Attenuation-based Light Field Displays

Bachelorarbeit

der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Adrian Wälchli

2016

Leiter der Arbeit:
Prof. Dr. Matthias Zwicker
Institut für Informatik und angewandte Mathematik

Abstract

This work reviews and extends an existing, glasses-free 3D display called *Layered 3D* which is based on light field technology. The purpose of the thesis is to understand this specific multi-layer architecture in terms of spectral properties and computational limitations and to adapt the previous work to support non-synthetic light fields. In the first part, the theoretical concept of light attenuation by the display is explained and presented as an optimization problem. The problem of finding the correct attenuation values is closely related to computed tomography and as such, a tomographic reconstruction technique is applied. Subsequently, the limitations of depth of field are presented with an analysis on the spectral support of the light field display. The second part covers the practical implementation of the software as well as the physical realization of an attenuation display with backlight. The developed software provides features that assist the user with importing a light field, finding optimal display parameters, simulating the projected images and finally, printing the attenuation masks on transparencies.

Acknowledgements

Special thanks goes to Prof. Dr. Matthias Zwicker for his supervision and support of this thesis and for giving me the opportunity to work together with the Computer Graphics Group at the University of Berne. A big thank you to secondary advisor Siavash Bigdeli for his time and effort to help me throughout the project. Lastly, I would like to thank my family, friends and colleagues for the great time I had with them during the time of my bachelor studies.

Contents

Chapter 1

Introduction

Over the last few years, devices capable of displaying 3D content have shown to become increasingly popular. Most of the moviegoers have long accustomed to the variety of movies releasing in 3D every year, and with affordable 3D television screens on the market, movies with the extra dimension can be enjoyed in the living room. It is obvious that for the viewer, the most important part of the display is the content shown by it. Current graphics processors together with state-of-the-art rendering algorithms bring the 3D experience to the video game consumer, allowing for a higher immersion into the virtual world. But there is also the desire to view real-world photos or videos on such a display, elevating the need for 3D capturing devices.

1.1 Light Fields

Light fields are the foundation for image based rendering, a technique to present different perspectives of a scene without the need to store geometry data, texture or lighting information. The light field is a simplified version of the more general plenoptic function, first characterized by Adelson and Bergen [1991]. It can be thought of as a snapshot of the light in the entire scene, a database storing the radiance for every possible ray in the scene. This data can be captured by a grid of cameras such as the one depicted in figure 1.1b. In fact, a light field is formed inside every conventional camera, but the per-ray radiance information is lost when the light striking the sensor is accumulated over all angles under the aperture. A camera that does not discard the additional radiance information is called a plenoptic camera or light field camera, shown in figure 1.1a. As described by Ng et al. [2005], an application for the plenoptic camera is digital refocusing, the process of refocusing an image after it was taken. Chapter 2 gives an overview of the properties of light fields and how they are used in this work.

The ideal 3D display should be able to display any light field, meaning it should emit light rays with radiance equal to the value in the database. It turns out it is not so easy to build these displays. On the one hand are physical challenges, such as the direction of light in different angles or the correct depiction of color, contrast and brightness. On the other hand comes a lot of data from the light field that needs to be processed, desirably in real-time and full reso-

(a) (b)

Figure 1.1: (a) Hand-held plenoptic camera from Lytro. Image courtesy D-Kuru, Wikimedia Commons. (b) The Stanford multi-camera array, holding 128 video cameras. Image courtesy Marc Levoy, Stanford Computer Graphics Laboratory, with permission.

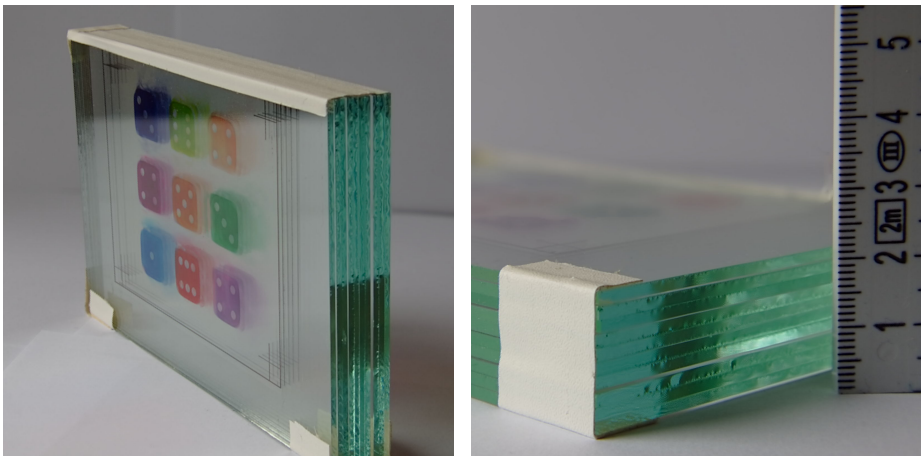
lution. Despite these challenges, many types of displays have been developed, although not necessarily based around light field technology, all having different trade-offs and limitations. There are two main categories, stereoscopic- and true 3D displays.

1.2 Stereoscopic Displays

Stereoscopic displays are based on the principles of binocular vision. The objective is to provide two distinct images to the human visual system, one for each eye, presenting the content from two slightly different perspectives. The disparities between the two images translate to depth cues in the human brain and allow for depth perception. The pair of images presented to the eyes remains constant when the viewer moves in front of the device. This effect distinguishes stereoscopic displays from 3D displays. Modern technologies include head-mounted displays, polarization systems, active shutter systems and autostereoscopy. Although not as comfortable to wear, head-mounted displays have separate high resolution screens for each eye allowing for a high degree of immersion. Polarization screens show the image pair superimposed with different polarization of the light, which is separated again by different polarization filters in the right and left side of the viewers eyeglasses. Active shutter systems use special eyeglasses that alternately block the light for one eye, letting the opposite eye see the corresponding image on the synchronized screen. Autostereoscopic displays present stereo content to the viewer without the need of special glasses. The technology is based on a lenticular lens or parallax barriers, which requires the viewer to be in a fixed and predefined position.

1.3 3D Displays

Real 3D displays ideally show the full 3D information to the observer. In contrast to stereoscopic displays, the person is able to move in front of the screen and view the content from a desired perspective. Present technologies include volumetric displays, holography, integral imaging and compressive light field



(a) (b)

Figure 1.2: Attenuation layers between glass plates. (a) Front view of the display. (b) Side view: Ten pieces of 2 mm thick glass plates hold the five layers of transparencies, with a 4 mm separation between them.

displays. Volumetric displays reproduce a physical volume emitting the light of virtual objects inside, allowing for a full 360 degree viewing angle. Holographic displays are based on conventional LCD panels equipped with a diffraction layer making it possible to project images in different directions in space. Integral imaging devices achieve the same result with a microlens array in front of the screen similar to lenticular lenses. Finally, compressive light field displays, also called tensor displays, consist of multiple LCD panels forming a stack of time multiplexed, light attenuating layers.

The work in this thesis is based on a much simpler version of these light field displays, called *Layered 3D*, which was first realized by Wetzstein et al. [2011]. An example of the final product is shown in figure 1.2. The depicted display is able to present a static, full 4D light field without the need of special glasses. It consists of masks, printed on transparencies, which attenuate light from a backlight in a multiplicative manner.

1.4 Related Work

This thesis was inspired by and builds upon the work of Wetzstein et al. [2011]. In their paper, they present a model for an inexpensive 3D display built from light attenuating, multiplicative masks. The simulated reconstructions they achieve are very convincing, although they only worked with synthetic light fields from oblique projections. The idea for this thesis was to extend their work in addition to support camera light fields, e.g. captured with a plenoptic camera or a camera array. With a conclusive spectral analysis, they show that multi-layer displays have increased depth of field and improved spatial resolution in comparison to other automultiscopic displays such as parallax barriers or integral imaging.

The Layered 3D architecture falls in the category of volumetric displays,

although the theoretical approach and physical realization is quite different from previous work. Blundell and Schwarz [2000] distinguish previous volumetric devices between swept- and static volume display units. Both types restrict the 3D content to lie within the physical enclosure of the device as opposed to light field displays. As a consequence, this type of display is not suited for most mobile applications. Also in both cases, the image is formed by the light emitting volume elements (voxels) in an additive manner in contrast to the multiplicative transport of Layered 3D, for example by temporally multiplexed presentation planes (Sullivan [2003]). Due to manufacturing reasons, it is challenging to produce high resolution voxel volumes to guarantee high image fidelity. The size of the voxels dictates the highest spatial frequency that can be depicted by the volume. As will be demonstrated, Layered 3D is capable of displaying higher frequency than allowed by the individual display mask inside the display.

As a followup to the original paper, Wetzstein et al. [2012] have built a display comprising a stack of liquid crystal panels that modulate the light similar to the static layered 3D for printed transparencies. To reduce artifacts from limited depth of field and to increase the displays field of view, they extend the degrees of freedom for the optimization by time-multiplexing a number of frames on each layer. As a result, the observer perceives a time-average over the multi-layer frame sequence. Their GPU-implementation solves the problem of finding the attenuation layers and frames by means of multiplicative update rules rather than tomographic reconstruction techniques, achieving interactive frame rates.

Multi-layer displays are not only suited for 3D presentation. Narain et al. [2015] use multi-plane displays to present focus cues instead of parallax. With their model for image formation inside the human eye, they describe the defocus effects produced in the eye when accommodating at different distances. They compute optimal presentation layers that create a focal stack for the viewer, which handles occlusion boundaries and specularity correctly. The proposed algorithm requires a series of images focused at different distances rather than a full 4D light field and thus requires less memory for computation. Contrary to this work, the display planes are additive and the optimization is performed in the frequency domain.

A different application for light field displays is vision correction. As shown by Huang et al. [2014], light field displays can be used to pre-distort imagery in order to compensate for visual aberrations in the observers eye and to obviate the need for eyeglasses or contact lenses. Although their physical prototype is based on parallax barriers, the concept applies for all light field displays and therefore it could also be adapted for layered 3D displays.

Chapter 2

Capturing a Light Field

2.1 The Plenoptic Function and the Light Field

The plenoptic function, as introduced by Adelson and Bergen [1991], is a 7D function that describes the intensity of light for every frequency, along every light ray in space, at any time. Formally, it is defined as

$$\begin{aligned} P\colon \mathbb{R}^3 \times [0, 2\pi) \times [0, \pi] \times \mathbb{R}^2 &\to \mathbb{R}^+ \\ (x, y, z, \theta, \phi, t, \lambda) &\mapsto P(x, y, z, \theta, \phi, t, \lambda), \end{aligned}$$

where the parameters (x, y, z) are the coordinates of a point in 3D space and the angles (θ, ϕ) describe the direction of an incoming light ray at time t. The light's intensity is given for every wavelength λ and thus, the plenoptic function not only captures the visible frequency spectrum but all electromagnetic waves. A commonly used measure for light is the radiance (power per area perpendicular to travel direction and per solid angle), which is obtained from P by integrating over all wavelengths: $R(x, y, z, \theta, \phi, t) = \int_{\mathbb{R}} P(x, y, z, \theta, \phi, t, \lambda)\, \mathrm{d}\lambda$.

In practice, it is impossible to acquire all the data needed to model the 7D plenoptic function and hence it is reasonable to consider only a subset of the parameters. Dropping the time parameter t in $R(x, y, z, \theta, \phi, t)$ yields a 5D function for the radiance in a static scene. As described by Levoy and Hanrahan [1996], this five dimensional representation can further be reduced to four dimensions in the following way. The radiance along a line is constant in free space and so, the 5D plenoptic function holds redundant information for the points on this line. Ignoring this redundancy leads to the equivalent 4D parameterization of the ray space. Levoy and Hanrahan [1996] propose a parameterization by two parallel planes, as seen in figure 2.1, where the coordinates of the lines (rays) are given by the intersections with the two planes. The **4D light field** $L(u, v, s, t)$ is therefore defined as the radiance along the line intersecting the two planes at coordinates (u, v) and (s, t). This two-plane parameterization of the light field is the most common one seen in literature, but there are many ways to choose a parameterization. For instance, one can use a plane and two angles to define each ray passing this plane, which would result in a light field $L(u, v, \theta, \phi)$, where $\theta, \phi \in (0, \pi)$.

