom row shows the red-cyan anaglyph image generated by overlapping two views in quilted light field. The fourth row shows the stitched EPI from the four individual EPIs. Without using elaborate setups, our approach is still able to successfully preserve the smooth transitions in both the spatial and angular dimensions. Moreover, unlike all-in-focus image based light field panorama, our approach does not rely on the estimation of local gradient of the focal stack or any other depth-estimation based schemes. This therefore minimizes the visual artifacts at the occlusion boundaries on the rendered image.

The garden scene shown in Figure 6.9 is particularly challenging for light field stitching due to rich depth layers and complex occlusion conditions. To capture this

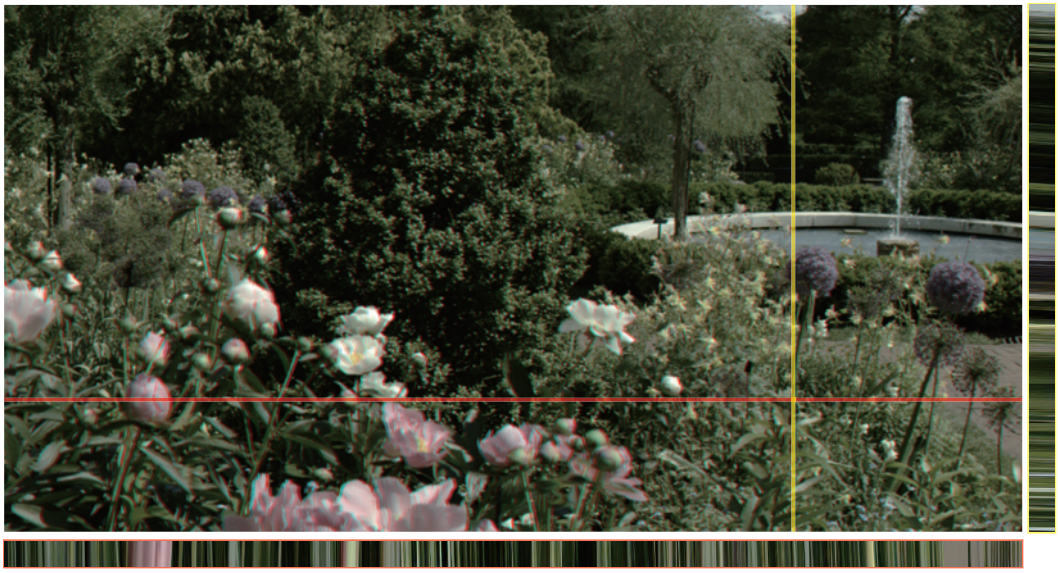


Figure 6.11: 2D light field panorama (in red-cyan anaglyph rendering) quilted by a 5×4 light field array using our light field quilting algorithm. The red and yellow lines highlight the u, x and v, t EPIs shown on the bottom and right respectively.

scene, we mount a Lytro camera on a tripod and horizontally rotate it 7 times at about 30 degree each time. To capture the dynamic fountain in the panorama, we purposely leave it outside the common FoVs of any two overlapped light fields so that we avoid dealing with quilting on the time domain. The first and second row shows the shallow DoF rendering with far and close focuses respectively. The red-cyan anaglyph image on the third row and the EPI image on the bottom demonstrate that even in such complex situations, our algorithm is still able to synthesize a visually pleasing light field with a much larger FoV.

6.5.2 Light Field Mosaic

Similar to light field panorama, our light field mosaic stitches a 2D grid of light fields with a large horizontal and vertical FoV. In Fig. 6.11 (a), we capture the garden scene by rotating a Lytro camera on a tripod horizontally 5 times and vertically 4 times, each time at 30 degree interview. This results in a 5×4 angular grid around

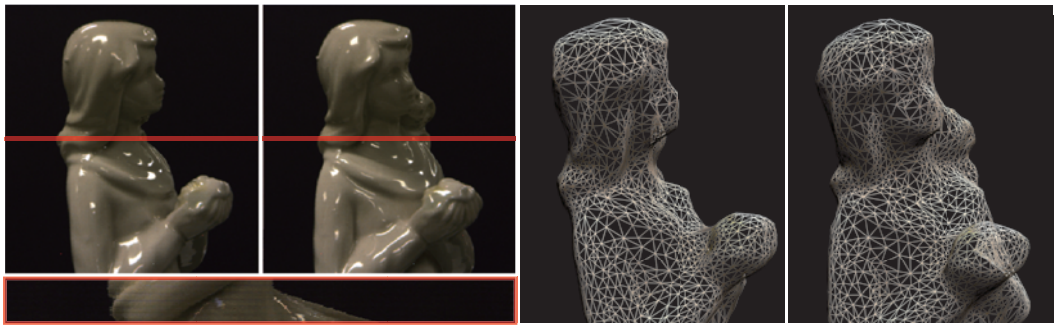


Figure 6.12: The quilted light field with increased orbiting parallax. (a) Two views from the quilted light field. The red line highlights the "orbiting" u, x EPI of the new light field. (b) and (c) show the reconstructed 3D mesh based on the new light field.

the tripod. In this case, we do not restrict the fountain in one light field but rely on the stitching technique to optimize the seam. The result shows red-cyan anaglyph image generated by overlapping two views in stitched light field. The two EPI images show that our algorithm is able to preserve angular consistency on both ux and vy dimensions. Moreover, although each individual light field captures the fountain at different time, our stitching technique is still able to synthesize water flows from the fountain with little visual artifacts.

6.5.3 Orbiting parallax enhancement

Rotational parallax enhancement aims to increase the freedom of viewpoints orbiting an object by quilting together multiple light fields. Our setup is similar to capturing and rendering concentric mosaic [99]. In [99], an horizontal array of cameras are rotated concentrically to capture a 3D concentric mosaics. Ours used a light field camera to orbit around an object. However, the direction of their cameras are tangential to the orbiting circle while ours is facing the center object. We aim to combine all the captured light fields into a single "circular" light field around the object, with each light field representing a piecewise linear approximation of the arc on circle. we call it

"orbiting light field".

To capture a orbiting light field, we place it on a rotation table in front of a constant color background. We fix the Lytro camera while rotating the object. Since the background is constant, this is equivalent to rotating the camera around the object (Fig. 6.10 (b)), which is more difficult to setup. Since the Lytro camera does not support continuous capture mode and the camera requires about 3 second to record each capture before it is ready for the next capture, we manually shoot each image in 5 seconds. The speed of the rotation table is at 0.1 degree per second, therefore we take each shot 0.5 degree after from its previous shot, so that all the light fields form a piecewise linear approximation of the circular light field.

In Fig. 6.12, we acquire 60 light fields to cover around 30 degree of the circular light field. The top row shows two views from at angle 0 and 30 rendered with a small aperture, and the bottom row shows the "rotational" EPI image of the red highlighted slice. Our algorithm is able to preserve the smooth transition on the rotational EPI image. With the increased rotational parallax, we can synthesize novel views rotating around the object with a large angle while a single light field in restricted in a small linear motion.

Finally, we apply our orbiting light field to the 3D reconstruction application. To get the 3D mesh, we first estimate the 3D point cloud from our light field with structure from motion [101] and then apply 3D Delaunay triangulation [13]. As show in Fig. 6.12, we are able to recover fine details such as the hair strands of the statue, demonstrating the benefits of having a lumigraph of light fields in 3D reconstruction.

6.5.4 Translating parallax enhancement

Translating parallax enhancement aims to increase the bokeh of the shallow DoF rendering and the freedom of viewpoints of the novel view rendering by quilting multiple translating light fields on the $[u, v]$ plane. The bokeh of traditional light field rendering is confined by the number of views in the light field, and the freedom of viewpoints is restricted by the small baseline between the first view and the last view

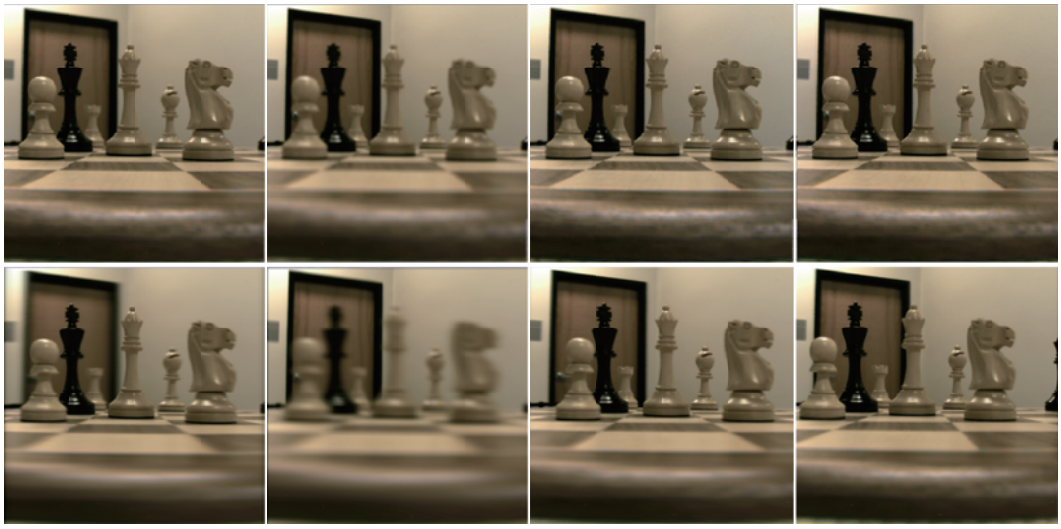


Figure 6.13: The quilted light field with increased parallax and bokeh. The top row is using the central light field of the captured light field array. The bottom row is using the quilted light field. (a) and (b) Shallow DoF renderings of the chess scene focusing at foreground queen chess piece and the background door respectively. (c) The leftmost view of the scene. (d) The rightmost view of the scene.

on u, v dimensions. By increasing the number of views of the scene by quilting multiple light fields captured at different locations on the $[u, v]$ plane, our algorithm is able to increase the bokeh and parallax at the same time.

As shown in Fig. 6.10 (b), To capture the light fields, we linearly translate the light field camera in front of the scene with each shot 0.5 mm away from the other. Note that this distance does not guarantee views from one captured light field L will match views from next captured light field $\tilde{L}$. In fact, as mentioned in Sec. 6.13, with a Lytro camera, translating the light field will not only involve angular shifting but also introduce spatial movement of the scene in each light field views. Therefore we apply light field warping to roughly match L and $\tilde{L}$ and rely on spatial angular stitching to find the best seam. The top row of Fig. 6.13 shows the dynamic DoF effect ((a) and (b)) and parallax effect ((c) and (d)) of a single view. Due to the sparse sampling of a single light field, the bokeh of the rendered image is very small and the parallax is hard to notice.

In the bottom row, we stitch 21 light fields together and the resulting light field can be used to synthesize a much shallower DoF. As shown in (a), notice the background door appears much blurrier compared with the result from a single light field when we focus at the foreground queen chess piece. We can also observe the smooth transition of the blurriness on the chess pieces on different planes, when we shift the focus to the background. This effect is harder to observe when using only one light field. (Fig 6.13 (b)).

We also use the resulting light field as an image-based rendering primitive to synthesize new views with extended DoF. Compared with the result using a single light field, our result is able to smoothly translate the viewpoint from the rightmost position when we see almost half of the black piece on the right (as shown in (d)), to the leftmost position when the black piece disappears. While the result with a single light field is restricted within a small region.

6.6 Discussions and Conclusions

We have presented a light field quilting technique which takes multiple light fields as inputs and generates new light fields with increased spatial and angular resolution as outputs.

Light Field Homography Currently we use the 5D homography to model the warping between light fields to preserve spatial-angular continuity. An immediate future direction is to explore the depth based warping to better represent the ray geometry. In our homography estimation, thresholds such as k, t, d are empirically chosen. We plan to leverage image statistics for automatically assigning those values.

Gradient Domain Composition Gradient domain composition by solving a Poisson equation is an effective way to match the colors in 2D image stitching. Theoretically, it could also be generalized onto the light field quilting case. However, due to the undersampling on the angular dimensions by conventional light field cameras, it is not reasonable to measure 4D gradients on each light field. Therefore in our experiment, we chose to lock the shutter speed and ISO during capturing to maintain the color tones of different light fields.

Chapter 7

ENHANCING TEMPORAL RESOLUTION: A COMPUTATIONAL CAMERA APPROACH

As discussed in Sec. 2.6.3, very little work has been conducted to improve the temporal resolution of light field imaging. In fact, by far nearly all acquired light field data are for static scenes. An exception is the data acquired by the light field camera array. The Stanford light field camera array [122, 123, 114, 115] is a two dimensional grid composed of 128 1.3 megapixel Firewire cameras which stream live video to a stripped disk array. The large volume of data generated by this array forces the DoF effect to be rendered in post processing rather than in real-time. Furthermore, the system infrastructure such as the camera grid, interconnects, and workstations are bulky, making it less suitable for on-site tasks. The MIT light field camera array [127] uses a smaller grid of 64 1.3 megapixel usb webcams instead of Firewire cameras and is capable of synthesizing real-time dynamic DoF effects. Both systems, however, still suffer from spatial aliasing because of the baseline between neighboring cameras. The camera spacing creates appreciable differences between the pixel locations of the same scene point in neighboring cameras producing an aliasing effect at the DoF boundary when their images are fused.

More recently, Agrawal et al. [10] proposed a mask based optical design to achieve spatial-angular-temporal tradeoffs using a time-varying aperture mask and a static mask close to the sensor. Their design allows variable resolution tradeoff depending on the scene. However, their method has three major limitations: 1) The output video sequence is at a much lower resolution than the captured image due to the resolution tradeoff. 2) The dynamic components in the scene lose refocus capability. 3)

The masks on the aperture and the sensor greatly reduce the light efficiency of the design.

In this chapter, I introduce a stereo based light field camera that can acquire dynamic light field videos and synthesize dynamic refocusing on the fly. Our solution builds upon a novel hybrid stereo-lightfield solution. Our goal is to first recover a high-resolution disparity map of the scene and then synthesize a virtual light field for producing dynamic DoF effects. Despite recent advances in stereo matching, recovering high-resolution depth/disparity maps from images is still too expensive to perform in real-time. We therefore construct a hybrid-resolution stereo camera system by coupling a high-res/low-res camera pair. We recover a low-res disparity map and subsequently upsample it via fast cross bilateral filters. We then use the recovered high-resolution disparity map and its corresponding video frame to synthesize a light field. We implement a GPU-based disparity warping scheme and exploit atomic operations to resolve visibility. To reduce aliasing, we present an image-space filtering technique that compensates for spatial undersampling using mipmapping. Finally, we generate racking focus and tracking focus effects using light field rendering. The complete processing pipeline is shown in Figure 7.2.