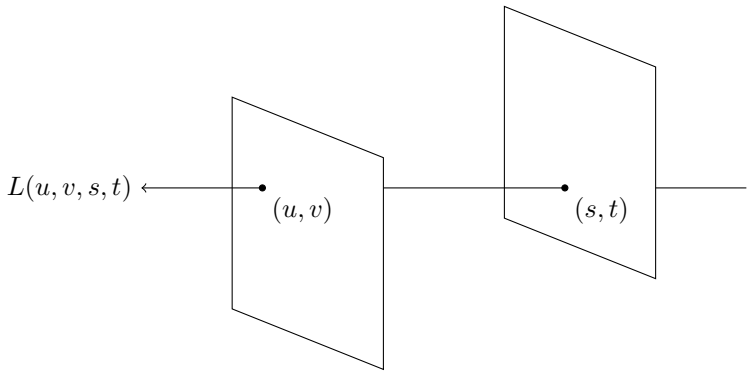


Figure 2.1: Parametrization of the light field with two planes.

2.2 Light Field Acquisition

For practical applications, the light field must be discretized and so, an appropriate sampling method needs to be chosen. This means that only a slice of the actual light field can be captured and the two planes are clipped to form rectangles. In this work, the term *light field* is used for both the infinite, continuous light field as well as the discrete collection of data samples.

Oblique Projection

Oblique projection, as shown in figure 2.2a, is a special case of orthographic projection: The parallel rays do not need to be perpendicular to the image plane of the camera. The advantage is that there is a one-to-one correspondence between camera position and ray angle, since all rays in one camera are parallel. This means that the angular resolution is simply the number of cameras, and the spatial resolution is the number of pixels in the image plane. The angular extent from $\theta_{\min}$ to $\theta_{\max}$ is called the **field of view** (FOV) of the light field and should not be confused with the field of view of a conventional camera. For a uniform angular sampling with resolution $N_\theta \times N_\phi$, the angles θ_i and ϕ_j are

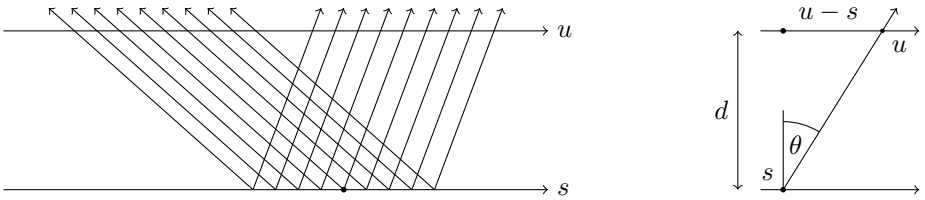
$$\theta_i = \theta_{\min} + (i-1)\frac{\mathrm{FOV}_\theta}{N_\theta - 1}, \qquad \phi_j = \phi_{\min} + (j-1)\frac{\mathrm{FOV}_\phi}{N_\phi - 1}, \tag{2.1}$$

where $i = 1, 2, \ldots, N_\theta$ and $j = 1, 2, \ldots, N_\phi$.

Given a light field $L(u, v, s, t)$ and the distance d between the two planes, a re-parameterization $L'(\theta, \phi, s, t)$ can be obtained according to figure 2.2b by the transformation

$$\theta = \arctan\left(\frac{u-s}{d}\right), \qquad \phi = \arctan\left(\frac{v-t}{d}\right). \tag{2.2}$$

Note that uniform sampling in angular dimension does not yield a uniform grid in the (u, v)-plane. Despite the simplicity of this projection type, it is not feasible to build cameras of this type and so, oblique projection is left to be used exclusively by computers for rendering synthetic scenes.



(a) (b)

Figure 2.2: (a) Light field acquisition using oblique projection. (b) Re-parameterization of the two-plane representation to angular coordinates.

Perspective Projection

Another way to capture the light field is with a grid of optical systems, e.g. cameras. Typically, the (u, v)-plane is sampled on a grid $G_{uv} = \{(u_i, v_j) \mid i = 1, \ldots, n, j = 1, \ldots, m\}$ on the (u, v)-plane with a resolution $n \times m$. The extent in horizontal (vertical) direction is called the horizontal (vertical) **baseline**. Although it is strictly speaking not correct, the resolution of the (u, v)-plane is often referred to as the angular resolution. The angles of the rays in a light field captured by perspective projections are determined by the focal length, the sensor size and the sensor resolution of the camera. For a camera light field, typically it is expected that

- All cameras are placed at grid positions in G_{uv} on the same plane, called the (u, v)-plane,
- The optical axes of the cameras are orthogonal to the (u, v)-plane,
- All cameras have the same intrinsic parameters (e.g. focal length).

In this case, the focal planes of all cameras coincide with a common focal plane, the (s, t)-plane. Figure 2.3a shows this scenario for three cameras in two dimensions. Given images $I_{uv}(x, y)$ with respect to a coordinate system centered at the camera position (u, v), the coordinates on the (s, t)-plane are $s = u + x$, and $t = v + y$. Thus, the light field in continuous coordinates is obtained by

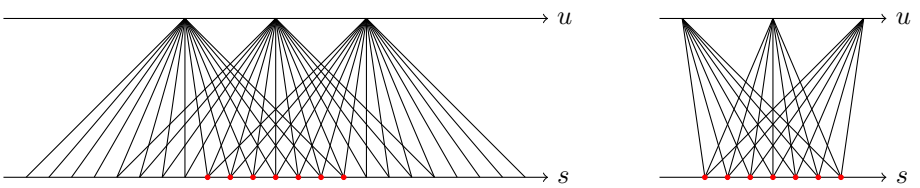
$$L(u, v, s, t) = L(u, v, u + x, v + y) = I_{uv}(x, y). \tag{2.3}$$

In the discrete case, each camera captures sample points on the (s, t)-plane, but not everyone of these sample points on the (s, t)-plane is captured by every camera. So, as demonstrated in figure 2.3b, the camera images need to be rectified such that all discrete coordinates (u, v, s, t) correspond to valid rays. This rectification process is equivalent to a re-parameterization L' of the continuous light field L, given by the formula

$$L'(u, v, s', t') = L\left(u, v, \gamma(s' - u) + u, \gamma(t' - v) + v\right), \tag{2.4}$$

where $\gamma = \frac{d}{d'}$ and d' is the distance between the (u, v)-plane and the new (s', t')-plane. As derived by Isaksen et al. [2000], this re-parameterization is equivalent to a 4D shear of the light field.

A different way to understand this coordinate change is to imagine the (u, v)- and (s, t)-plane being the aperture and sensor planes respectively, resulting in



(a) (b)

Figure 2.3: Perspective projections of a scene. (a) Projections with three pinhole cameras. (b) Discarding unused rays corresponds to cropping the camera images.

one big camera in which a light field is formed. Changing the distance between the two planes is now equivalent to changing the focal length of this one camera. The effect on the light field inside is similar to refocusing, except that in a conventional camera, the image on the sensor is formed by a weighted integral over u and v such that the angular information vanishes. Objects at focal distance from the camera would appear sharp and objects away from the focal point would become blurred. Section 2.4 discusses this type of camera in more detail.

From stereo vision, it is known that the displacement of the projections in the image planes of two cameras is only dependent on the focal length f, the baseline Δu and the distance z, and the relation is given by $\Delta x = f\Delta u/z$. This knowledge can directly be applied to the two-plane parameterization. For the continuous light field, it amounts to

$$\mathrm{d}s = \frac{z - Z_{st}}{z - Z_{uv}}\,\mathrm{d}u \qquad \text{and} \qquad \mathrm{d}t = \frac{z - Z_{st}}{z - Z_{uv}}\,\mathrm{d}v, \tag{2.5}$$

with Z_{uv} and Z_{st} denoting the placement of the (u, v)- and (s, t)-planes in Z-direction. Usually, the coordinate system is chosen such that $Z_{uv} = 0$. In the discrete case, the displacement Δs or Δt is also called the **disparity** and is often measured in pixel units.

2.3 Visualization

The epipolar-plane image (EPI) allows for a very intuitive visualization of depth from a 4D light field. It was first defined by Bolles et al. [1987] as follows. Consider a point P in 3D space and a pair of cameras with the optical axis pointing in the same direcion. The plane passing through P and the two centers of projection is called the **epipolar plane**. The epipolar plane projects to a line on each of the camera image planes, named the **epipolar line**. This line represents a constraint for the projection of P in each of the images and it is used to solve the correspondance problem in computer vision. The notion of epipolar lines can be directly applied to a multiple camera setup. In figure 2.4, a synthetic scene is rendered in 500 different positions along a horizontal baseline. Since the camera movement is in horizontal direction only, the epipolar lines correspond to a fixed pixel row in each image. The EPIs shown in figures 2.4b and 2.4c are created by collecting the chosen pixel row (scanline) in every image and stacking it up.

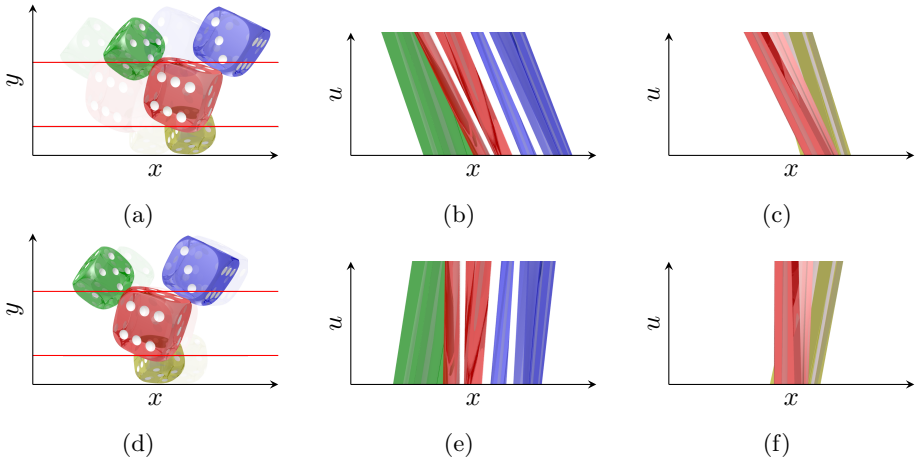
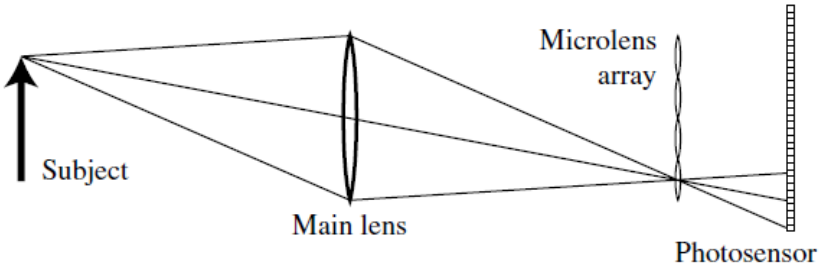


Figure 2.4: (a) Raw 3D light field rendered from 500 positions along a horizontal baseline. Two scanlines are extracted from every image. (b) The feature paths of the blue and green dice have a steeper slope than those of the red die. (c) Feature paths of the yellow die have an even steeper slope, indicating greater depth. (d) The light field is rectified according to figure 2.3b such that the disparities of the red die are approximately zero. (e) - (f) EPIs from the same scanlines. The slopes of the feature paths stay the same relative to each other.

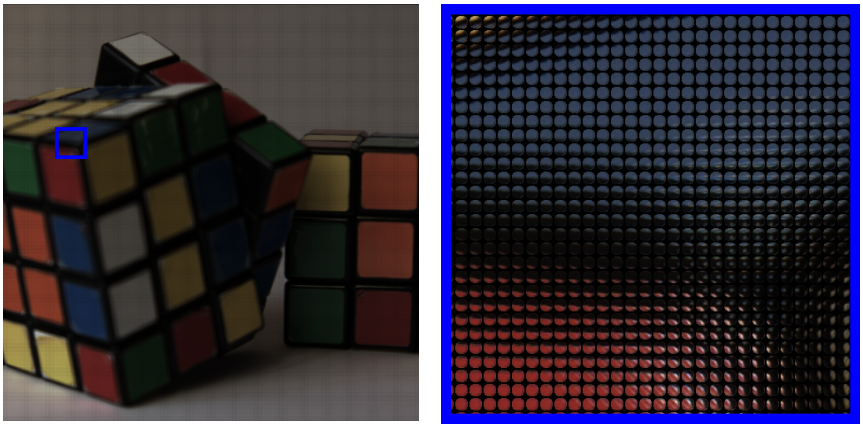
As described in the previous section, the depth component of P occurs as a displacement of the projections in consecutive images. Under the assumption that the (u, v)-plane is sampled uniformly, the disparity D with respect to P stays constant from one image to the next. Thus, following the projection of the point P in every image corresponds to a line in the EPI with a slope proportional to $1/D$. Bolles et al. [1987] refer to this line as the **feature path**. This means that points farther away from the camera will appear as a feature path in the EPI with steeper slope than points close to the camera. Note that the depth range in the light field can immediately be determined by identifying the maximum and minimum slope in the EPI. Also, for a perfectly Lambertian scene, each line in the EPI has a uniform color.

2.4 The Plenoptic Camera

The plenoptic camera as depicted in figure 1.1a is a hand-held device capable of capturing a light field similar to a camera array but with a much smaller baseline. The design of Lytro's camera is based on the work of Ng et al. [2005]. In principle, it functions like a conventional hand-held camera with the only major difference being that it has an array of microlenses placed in front of the image sensor as it is shown in figure 2.5a. A light ray entering the aperture and hitting a microlens is mapped to one of the sensor's pixels behind the microlens. By knowing the position of the microlens array, focal length of the lenses and size of the aperture, each pixel in the final image can be identified with the direction of the incident ray. Hence the plenoptic camera directly adopts the two-plane parameterization with the microlens-array being the (s, t)-plane and



(a) The layout of the plenoptic camera. The rays starting from the same point in the scene are mapped to different pixels on the photosensor via the microlens array. Image taken from Ng et al. [2005].



(b) Raw light field image captured by the Lytro Illum plenoptic camera. The circularly shaped patches are the images that form behind each microlens.

Figure 2.5: The inner workings of the plenoptic camera.

the main lens representing the (u, v)-plane. Figure 2.5b shows an enlarged part of the image formed on the sensor. It consists of rectangular patches corresponding to the images behind the microlenses. Because of the circular shape the main lens has, only a disk of pixels pictures the light coming from outside the camera. A popular application of the plenoptic camera is digital refocusing. After the light is captured by the camera, refocused photographs can be created by re-parameterization of the light field with equation 2.4 as described by Ng et al. [2005]. Although the plenoptic camera was specifically designed for digital refocusing, the captured light field can be used for 3D display like any other light field.

Chapter 3

Light Field Tomography

3.1 A Model for Light Attenuation

The light field display is modeled by a volumetric attenuator $\mu(x, y, z)$ that attenuates the light traveling through its material. According to the Beer-Lambert law, the intensity of a light ray $\mathcal{R} \subset \mathbb{R}^3$ passing through the material decreases exponentially over distance:

$$I = I_0 e^{-\int_{\mathcal{R}} \mu(r)\,\mathrm{d}r}. \tag{3.1}$$

The incident intensity I_0 is the intensity of the ray before it enters the attenuator. Equation 3.1 can be rewritten into

$$\bar{I} \coloneqq \log\left(\frac{I}{I_0}\right) = -\int_{\mathcal{R}} \mu(r)\,\mathrm{d}r. \tag{3.2}$$

Now, let the attenuator $\mu(x, y, z)$ be a cubic slab of height d in Z-direction and let $L(u, v, s, t)$ be the two-plane parameterization of the light field such that the (s, t)-plane coincides with the (x, y)-plane of the attenuator and the (u, v)-plane is at distance d. The set of points describing the ray defined by the coordinates (u, v, s, t) is

$$\mathcal{R} = \left\{ \lambda a + b \;\middle|\; a = \begin{pmatrix} u - s \\ v - t \\ d \end{pmatrix}, b = \begin{pmatrix} s \\ t \\ 0 \end{pmatrix}, \lambda \in \mathbb{R} \right\}. \tag{3.3}$$

A point $p = (x, y, z)^T$ is part of the ray $\mathcal{R}$ if and only if

$$\exists \lambda \in \mathbb{R} : p = \lambda a + b \qquad \Longleftrightarrow \qquad a \times (p - b) = 0, \tag{3.4}$$

where $\times$ denotes the cross product. Now, I can be replaced with the light field L and the right hand side of equation 3.2 can be written as an integral over $\mathbb{R}^3$:

$$\bar{L}(u, v, s, t) = -\int_{\mathbb{R}^3} \mu(p)\delta(a \times (p - b))\,\mathrm{d}p. \tag{3.5}$$

Here, δ denotes the Dirac delta function on $\mathbb{R}^3$ and μ is zero outside the boundaries of the slab. This means that the integrand is only non-zero for points on the ray with coordinates (u, v, s, t).

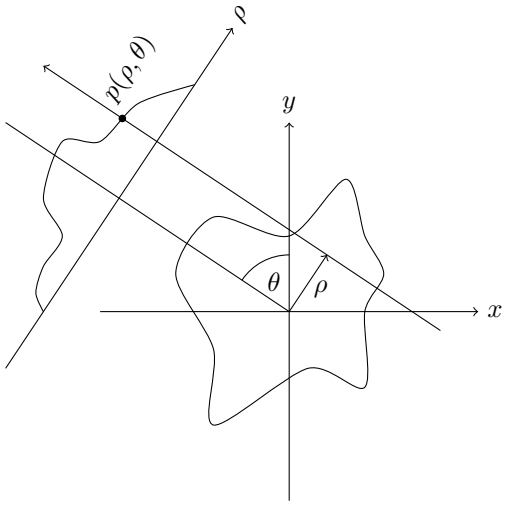


Figure 3.1: The 2D Radon transform of the ray (ρ, θ) passing a material with density $f(x, y)$.

Combining equation 3.1 and 3.5 gives the light field emitted by the attenuator. The goal is to produce such an attenuation display that emits a given target light field.

In computed tomography, the **Radon transform** of a real valued and compactly supported, continuous function $f(x, y)$ on $\mathbb{R}^2$ is defined as

$$p(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(x \cos\theta - y \sin\theta - \rho)\, \mathrm{d}x\, \mathrm{d}y, \tag{3.6}$$

where $(\rho, \theta) \in \mathbb{R} \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ defines a ray as shown in figure 3.1. Because the Radon transform is essentially a line integral, it can be generalized to three or more dimensions. Adapting the notation from the two-plane parameterization, the Radon transform of the attenuation map μ along ray $\mathcal{R}$ becomes

$$p(u, v, s, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y, z)\delta\left(a \times \left((x, y, z)^T - b\right)\right)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z, \tag{3.7}$$

which is equivalent to equation 3.5. This shows that

$$\bar{L}(u, v, s, t) = -p(u, v, s, t), \tag{3.8}$$

or with the words of Wetzstein et al. [2011]: "The logarithm of the emitted light field is equivalent to the negative Radon transform of the attenuation map."

3.2 Discrete Attenuation Layers

The previous section introduced a continuously varying attenuation map to model the display. Wetzstein et al. [2011] propose to represent the attenuator with a set of N two-dimensional layers, also called masks.

Let $L_{ijkl} = L(u(i), v(j), s(k), t(l))$ be the matrix of samples from the light field and for simplicity, let $m \coloneqq m(i, j, k, l)$ be a linear index of the 4D indices. Equation 3.1 suggests a per-ray constraint in the form

$$L_m = L_0 \prod_{n=1}^{N} t^{(n)}(h(m, n)), \tag{3.9}$$

where $h(m, n)$ is the (discrete) 2D coordinate of the intersection of the m-th ray with the n-th layer, and $t^{(n)}(\xi)$ is the **transmittance** of layer n at that coordinate. Having a constraint for each ray, the goal is to solve for the transmittance t. However, the system of equations in 3.9 is non-linear and cannot directly be solved. One can obtain a linear system of equations by taking the logarithm in 3.9:

$$\bar{L}_m = \sum_{n=1}^{N} \log\left(t^{(n)}(h(m, n))\right) = -\sum_{n=1}^{N} a^{(n)}(h(m, n)) = -P_m\alpha. \tag{3.10}$$

Here, $a^{(n)} \coloneqq -\log t^{(n)}$ denotes the **absorbance** of layer n. This relation between transmittance and absorbance also directly follows from the Beer-Lambert law. Here, $P_m = \left(P_m^{(1)}, \ldots, P_m^{(N)}\right)$ is a binary row vector, encoding the intersection of the ray with the pixels on each layer. The unknown absorbance is represented by the column vector $\alpha = \left(\alpha^{(1)}, \ldots, \alpha^{(N)}\right)^T$. Each $\alpha^{(i)}$ is just a flattened representation of the absorbance matrix $a^{(i)}$. Note that equation 3.10 is the equivalent of the continuous version in 3.8, since P_m encodes the Radon transform. Finally, the above equations indexed by m can be combined into one large linear system $P\alpha = -\bar{L}$.

In most cases, P is not a square matrix and the system can become overdetermined, which means that it has no solution in general. However, it is still possible to find values for α such that the error $\left\|P\alpha + \bar{L}\right\|$ is small. Thus, the objective becomes

$$\begin{aligned} &\underset{\alpha}{\text{argmin}} && \left\|P\alpha + \bar{L}\right\|^2 \\ &\text{subject to} && \alpha \geq 0. \end{aligned} \tag{3.11}$$

Finally, when optimal values α are found, the transmittance used to fabricate the layers is obtained by calculating $e^{-\alpha}$. Also note that the matrix P is very sparse because it is assumed that a ray passes through each layer at exactly one pixel no more than once and inter-reflections between the layers are not supported by the model. Thus, P can be efficiently stored using an appropriate data structure.

3.3 Ray Casting

To obtain the linear system P, the intersections between the rays and the attenuation layers have to be calculated. This calculation depends on the parameterization of the light field. For continuous light fields, it is always possible to apply a re-parameterization to get the desired representation (e.g. the two-plane parameterization) and then compute the intersection in a standard way. For discrete light fields however, this would require a suitable interpolation in

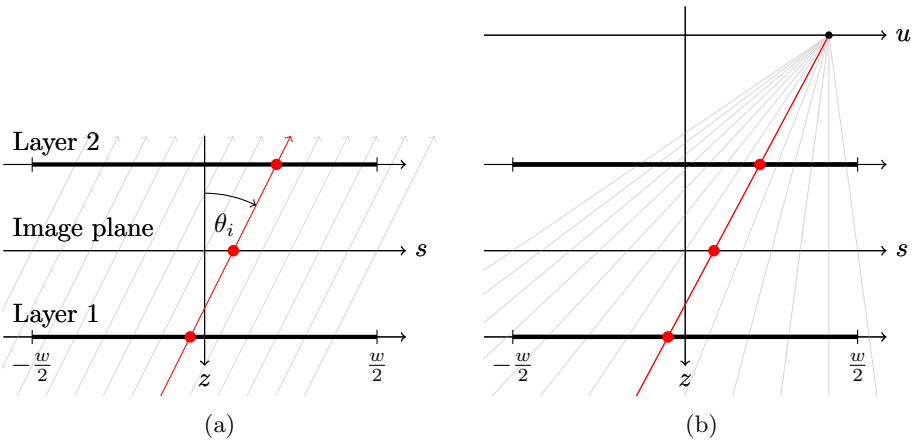


Figure 3.2: Computation of the ray-layer intersections from oblique (a) and perspective (b) projections. Two attenuation layers are drawn (top and bottom) with the virtual image plane in the center. Light rays intersect the layers at positions to be calculated.

ray-space, which gives poor results when the distribution of samples in the target space becomes too sparse.

What follows is a description of two methods to compute the indices for the non-zero entries in P. For simplicity, only a two-dimensional attenuator of size w is assumed, consisting of N layers at various depths $Z_{\min} = z_1 < z_2 < \cdots < z_N = Z_{\max}$. It is also assumed that the image plane s of the virtual cameras is bisecting the attenuator in the middle, at depth $Z_s = \frac{Z_{\max} - Z_{\min}}{2}$.

Oblique Projection

The setup for the oblique projection type is illustrated in figure 3.2a. Let θ_i denote the angle of the i-th oblique view from the light field. Following the notation in previous sections, the linear index $m = m(i, k)$ identifies the ray $(\theta_i, s(k))$. The intersection of ray m with the n-th layer is simply

$$h(m, n) = s(k) + \Delta z \tan(\theta_i), \tag{3.12}$$

where $\Delta z = Z_s - z_n$ is the displacement of layer n from the image plane. The next step is to compute the pixel index at the point $h(m, n)$. The shift in pixel units can directly be derived from the shift in equation 3.12 given the pixel size Δs, yielding

$$\Delta k = \left[\frac{\Delta z \tan(\theta_i)}{\Delta s} \right], \tag{3.13}$$

and the new index is $k' = k + \Delta k$. The brackets in the above equation denote the rounding operation. Finally, this information is stored in the propagation matrix with the assignment $P^{(n)}_{mk'} = 1$.