We map all processing stages onto NVIDIA's CUDA architecture. Our system can produce racking focus and tracking focus effects with arbitrary aperture sizes and focal depths for the resolution of 640×480 at 15 fps, as shown in the supplementary video. This indicates that if we capture the video streams at the same frame rate, we can display the refocused stream simultaneously. Our system thus provides a low-cost, computational imaging solution for runtime refocusing, an effect that is usually the domain of expensive movie cameras with servo-controlled lenses. Experiments on both indoor and outdoor scenes show that my framework can robustly handle complex, dynamic scenes and produce high quality results. Figure 7.1 shows the result of my system on a parking lot scene.

Figure 7.1: Depth of Field effect on a parking car scene using my system.

7.1 Hybrid-Resolution Stereo Camera

We first construct a hybrid stereo camera for recovering high-resolution disparity map in real-time. Our system uses the Pointgrey Flea2 camera pair to produce one *high-resolution color* video stream and one *low-resolution gray-scale* video stream. We synchronize frame capture to within 125μs by using the Pointgrey camera synchronization package. A unique feature of my approach is coupling my Hybrid-Resolution Stereo Camera with a CUDA processing pipeline for real-time DoF synthesis. Sawhney et al. proposed a hybrid stereo camera for synthesis of very high resolution stereoscopic image sequences [93]. Li et al. also proposed a hybrid camera for motion deblurring and depth map super-resolution [65]. Our configurations, however, have many more advantages. First and foremost, it provides a multi-resolution stereo matching solution that can achieve real-time performance (Section 7.2). Second, the lower bandwidth requirement also allows my system to be implemented for less expense on a greater number of platforms. Stereo systems that stream two videos at 15 fps and 640×480 resolution can produce up to 27.6 MB of data per second. By comparison, my hybrid-resolution stereo camera only produces slightly more than half that rate of data. Although my current implementation uses Firewire cameras, the low bandwidth demands of my solution make it possible to use a less expensive and more common alternative like USB 2.0, even for streaming higher resolutions such as 1024×768. Finally, compared to off-the-shelf stereo cameras such as Pointgrey's Bumblebee, my system has several advantages in terms of image quality, cost, and flexibility. For example, the form factor

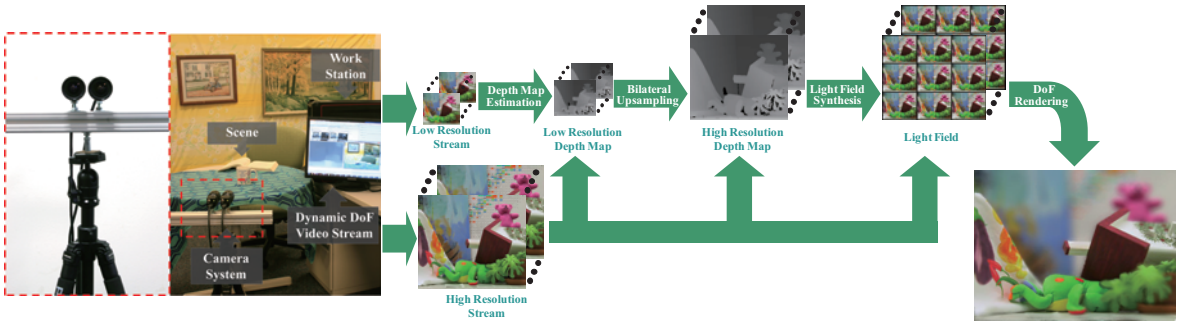


Figure 7.2: The imaging hardware and the processing pipeline of my dynamic DoF video acquisition system. All processing modules are implemented on NVIDIA's CUDA to achieve real-time performance.

of the Bumblebee forces its lenses to be small and it produces image with severe radial distortion. Our system is also less expensive ($1500 vs. $4000), and my setup allows me to dynamically adjust the camera baseline to best fit different types of scenes unlike the Bumblebee. We calibrate the stereo pair using a planar checker board pattern the algorithm outlined by Zhang [137]. It is not necessary, however, that the calibration be absolutely accurate as the disparity map is recovered from a severely downsampled image pair. Our experiments have shown that disparity map recovery using belief propagation on the low-resolution image pair is not affected by slight changes in the camera pair geometry. The intensity calibration on the camera pair is performed prior to capture via histogram equalization. The mappings for these processes are retained and applied to each incoming frame prior to stereo matching.

7.2 Real-time Stereo Matching

In order to efficiently generate a high-resolution disparity map from the input low-res/high-res image pairs, we implement a GPU-based stereo matching algorithm on CUDA.

7.2.1 CUDA Belief Propagation

Stereo matching is a long standing problem in computer vision [94]. Global methods based on belief propagation (BP) [106] and graph-cut [53, 20] have been known to produce highly reliable and accurate results. These methods, however, are more expensive when compared to local optimization methods such as dynamic programming. Fortunately, BP lends itself well to parallelism on the GPU [21, 43], where the core computations can be performed at every image pixel in parallel on the device.

We utilize the methods presented by Felzenwalb [38] to speed up my implementation without affecting the accuracy: We use a hierarchical implementation to decrease the number of iterations needed for message value convergence; We apply a checkerboard scheme to split the pixels when passing messages in order to reduce the number of necessary operations and halve the memory requirements; and we utilize a two-pass algorithm to reduce the running time to generate each message from $O(n^2)$ to $O(n)$ using the truncated linear model for data/smoothness costs.

Our CUDA BP implementation uses five separate kernels, whereas the CPU only calls the appropriate kernels and adjusts the current parameters/variables. A kernel is used to perform each of the following steps in parallel, with each thread mapping to computations at a distinct pixel:

1. Compute the data costs for each pixel at each possible disparity at the bottom level.
2. Iteratively compute the data cost for each pixel at each succeeding level by aggregating the appropriate data costs at the proceeding level.
3. For each level of the implementation:
   (a) Compute the message values at the current 'checkerboard' set of pixels and pass the values to the alternative set. Repeat for i iterations, alternating between the two sets.
   (b) If not at the bottom level, copy message values at each pixel to corresponding pixels of the succeeding level.
4. Compute the disparity estimate at each pixel using the data cost and current message values corresponding to each disparity.

Table 7.1 shows the performance of my algorithm on some of the Middlebury datasets at different resolutions. Despite the acceleration on the GPU, we find that it is necessary to use the lower resolution images (320×240 or lower) as inputs to my stereo algorithm in order to achieve real-time performance.

Data sets	Resolutions		
	128×96	320×240	640×480
Teddy	13ms	78 ms	446 ms
Tsukuba	8ms	55ms	357 ms
Cones	11ms	69 ms	424 ms

Table 7.1: Performance of my CUDA stereo matching at different resolutions. Note that the number of disparity levels is proportionally scaled to the resolution. The levels of belief propagation are all set to 5 and iterations per level are all set to 10.

In my experiments described in the rest of the chapter, we first smooth these low-resolution image pairs using a Gaussian filter where σ equals 1.0, then process them using my implementation with a disparity range from 0 to 35, maximum data cost and smoothness costs of 15.0 and 1.7, respectively, a data cost weight of 0.7 in relation to the smoothness cost, with 5 levels of belief propagation and 10 iterations per level. Each kernel is processed on the GPU using thread block dimensions of 32×4.

7.2.2 Fast Cross Bilateral Upsampling

Given a low-resolution disparity map D' and a high-resolution image I, we intend to recover a high-resolution disparity map D using cross bilateral filters [128], where we apply a spatial Gaussian filter to D' and a color-space Gaussian filter to I. Assuming p and q are two pixels in I; W is the filter window size; I_p and I_q are the color of p and q in I; and q' is the corresponding pixel coordinate of q in D'. We also

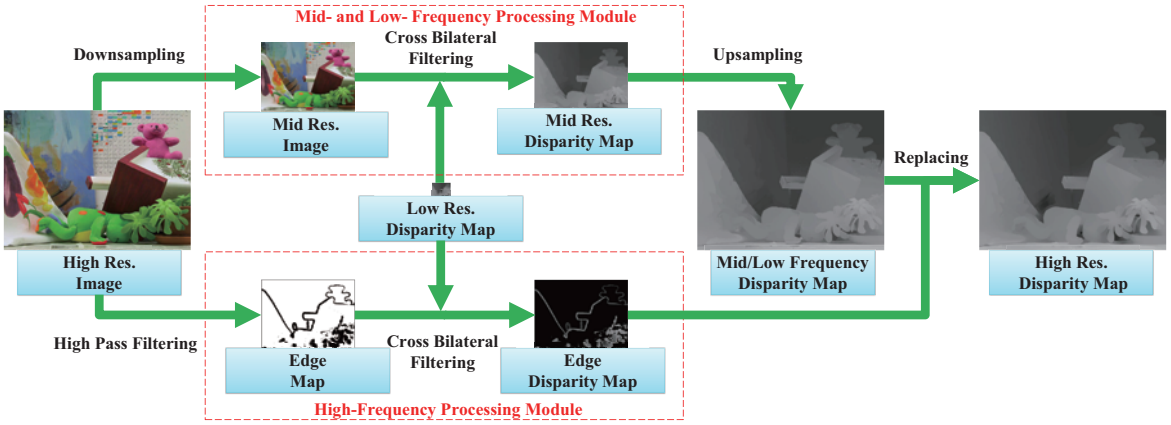


Figure 7.3: Our fast cross bilateral upsampling scheme synthesizes a high-resolution disparity map from the low-resolution BP stereo matching result on CUDA.

use σ_c and σ_d as constants to threshold the color difference and filter size. We compute the disparity of pixel D_p as:

$$D_p = \frac{\sum_{q \in W} G_d(p,q) G_c(p,q) D'_q}{K_p}, \tag{7.1}$$

where $K_p = \sum_{q \in W} G_d(p,q) G_c(p,q)$, $G_d(p,q) = exp(\frac{-||p-q||}{\sigma_d})$, and $G_c(p,q) = exp(\frac{-||I_p - I_q||}{\sigma_c})$.

The complexity of cross bilateral upsampling (CBU) is $O(NW)$ where N is the output image size and W is the filter window size. Therefore the dominating factor to the processing time is the number of pixels that need to be upsampled, i.e., the resolution of the high-res image in the brute-force implementation.

To accelerate my algorithm, we implement a fast CBU scheme that effectively reduces the pixels to be upsampled. Paris *et al.* [79] have shown that the mid and low frequency components of an image remain approximately the same when downsampled. We therefore treat the high-frequency and the mid- and low- frequency components separately. Our method first applies a Gaussian high-pass filter to identify the pixels of high frequency in I and then uses a standard cross bilateral filter to estimate the disparity values at only these pixels. We store the resulting disparity map as D_{high}. We call this step the high-frequency processing module. In parallel, we downsample

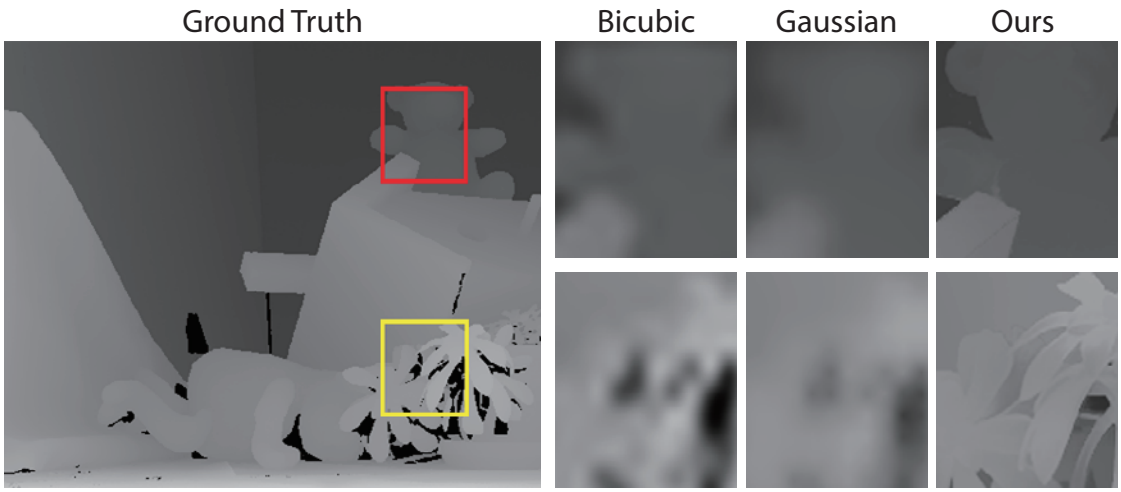
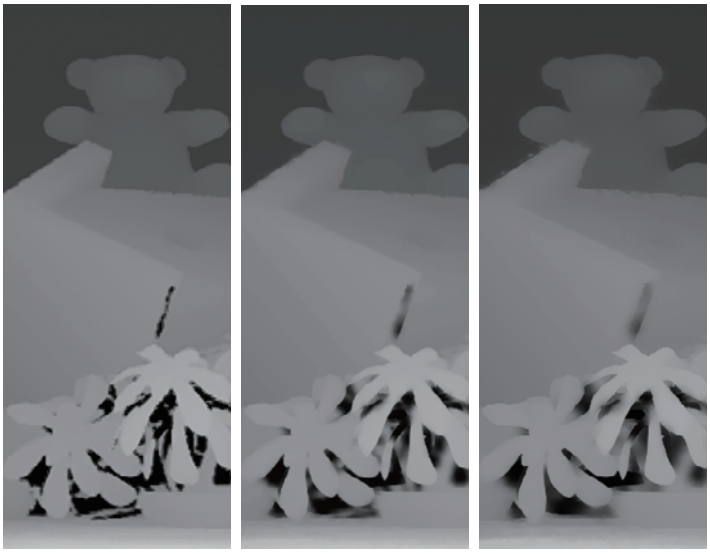


Figure 7.4: Comparison of my method and other upsampling schemes on synthesize data. Both patches in the disparity map are upsampled from a resolution of 30×25 to 450×375.

the color image to mid-resolution I_{mid}, apply CBU between D' and I_{mid} to obtain the mid-res disparity map D_{mid}; and subsequently upsample D_{mid} to D_{high} using standard bilinear upsampling. We call this step the mid- and low- frequency processing module. Finally, we perform high frequency compensation by *replacing* the disparity value at the identified high frequency pixels $\tilde{I}$ with D_{high}. Figure 7.3 shows the complete processing pipeline of my algorithm. Compared with standard CBU, my scheme only needs to upsample a small portion of the pixels and hence is much faster.

We also added a refining stage for sharpening the boundary regions and smoothing the surface regions with a cross bilateral filter, after the unsampling is done. The stage is basically the same as the upsamling stage except the input disparity map is the same size as the color image. Since the output disparity map could be treated as the input of another stage, this refining stage can be performed iteratively.