Perspective Projection

For the perspective projection, it is assumed that the (u, v)-plane (or u-plane here) is at depth Z_u, with a distance $Z_s - Z_u$ from the image plane. Again,

let $m = m(i,k)$ be the index that identifies the ray $(u(i), s(k))$. Finding the coordinates of the ray (u,s) on a layer is similar to the re-parameterization in equation 2.4. Setting $\gamma = \frac{Z_s - Z_u}{z_n - Z_u}$ for layer n, the formula for the intersection is

$$h(m,n) = \gamma(s(k) - u(i)) + u(i). \tag{3.14}$$

As before, the new pixel index $k' = k + \Delta k$ is computed from the shift in the number of pixels

$$\Delta k = \left[\frac{h(m,n) - s(k)}{\Delta s} \right], \tag{3.15}$$

and the assignment $P^{(n)}_{mk'} = 1$ is made. Although these equations are straightforward to implement, there are some improvements that can be made. These and the challenges that come with them are discussed in section 5.3.

3.4 Iterative Reconstruction

The optimization problem in equation 3.11 is essentially a fitting problem. Theoretically, the unconstrained problem can be solved in a least squares sense using the normal equation $P^T P\alpha = P^T \bar{L}$ and by inverting the matrix $P^T P$. For high resolution light fields, the matrix P becomes extremely large and it is unfeasible to compute the inverse of $P^T P$. And because the problem in 3.11 is constrained by the inequality $\alpha \geq 0$, a different strategy is needed anyhow. In general, the approach to solve these kind of problems is to use iterative methods. The choice of the method depends on the type of problem and the structure of the design matrix. In the following, the two iterative solvers used in this work are presented and later compared in section 5.7 in terms of reconstruction quality.

Simultaneous Algebraic Reconstruction Technique

In computed tomography, a variety of iterative solvers have been developed to solve the exact same problem. Among the different methods is the Simultaneous Algebraic Reconstruction Technique (SART) first proposed by Andersen and Kak [1984]. The update rule of SART for iteration $k = 0, 1, 2, \ldots$ is

$$\alpha^{(k+1)} = \alpha^{(k)} + \lambda C P^T R \left(\bar{L} - P\alpha^{(k)} \right), \tag{3.16}$$

where λ is a relaxation factor. R and C denote the diagonal matrices with entries $R_{ii} = \frac{1}{r_i}$ and $C_{ii} = \frac{1}{c_i}$, where r_i and c_i are the sum of the elements in the i-th row and column of P respectively. To take care of the constraint $0 \leq \alpha < \infty$, the values $\alpha^{(k)}$ are simply clamped to this range in each iteration. The parts in 3.16 involving P and P^T are also referred to as the **forward-** and **back-projection** respectively. The convergence of SART has been studied by Jiang and Wang [2001]. They have proven that it converges to a weighted least squares solution.

Trust-Region-Reflective Least Squares

With the function *lsqlin*, MATLAB provides algorithms to solve constrained linear least squares problems. The concrete algorithm it uses to solve 3.11 is

named Trust-Region-Reflective Least Squares. To summarize the official documentation[1], the algorithm performs the following steps.

The function to minimize is $f(\alpha) = \|P\alpha + \bar{L}\|^2$. Let $\alpha^{(k-1)}$ be the result of the previous iteration. The goal for the current iteration is to find $\alpha^{(k)}$ in the neighborhood of $\alpha^{(k-1)}$ for which f improves:

1. Approximate f in the neighborhood N (the trust region) with a simpler function q.
2. Compute a step s by minimizing $q(s)$ in that region.
3. If $f(\alpha^{(k-1)} + s) < f(\alpha^{(k-1)})$, set $\alpha^{(k)} := \alpha^{(k-1)} + s$ and the iteration is completed. Otherwise, shrink the region N and repeat from 2.

The incorporation of constraints is a bit more complicated, but the underlying idea is the same. In the standard case, q is defined by the first two terms of the Taylor approximation.

[1]The official documentation can be found here: `http://goo.gl/XfhhaR`

Chapter 4

Spectral Analysis

This chapter is intended to give an overview of the spectral properties and limitations specific to multiplicative light field displays. Spectral analysis is a crucial method for the quality assessment and it is the origin of a comprehensive understanding of 3D displays. A light field emitted by the display can be interpreted as a signal that is composed of sine waves with different amplitude, phase and frequency. Section 4.1 introduces the Fourier transform, an operation that decomposes such a signal into the frequencies that produce it. The spectral support, i.e. the range of frequencies the display is able to produce, is analyzed in section 4.3.

4.1 Definitions

The **Fourier transform** $\widehat{f}$ of an integrable function $f\colon \mathbb{R}^n \to \mathbb{C}$ is defined as

$$\widehat{f}(\xi) = \mathcal{F}(f)(\xi) \coloneqq \int_{\mathbb{R}^n} f(x) e^{-2\pi \mathrm{i} x \cdot \xi} \,\mathrm{d}x \tag{4.1}$$

for any $\xi \in \mathbb{R}^n$. According to the Fourier integral theorem, if both f and $\widehat{f}$ are absolutely integrable and f is continuous, then the inverse transform

$$f(x) = \mathcal{F}^{-1}(\widehat{f})(x) \coloneqq \int_{\mathbb{R}^n} \widehat{f}(\xi) e^{2\pi \mathrm{i} x \cdot \xi} \,\mathrm{d}\xi \tag{4.2}$$

is well-defined. The domain of f is called the **spatial domain** and the domain of $\widehat{f}$ is referred to as the **frequency domain**. An important property of the Fourier transform is that a convolution in the spatial domain becomes a multiplication in the frequency domain, or in other words,

$$\widehat{(f * g)}(\xi) = \widehat{f}(\xi) \cdot \widehat{g}(\xi) \tag{4.3}$$

for integrable functions $f, g\colon \mathbb{R}^n \to \mathbb{C}$. On the other hand, a multiplication in the spatial domain becomes a convolution in the frequency domain after applying the Fourier transform, that is

$$\widehat{(f \cdot g)}(\xi) = (\widehat{f} * \widehat{g})(\xi). \tag{4.4}$$

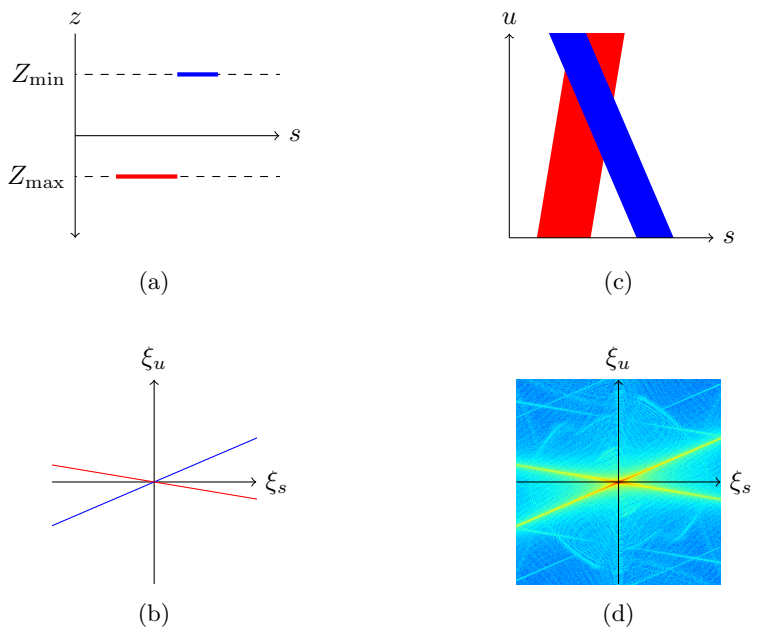


Figure 4.1: (a) Two objects (red and blue) placed at the bounds of the depth range. (b) The EPI representing the 2D light field of the scene. (c) Fourier transform of the EPI. The red and blue line mark the bounds for the spectral support. (d) Discrete Fourier transform of the EPI. Absolute values (magnitude response) are presented with colors on a logarithmic scale. Blue indicates small magnitude and red color indicates high magnitude.

These two properties of the Fourier transform are known as the convolution theorem. The Fourier transform and its inverse can also be discretized so that the convolution theorem still holds. Hence, for the following analysis it is immaterial which form of the Fourier transform is used.

4.2 Spectral Support of Light Fields

Consider a scene with a bounded depth range between $Z_{\min}$ and $Z_{\max}$. The two objects at the boundaries are shown in figure 4.1a, with the virtual image s plane between them. The consequent 2D light field $L(u, s)$ (or EPI) is depicted in figure 4.1c. From equation 2.5 it follows that objects appear in the EPI with a slope $\frac{\mathrm{d}u}{\mathrm{d}s} = \frac{z-Z_u}{z-Z_s}$. Substituting z with $Z_{\min}$ and $Z_{\max}$ gives the slopes for the red and blue objects at the boundary, defining the range of slopes in the EPI for objects between the two.

Applying the Fourier transform to the continuous light field reveals that the frequency response is non-zero on lines $\frac{\mathrm{d}s}{\mathrm{d}u}\xi_s + \xi_u = 0$. Again, for the scene with bounded depth range, this yields two lines representing the limits of the spectral support as shown in figure 4.1b. Objects between the red and blue ones will also have a frequency response within the fan spanned by the two lines. Therefore, the region of support for a continuous light field with bounded depth range can

be defined in the following way.

$$\mathcal{S}(\xi_u, \xi_s) \coloneqq \begin{cases} 1, & \text{if } Z_{\min} \leq \dfrac{Z_u \xi_u + Z_s \xi_s}{\xi_u + \xi_s} \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

A similar expression follows for the 4D light field, defining a 4D hyperfan for the region of support as derived by Dansereau et al. [2015]. Note that occlusions as well as specular reflections are not incorporated in the above expression. These effects introduce additional discontinuities in the EPI that result in a high frequency response possibly outside the fan defined in equation 4.5.

In the case of sampled light fields, aliasing can occur due to a small sampling rate in either angular- or spatial direction. Chai et al. [2000] analytically derived the minimum sampling rate required for alias-free light field rendering and proposed a reconstruction filter from known depth boundaries. The region of support $\mathcal{S}(\xi_u, \xi_s)$ can also be thought of as an ideal filter. As equation 4.3 shows, multiplying $\mathcal{S}(\xi_u, \xi_s)$ in the frequency domain is equivalent to a convolution in the spatial domain.

4.3 Spectral Support of Layered 3D Displays

With light field displays, it is of course desirable to achieve the same spectral coverage for the emitted light field as for the original. Again, the analysis starts with the assumption of a continuous light field and an attenuator with N continuously varying layers. Each layer by itself creates a light field, and since the layer is at constant depth, the frequency response is non-zero along a slanted line as demonstrated before. Let $L_1, \ldots, L_N$ denote the constant depth light fields per layer and let's assume all are parameterized with respect to the same (u, v)- and (s, t)-plane. The light field produced by all layers together is $L' = L_0 \cdot L_1 \cdots L_N$, where L_0 is the uniform illumination from the backlight. This directly follows from equation 3.9. With the multiplication theorem from equation 4.4, the Fourier transform of L' can be expressed as

$$\widehat{L'}(\xi) = (\widehat{L_0} * \widehat{L_1} * \cdots * \widehat{L_N})(\xi), \tag{4.6}$$

where $\xi = (\xi_u, \xi_v, \xi_s, \xi_t)$, or $\xi = (\xi_u, \xi_s)$ for the two dimensional case. For the case of discretely sampled layers, the frequency support of the individual layer will be limited by its spatial cutoff frequency, that is the highest frequency it can produce with a given pixel size p. A signal with a period smaller than two pixels can not be reproduced by the layer's pixel grid and thus, the spatial cutoff frequency is defined as $\xi_0 = \frac{1}{2p}$ cycles/m. The sketch in figure 4.2a illustrates this for the case of three layers that are bandlimited by $\pm\xi_0$. The three lines convolved produce a diamond shaped region of support as shown in figure 4.2b, which is the effective spectral support of the display. This means that a light field with high frequencies outside the spectral support of the display will not be correctly displayed, or in other words, the display acts as a low-pass filter.

The **depth of field** of an automultiscopic display, as explained by Zwicker et al. [2006], is the depth range that can be reproduced by the display in full spatial resolution. Thus, the boundary of the spectral support describes an

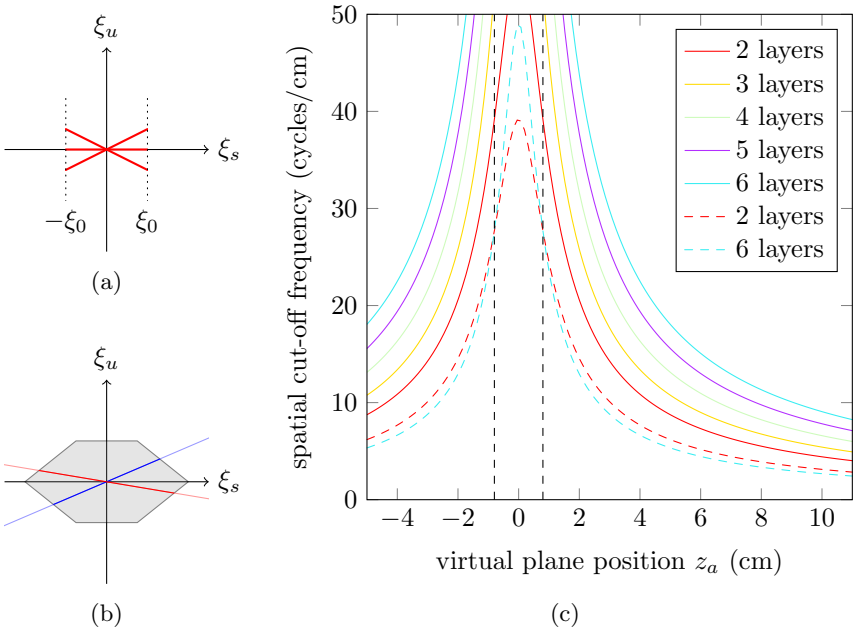
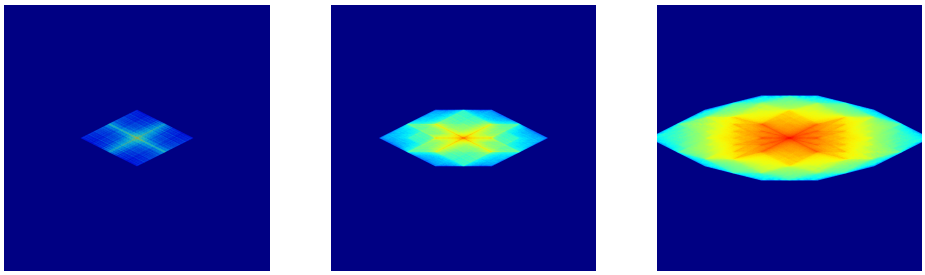


Figure 4.2: Spectral analysis for layered 3D displays. (a) Spectral support of individual layers (red) from a display with three layers, superimposed in the same frequency domain. The dashed lines mark the spatial cutoff frequency ξ_0. (b) Combined spectral support of all three layers (gray), obtained by the convolution. The light field from figure 4.1 can be displayed with frequencies within the region of support. (c) Approximate upper bound on the depth of field for layered 3D displays with different a number of layers (Wetzstein et al. [2011]) and constant display thickness. The *expected* upper bound is shown as dashed lines for two and six layers. The displays extent is depicted by the vertical dashed lines.

upper bound on the depth of field for any automultiscopic display, including layered displays. It turns out to be quite hard to analytically derive an exact expression for the upper bound. Wetzstein et al. [2011] present a statistical approach and give an approximation for the upper bound on the depth of field $|\xi_a|$ for a plane a placed at depth z_a from a N-layer display with a thickness $h = z_N - z_1$:

$$|\xi_a| \leq N\xi_0 \sqrt{\frac{(N+1)h^2}{(N+1)h^2 + 12(N-1)(z_a - Z_s)^2}} \tag{4.7}$$

This approximation is based on the observation that the region of support approaches the shape of an ellipse when increasing the number of layers as seen in figure 4.3. The right side of the equation is plotted in figure 4.2c for different positions z_a of the virtual plane and with $Z_s = 0$. It shows that the spatial cut-off frequency drops rapidly when moving the virtual plane away from the display. In fact, the drop-off is inversely proportional to z_a as equation 4.7 shows. As figure 4.2b suggests, the display is theoretically able to produce spatial frequency that exceeds the layers cut-off ξ_0, even for content outside the



(a) (b) (c)

Figure 4.3: Spectral support of layered 3D displays. The magnitude response is plotted on a logarithmic scale for a two (a), three (b) and five layer (c) display.

display enclosure. The highest spatial frequency is achieved in the middle of the display ($z_a = Z_s$), bounded by $N\xi_0$ cycles/cm as it can be deducted from equation 4.7.

Although this theoretical upper bound points out the limits on achievable depth of field, it is only a good reference for the ideal display and a high number of layers. In practice, the upper bound can not be achieved in most cases due a number of reasons, including: Simplifications in the attenuation model, approximate solutions to equation 3.11, pixel diffraction or the restriction to positive transmission values. Wetzstein et al. [2011] also give a more conservative expression that more closely qualifies the behavior under the mentioned restrictions:

$$|\xi_a| \leq \xi_0 \sqrt{\frac{2(2N-1)h^2}{(N+1)h^2 + 12(N-1)(z_a - Z_s)^2}}. \tag{4.8}$$

This is the expected upper bound on the depth of field, which is drawn as a dashed line in figure 4.2c for a two- and six layer display. In particular, it shows that adding more layers to the display alone does not necessarily increase the potential for higher depth of field. As Wetzstein et al. [2011] explain in their supplement to the paper, the derivation of this inequality is based on the assumption that the display designer does not attempt to show content on the outer layer with spatial frequency higher than ξ_0. This is where the two dashed lines in figure 4.2c intersect the display boundary ($z_a = \frac{h}{2}$). In other words, it is assumed that the input light field has a depth of field equal to the displays thickness h, which means that content outside this range is already blurred. Since the expected upper bound for six layers is lower than for two, one may conclude that a two layer display is optimal. But as will be discussed in sections 5.7 and 5.10, the addition of more layers has other benefits such as increased reconstruction quality. What can be concluded though is that for the best display performance, not only the number of layers but also the optimal thickness of the attenuator needs to be found for each new light field. The latter may not be possible if the display is built in a way that the thickness cannot be changed dynamically.

Chapter 5

Implementation and Assessment

All algorithms discussed in this work are realized in MATLAB, a programming language specialized on matrix manipulations. It is especially useful for high-dimensional matrix operations, such as the ones for this work. The following text explains the implementation of the theoretical models and concepts discussed in the previous chapters. For the implementation, the goal was to make the pipeline as flexible as possible to support all kinds of light fields in different formats.

5.1 Requirements

For the implementation and physical realization, some assumptions and requirements have to be formulated. The input light field for the optimization algorithm is expected to be a five dimensional array L with entries L_{ijklc}, where pairs (i, j) and (k, l) correspond to the angular- and spatial coordinates, and c indexes the color channel. The data is normalized such that $L_{ijklc} \in [0, 1]$. It is also assumed that the light field is rectified such that indices i, j conform to global coordinates on the (s, t)-plane as explained in section 2.2. In addition, the baseline as well as the distance between the two planes are a required input for the system.

The attenuator is defined by the number of layers, resolution, size and thickness. Each layer has the same dimensions and resolution and is modeled to be infinitely thin. Also, the backlight is modeled as a constant white light field, $L_0 \equiv 1$.

5.2 The Basic Procedure

As described in section 3.3, the two virtual planes that parameterize the light field are placed relative to the attenuator and by ray casting, the entries of the propagation matrix P are computed. Next, the constrained optimization problem given in equation 3.11 is solved independently for each color channel using an iterative solver of choice, e.g. SART. The outcome of each optimization

is a vector α_c containing the attenuation values in the interval $[0, \infty)$ where $c = 1, 2, 3$ (red, green, blue) denotes the color channel. The transmittance values are then obtained by element-wise exponentiation, $t = \exp(-\alpha)$, which holds values between zero and one. Finally, the linearly indexed vector t is reshaped so that the layers can be extracted as three-dimensional matrices and printed on transparencies.

In order to evaluate the attenuation masks, one has to compare the emitted light field $L^* = \exp(-P\alpha)$ with the original, L. For instance, one could evaluate the squared 2-norm of the difference $L - L^*$ in each color channel. However, this resulting number is not very meaningful because it also varies with the size of the light field, i.e. the angular and spatial resolution. This makes it harder to compare the display quality between different light fields. Thus, it is better to use a normalized figure such as the mean squared error (MSE) or the root-mean-square error (RMSE) defined as

$$\mathrm{MSE} \coloneqq \frac{1}{n}\sum_{i=1}^{n}(X - X^*)^2 = \frac{1}{n}\|X - X^*\|^2 \quad \text{and} \quad \mathrm{RMSE} \coloneqq \sqrt{\mathrm{MSE}}$$

for vectors $X, X^* \in \mathbb{R}^n$. To deal with the color channels, the MSE is computed for each color component and then averaged. Another important quality measure in signal processing is the peak signal-to-noise ratio (PSNR) defined as

$$\mathrm{PSNR} \coloneqq 10 \log_{10}\left(\frac{X_{\max}^2}{\mathrm{MSE}}\right),$$

where $X_{\max}$ is the maximum intensity value the image or color channel can hold, e.g. for 8-bit color channel representation this value would be 255. It measures the loss of power in a reconstructed signal, with an additional adjustment for the human perception of error. The use of this measure is motivated by the fact that Layered 3D is also about signal reconstruction, and the goal is to compare the original signal (light field) with the reconstructed one. As opposed to the MSE, it is easier to compare the PSNR of different images because it does not depend on the pixels intensity scale given by the bit depth. Generally, a higher PSNR indicates greater image fidelity and conversely, a highly corrupted signal produces low PSNR.

The following sections present a variety of ideas and improvements that were implemented on top of the standard procedure explained above. Next to that, the results are evaluated using the introduced error measures.

5.3 Challenges with Ray Casting

In section 3.3 it was explained how to compute the pixel indices on the layers for a ray that passes the attenuator. But it is not entirely obvious how to sample the rays in the first place. One of the simplest methods is to cast one ray per image pixel for each of the virtual cameras as it is illustrated in figure 5.1a. The sketch shows a one-dimensional pinhole camera with three pixels of equal width and the associated projection volumes. Because of the perspective projection it is possible that multiple pixel elements on the layers project to a single pixel on the image plane ((s, t)-plane). If the strategy is to only cast one ray per image pixel, many layer pixels inside the respective projection volume are missed and

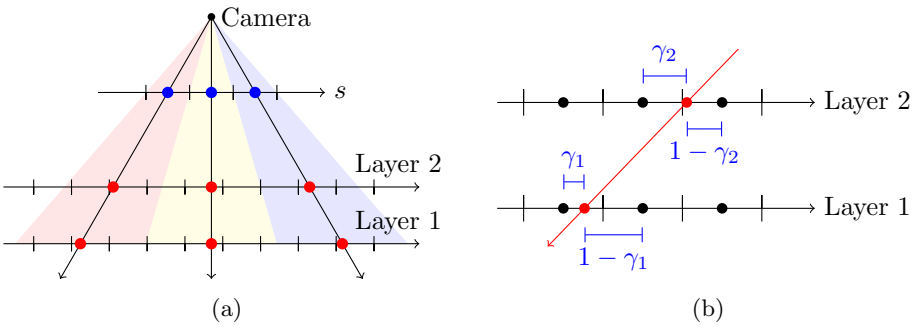


Figure 5.1: Intersecting light rays with attenuation layers. (a) Alternative ray casting method. The rays are cast from the camera center through the pixel centers (blue) and intersect with the pixels on the layers (red). The colored areas indicate the projection volume of an individual pixel on the image plane. (b) Linear interpolation of the intersections between adjacent pixel centers marked by the black dots.

thereby left unconstrained. But precisely because they lie in the same projection volume, the missed pixels are equally important and should be considered when setting up the linear system of equations. Note that although the proportions in the illustration are different for a real scenario, the problem exists in any case and the severity of it mainly depends on the resolution and placement of the layers.

Based on this observation, it seems natural to sample more than one ray per camera pixel. But how many rays are necessary? Again because of the perspective projection, a uniform sampling of the image plane does not imply uniform distribution of intersections on the layers. It stands to reason that the rays should be sampled from a plane near the attenuation layers such that the intersections are approximately uniformly distributed. Thus, the final method implemented in this thesis introduces a new virtual plane that is parallel to the attenuation layers but can be placed at an arbitrary distance. Placing this so called **sampling plane** at the same location as the bottom-most layer (farthest from the camera) ensures that the ray-density is no smaller on the other layers. Because the rays that are cast from the sampling plane to the image plane are different from the original rays, the light field is resampled to a higher resolution which also requires interpolation. On top of that, the sampling plane can have a different resolution than the attenuation layers and increasing it results in an increased ray density. If this density is higher than the resolution of the input light field, the outcome is a oversampled light field that requires more memory. Because one ray is equivalent to a constraint in the system of equations, the resolution of the sampling plane directly affects the number of equations.