As shown in Figure 7.5, if the refining stage is not performed, the edges and surfaces of the disparity map looks noisy due to the imperfection of the low resolution disparity map and the textures in the color image. However, if the refining stage contains too many iterations, then the disparities of one side of edges starts to bleed



(a) (b) (c)

Figure 7.5: Comparison of three results using different number of refining iterations. Result (a), (b), (c) are using 0, 3, and 10 iterations respectively.

into the other side, which is the effect of over-smoothing. Therefore, a compromise number of iterations must be chosen at run time, using my interactive parameter interface (Section 8.6).

Note that the upsampled depth edges may not be consistent with the depth edges compute using the high frequency map. Here we experimented with the following solutions: 1) Use the unsampled depth edges. 2) Use the high frequency depth edges. 3) Blend the two results. We found out that we can preserve more accurate edges and render better results using the second way.

7.2.3 CUDA Implementation.

We developed a GPU implementation of my algorithm on the CUDA architecture to tightly integrate with my CUDA BP stereo mapping algorithm. In my experiments, we found that it is most efficient to assign one thread to upsample each pixel in the disparity map. To further evaluate the throughput of my implementation,

Figure 7.6: Comparison between my method and bicubic upsampling on real scenes. The disparity map is upsampled from 320×240 to 640×480. Our method preserves sharp edges and maintains smoothness, which is critical to reliable DoF synthesis.

we upsampled 128×128 disparity maps with 1280×1280 color images. Our implementation achieves a processing speed of 22 ms per frame or 14 ms per megapixel with a 5×5 filter window, a significant speedup to the CPU-based scheme [55] (which was 2 seconds per megapixel). To measure the accuracy of my scheme, we performed experiments using various stereo data sets. In Figure 7.7, we show using the Teddy data set that reintroducing high frequency compensation produces sharper edges and smoother surfaces. Figure 7.4 illustrates my results in three regions on the Teddy data set. They are upsampled from 30×25 to 450×375. Compared with standard bicubic or Gaussian upsampling, my method preserves fine details near the edges. It is important to note that preserving edges while removing noise in the disparity map is crucial to my DoF synthesis as DoF effects are most apparent near the occlusion boundaries. Figure 7.6 gives the results on an indoor scene using bicubic upsampling and my method. To further measure the accuracy, we compared my estimation with the ground truth by computing the mean squared errors over all pixels. Table 7.2 compares the error incurred by my method under different upsampling scales on a variety of Middlebury stereo data sets, and the results show that my method is reliable and accurate even with very high upsampling scales.

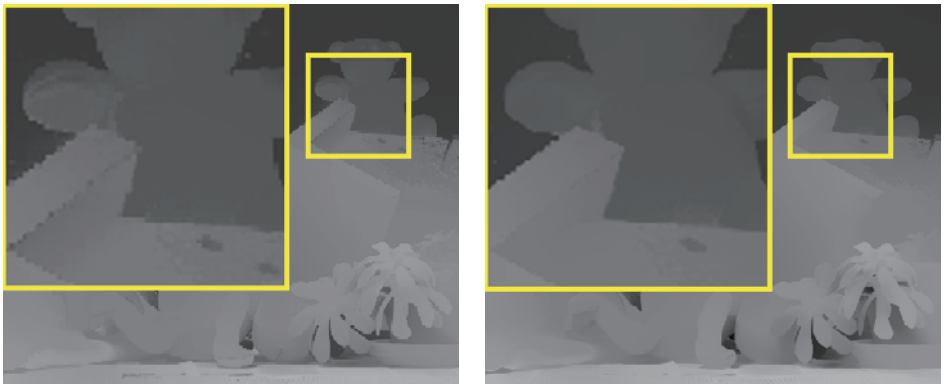


Figure 7.7: Comparison of the result with(right) and without(left) high frequency compensation.

In my indoor and outdoor experiments, good results of disparity maps can be achieved when there are 10 iterations in refining stage. Since the system runs at interactive speed, it is impossible to use standard CBU to upsample the low resolution disparity because it would take 0.1 second (10 frame per second) to compute a single frame of resolution 640×480 with CUDA implementation. However with my fast CBU framework, the speed quickly goes up to 40 frame per second with the downsampling factor 2×2.

7.3 Real Time DoF Synthesis

Once we obtain the high-resolution disparity map, we set out to synthesize dynamic DoF effects. Previous single image based DoF synthesis algorithms attempt to estimate the circle of confusion at every pixel and then apply the spatially varying blurs on the image. These methods produce strong bleeding artifacts at the occlusion boundaries, as shown in Figure 7.8. In computer graphics, the distributed ray tracing and the accumulation buffer techniques have long served as the rendering method for dynamic DoF. Both approaches are computationally expensive as they either require tracing out a large number of rays or repeated rasterization of the scene. Furthermore, to apply ray-tracing or accumulation buffer in my application requires constructing

Data sets	Upsampling Scales			
	20×20	10×10	5×5	2×2
Teddy	10.41%	3.56%	1.71%	0.52%
Plastic	8.36%	4.23%	2.05%	0.91%
Monopoly	11.76%	5.35%	2.96%	1.14%
Books	9.28%	6.12%	2.63%	1.02%
Baby2	5.76%	2.38%	1.61%	0.69%
Aloe	15.12%	7.83%	3.40%	1.17%
Cones	11.51%	5.87%	3.25%	1.28%
Art	13.47%	7.15%	3.43%	1.41%

Table 7.2: Pixels with disparity error larger than 1 under different upsampling factors on the Middlebury data sets.

a triangulation of the scene from the depth map, which would incur additional computational cost. In this paper, we adopt a similar approach to [132] by dynamically generating a light field from the high-resolution video stream and its depth stream, as shown in Figure 7.9. Our technique, however, differs in that we directly use the disparity map for warping and filtering whereas [132] builds upon the depth map. As follows, we briefly reiterate the main steps of this light-field based DoF rendering technique.

7.3.1 The Lens Light Field

The light field is a well known image based rendering technique. It uses a set of rays commonly stored in a 2D array of images to represent a scene. Each ray in the light field can be indexed by an integer 4-tuple (s, t, u, v), where (s, t) is the image index and (u, v) is the pixel index within a image.

Our first step generates a light field from the stereo pair. The high resolution camera in my stereo pair is used as the reference camera R_{00}.

To synthesize the light field, we use the high-resolution camera in my stereo pair as the reference camera R_{00} (i.e., $(s, t) = (0, 0)$). We can then easily find all rays that pass through a 3D point A in terms of its disparity γ from the reference view.

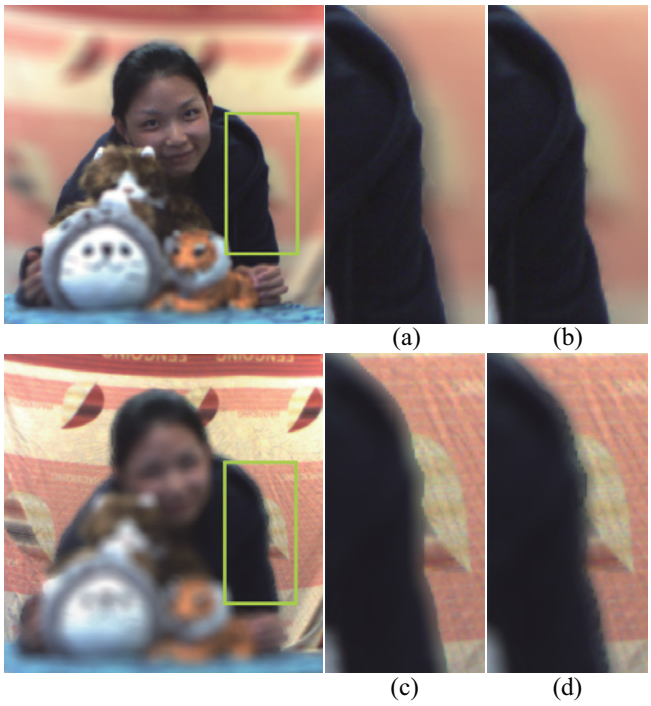


Figure 7.8: Comparing results generated by image space blurring (a, c) and my light field synthesis method (b, d). Our approach effectively reduces both the intensity leakage (a) and boundary discontinuity (c) artifacts.

Assuming A's image is at pixel (u_0, v_0) in the reference camera, we can compute its image (pixel coordinate) in any light field camera R_{st} as:

$$(u, v) = (u_0, v_0) + (s, t) \cdot \gamma \tag{7.2}$$

We use $L_{out}(s, t, u, v)$ to represent the out-of-lens light field and $L_{in}(x, y, s, t)$ to represent the in-camera light field. The image formed by a thin lens is proportional to the irradiance at a pixel a [104], which can be computed as a weighted integral of the incoming radiance through the lens:

$$a(x, y) \approx \sum_{(s,t)} L_{in}(x, y, s, t) cos^4\phi \tag{7.3}$$

To map the in-lens light field to the out-of-lens light field, it is easy to verify that pixel $a(x, y)$ on the sensor maps to pixel $(u_0, v_0) = (w - x, h - y)$ in R_{00}. Therefore,

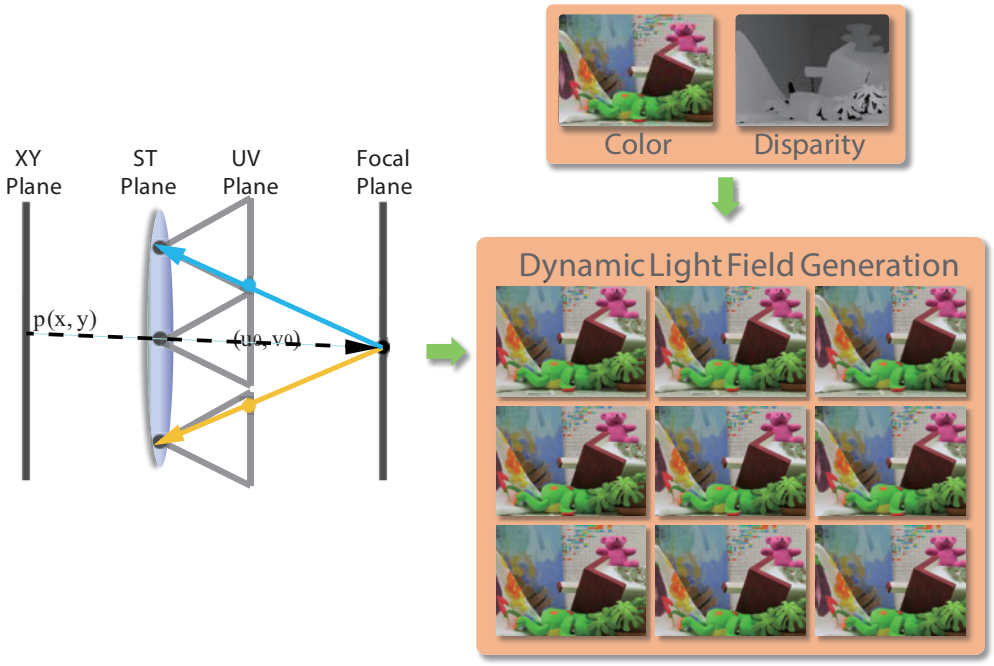


Figure 7.9: We synthesize an in-lens light field (left) from the recovered high-resolution color image and disparity map (right).

if we want to focus at the scene depth whose corresponding disparity is γ_f, we can find the pixel index in camera R_{st} using Equation 7.2. The irradiance at a can be approximated as:

$$a(x,y) = \sum_{(s,t)} L_{out}(s,t,u_0 + s \cdot \gamma_f, v_0 + t \cdot \gamma_f) \cdot cos^4\phi$$

To estimate the attenuation $\cos^4\phi$ term, we can directly compute $\cos^4\phi$ for each ray (s,t,u,v). Notice that the ray has direction $(s,t,1)$. Therefore, we can compute $\cos^4\phi = \frac{1}{(s^2+t^2+1)^2}$.

7.3.2 CUDA Implementation

To synthesize the light field from the reference camera R_{00} and its disparity map, we warp it onto the rest light field cameras using Equation 7.2. Note that inverse

warping is impractical here because the disparity maps of target light field cameras are unknown. Therefore we choose to forwardly constructing those cameras.

A naive approach would be to directly warp the RGB color of each pixel $a(u_0, v_0)$ in R_{00} onto other light field cameras. Specifically, using a's disparity value, we can directly compute its target pixel coordinate in camera R_{st} using Equation 7.2. Since the CUDA architecture supports parallel write, we can simultaneously warp all pixels in R_{00} onto other light field cameras.

Although the warping process is straight forward, attention needs to be paid to the correctness of the light field. Since multiple pixels in R_{00} may warp to the same pixel a in the light field camera R_{st}, a depth comparison is necessary to ensure the correct visibility. Thus each light field camera requires an additional depth buffer. To avoid write-write conflicts in the warping process, we use atomic operations. However, current graphics hardware cannot handle atomic operations on both color and depth values at the same time. To resolve this issue, we only choose to warp the disparity value. We can easily index the RGB value for each light field ray using the stored disparity value and the camera parameters. This solution requires less video memory as the RGB value does not need to be stored in the light field.

Due to speed requirements, we can only render a small light field with 36 to 48 cameras at a 640×480 image resolution. The low spatial resolution leads to strong aliasing artifacts due to undersampling. Since my reference view does not contain information from the occluded regions, the warped light field camera images will contain holes.

To reduce the image artifacts caused by undersampling and occlusions, we develop a simple technique similar to the cone tracing method to pre-filter the reference view [60]. Our method is based on the observation that out-of-focus regions exhibit most severe aliasing artifacts and occlusion artifacts since they blend rays corresponding to different 3D points.

Our method compensates for undersampling by first blurring the out-of-focus rays and then blending them. A similar concept has been used in the Fourier slicing

photography technique for generating a band-limited light field [77].

To simulate low-pass filtering in light field rendering, we first generate a Mipmap from the reference image using a 3×3 Gaussian kernel [61].

Gaussian Mipmaps eliminate the ringing artifacts and produce smoother filtering results than the regular box-filters . We then integrate the Gaussian Mipmap into the light field ray querying process.

Assume the scene is focused at depth d_f. For a ray (u, v) in camera R_{st} that has depth value d_r, we have a similitude relationship:

$$C_{lens}/C_{blurdisk} = d_f/(d_r - d_f) = (\gamma_r - \gamma_f)/\gamma_f \tag{7.4}$$

where C_{lens} is the diameter of the aperture and $C_{blurdisk}$ is the size of the blur disk in world space. The MipMap level for the ray can be calculates as:

$$\begin{aligned} l &= log_2(\gamma_r \cdot C_{blurdisk}/N) \\ &= log_2(\gamma_r \cdot (C_{lens} \cdot \gamma_f/(\gamma_r L - \gamma_f))/N) \\ &= log_2(C_{lens} \cdot (\gamma_f - \gamma_r)/(B \cdot N)) \end{aligned} \tag{7.5}$$

where N is the number of samples, γ_r gives the pixel per length ratio which transform the size of the ray cone $C_{blurdisk}/N$ into number of pixels on the image.