5.4 Interpolation

In addition to nearest neighbor interpolation indicated in equation 3.15, bilinear interpolation is incorporated into the pipeline in order to reduce artifacts. For every ray, the value of the intersection point on the attenuation layer is

represented as a linearly weighted sum of four neighboring pixels corresponding to the rounded indices. Figure 5.1b shows how the weights for linear interpolation are computed in the one-dimensional case. For each intersection, there are two nearest pixels that represent the unknown attenuation values. The weights are deduced simply based on the distance between intersection point and pixel centers. This yields two weights γ_n and $1-\gamma_n$ for each layer n, or four weights in the case of two-dimensional attenuation layers. Accordingly, instead of storing N binary weights, each row of the propagation matrix now holds $4N$ values. Essentially, this just means that each ray in the log-light field is a linear combination of $4N$ absorbance values. Due to the exponential propagation model, the interpolation in the log-domain translates to a multiplication of the transmittance simply by the exponential law,

$$\exp(\gamma\alpha_i + (1-\gamma)\alpha_j) = \exp(\gamma\alpha_i)\exp((1-\gamma)\alpha_j) = t_i^{\gamma} t_j^{1-\gamma},$$

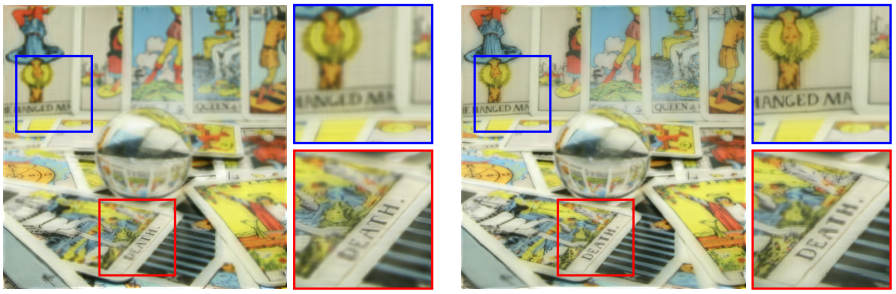
with $\gamma := \gamma_n$ being the linear interpolation weight, i and j identifying the neighboring pixels which are the query points for the interpolation.

The attempt of interpolation does not exhibit a significant increase in reconstruction quality. Alternatively, interpolation could be applied to the transmittance instead of the absorbance, but this requires a different optimization strategy. Because of the non-linearity of the logarithm, the optimization problem can not simply be transformed into a linear equation.

In general, it is not entirely clear which interpolation strategy one should choose for such a problem. The choice depends on a variety of physical factors as well as implementation specific restrictions. For example, an ink-jet printer produces slightly different attenuation masks than a laser printer. One could estimate the point spread function (PSF) of the printer by observing the distribution of a single ink drop on the surface. The interpolation weights should then be chosen according to the PSF. This approach would take into account the very physical aspects of printing. However, an accurate validation of this idea would introduce a lot of challenges because it requires accurate calibration and fabrication in order to compare the input light field with the displayed light field: The measuring process would require two iterations, one with PSF interpolation and one without. The attenuation layers need to be precisely aligned and placed in front of the capturing device (camera array, plenoptic camera etc.) such that the emitted light rays directly correspond to the original light rays.

5.5 Baseline Scaling and Back Projection

Often it is the case that the light field of interest has a depth range that does not match the depth of field of the display because of its fixed thickness. As a consequence, objects outside the depth of field appear blurry as demonstrated in figure 5.2a The reconstruction shows that cards in the front and back are blurred because they are virtually further away from the display, while the objects in the center are sharper. This problem can be solved by virtually scaling the baseline while keeping all other distances the same. The effect is that depth is compressed and hence objects appear to be squeezed in Z-direction. But the question is: How should one chose the scale such that a desired range is sharp? It is possible to solve this problem analytically, for example by using equation 4.7 or 4.8. But often it is the case that the exact baseline is unknown,



(a) (b)

Figure 5.2: Depth compression by means of baseline scaling. Shown is the reconstructed central angular view without proper baseline scaling (a) and with scaling (b). The light field used here is from the Stanford light field archive, `http://lightfield.stanford.edu`.

as it is for nearly all light fields used in this work. The alternative, more visual way, is to back-project the light field to the layers by trial-and-error using the propagation matrix P for a guessed baseline. Although the propagation matrix must be computed for every trial, the back projection itself is fast because it is simply a matrix multiplication,

$$\beta = P^T L.$$

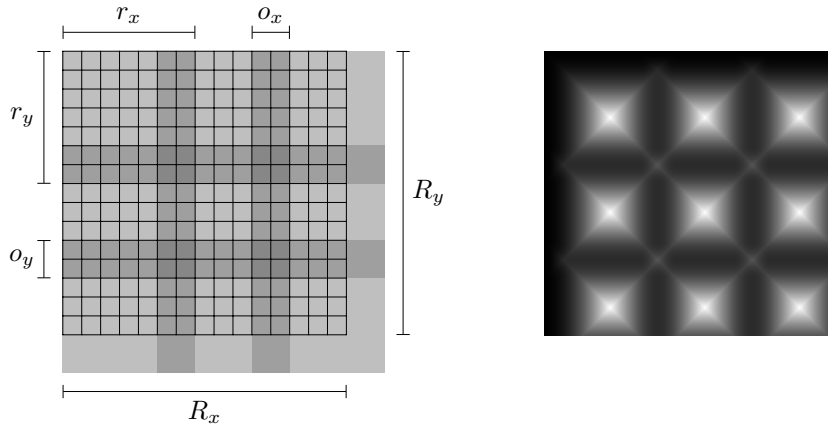
The outcome of this operation is a vector β holding the values of N refocused images where N is the number of layers used. In this way, one can control the depth compression by observing the focused parts of the light field in the top- and bottommost layer. This method can also be used to align the display center with the center of the scene, or an arbitrary position if desired.

5.6 Attenuator Tiling and Blending

High resolution light fields can take up a significant amount of space in memory. For example, a light field taken with a Full HD camera from 17×17 angles would take up $1920 \cdot 1080 \cdot 17^2 \cdot 3 \cdot 8/(1024^3) = 13.3947$ Gigabyte of memory, assuming 8-bit color channels. In addition, the propagation matrix stores information about every pixel in the light field and thus, can take up Gigabytes of space depending on the resolution of the attenuation layers. The proposed approach divides the attenuation layers into tiles. Figure 5.3a shows how the tiles are laid out. The inputs for the tiling algorithm are the resolution of the tiles $r = (r_x, r_y)$ and the overlap in horizontal and vertical direction, $o = (o_x, o_y)$. The tiles are then laid out in a grid beginning in the top left corner of the layer. The number of tiles needed to cover the plane can be calculated by

$$N_x = \left\lceil \frac{R_x - o_x}{r_x - o_x} \right\rceil \quad \text{and} \quad N_y = \left\lceil \frac{R_y - o_y}{r_y - o_y} \right\rceil. \tag{5.1}$$

The combination of the same tile from each layer forms a subsection of the original layer stack and so, essentially a new attenuator of smaller size and



(a) (b)

Figure 5.3: (a) Layout of the tiles that cover the attenuation layers. The pixel grid of size $R_x \times R_y$ is covered by tiles of $r_x \times r_y$ pixels with an overlap of o_x in horizontal and o_y in vertical direction. (b) The sum of the per-tile quadratic blending masks used for the normalization.

lower resolution. The optimization is then performed on all of the subsections with a smaller propagation matrix per tile (fewer columns). As a consequence, less memory is used to store attenuation layers and propagation data in each step. In the end, the optimized tiles are put together to form the complete attenuation layers.

In general, the borders of the attenuator contain less ray-propagation information and thus provide fewer constraints for the optimization. This introduces artifacts that are clearly visible in the reassembled layers as shown in figure 5.4. To solve this issue, the tiles have to overlap. In this case, when reassembling the layers from the tiles, the overlaps need to be blended with a mask: After the optimization, each tile gets multiplied with a quadratic blending mask. The finished layers are then obtained by summing the tiles and dividing by the sum of the blending masks shown in figure 5.3b. For the results shown in this work, a quadratic blending mask was used (weights increase quadratically towards the middle of the tile).

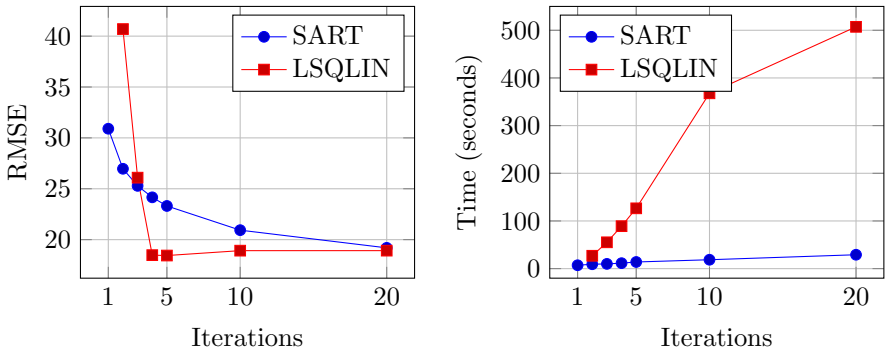
5.7 Performance of SART

The two iterative solvers described in section 3.4 are compared in figure 5.5 in terms of reconstruction error and runtime. Although the standard least squares solver uses fewer iterations compared to SART, the computation time is significantly longer. For the specific experiment in the figure, both methods achieve the same RMSE with twenty iterations, but SART performs the twenty iterations in the same time as LSQLIN solves two iterations. Furthermore, the simple update rule of SART allows for concurrent optimization in each color channel as opposed to LSQLIN, which needs to be run on every channel separately. This shows that SART is a superior solver for this large scale layered 3D problem.

(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Blending masks

Figure 5.4: Impact of tile overlap on attenuation layers. Top: Tiles have no overlap and grid artifacts are visible. Bottom: With a 50% overlap, the artifacts are no longer noticeable, but more tiles are needed.

Next, the reconstruction quality of SART is compared against the number of layers used to display. Adding more layers means adding more pixels and hence there are more degrees of freedom for the optimization. In theory the more layers there are, the closer is the display to a full attenuation volume. The plot in figure 5.6 shows how the PSNR of the simulated projections behaves with up to twenty layers. The quality measured by the PSNR increases with more layers added to the display, but above five layers, it is nearly constant. As it can be seen in the figure, the PSNR even drops above ten layers. There are two possible reasons for that: The first is the limitation of the attenuator to have a minimum transmission: It is assumed that the attenuation masks cannot produce total absorbance, that is completely black pixels. Hence, the transmitted light field will also have a minimum brightness and completely black regions cannot be reproduced by the display. To incorporate this property, the values in the input light field that are smaller than the minimum transmission allowed by the attenuator are replaced by this constant and the resulting clamped light field is the new input for the optimization algorithm. Because of the logarithms singularity at zero, the elements in the light field vector need to be strictly positive and thus the clamping becomes necessary anyway. This alteration introduces a small error when computing the PSNR between simulated projections and the unmodified light field. The second reason for the decrease in PSNR could be that the increase of layers changes their placement slightly because the display thickness is kept constant. The displacement affects the approximation of the scene volume and depending on the light field, one setting may be better than the other.



(a) (b)

Figure 5.5: Performance assessment of the optimization. The two iterative methods SART and MATLAB's linear least squares solver LSQLIN are compared in terms of RMSE (a) and runtime (b). The input light field is the same as in figure 5.2 and five attenuation layers were used.

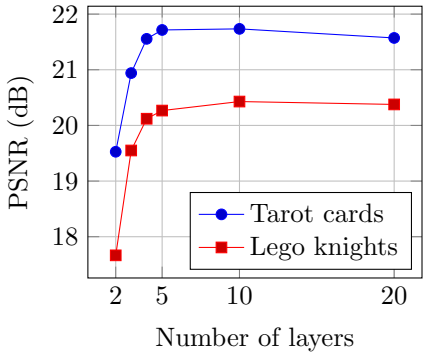


Figure 5.6: Impact of layer count on the PSNR. Performance comparison of the attenuation display for two different scenes from the Stanford light field archive. Ten iterations of SART were performed for each scene.

5.8 Contrast Sensitivity

The contrast sensitivity is a very important measure for human visual perception. The human visual system does not only perceive light intensity in a non-linear fashion, but it is also very sensitive to contrast. Contrast sensitivity is defined to be the threshold at which the individual can no longer distinguish between contrast differences, and this threshold changes with spatial frequency. The top left image in figure 5.7 shows the contrast sensitivity pattern which changes horizontally in frequency and vertically in contrast. In context of Layered 3D, it is of course desirable that the loss of contrast in the emitted images is as little as possible. The experiments in figure 5.7 quantify the absolute error of contrast with a light field comprising the contrast sensitivity pattern in spatial dimension which is constant in angular dimension. It shows that adding more layers reduces the magnitude of the error, and increasing the layer resolution reduces the error for higher frequencies.