7.3.3 Our Technique vs. Single-Image Blurring

Compared with single-image methods that apply spatially varying blurs, my light field based DoF synthesis technique significantly reduces two types of boundary artifacts. In instances where the camera focuses at the foreground, the ground truth result should blend points on the background. Conversely, single-image filtering techniques use a large kernel to blend the foreground and background pixels and hence, produce the *intensity leakage* artifact. Consider a point A_b lying on the background near the boundary, as shown in Figure 7.10. Our method attempts to blend rays originating from the background. Although my technique can only access a portion of them Due to occlusions, it still produces reasonable approximations.

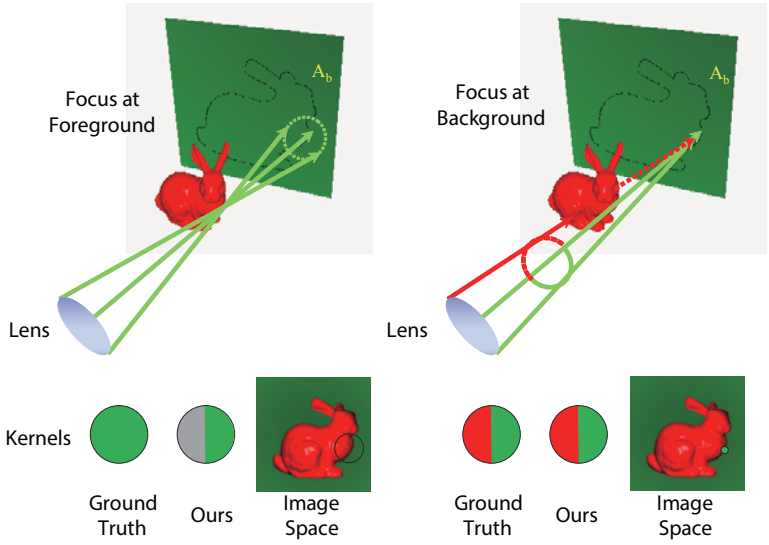


Figure 7.10: Illustrations of two types of boundary artifacts. See Section 7.3.3 for details.

In instances where the camera focuses on the background, the ground truth result should blend both the foreground and background points. Single-image filtering techniques, however, would consider A_b in focus and hence directly use its color as the pixel's color. In this case, the transition from the foreground to the background appears abrupt, causing the *boundary discontinuity* artifacts. Consider a point A_b on the background near the occlusion boundary in the image as shown in Figure 7.10. Since rays originating from both the foreground and background are captured by the synthesized light field, my technique will produce the correct result.

Figure 7.8 compares the rendering results using my method and the single-image filtering approach on an indoor scene. Our technique exhibits fewer visual artifacts compared to the single-image filtering method, especially near the boundary of the girl. When examining the boundary of the sweater, the single-image method blurs the black sweater regions into the background and thus causes color bleeding, whereas my technique prevents such leakage. When focusing at the background, the single-image method exhibits discontinuous transitions from the girl to the background while my method preserves the smooth transition.

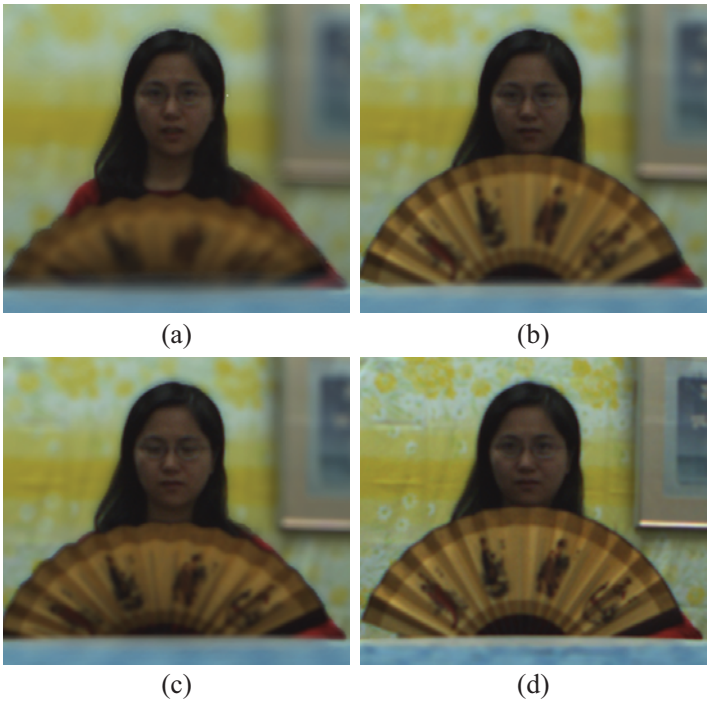


Figure 7.11: Results of synthesizing changing aperture sizes. The aperture size gradually decreases from (a) to (d).

Our method also correctly preserves the boundaries between the in-focus and out-of-focus regions when synthesizing changing aperture sizes. As shown in Figure 7.11, we fix the focus at the woman. With the aperture fully open in (a), the blur level decreases as we decrease the aperture size.

7.4 Applications: Real-time Tracking and Racking Focus

In cinematography, a rack focus is the practice of changing the focus of the lens during a shot. The term can refer to small or large changes of focus. If the focus is shallow, then the technique becomes more noticeable. In professional films, a camera assistant called a focus puller is responsible for rack focusing. The director Richard Rush developed the technique in his documentary film "The Sinister Saga of Making The Stunt Man" in the 1960s. Our system provides real-time dynamic DoF effects which is comparable to racking focus of the movie camera.

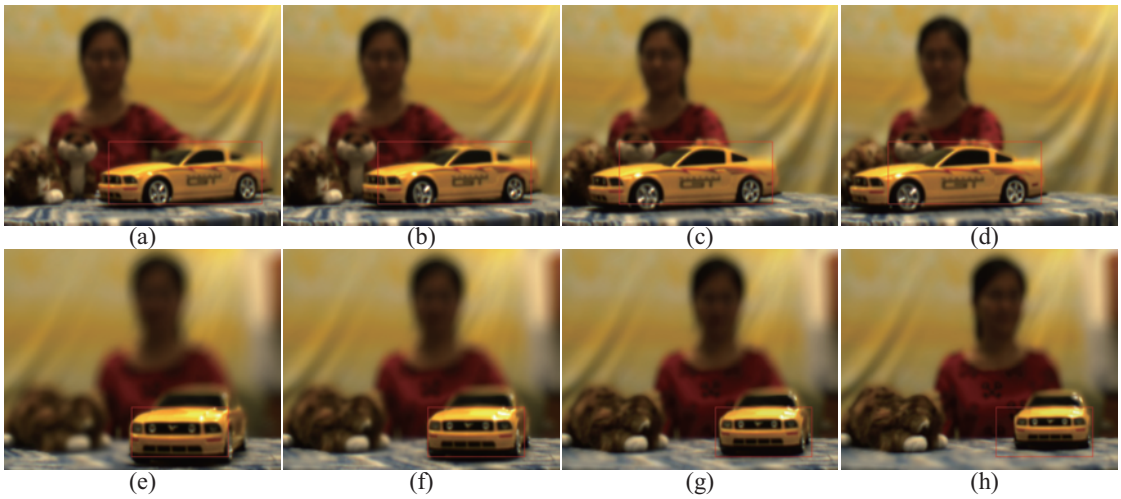


Figure 7.12: Results using my tracking algorithm. Notice that with the auto-refocusing functionality, the cat on the right hand side of the girl is becoming sharper as the toy car moves closer to its plane.

A tracking focus is the practise of keeping the object of interest in focus by tracking its 3D location. Tracking focus aims to resolve the challenging task of focusing exactly on moving objects while shooting a dynamic scene. Since the resolution of movie camera's viewfinder is relatively small, it is hard to tell whether the object of interest is sharp or blurry until the postprocessing stage. Our system couples the tracking on the color image with the tracking on depth map to location the object in 3D to dynamically change focus with the moving object.

7.4.1 Tracking

Like all the other classic tracking algorithms, we model my problem by reasoning probabilistically about the world based on the Bayesian rule. Since we have two images as the input, the posterior probability can be represented as

$$p(W|I_1, I_2) = \frac{p(I_1, I_2|W)p(W)}{p(I_1, I_2)},$$

where W is the latent scene, $p(I_1, I_2)$ is treated as normalizing constant, I_1 and I_2 are the images seen, and

$$p(I_1, I_2|W) = p(I_1|I_2, W)p(I_2|W)p(W),$$

Here we use the *maximum a posteriori* estimate to find the result. Since the underline scene W does not change during one shot, $p(I_1|I_2, W)$ could be interpreted as the warping result from one of the images using the disparity map. Therefore, instead of dealing with multiple images, we use both images and a disparity map as inputs. The result is estimated by $argmax_W p(W|I_1, I_2)$.

We use Sum of Squared Differences (SSD) as the error function in my calculation. The estimated location of object in frame i is computed using the following algorithm:

Algorithm 2 Compute current tracking postion

```
if i = 0 then
  pos[i] ← pos[i]
else
  MinError ← INFINITE
  while p ← nextposition do
    n ← 0
    for j = i → max(i − MaxLength, 0) do
      e ← e + DisparitySSD(p, pos[i]) × ColorSSD(p, pos[i])
      n ← n + 1
    end for
    e ← e/n
    if e < MinError then
      MinError ← e
      pos[i] ← p
    end if
  end while
end if
```

With the additional disparity information, the tracking result becomes very stable even though the object of interest and the background have similar colors, as shown in Row 1 of Figure 7.12. The search of tracking position is also parallelized with CUDA, so the computation overhead of this step is negligible.

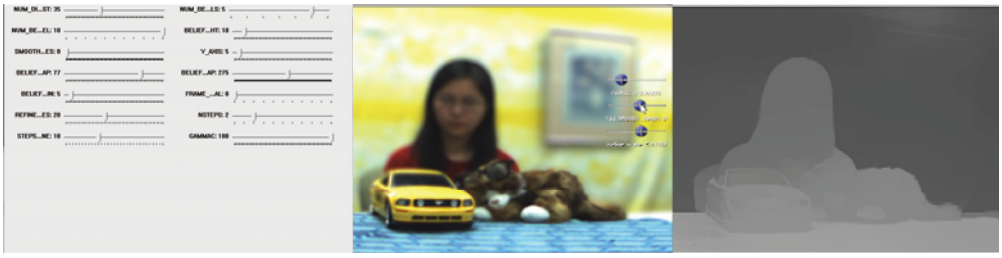


Figure 7.13: The real time DoF effects (middle) and disparity map (right) given by my system after fine tuning the parameters using my interface (left).

7.4.2 Auto-Refocusing

With the object of interest being estimated on certain frame i, we assume that all pixels p_{ij} around pixel j inside the object should have the same disparity. A straight forward approach will be calculating the focusing disparity value $Disp$ by averaging all disparity values in this region. The result, however, is subject to noise and not robust to pixels which are incorrectly marked as the object. To overcome these problems, we first assign different weights for pixels. Therefore, $Disp$ is computed by

$$Disp = \frac{\sum_j D_{ij} w_{ij}}{\sum_j w_{ij}},$$

where D_{ij} is the disparity of pixel p_{ij} on frame i and w_{ij} is the weight for pixel p_{ij}. The straight forward approach is assigning constant weights for all pixels. Note that user is defining the object of interest by a rectangle, pixels with different disparities or even occlusions may appear on the boundary when shooting the video. To make my disparity computation robust, here we use Gaussian weight. For each pixel in the object, we keep track of the previous assigned disparity. Since my system runs at interactive speed, we can safely assume that large disparity jumps do not occur on any pixel. If the difference between the previous and current disparities is larger than a certain threshold, we claim that this pixel is noisy and do not use it in the computation of current focused disparity $Disp$. Row 2 of Figure 7.12 shows results of my auto-refocusing algorithm. The moving car stays in focus while the out of focus regions are getting sharper, such as the girl

Figure 7.14: Screen captures of live video streams produced by my system on both indoor (top two rows) and outdoor (bottom row) scenes.

and the cat, or more blurry, such as the tablecloth in the front, as the car moves closer or further away to their planes, respectively.

7.5 Results and Discussions

Our hybrid-resolution stereo system is connected to a work station through a single PCI-E Firewire card. The workstation is equipped with a 3.2GHz Intel Core i7 970 CPU, 4GB memory and an NVIDIA Geforce GTX 480 Graphic Card with 1.5GB memory. We implement all three processing modules (the disparity map estimation, fast CBU, dynamic DoF rendering) using NVIDIA's CUDA 3.1 with compute capability 2.0. Our system runs at the resolution of 640×480 with 15 fps. Compared with a equivalent CPU implementation at 0.2 fps, the overall speed up is over $\times 30$. Table 7.3 gives detailed speed up of each component in my system.

A crucial step in my real-time stereo matching module is choosing the proper parameters (e.g., the weight for the smooth/compatible terms of the energy function)

to fit different types of scenes (indoor vs. outdoor). We have developed an interface to dynamically change the parameters, As shown in Figure 7.13.

We have conducted extensive experiments on both indoor and outdoor scenes. We first demonstrate my system on indoor scenes with controlled lighting. Figure 7.14 row 1 shows four frames captured by my system of a girl drinking coffee while reading. The coffee cup in the scene is textureless and very smooth. Our fast CBU scheme, however, still preserves the disparity edges, as shown in Figure 7.14 row 1. We then dynamically change the depth of the focal plane: column (a) and column (d) focus on the front of the table, column (b) focuses on the girl, and column (c) focuses on the background. Notice how the blur varies and the in-focus regions fade into the out-of-focus regions.

Figure 7.14 row 2 displays a scene of a girl moving a toy car on a table. The surface of the car is specular, and the background and car have similar colors, making it challenging to prevent the disparity of the background from merging with the disparity of the car. Moreover, the motion of the car is towards the camera, causing the body of the car to have several different disparities. This makes labeling each pixel using stereo correspondence methods even more difficult. Nevertheless, my algorithm preserves the edges of the car when it is in focus and correctly blurs portions of the scene outside of the focal plane. Our system performs well indoors because the background distance is often limited, therefore allowing one baseline to produce accurate disparity labels for the entire scene. In addition, artificially lit indoor scenes with diffuse walls and

Component	CPU	GPU
Depth Estimation	200 ms	30 - 50 ms
Bilateral Upsampling	100 ms	5 ms
Light Field Rendering	200 ms	15 ms

Table 7.3: Speed up of each component in the system.

surfaces tend to have moderate dynamic range and have few poorly lit or saturated regions.

Indoor scenes undoubtedly aid the performance of my system. Our experiments on outdoor scenes, however, show promising results as well. Row 3 of Figure 7.14 shows an outdoor sequence with a distant background under dynamically varying lighting conditions. Notice that in column (a), the image is brighter than the rest of the frames in the sequence and the background contains noticeable shadows. In addition to incoherent illumination, large portions of the scene such as the sky and the ground are textureless, making it difficult to achieve robust stereo matching. Since my system allows us to dynamically change the camera baseline, we use its real-time feedback to tune the parameters and increase the camera baseline to obtain a satisfactory disparity map, as shown in the supplementary video. The use of large baseline may lead to holes near the occlusion boundaries on full-resolution images. These holes are, however, less significant in low-resolution stereo pairs and my upsampling scheme is able to borrow information from the color image to fill in the holes. The extracted frames show that we am able to correctly change the focus between the moving targets in both foreground and background.

7.6 Discussions and Future Work

We have presented an affordable stereo solution for producing high quality live DoF effects. Our system shows promising results on indoor and outdoor scenes although it still has several limitations. First, 15 fps is a low frame rate and our resolution of 640×480 precludes our system from immediately being used in high quality HD video applications. Using multiple GPUs may address this problem as they allow greater exploitation of inherent parallelism in our computational pipeline. Second, although the high quality sensor and lens system on our camera pair significantly reduces image noise and optical distortions, this comes with a higher price. While less expensive than existing commercial movie cameras, our system is still twice the cost of most base level video cameras. Integrating existing real-time techniques to correct optical distortions

Figure 7.15: Observed artifacts (high lighted with red rectangle) at specular regions on a computed disparity map.

and sensor noise into our pipeline would make it feasible to use lower cost webcams instead of the Firewire Flea cameras.

Since our approach is stereo-based, it suffers from the same problem in classical stereo matching. It is well known that regions without any texture are difficult to handle. Results using our Belief Propagation scheme exhibit visual artifacts in such regions, especially for outdoor scenes which contain the sky in the background. Recall that our bilateral upsampling algorithm assumes that adjacent boundary regions of different depth should have different colors to avoid color bleeding. In practice, when a large filter kernel is used on on adjacent regions with difficult but still similar colors, our results still exhibit the bleeding artifacts. Occlusions boundaries of objects are also problematic since it is extremely challenging to synthesize the occluded regions when warping the central view to other views in the light field. Specular highlights, due to view dependency, can also introduce artifacts, as shown in Figure 7.15. Finally, translucent objects such as a person's hair or face boundary can exhibit visual artifacts, as shown in Figure 7.16. A potential solution is to apply image matting to first extract the region and then separately synthesize the foreground and background.

Our other future efforts include adapting our system to functional applications such as privacy protected surveillance. We plan to demonstrate the usefulness of our system in urban spaces to limit the focal plane to public areas, e.g., the sidewalks, while

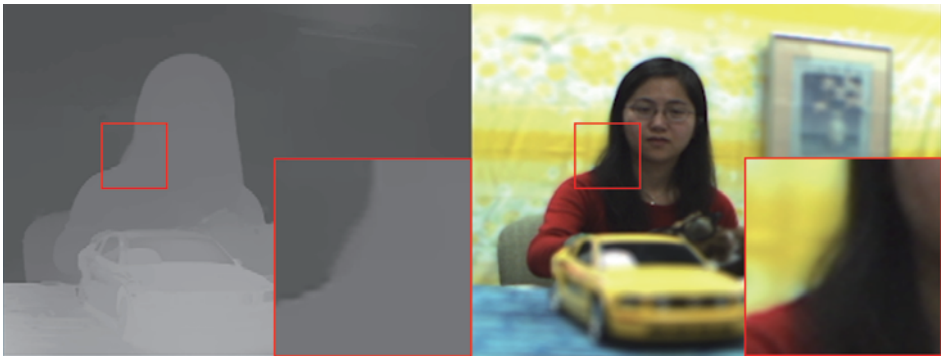


Figure 7.16: Observed artifacts at translucent regions.

blurring more distant private areas like the interior of homes. Current urban surveillance networks are augmented with real-time recognition algorithms to detect illegal activity. When illegal activity is detected, our system could provide more information to law enforcement by removing the DoF effect using the stored disparity map stream for subsequent scene reconstruction. We can also Since our approach is leverage future gains in ubiquitous computing to produce a truly mobile platform which utilizes, for example, two camera phones for producing DSLR quality imagery. On the algorithm side, instead of performing a straight forward high-pass Gaussian filter to acquire high frequency information in the image, we can also perform other sophisticated frequency decomposition method such as wavelet transform.

Chapter 8

STEREO BASED LIGHT FIELD CAMERA : MIRROR BASED APPROACH

In this Chapter, I present an alternative light field imaging solution using a catadioptric mirror array and discuss its unique advantage on low light imaging.

8.1 Catadioptric Light Field Camera

A classical catadioptric camera positions a regular pinhole camera in front of a curved mirror to capture images with a much wider field-of-view. Recent developments further replace the single mirror with a mirror array [34, 108] so that a single photograph will contain multiple views towards the scene. This new catadioptric array photography (CAP) technique is capable of recovering scene depth from a single image [34, 50] and has shown promising results in special photographic effects such as dynamic Depth-of-Field [108]. In this paper, we present a new technique that uses CAP for high quality imaging under low light.

Imaging under dark illuminations has been challenging since it is difficult to simultaneously preserve intrinsic lighting and maintaining low noise. The simplest approach of increasing the exposure through aperture, shutter or flash produces undesirable visual artifacts: wide apertures lead to shallow depth-of-field, long shutters cause motion blurs [78], and the use of flash corrupts intrinsic lighting and generates sharp shadows [36]. The seminal flash/no-flash photography [8, 82] captures one flash image and one non-flash image at the same viewpoint. We show that CAP can produce similar results without using the auxiliary lighting(Fig. 8.2).

Capturing the scene from multiple viewpoints in a single shot has a number of advantages. Each view is slightly different from the other therefore providing depth

cues for correspondence matching across views. At the same time, the similarity across the view affords noise reduction [136]. In addition, all views are captured in a single shot and therefore naturally resolves synchronization and motion blurs. An apparent shortcoming, however, is the sacrifice of resolution: the effective resolution will be reduced to 1/N of the full camera resolution where N corresponds to the number of mirrors. For static scenes, we address this issue by using a multi-resolution fusion approach: the users can further zoom in the camera to capture only the central mirror at a much higher resolution and then apply our depth-guide image denoising/superresolution. The recovered scene geometry can also be used for other imaging applications.

8.2 Related Work in Low-Light Photography

Photography under poor lighting conditions is a long standing problem and has attracted a lot of attention from image processing, computer vision, and computational photography. We categorize existing solutions into two categories: pure image processing techniques and computational photography/imaging techniques.

8.2.1 Image Processing

The image denoising literature is huge and we refer the readers to the survey by Buades et al. [22]. We here only review the most relevant ones.

8.2.1.0.1 Single-image Denoising

These approaches aim to reduce image noise using an image captured from a single viewpoint. The wavelet-based method proposed by Portilla et al. [85] had been the state-of-the-art until the more recent non-local patch-based algorithms such as BM3D [23, 27, 134]. Built on the concept of non-local means [27], BM3D exploits self-similarity within an image and applies the optimal Wiener filtering for image denoising. However, these algorithms can break down when the image does not exhibit such self-similar patterns.

8.2.1.0.2 Multi-image Denoising

To resolve the lack of similarity issue, a number of approaches attempt to leverage multi-view acquisitions. Vaish et al. [117] explored the problem of using a camera array for denoising by exploiting redundancies from multiple viewpoints. Heo et al. [46] combined stereo matching with non-local denoising on each view of a stereo pair. Zhang et al. [136] extended the two-view stereo approach to N-views by selecting patches from multiple cameras with depth constraints and approximating the noiseless patch using Principal Component Analysis or Tensor Analysis. While producing promising results, their method is not directly applicable to dynamic scenes. Bennett and McMillan [15] demonstrated a per-pixel exposure model to enhance underexposed visible spectrum video through spatial temporal bilateral filtering. Their model assumes the difference between consecutive frames is small whereas we allow strong disparity between the mirror views.

8.2.2 Computational Photography

We develop a computational photography solution that acquires multiple views in a single shot and then denoise the images. Over the past decade, a number computational camera systems have been presented to tackle the problems of image denoising or image enhancement. A comprehensive survey can be found at [110].

8.2.2.0.3 Active Illumination

Flash based photography [36, 82, 8, 56] uses images captured under flash to enhance the ones without flash using spatial filters or optimization schemes. The seminal work by Eisemann and Durand [36] and Petschnigg et al. [82] used the non-flash image to preserve the ambiance of the original lighting while inserting sharpness from the flash image. Krishnan and Fergus [56] replaced the regular flash with the UV spectrum flash to reduce the bursts of light. They also exploited using the correlations between images captured at different wavelengths for robust denoising. We, in contrast, aim to completely eliminate flash lights during acquisition.

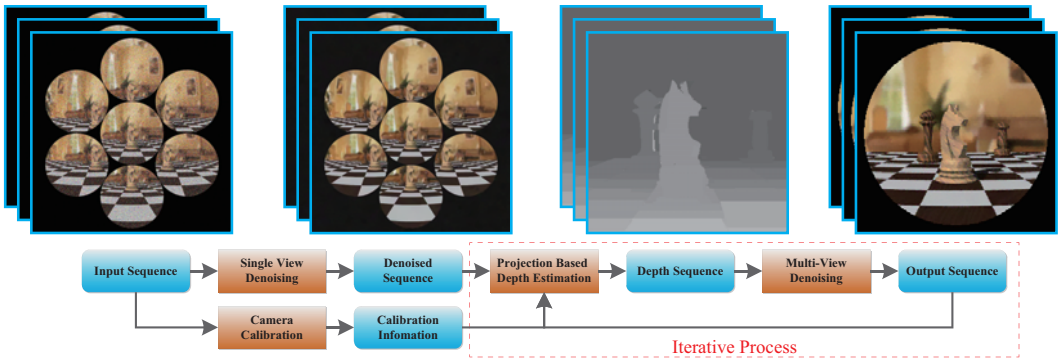


Figure 8.1: The processing pipeline of our CAP-based low light imaging technique. We adopt iterative processing: the denoised results are used to improve correspondence matching and vice versa.

8.2.2.0.4 Catadioptric Mirror Array

Our solution is inspired by the recent catadioptric mirror array for omni-directional computer vision [32, 7, 59]. In these systems, a pinhole camera is positioned in front of an array of curved mirrors to acquire the scene from multiple viewpoints. Each mirror view corresponds to a multi-perspective camera, which adds challenges to correspondence matching across views. To address the issue, Ding et al. [34] developed a mirror surface decomposition scheme by approximating the mirror camera as piecewise General Linear Cameras, a class of primitive multi-perspective cameras that have closed-form projections. Taguchi et al. [108] presented a novel Axial-Cone decomposition scheme that is particularly effective for rotationally symmetric mirrors. Their decomposition models each mirror camera as a set of pinhole cameras whose center-of-projection (CoP) lies on the rotational axis. Agrawal et al. [9] derived the analytical forward projection for axial non-central dioptric and catadioptric cameras. They proved that a closed form solution can be found by solving a 4th degree equation for forward projecting in case of spherical mirrors. By far, the applications of catadioptric systems have been focused on 3D scene reconstruction and image-based modeling and rendering. We investigate using it for low-light imaging.

8.3 Catadioptric Array Photography

8.3.1 System and Algorithm Overview

We first construct a single shot catadioptric camera system. As shown in Fig. 8.2, our prototype uses an array of 7 identical convex mirrors coated with enhanced Aluminum. Each mirror is a spherical cap of radius 51.68mm and height 6.45mm to avoid self-reflection when tightly packed together. We employ a Canon T2i digital camera with focal length 75mm and 300mm to capture the image of the mirror array and the central mirror respectively at the resolution of 3400×3500 (70% of the sensor resolution). We can also use different mirror setups to adapt other sensor types.

Compared with the multi-view acquisition systems, our configurations have many advantages. First, we only need to use a commodity camera without any modification. The captured multi-view multi-perspective (MVMP) image is also self-contained for processing, i.e., each denoised frame can be generated from a single shot. Second, the system is easy to setup. One only needs to set up the mirror array in front of the scene and orient the camera to capture the desirable views. Finally, our system is inexpensive to build and does not require complex setups. We calibrate camera and the mirror using a checkerboard during acquisition.

Fig. 8.1 shows our processing pipeline. We start with performing single-image based denoising on each captured MVMP image and camera calibration in parallel. Next, we model the ray geometry of the scene and find the voxel-pixel correspondence with forward projection. We then conduct dense stereo matching via graph cuts. Finally, we denoise each patch by using its corresponding patches across mirrors. We can iterate the steps to refine the results.

8.3.2 Stereo Matching

In order to estimate correspondences across different views, we adopt the space carving approach. Most existing algorithms in this category can be considered variations of the foundational framework by [57], in which a set of N perspective input images are used to determine 3D volumetric scene geometry. A crucial step is to

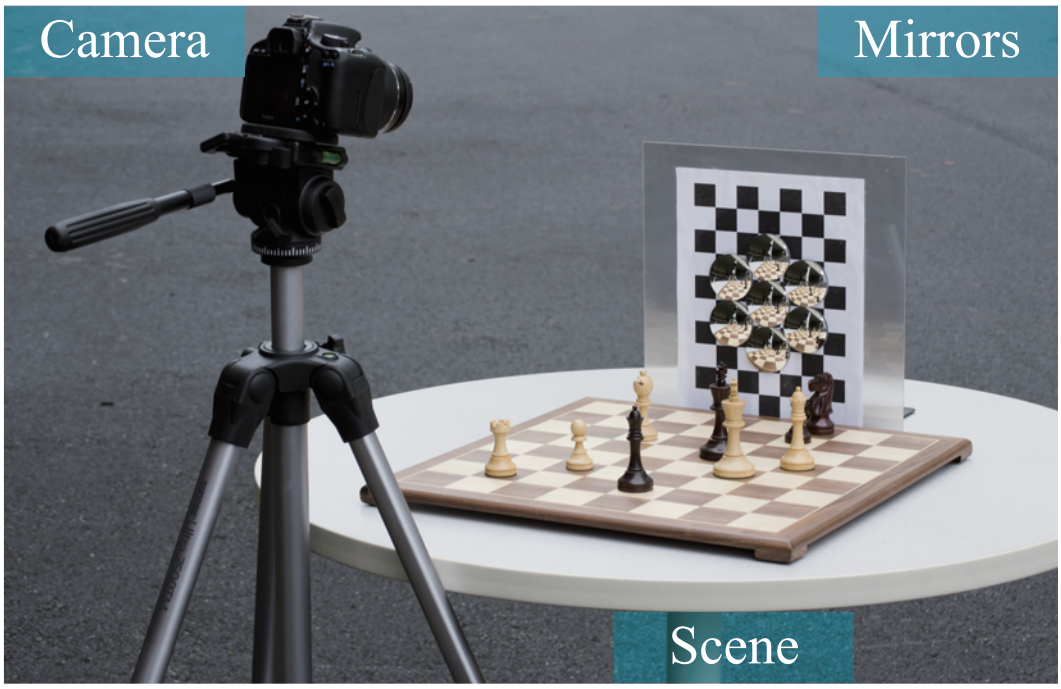


Figure 8.2: The catadioptric mirror array and our CAP setup.

project each voxel onto the mirror, a forward ray-tracing problem that does not have closed form solutions for general mirrors. Thanks to the special property of spherical mirrors, we adopt the closed form solution by Agrawal et al. [9] to efficiently forward trace the rays.