5.9 Graphical User Interface

All functionalities of the pipeline are made available through a graphical user interface (GUI) for ease of use. The main window is shown in figure 5.8 for a particular use case. A typical workflow involves three main steps, which correspond to the three columns in the window. In the first step, the user imports a light field from either a folder of images or a Lytro container file. The user can specify the projection type, camera- and image plane parameters as well as spatial- and angular downsampling in case memory is scarce. After a successful import, the individual angular views are displayed in the preview window below. The seconds step involves the configuration of the attenuator where the user enters the desired display thickness, size and resolution. Before running the optimization, the user has the option to back-project the light field in case he/she want to ensure that the light fields depth range is aligned with the display. In the last step, after optimization has completed, the user can preview the results (attenuation layers, reconstructed views or error images) in the window below or save them to disk. Finally, a PDF file with the attenuation masks can be generated, ready to print on transparencies.

The software together with the GUI is available as an executable[1] and does not require a MATLAB installation in order to run.

5.10 Benefits and Limitations

From a theoretical standpoint, the layered 3D architecture seems to be very promising. The analysis shows that multiplicative displays encompass extended spectral support resulting in a higher depth of field compared to other auto-multiscopic systems. The multi-layer design eliminates the trade-off between angular- and spatial resolution that is present with parallax barriers or integral imaging. In practice, there are a few limitations that need to be addressed. First off, to solve the multiplicative problem given by equation 3.9 it is assumed that the transmission values are positive in order to solve the problem in a linear manner with the logarithm applied. For the physical realization, this is of course a well justified assumption since negative transmission is not possible to achieve. Nonetheless, the restriction to positive transmittance is reducing the space of solutions for optimal attenuation layers. Although it would require an entirely different approach, it is plausible to achieve better results if real valued transmission values were permitted.

The prototypes produced in the context of this thesis are suited for demonstration purposes, though the viewing angles are limited to a small range. Unfortunately, experiments to show objects in virtual planes outside the displays enclosure were not successful. Another challenge with display fabrication is the manual layer alignment. The marks printed on the border of the layers help with alignment, but the process remains tedious and becomes increasingly harder with more layers. Moreover, there are no universally best printer settings (amount of ink, drying time etc.) for printing on transparencies. The settings have to be tuned by trial-and-error depending on the particular printer model and transparencies. Also, Moiré does not seem to be a problem because of the natural blending of ink.

[1] The software available at `https://github.com/awaelchli/bachelor_project`.

5.11 Improvements and Future Work

The current implementation in MATLAB provides the necessary features in order to produce a static layered 3D display. However, with greater effort a lot of operations can be made more efficient. A GPU implementation of SART like the one from Lu et al. [2009] could be incorporated in the software to accelerate the optimization process. A parallelization of SART is possible because the update rule depends solely on matrix multiplication and addition. The parallel approach would also eliminate the need of explicitly pre-computing the propagation matrix, which is a computationally- and memory intensive process. Still, the final printing and layer alignment remains time consuming and must be attacked with patience. Alternatively the masks could be printed directly on glass which would result in a higher build quality and precise layer alignment.

Some assumptions that are incorporated into the software could be generalized. Among them is the pinhole camera model that could be extended to a more realistic lens model. Further, only regularly sampled light fields are supported which would make it harder to adopt light fields from unconventional capturing devices.

As Wetzstein et al. [2012] have shown, LCD panels can be used to modulate light instead of printed layers. The benefit here is that the panels can be accurately aligned and calibrated once, and no further adjustments to the hardware are necessary in order to view different content. Building the display however requires advanced technical knowledge in developing custom electronics to control and interface with the panels.

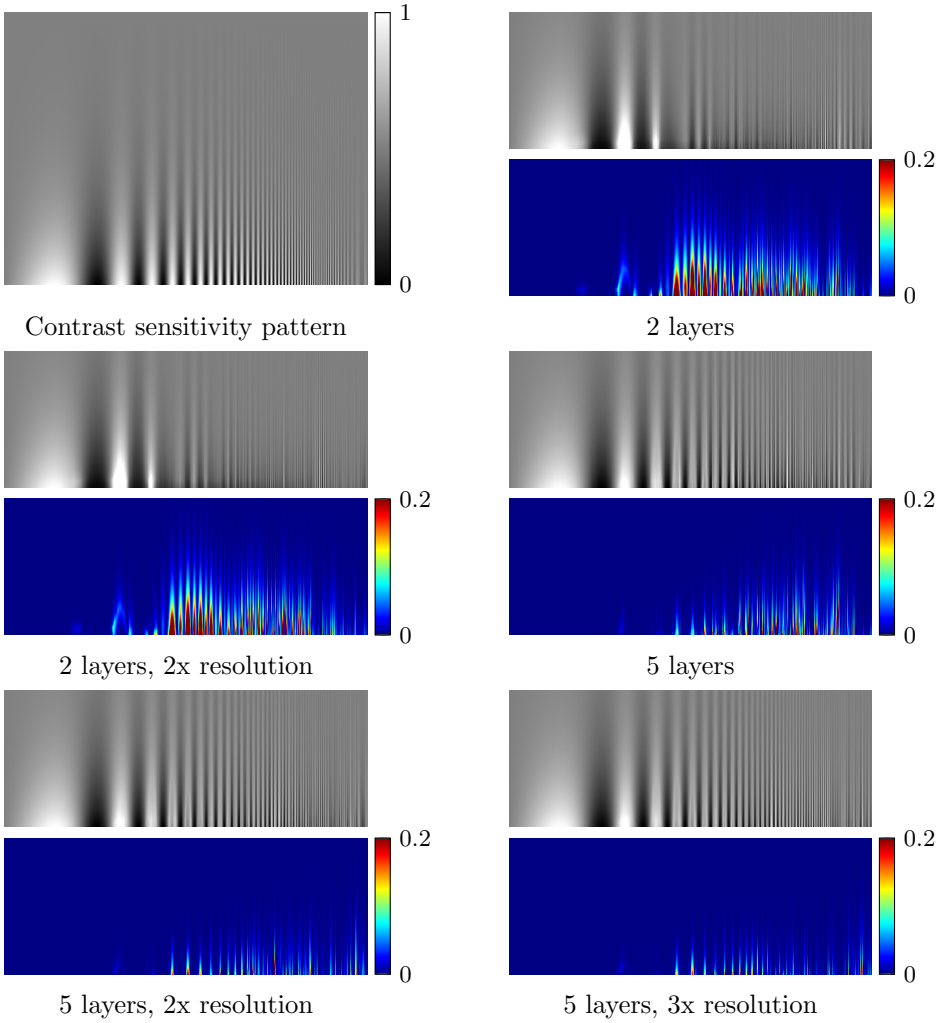


Figure 5.7: Contrast sensitivity analysis for layered 3D displays. Shown are the simulated display projection from a viewing direction perpendicular to the display (top) and the absolute error (bottom) for a two-layer and five-layer display with different layer resolution. The light field is constant in angle, contains increasing spatial frequency from left to right and increasing contrast from top to bottom.

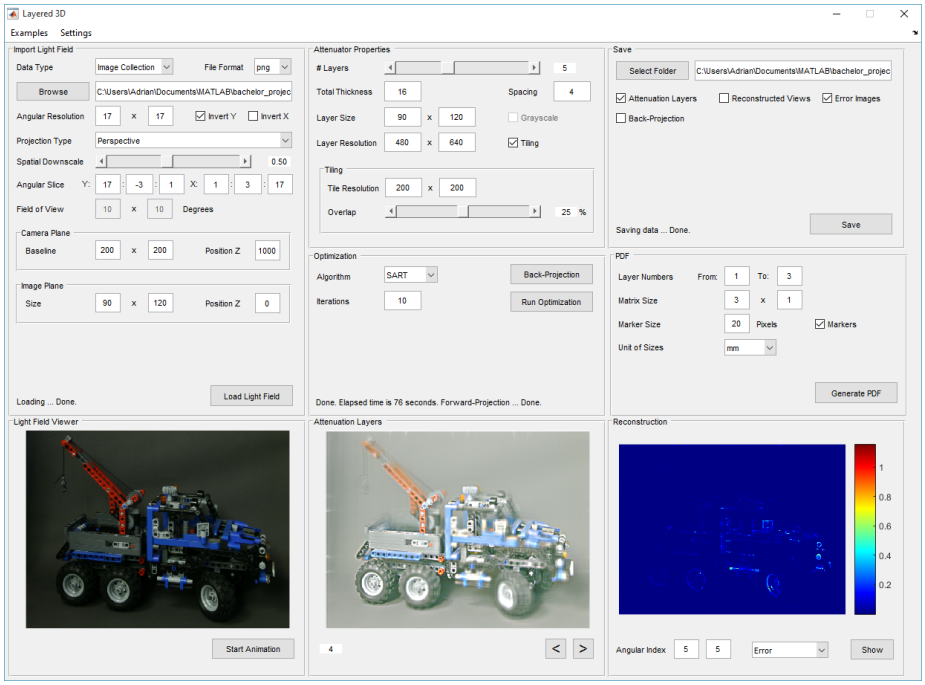


Figure 5.8: Graphical user interface for the *Layered 3D* software developed with MATLAB.

Appendix A

Appendix

A.1 Printing

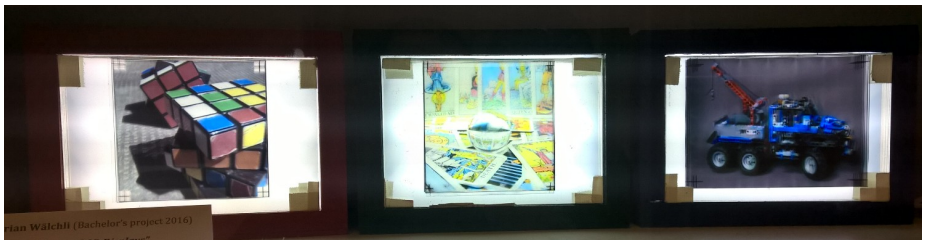
The exported PDF file(s) with the attenuation masks can directly be printed with any device that supports transparencies. The attenuation layers for this project have been printed with an old HP Photosmart C5180 ink-jet printer. The built-in photo-printing mode turned out to work best, but all automatic image enhancement settings (e.g. red-eye removal) must be turned off in the advanced settings. Also, ink-jet printers require special types of transparencies that have a rough surface such that the ink can hold on to the sheet.

A.2 Backlight Fabrication

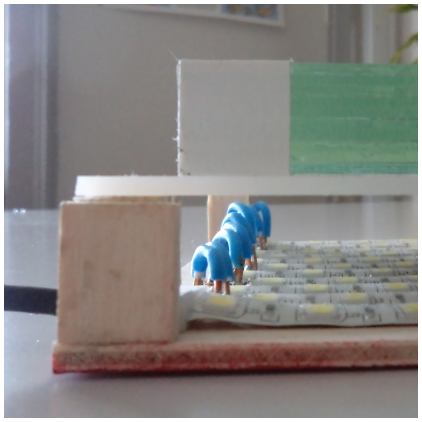
For an optimal viewing experience, a uniform backlight is needed to place behind the glass plates which hold the printed transparencies. LEDs are the optimal choice for a simple white backlight: They are small sized, power saving, affordable and don't produce a lot of heat when turned on. The ones used for this build are average, consumer grade quality LEDs bought on Amazon. A detailed specification of the product is given in table A.1. A five meter long and 10 mm wide LED strip is cut into pieces of 15 cm length, each having nine LEDs. These smaller strips are placed next to each other and glued onto a wooden

LED Chip	SMD 5050
Electric current	12V DC
Color	Cold white
Color temperature	6000 Kelvin
Luminous flux	4500 Lumen
Emission angle	120°
Power consumption	60 Watt
Average lifespan	50000 hours

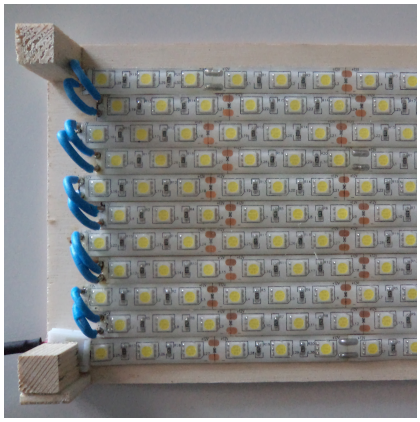
Table A.1: Specification of LEDs used for the backlight.