8.3.2.1 Forward Projection

Given the position (0,0,d) of the camera CoP and a 3D point P=(X_p, Y_p, Z_p), our goal to find the image of P on the camera sensor via a single reflection on the spherical mirror M. We denote the reflection point m as (x, y) on plane Π formed by CoP, P, and center of M. According to the derivation by Agrawal et al. [9], we can compute y by solving the following 4^{th} order equation:

$$\sqrt{X_p^2 + Y_p^2}(r^2(d+y) - 2dy^2)^2 - (r^2 - y^2)(r^2(d+Z_p) - 2dZ_p y)^2 = 0 \quad (8.1)$$

And $x = \pm\sqrt{r^2 - y^2}$. Therefore, for each P in the scene, we can instantly get the corresponding incident ray v from the camera by linking the CoP and m, and subsequently

map P onto a pixel on the sensor by intersecting v with the sensor plane.

8.3.2.2 Voxel-Pixel Mapping

Next, we show how to use the forward projection to reconstruct the scene. We first partition the 3D space into voxels by placing a virtual perspective camera close to the central mirror and uniformly discretize its viewing frustum . Next, for each voxel in the viewing frustum, we find the corresponding pixel on the captured image using the forward projection. Finally, we map the problem of reconstructing scene geometry onto a graph-cut framework. Specifically, We first build a X×Y graph (X and Y are the user defined resolution of the virtual perspective camera) and loop over all depth values in the viewing frustum of the virtual perspective camera. For each depth value D_i, we reassign the data term D_{ixy} and smoothness term of S_{ixy} for each node N_{ixy} and perform max-flow min-cut algorithm. We select the optimal labels of the nodes in the graph as the depths of the pixels on the captured image by the virtual perspective camera. Note that the captured low light image is noisy and affects color consistency of the data term, we first apply a single image denoising process (BM3D in our case) on the captured image to get an initial guess of the denoised result. Fig. 8.3 shows two depth maps generated by our algorithm (after 5 iterations) for synthetic scenes.

8.3.2.3 Pixel-Pixel Correspondence

With the estimated Voxel-Pixel Correspondence and the depth map, for a given pixel p in a given mirror M, we can find the corresponding pixels in other views. Specifically, for each voxel V_i corresponding to p, we map it onto the depth map of the virtual perspective camera and check if the depth of V_i is coherent with the value on the depth map. Note that multiple V_i could be found in this step, we choose the V_i with the closest distance d_r to M since all the voxels further away should be occluded by V_i. Finally, we use the Voxel-Pixel Correspondence to locate all images of V across mirrors. Inaccuracies in the recovered depth map due to noise can cause problems when finding V_i. We address the problem by using a search window to find the corresponding patches

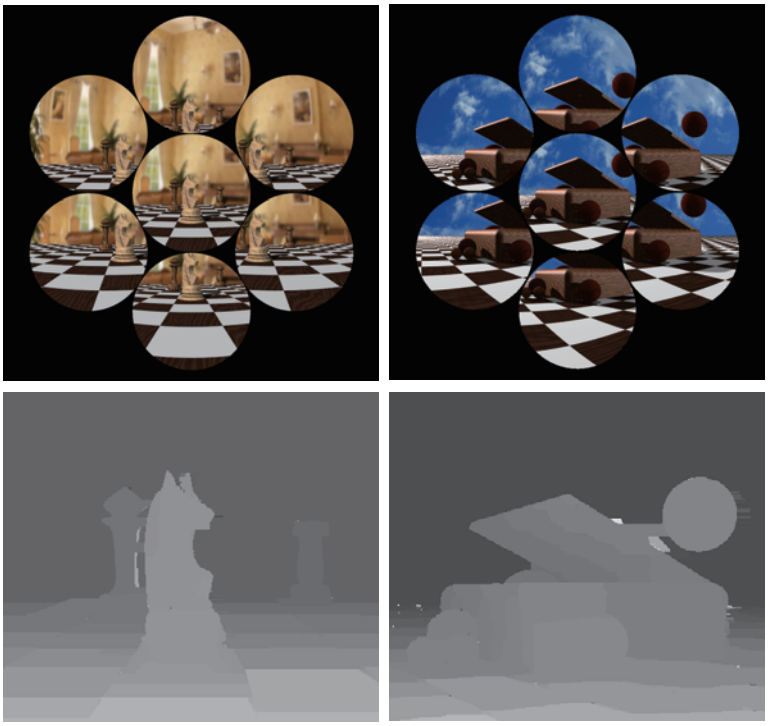


Figure 8.3: Our recovered depth maps of three synthetic scenes using the MVMP space carving scheme.

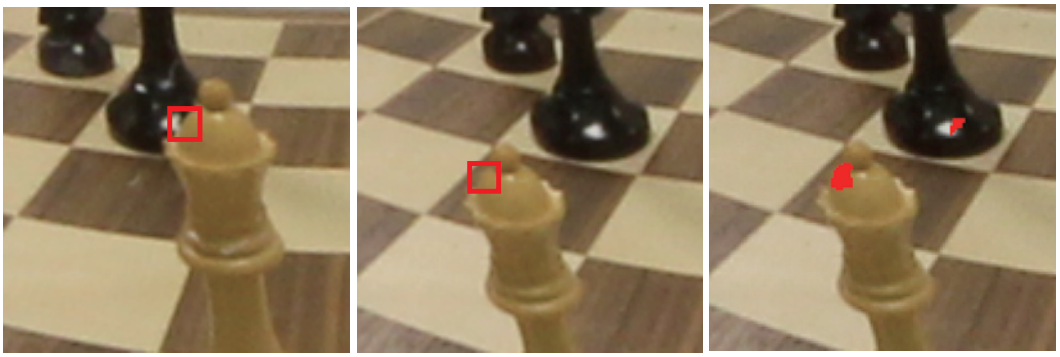
in other views (Section 8.4.1). Further, since our ultimate goal is noise reduction, as far as the correspondence maps are relatively accurate, the outliers will be assigned with a smaller weight in our denoising method.

8.4 MVMP Denoising

Given our estimated pixel correspondences, we set out to denoise the captured MVMP image. Our approach is inspired by recent Multi-View image denoising [134].

8.4.1 Patch Matching

Similar to conventional patch-based denoising, for each pixel p in a mirror M on the image, we first select the $n \times n$ patch (n is user defined) centered at p as the *reference patch* $\boldsymbol{b}_r$ and then search for all similar patches within M and across other mirrors.



(a) (b) (c)

Figure 8.4: Demonstration of patch warping on two scenes (one synthetic, one real) under the MVMP context. (a) Reference patch on the central mirror. (b) Corresponding patch on another mirror without considering patch warping.[134] (c) Corresponding pixels found with our approach.

To find similar patches within M, a brute-force approach is to consider all the patches $\boldsymbol{b}_i$ corresponding to $\boldsymbol{b}$ over different views with the depth constraint. Although it works well in the perspective multi-view case, it is less optimal in the MVMP case because the warping of $\boldsymbol{b}_i$ from one view to another is non-linear (perspective). This can be demonstrated by purposely select $\boldsymbol{b}_r$ to check the similarity with itself. As shown in Fig. 8.4(b), without patch warping, on another mirror, pixels from an alien object are incorrectly selected to compute the weight, hence degrading the weight of the correct patch.

We resolve the problem by using a more robust patch matching scheme. Instead of locating the corresponding patch with the depth of p directly, for each pixel q in patch $\boldsymbol{b}$, we find the corresponding pixels in other views to compensate for the severe deformation of the patch. As shown in Fig. 8.4(c), given $\boldsymbol{b}_r$, the corresponding pixels will be selected correctly from the other mirrors. The distance from $\boldsymbol{b}$ to $\boldsymbol{b}_r$ is then computed as:

$$\Phi(\boldsymbol{b}, \boldsymbol{b}_r) \quad = \quad \sum_i \sum_{q \in \boldsymbol{b_i}} ||q - q_r||_2, \tag{8.2}$$

where q_r denote a pixel in $\boldsymbol{b}_r$, and the $L2$ norm computes the squared color difference

Figure 8.5: Comparison of different denoising schemes on a logo scene.

between the two pixels. The weight of $\boldsymbol{b}$ is then computed by $\exp\{\frac{-\Phi(\boldsymbol{b},\boldsymbol{b}_r)}{2\sigma^2}\}$, i.e., the smaller the distance, the higher the weight. We determine σ by estimating noise level of the image. To make use of similar patches over all mirrors, we first map p onto other mirrors based on the pixel correspondences. Within each mirror, we perform our MVMP matching method to weight patches based on the new pixel location. Assume p is captured by n mirrors, and we select k most similar patches from each mirror, then we get kn patches for denoising p.

8.4.2 Patch-based Denoising

Assuming all kn patches have a similar underlining structure, to denoise each pixel p on the captured image, our approach takes advantage of the recent patch based denoising using Principal Component Analysis (PCA) [136]. Instead of trying to cancel out the noise on the dimension of the patch itself, we assume that the noiseless patch lies in a lower dimensional subspace to increase the robustness against outliers. Specifically, we first find the dimensionality of the subspace by matching the average squared residuals of noisy patches and the denoised patches to the noise variance, then estimate the subspace by minimizing the difference between the noisy patches and the denoised patches. We choose the PCA based approach simply because it generates the highest PSNR during our synthetic experiments.

Once we conduct patch-based denoising to all mirror images, we then re-estimate the depth map and use the refined depth map to re-apply patch-based denoising. To determine the rounds of iterations that would be needed, we measure the difference

between the results of two consecutive rounds and if it is small enough, we stop the iteration and output the final image. In all our experiments, our first iteration use the BM3D denoised results to estimate the depth map and we find that in nearly all cases it takes less than 5 iterations for our algorithm to converge.

8.5 Multi-resolution Enhancement

A major disadvantage of our CAP photography is the loss of resolution. With 7 mirrors covering 12M pixels on the sensor, the denoised result has only less than 1.8M pixel resolution. To compensate that, we present a multi-resolution denoising scheme. A closely related work is the image deblurring with blurred/noisy image pair technique developed by Yuan et al. [133]. Their model, however, does not apply directly to our problem since they use an image pair with the same resolution.

Our strategy is to zoom in onto the central mirror M and capture a full resolution image of M without changing the CoP of the view camera, i.e. the ray geometry remains the same for the central mirror but is sampled at a much higher resolution. We denote the low resolution image of M as I_l and full resolution image of M as I_h. Our approach is to denoise I_h by imposing the denoised I_l as a prior.

Given the denoised I_l, we find similar patches of a reference patch $\boldsymbol{b}_{rh}$ at pixel p within I_h, and use them to denoise p. The simplest approach is to first map all patches to I_l and then compute the similarities. This approach is robust against large noise on I_h with the expense of accuracy on edges, due to the low resolution of I_l. Our method improves this basic approach by combining the distances on I_l and I_h so that only patches close to $\boldsymbol{b}_r$ on both I_l and I_h are treated as similar ones. Specifically, we compute the weight of each patch $\boldsymbol{b}_h$ in I_h by:

$$w(\boldsymbol{b}_h) = \exp\{-\frac{\Phi(\boldsymbol{b}_h, \boldsymbol{b}_{rh})}{\sigma_h^2}\} \exp\{-\frac{\Phi(\boldsymbol{b}_l, \boldsymbol{b}_{rl})}{\sigma_l^2}\}, \tag{8.3}$$

where $\boldsymbol{b}_l$ and $\boldsymbol{b}_{rl}$ denote the corresponding patches of $\boldsymbol{b}_h$ and $\boldsymbol{b}_{rh}$ in I_l respectively. σ_h is determined by the noise level of I_h and σ_l is computed from the resolution difference between I_h and I_l. To denoise each pixel on I_h, we select k most similar patches and

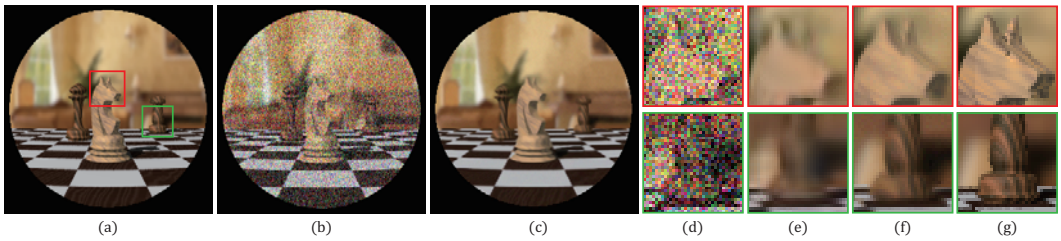


Figure 8.6: Comparison of our result with BM3D on a synthetic scene. (a) The ground truth images. (b) The synthetic noisy images. (c) Our denoised low resolution results. (d), (e), (f), and (g) are the closeup views of the highlighted regions from the noisy images, BM3D denoised results, our results, and the ground truth respectively. PSNR (computed by cropping out the central view): Ours: 33.25, BM3D: 30.78.

conduct our PCA based denoising. Fig. 8.5 shows our results comparing with BM3D on a synthetic scene.

8.6 Experimental Results

We have conducted thorough experiments of our CAP-based low light imaging on both synthetic and real low light scenes. For synthetic scenes, we render the reflections of 7 spherical mirror caps of radius 1 and height 0.2588 (which covers 30° of the sphere) from a perspective camera of resolution 1200×900. To emulate low lights, for each pixel on the synthesized noisy image, we add Poisson noise using a Poisson random number with mean and variance both equal to $\frac{1}{18}$ of the clean pixel intensity. To demonstrate the robustness of our approach, we compare our results with the ones from BM3D (Fig. 8.6), the *de facto* benchmark for image denoising. In all following examples (synthetic or real), we apply BM3D to the complete image rather than a small mirror view for fairness. In the horse scene (Fig. 8.6), our technique achieves a higher PNSR ratio than BM3D. Further, CAP is able to better preserve fine details as shown near the ears of the horse and around the triangle patterns on the background. This is mainly because BM3D only attempts to locate the nearby similar patches whereas CAP establishes and then utilizes similar patches across mirrors.