(a)



(b)



(c)

Figure A.1: (a) Three fully assembled displays with backlight. (b) Close up of the inside with the outer frame removed. Top to bottom: Glass plates holding attenuation layers, diffusion plate, LED grid. (c) Close up of the LED grid with the diffuser removed.

plate forming the base for the backlight. Corresponding ends of the strips are reconnected by soldering small pieces of wire onto the contacts of the strips. For an easier soldering process and to reduce the risk of an electrical short, the strips are alternately offset by a small amount as shown in figure A.1c. A connector for the power supply is mounted on one end of the strip. Wooden stand-offs glued to the base hold the diffuser plate 17 mm above the LEDs. The diffuser is simply a white, milky acrylic plate from a hardware store, cut to the right size. Finally, the build is concluded with a wooden frame covering the LEDs and wires and holding the glass plates in place. The frame has a diagonal of 23 cm, is 5 cm thick and is painted with a color varnish to protect the wood from scratches and to beautify the product.

A total of three backlights were produced for this project. All three displays are powered by a 12V/6A DC power supply. The set also includes a remote control to turn the displays on and off or to adjust the brightness level.

A.3 More Simulated Projections

The following pages show simulated projections from various scenes displayed with the attenuation display. All the examples here have display parameters that

correspond to the displays shown in figures 1.2 and A.1. Five layers are equally spaced to form a 1.6 cm thick stack. The baseline has been manually adjusted such that the desired depth range is approximately within the displays depth of field. Each simulated view is labeled with an index (i, j) that corresponds to the location of the virtual camera on the (u, v)-plane where i denotes the vertical placement from top to bottom and j denotes the horizontal placement from left to right. The images below each simulation show the per-pixel RMSE over the three color channels. High error is visualized with yellow and red color and bluish colors indicate low error.

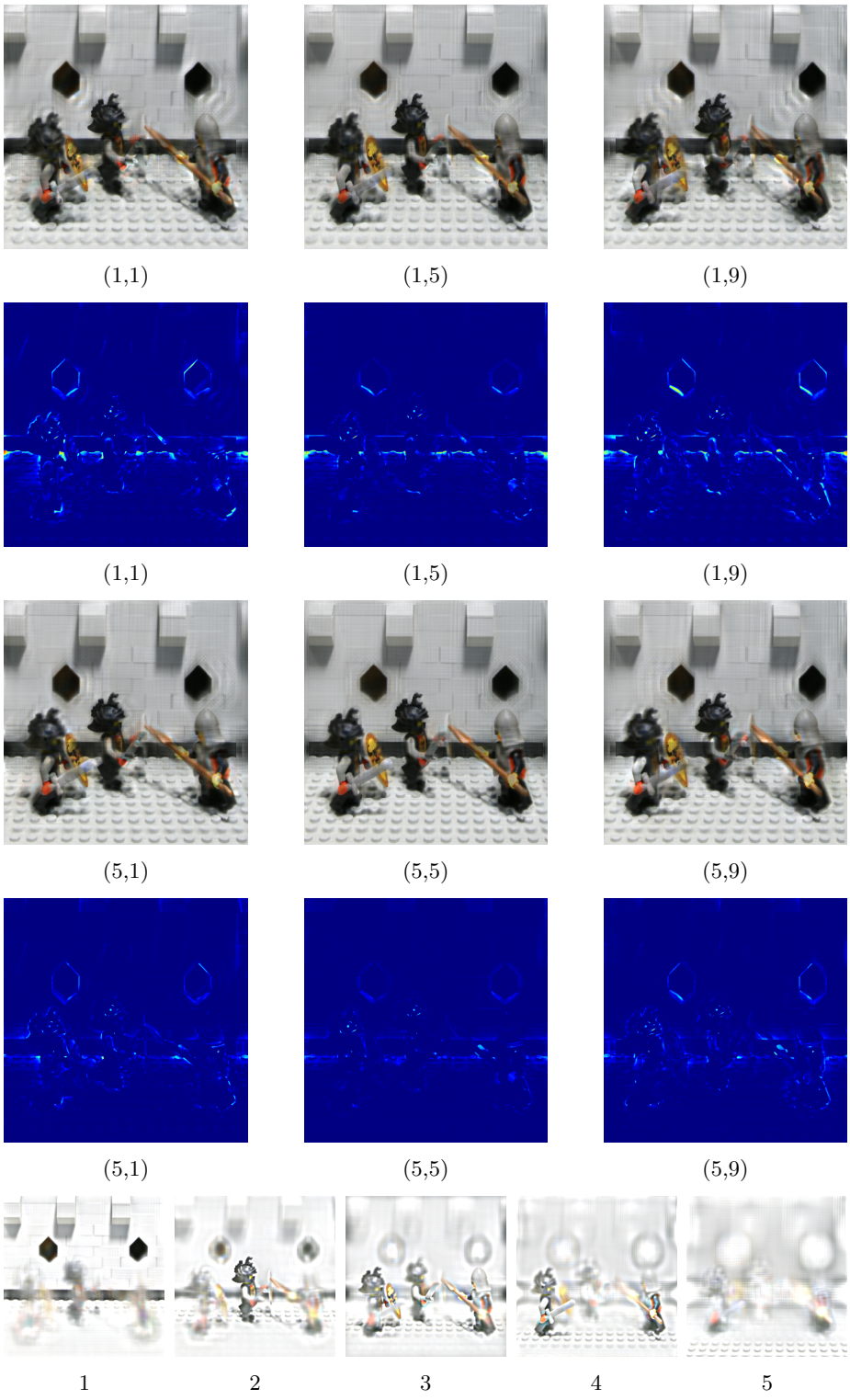


Figure A.2: Simulated projections of the lego knight scene. The light field extends over a baseline of about 14 cm with a distance of 20 cm between the two reference planes. The camera plane is sampled at 9×9 positions. Shown are the simulated views at different positions on the camera plane and the respective visualization of the error compared to the input light field. All five attenuation layers are displayed in the bottom row.

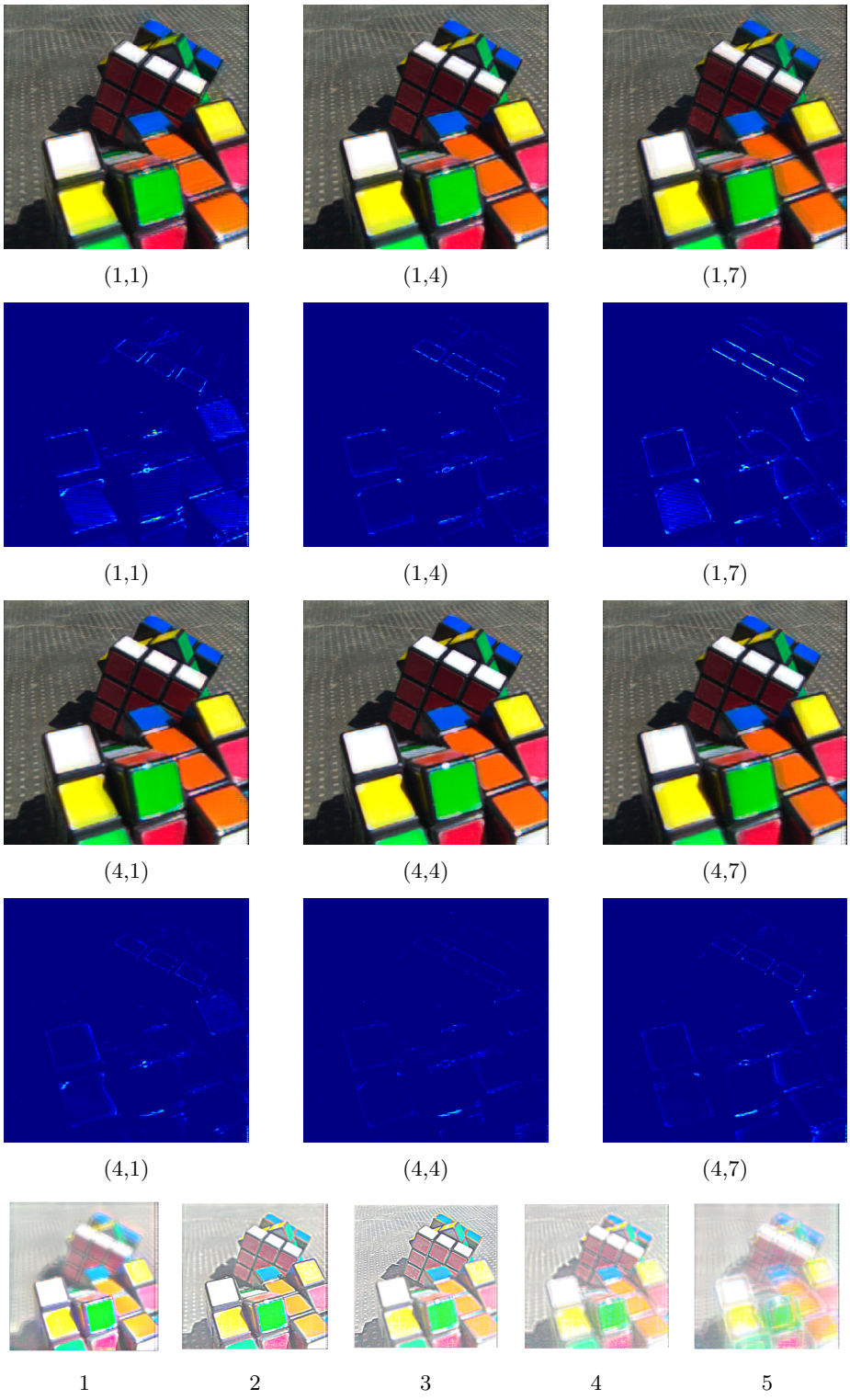


Figure A.3: Simulated projections of a scene taken with the Lytro camera. The light field extends over a baseline of about 5 cm with a distance of 1 m between the two reference planes. The camera plane is sampled at 7×7 positions. Shown are the simulated views at different positions on the camera plane and the respective visualization of the error compared to the input light field. All five attenuation layers are displayed in the bottom row.

List of Figures

List of Tables

Bibliography

E. H. Adelson and J. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, pages 3–20, 1991.

A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging*, 6(1):81–94, 1984.

B. G. Blundell and A. J. Schwarz. *Volumetric Three-Dimensional Display Systems*. Wiley-VCH, Mar. 2000. ISBN 0-471-23928-3.

R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, pages 7–55, 1987.

J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 307–318. ACM, 2000.

D. G. Dansereau, O. Pizarro, and S. B. Williams. Linear volumetric focus for light field cameras. *ACM Trans. Graph.*, 34(2):15:1–15:20, 2015.

F.-C. Huang, G. Wetzstein, B. A. Barsky, and R. Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Trans. Graph.*, 33(4):59:1–59:12, 2014.

A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 297–306. ACM, 2000.

M. Jiang and G. Wang. Convergence of the simultaneous algebraic reconstruction technique (SART). In *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 360–364, 2001.

M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pages 31–42, 1996.

Y. Lu, W. Wang, S. Chen, Y. Xie, J. Qin, W.-M. Pang, and P.-A. Heng. Accelerating algebraic reconstruction using CUDA-enabled GPU. In *Sixth International Conference on Computer Graphics, Imaging and Visualization*, pages 480–485. IEEE, 2009.

R. Narain, R. A. Albert, A. Bulbul, G. J. Ward, M. S. Banks, and J. F. O'Brien. Optimal presentation of imagery with focus cues on multi-plane displays. *ACM Trans. Graph.*, 34(4):59:1–59:12, 2015.

R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

A. Sullivan. A solid-state multi-planar volumetric display. In *SID Symposium Digest of Technical Papers*, volume 34, pages 1531–1533. Wiley Online Library, 2003.

G. Wetzstein, D. Lanman, W. Heidrich, and R. Raskar. Layered 3D: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. *ACM Trans. Graph.*, 30(4):95:1–95:12, 2011.

G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph.*, 31(4):80:1–80:11, 2012.

M. Zwicker, W. Matusik, F. Durand, H. Pfister, and C. Forlines. Antialiasing for automultiscopic 3D displays. In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

Erklärung

gemäss Art. 28 Abs. 2 RSL 05

Name/Vorname: .....

Matrikelnummer: .....

Studiengang: ……

Bachelor ☐ Master ☐ Dissertation ☐

Titel der Arbeit: .....

.....

.....

LeiterIn der Arbeit: .....

.....

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe o des Gesetztes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.

.....
Ort/Datum

.....
Unterschrift