For real scenes, the setup of the CAP system is discussed in Sect. 8.3.1. We

Figure 8.7: Results on a stair scene. (a) The noisy input image. (b) The BM3D result. (c) Our CAP-based low light imaging results. (d) The reference image acquired with a long exposures. (d) Recovered depth map.

use lens speed F-14 to minimize defocus blurs with a high ISO of 6400. To capture the mirror array images and later the zoomed-in central mirror image, we use 100 and 300 mm focal lengths. For static scenes, the view camera captures at a resolution of 5184×3456. However, since the catadioptric mirror array only occupies a portion rather than the complete image, the effective CAP resolution is reduced to about 3400×3500 or 12M pixels. For dynamic scenes, we use the high definition video mode on the camera that captures at a resolution of 1920×1080 at 60 fps. For static scenes, we capture the images without using a tripod and use 1/60 second shutter speed to minimize motion blurs. For indoor scenes, we simply use the image captured under flash as the reference image. For outdoor scenes, we capture the reference image by facing the camera directly towards the scene using a tripod with a slow shutter (6 seconds).

8.6.0.0.1 Static Scenes

Fig. 8.7 shows our results on a real static scene of a building. Fig. 8.7(a) shows the central mirror image of the raw data. The image exhibits high levels of noise.

For example, comparing with the long exposure result (c), the grids of the window and the door are corrupted. We then apply our MVMP denoising algorithm. As the surfaces are mostly Lambertian, our multi-perspective multi-view stereo algorithm is able to robustly establish correspondences. Fig. 8.7(b) shows the denoised central mirror image and we are able to effectively remove the noise and faithfully recover details such as the grids of the window and the door using multiple views.

In Fig. 8.8, we show a candle scene similar to the teaser in the flash/no-flash paper [36]. By applying our MVMP denoising and super-resolution schemes, we are able to produce the warm atmosphere under ambient lighting whereas the flash image leads to harsh, opaque lighting. Further, our technique does not require special handling of shadows caused by flash. Our MVMP result already shows similar quality with the result achieved by applying BM3D and downsampling on the close up view of the central mirror at resolution of 3000×2500. Moreover, our multi-resolution denoising result is able to recover fine details such as the characters on the stones which could only be captured by using a long exposure on the close up view of the central mirror. Our correspondence matching, however, fails on the wine glass as we can only handle Lambertian surfaces. It is worth noting that even though our low-resolution central mirror image contains such artifacts near the specular wine glass, the super-resolution scheme is able to partially reduces the artifacts thanks to the cross-bilateral weighting scheme.

In the chess scene (Fig. 8.9), our result faithfully preserves the intrinsic color of the chessboard and the chess pieces whereas flash illumination easily corrupts the color and introduces shadows. Although flash/no-flash photography can be potentially used to address this issue, it will be difficult to resolve artifacts caused by strong discontinuities around shadow edges.

8.6.0.0.2 Dynamic Scenes

Fig. 8.10 shows our result on a real dynamic scene. This fountain scene is particularly challenging: to reduce motions, we have to use fast shutters; however, the

Figure 8.8: The CAP-based results on a candle scene.

environment lighting is sufficient, therefore the resulting video stream is highly noisy. Recall that none of the traditional approaches would work well in this scenario: long shutter leads to motion blurs, wide aperture leads to defocus blurs, and flash cannot be continuously applied to videos. By applying our CAP-based low light photography, we are able to not only acquire blur free video streams but also significantly reduce the noise and at the same time preserve fine details. A downside of our approach, however, is that we are unable to recover a high resolution video as our multi-resolution enhancement technique is only suitable for static scenes. We refer the reviewers to the supplementary videos for more results.

8.7 Discussions and Future Work

We have presented a low-light photography scheme based on a relatively simple and reusable catadioptric array setup. Our post-processing technique first finds pixel (patch) correspondences across multiple mirror views on the photography and then conducts MVMP denoising and multi-resolution enhancement to produce high quality,

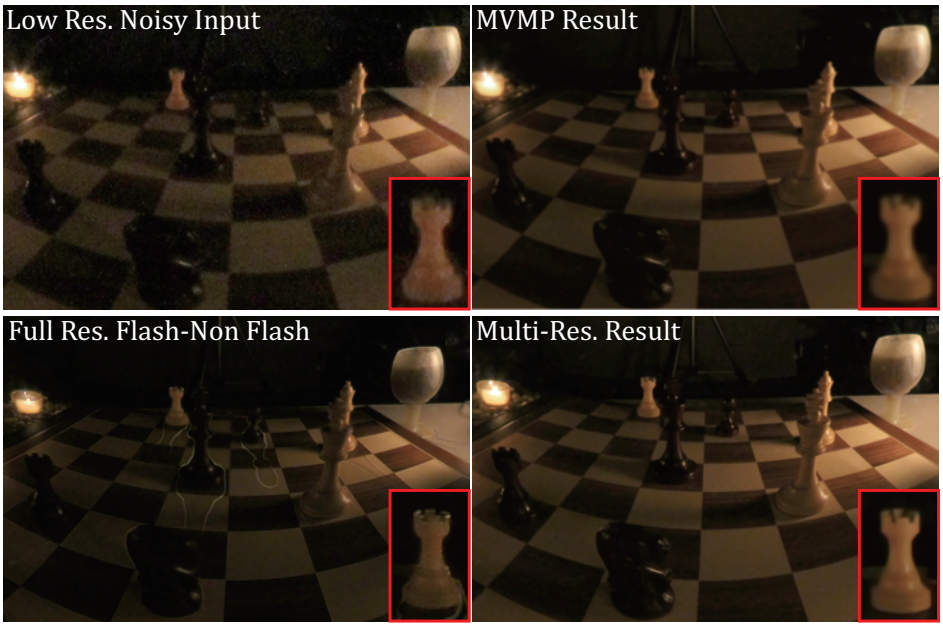


Figure 8.9: Results on a chess scene compared with Flash-Non Flash approach focusing on the central mirror at resolution of 3000×2500 and downsample.

high resolution imagery while preserving intrinsic lighting.

8.7.0.0.3 CAP vs. Light Field Photography

Similar to CAP, Light Field Photography (LFP) also aims to capture scenes from multiple views in a single photograph. A significant advantage that LFP has over CAP is that each microlens image in LFP directly corresponds to a perspective image and therefore multi-view stereo matching can be directly applied to establish pixel (patch) correspondences. This would avoid the complex multi-perspective back-projection step in CAP. However, there are a number of factors that prohibit the direct use of LFP in low light imaging. First and foremost, the aperture size of each microlens "camera" is ultra-small. For example, the commercial light field camera Lytro uses a main lens with aperture F/2. However, when coupled with a dense microlens array (consisting of thousands of microlenses in Lytro camera), the effective aperture size drops to f/2 ($f < \frac{F}{100}$). The ultra-small aperture will lead to low SNR and add significant challenges to any state-of-the-art denoising algorithm. Second, while our

Figure 8.10: Comparison of CAP-based solution vs. Flash photography on dynamic scenes. (Left) The raw noisy input image. (Mid) Our CAP-based imaging results (gamma corrected for the fountain scene to match the intensity with long exposure). (Right) Images acquired with long exposure (top). More results can be found in the supplementary video.

CAP supports dynamic zoom-in/zoom-out for trading off between spatial and angular resolutions and super-resolution, LFP does not support similar mechanism. In fact, the light field super-resolution remains as the one of the most challenging problems in computational photography.

Our CAP low-light photography also has to use a relatively small aperture (F/14) to avoid multi-perspective defocusing [33]. However, unlike LFP, the aperture size does not scale down by the number of mirrors, i.e., each mirror has the same aperture F/14, or at least 51 times larger than the microlens camera in Lytro. Therefore, CAP fits better for low-light photography. It is worth noting that if future LFP can accommodate a smaller (and controllable) set of microlenses, e.g., with F/2 in LFP and no more than one thousand microlenses, it would be able to achieve comparable performance to CAP.

8.7.0.0.4 Future Directions

There are a number of future directions that we plan to explore. As mentioned above, our CAP still uses a small aperture to avoid defocusing artifacts. However, it also boosts the noise level due to insufficient light. In the future, we plan to study the tradeoff between defocus and denoise in CAP. Recent studies [96, 135] have shown that defocus may be a better option than denoise and we will investigate which approach is more suitable for our super-resolution algorithm. In addition, our current setup only uses a single view camera. For dynamic scenes, our approach can effectively denoise

but cannot super-resolve. An important future direction is to use an auxiliary camera to capture the high resolution video stream and fuse it with our denoised low-resolution videos.

Chapter 9

CONCLUSION AND FUTURE WORK

9.1 Conclusions

In this dissertation, I have presented new image processing algorithms and camera designs to improve the spatial, angular, and temporal resolution of light field imaging.

9.1.1 Spatial Resolution

To improve the image resolution of the light field camera, we have presented a well-principled plenoptic demosaicing and rendering framework, which preserves more high frequency information from the captured light field and generates less aliasing artifacts compared with the classical approach.

Our framework does not apply demosaicing directly to the image captured by the plenoptic camera. Instead, with a resampling scheme which helps achieve constant spacing on each dimension, it dynamically performs demosaicing after integral projection. Extensive experiments show that this framework could produce photographs with commercially acceptable resolution.

9.1.2 Angular Resolution

To increase the angular resolution, we have presented a light field triangulation approach by imposing ray geometry of 3D line segments as constraints. We utilize CDT and by far our solution is restricted to 3D and pseudo 4D light fields since 4D CDT is still an open problem in computational geometry.

Improved Light Field Stereo Matching An accurate depth map is crucial in light field superresolution. To improve current light field stereo matching algorithms, we

have presented two novel solutions. To resolve the occlusion problems. We apply an iterative plane sweeping from the closest depth layer to the furthest, so that the occlusion pixels will be masked out when estimating local minima. We further propose a global optimization solution and an edge mask solution to avoid trivial solutions on textureless surfaces.

Based on our ray geometry analysis of 3D lines, we introduce a new $\mathcal{F}^3$ energy term to preserve disparity consistency along line segments in light field stereo matching. We then modify the binocular stereo graph via the general purpose graph construction framework and solve it using the extended Quadratic Pseudo-Boolean Optimization algorithm. Experiments show that both our light field triangulation and stereo matching algorithms outperform state-of-the-art solutions in accuracy and visual quality.

9.1.3 A Unified Spatial-Angular Resolution

To enhance spatial and angular resolution of the light field under a unified framework, we have presented a high-dimensional image based rendering technique which takes a set of 4D light fields as inputs and produces novel 4D light fields depending on user defined hyper-geometry proxies in the light field space. Since the input light fields have denser angular samples, the result light field is no longer restricted by the angular limitation. Therefore, with this new technique, we can produce novel effects such as panorama with dynamic DoF, panorama with parallax, and novel views with larger parallax and shallower DoF.

We have demonstrated the approach for enhancing the light field resolution at different dimensions. For example, we can create a wide horizontal FoV light field from a series of light fields captured by rotating the Lytro camera on a tripod. We can also create an ultra-high spatial resolution light field using an array of Lytro cameras. The same structure allows us to increase the size of the virtual aperture and hence the bokeh. Finally, we can increase the parallax between light field views by orbiting the Lytro camera around the object of interest.

9.1.4 Temporal Resolution

To improve the temporal resolution of current light field imaging, we have presented an affordable stereo solution for generating light field and producing high quality racking focus and tracking focus effects. Specifically, we have constructed a hybrid-resolution stereo camera system by coupling a high-res/low-res camera pair. We recover a low-res disparity map and subsequently upsample it via fast cross bilateral filters. We then use the recovered high-resolution disparity map and its corresponding video frame to synthesize a light field. We implement a GPU-based disparity warping scheme and exploit atomic operations to resolve visibility. To reduce aliasing, we present an image-space filtering technique that compensates for spatial undersampling using mipmapping. Finally, we generate racking focus and tracking focus effects using light field rendering. Our system shows promising results on indoor and outdoor scenes.

9.2 Future Work

There are a number of future work I would like to explore.

9.2.1 Spatial Resolution

On the demosaicing front, the resolution enhancement of each plane in the scene achieved by our algorithm varies according to the depth of the plane. We plan to explore new light field camera designs that automatically find the best spatial angular tradeoff on planes of interests based on the scene content analysis such saliency detection.

Moreover, since each microlens is not equivalent to a pinhole, our thin ray assumption is not ill posed. We plan to further analyze the effect of defocus on the different planes and how it affects the resolution enhancements.

9.2.2 Angular Resolution

An immediate future direction of our light field triangulation is to experiment our scheme on irregularly sampled light field, e.g., the ones captured by a catadioptric

mirror array or by a hand-held camera. Our current super-resolution scheme requires rasterizing ray simplices into voxels.

An alternative approach is to use a walk-through algorithm that picks one face of the ray simplex at a time and does the orientation test for locating the simplex, a process can be accelerated using parallel processing on the graphics hardware. Finally, the triangulated light field can be potentially compressed via geometric compression. For example, half-edge collapse operator in progressive meshes can be used to remove edges and vertices while maintaining a continuous simplex-tiled structure.

9.2.3 A Unified Spatial-Angular Resolution

Our light field quilting algorithm uses the 5D homography to model the warping between light fields to preserve spatial-angular continuity. An immediate future direction is to explore the depth based warping to better represent the ray geometry. Moreover, thresholds such as k, t, d in our homography estimation are empirically chosen. We plan to leverage image statistics for automatically assigning those values.

Currently we fix the shutter speed and ISO of the light field camera to match color between different light fields. In the furture, we also plan to explore the gradient domain composition on the light field quilting using sparsely sampled light fields as inputs.

9.2.4 Temporal Resolution

For our stereo based light field acquisition system, we plane to integrating existing lower cost webcams with realtime lens correction algorithm to further reduce our form factor. Moreover, most of current disparity estimation algorithms can not handle transparent objects such as planar glass and hair. We plan to improve the performance of our disparity estimation algorithm on those regions by proposing a new statistical model. For example, we could assume each pixel of the captured image may contain information from multiple depths. In this case, we can separately estimate different depth values for a single pixel.

BIBLIOGRAPHY

[1] James Adams and John Hamilton Jr. Adaptive color plan interpolation in single sensor color electronic camera, patent us 5506619, 1996.

[2] E. Adelson and J. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:99–106, 1992.

[3] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[4] Aseem Agarwala, Maneesh Agrawala, Michael Cohen, David Salesin, and Richard Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH 2006 Papers*, 2006.

[5] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3), August 2004.

[6] Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. Panoramic video textures. *ACM Trans. Graph.*, 24(3), July 2005.

[7] A. Agrawal, Y. Taguchi, and S. Ramalingam. Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors. In *IEEE Conference on CVPR*, pages 2993 –3000, june 2011.

[8] Amit Agrawal, Ramesh Raskar, Shree K. Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.*, July 2005.

[9] Amit Agrawal, Yuichi Taguchi, and Srikumar Ramalingam. Analytical forward projection for axial non-central dioptric and catadioptric cameras. In *Proceedings of ECCV*, 2010.

[10] Amit K. Agrawal, Ashok Veeraraghavan, and Ramesh Raskar. Reinterpretable imager: Towards variable post-capture space, angle and time resolution in photography. *Comput. Graph. Forum*, 29, 2010.

[11] D. Alleysson, S. Susstrunk, and J. Herault. Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4):439–449, apr. 2005.

[12] Dominique Attali and Jean-Daniel Boissonnat. Complexity of the delaunay triangulation of points on polyhedral surfaces. *Discrete & Computational Geometry*, 30(3):437–452, 2003.

[13] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4), December 1996.

[14] B. E. Bayer. Color imaging array. US Patent 3,971,065, 1976.

[15] Eric P. Bennett and Leonard McMillan. Video enhancement using per-pixel virtual exposures. SIGGRAPH 2005. ACM, 2005.

[16] Clemens Birklbauer and Oliver Bimber. Panorama light-field imaging. In *ACM SIGGRAPH 2012 Talks*, SIGGRAPH '12, 2012.

[17] Tom E. Bishop, Sara Zanetti, and Paolo Favaro. Light field superresolution. In *IEEE ICCP*, 2009.

[18] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR 2010*.

[19] Endre Boros, Peter L. Hammer, and Gabriel Tavares. Preprocessing of unconstrained quadratic binary optimization. Technical report, 2006.

[20] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE TPAMI*, 23:2001, 2001.

[21] A. Brunton, Chang Shu, and G. Roth. Belief propagation on the gpu for stereo vision. In *Computer and Robot Vision, The 3rd Canadian Conference on*, 2006.

[22] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Simul*, 4, 2005.

[23] Antoni Buades and Bartomeu Coll. A non-local algorithm for image denoising. In *In CVPR*, pages 60–65, 2005.

[24] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. SIGGRAPH '01, 2001.

[25] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 307–318, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[26] Vincent Couture, Michael S. Langer, and Sebastien Roy. Panoramic stereo video textures. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 2011.

[27] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising with block-matching and 3d filtering. In *IN ELECTRONIC IMAGING*, 2006.

[28] Donald G. Dansereau, Oscar Pizarro, and Stefan B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Computer Vision and Pattern Recognition, 2013. IEEE Computer Society Conference on*, 2013.

[29] Middlebury Stereo Datasets. http://vision.middlebury.edu/stereo/data/.

[30] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. *Comp. Graph. Forum*, May 2012.

[31] Boris N. Delaunay. Sur la sphére vide. In *Bulletin of Academy of Sciences of the USSR*, pages 793–800, 1934.

[32] Yuanyuan Ding, Jing Xiao, Kar-Han Tan, and Jingyi Yu. Catadioptric projectors. In *IEEE Computer Society Conference on CVPR*. IEEE, 2009.

[33] Yuanyuan Ding, Jing Xiao, and Jingyi Yu. A theory of multi-perspective defocusing. In *IEEE Conference on CVPR*, 2011.

[34] Yuanyuan Ding, Jingyi Yu, and P. Sturm. Multiperspective stereo matching and volumetric reconstruction. In *IEEE ICCV*, 2009.

[35] E. Dubois. Filter design for adaptive frequency-domain bayer demosaicking. In *IEEE International Conference on Image Processing*, oct. 2006.

[36] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 2004.

[37] Sina Farsiu, Michael Elad, and Peyman Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15(1), jan. 2006.

[38] P.F. Felzenszwalb and D.R. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004.

[39] Todor Georgeiv, Ke Colin Zheng, Brian Curless, David Salesin, Shree Nayar, and Chintan Intwala. Spatio-angular resolution tradeoff in integral photography. In *Proceedings of Eurographics Symposium on Rendering*, pages 263–272, 2006.

[40] T. Georgiev and A. Lumsdaine. Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, 19, 2010.

[41] Todor Georgiev, Georgi Chunev, and Andrew Lumsdaine. Superresolution with the focused plenoptic camera. In *Proc. SPIE 7873*, 2011. doi:10.1117/12.872666.

[42] Steven Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael Cohen. The lumigraph. In *Proceedings of ACM SIGGRAPH*, pages 43–54, 1996.

[43] S. Grauer-Gray, C. Kambhamettu, and K. Palaniappan. Gpu implementation of belief propagation using cuda for cloud tracking and reconstruction. In *Pattern Recognition in Remote Sensing (PRRS 2008), 2008 IAPR Workshop on*, pages 1 –4, December 2008.

[44] Kaiming He, Huiwen Chang, and Jian Sun. Rectangling panoramic images via warping. *ACM Trans. Graph.*, July 2013.

[45] Felix Heide, Gordon Wetzstein, Ramesh Raskar, and Wolfgang Heidrich. Adaptive image synthesis for compressive displays. *ACM Trans. Graph.*, 32(4), July 2013.

[46] Yong Seok Heo, Kyoung Mu Lee, and Sang Uk Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *Computer Vision and Pattern Recognition*, 2007.

[47] R. Hibbard. Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients, patent us 5506619, 1995.

[48] Aaron Isaksen, Leonard McMillan, and Steven Gortler. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH*, pages 297–306, 2000.

[49] R. Kakarala and Z. Baharav. Adaptive demosaicing with the principal vector method. *IEEE Transactions on Consumer Electronics*, 48(4):932–937, nov. 2002.

[50] Masayuki Kanbara, Norimichi Ukita, Masatsugu Kidode, and Naokazu Yokoya. 3d scene reconstruction from reflection images in a spherical mirror. In *International Conference on Pattern Recognition*, pages 874–879, 2006.

[51] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *TPAMI*, 29(7):1274–1279, 2007.

[52] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *In the proceedings of ICCV*, volume 2, 2001.

[53] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *PAMI, IEEE Transactions on*, 26(2):147 –159, feb. 2004.

[54] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the ECCV*, 2002.

[55] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *SIGGRAPH*, 2007.

[56] D. Krishnan and R. Fergus. Dark flash photography. *ACM SIGGRAPH*, 2009.

[57] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3), July 2000.

[58] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.

[59] Douglas Lanman, Daniel Crispell, Megan Wachs, and Gabriel Taubin. Spherical catadioptric arrays: Construction, multi-view geometry, and calibration. In *Proceedings of 3DPVT*, 3DPVT '06, 2006.

[60] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. Depth-of-field rendering with multiview synthesis. In *SIGGRAPH Asia*, 2009.

[61] Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):453–464, 2009.

[62] Jaakko Lehtinen, Timo Aila, Jiawen Chen, Samuli Laine, and Frédo Durand. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.*, 30(4), 2011.

[63] Anat Levin and Fredo Durand. Linear view synthesis using a dimensionality gap light field prior. In *In Proc. IEEE CVPR*, pages 1–8, 2010.

[64] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of ACM SIGGRAPH*, pages 31–42, 1996.

[65] Feng Li, Jingyi Yu, and Jinxiang Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *Computer Vision and Pattern Recognition,CVPR 2008. IEEE Conference on*, 2008.

[66] X. Li and M. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10:1521–1527, 2001.

[67] N. Lian, L. Chang, Y. Tan, and V. Zagorodnov. Adaptive filtering for color filter array demosaicking. *IEEE Transactions on Image Processing*, 16(10):2515 –2525, oct. 2007.

[68] G. Lippmann. La photographie intgrale. *Comptes-Rendus, Acadmie des Sciences*, 146:446–451, 1908.

[69] W. Lu and Y. Tan. Color filter array demosaicking: new method and performance measures. *IEEE T-IP*, 12:1194–1210, oct. 2003.

[70] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *In Proc. IEEE ICCP*, 2009.

[71] Lytro. www.lytro.com.

[72] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4), 2013.

[73] D. Menon and G. Calvagno. Demosaicing based on wavelet analysis of the luminance component. In *IEEE International Conference on Image Processing*, volume 2, pages 181–184, 2007.

[74] D. Muresan and T. Parks. Demosaicing using optimal recovery. *IEEE Transactions on Image Processing*, 14(2), feb. 2005.

[75] Ren Ng. Fourier slice photography. *ACM Trans. Graph.*, 24:735–744, July 2005.

[76] Ren Ng. Fourier slice photography. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 2005.

[77] Ren Ng. Fourier slice photography. In *SIGGRAPH*, 2005.

[78] Ren Ng. Digital light field photography. *Doctoral Dissertation*, 2006.

[79] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision*, 81(1):24–52, 2009.

[80] S. Peleg and M. Ben-Ezra. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, 1999.

[81] Jörg Peters and Ulrich Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Graph.*, 16(4), October 1997.

[82] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. SIGGRAPH, 2004.

[83] Sören Pirk, Michael F. Cohen, Oliver Deussen, Matt Uyttendaele, and Johannes Kopf. Video enhanced gigapixel panoramas. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, 2012.

[84] Jean Ponce and Yakup Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. *IJCV 1996*.

[85] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process*, 12:1338–1351, 2003.

[86] POV-Ray. www.povray.org.

[87] Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 199–206, New York, NY, USA, 1998. ACM.

[88] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaics: video mosaics with non-chronological time. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.

[89] Raytrix. www.raytrix.com.

[90] Christian Richardt, Yael Pritch, Henning Zimmer, and Alexander Sorkine-Hornung. Megastereo: Constructing high-resolution stereo panoramas. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[91] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR 2007*.

[92] B. Goldlücke S. Wanner. Globally consistent depth labeling of 4d light fields. In *Proceedings of IEEE CVPR*, 2012.

[93] Harpreet S. Sawhney, Yanlin Guo, Keith Hanna, and Rakesh Kumar. Hybrid stereo camera: an ibr approach for synthesis of very high resolution stereoscopic image sequences. In *SIGGRAPH*, pages 451–460, 2001.

[94] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47:7–42, April 2002.

[95] Steven M. Seitz and Jiwon Kim. The space of all stereo images. *Int. J. Comput. Vision*, 48(1), June 2002.

[96] Qi Shan, Jiaya Jia, Sing Bing Kang, and Zenglu Qin. Using optical defocus to denoise. In *CVPR*, 2010.

[97] Jonathan Richard Shewchuk. General-dimensional constrained delaunay and constrained regular triangulations i: Combinatorial properties. In *Discrete and Computational Geometry*, 2005.

[98] C. Shi, G. Wang, X. Pei, H. Bei, and X. Lin. High-accuracy stereo matching based on adaptive ground control points. *Submitted to IEEE TIP*, 2012.

[99] Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.

[100] Hang Si. Tetgen: A 3d delaunay triangulator.

[101] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3), July 2006.

[102] J. Stewart, J. Yu, S. J. Gortler, and L. McMillan. A new reconstruction filter for undersampled light fields. EGSR '03, pages 150–156, 2003.

[103] Leslie Stroebel, John Compton, Ira Current, and Richard Zakia. Photographic materials and processes. In *Focal Press*, 1986.

[104] Leslie Stroebel, John Compton, Ira Current, and Richard Zakia. *Photographic Materials and Processes*. 1986.

[105] Brian Summa, Julien Tierny, and Valerio Pascucci. Panorama weaving: fast and flexible seam processing. *ACM Trans. Graph.*, 31(4), July 2012.

[106] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *PAMI, IEEE Transactions on*, 25(7):787 – 800, 2003.

[107] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.

[108] Yuichi Taguchi, Amit Agrawal, Ashok Veeraraghavan, Srikumar Ramalingam, and Ramesh Raskar. Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. In *ACM SIGGRAPH Asia*, 2010.

[109] R. Tsai and Thomas Huang. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, 1, 1984.

[110] J. Tumblin and R. Raskar. State of the art report (star): Computational photography. In *ACM/Eurographics 2006*, 2006.

[111] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. EGRW '03, 2003.

[112] Stanford University. Stanford light field.

[113] Stanford University. Stanford spherical gantry.

[114] V. Vaish, M. Levoy, R. Szeliski, C.L. Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *CVPR*, 2006.

[115] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *CVPR*, 2004.

[116] Vaibhav Vaish, Gaurav Garg, Eino-Ville Talvala, Emilio Antunez, Bennett Wilburn, Mark Horowitz, and Marc Levoy. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03*, CVPR, pages 129–, 2005.

[117] Vaibhav Vaish, Richard Szeliski, C. L. Zitnick, Sing Bing Kang, and Marc Levoy. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *Computer Vision and Pattern Recognition*, 2006.

[118] 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 Trans. Graph.*, July 2007.

[119] Videocube. Microsoft.

[120] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *PAMI, IEEE Transactions on*, 32(4):722 –732, april 2010.

[121] S. Wanner and B. Goldluecke. Spatial and angular variational super-resolution of 4d light fields. In *ECCV*, 2012.

[122] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *CVPR*, 2004.

[123] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24:765–776, July 2005.

[124] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 287–296, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[125] O.J. Woodford, P.H.S. Torr, I.D. Reid, and A.W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR, IEEE Conference on*, june 2008.

[126] Yingen Xiong and K. Pulli. Fast image stitching and editing for panorama painting on mobile phones. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010.

[127] Jason C. Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, pages 77–86, 2002.

[128] Qingxiong Yang, Ruigang Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007.

[129] Jingyi Yu. *General linear cameras: theory and applications*. PhD thesis, MIT, 2005.

[130] Jingyi Yu and Leonard McMillan. General linear cameras. In *ECCV (2)*, pages 14–27, 2004.

[131] Jingyi Yu, Leonard McMillan, and Steven Gortler. Scam light field rendering. In *IN: PACIFIC GRAPHICS*, 2002.

[132] Xuan Yu, Rui Wang, and Jingyi Yu. Real-time depth of field rendering via dynamic light field generation and filtering. *Comput. Graph. Forum*, 29(7):2099–2107, 2010.

[133] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. SIGGRAPH. ACM, 2007.

[134] Lei Zhang, Weisheng Dong, David Zhang, and Guangming Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recogn.*, 2010.

[135] Li Zhang, Alok Deshpande, and Xin Chen. Denoising vs. deblurring: Hdr imaging techniques using moving cameras. In *CVPR*.

[136] Li Zhang, S. Vaddadi, Hailin Jin, and S.K. Nayar. Multiple view image denoising. In *Computer Vision and Pattern Recognition*, 2009.

[137] Z. Zhang. A flexible new technique for camera calibration. *PAMI, IEEE Transactions on*, 22(11):1330 – 1334, November 2000.

[138] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2003.

[139] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(6):741–754, June.