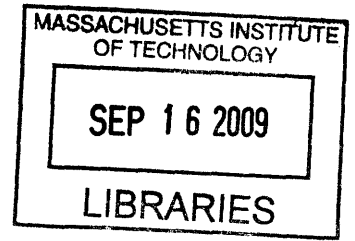


Methods of Invisibility and Multi-Dimensional Display

by
Jason Ku



Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Bachelor of Science in Mechanical Engineering

at the

ARCHIVES


MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

© Jason Ku, MMIX. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis
document in whole or in part.

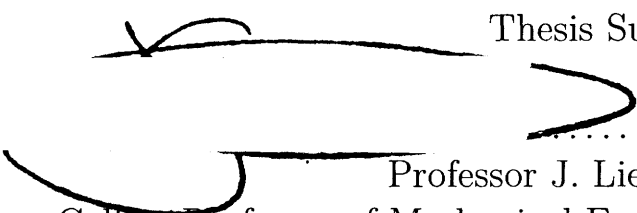
Author.....


Department of Mechanical Engineering
May 8, 2009

Certified by

Anette Hosoi
Associate Professor
Thesis Supervisor

Accepted by.....


Professor J. Lienhard V
Collins Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

Methods of Invisibility and Multi-Dimensional Display

by

Jason Ku

Submitted to the Department of Mechanical Engineering
on May 8, 2009, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Mechanical Engineering

Abstract

Modern methods of multidimensional display are extremely limited in their depiction of three dimensional environments. Improving upon modern integral imaging displays with respect to viewing angle and angular resolution, a new method is proposed consisting of spherical lenses to focus incident light to an imaging surface. A theoretically exact model is provided in addition to a number of feasible physical approximation alternatives for different components of the model. This model can be extended for applications in multidimensional display and invisibility.

Thesis Supervisor: Anette Hosoi
Title: Associate Professor

Acknowledgments

Much of this thesis was directly inspired by conversations with my good friend Brian Chan. Without him, I would never have begun to explore the limits of multidimensional projection. Special thanks to Prof. Barbastathis for introducing me to Lüneburg’s Lens. Additional thanks to my thesis advisor, Prof. Hosoi for allowing me the freedom to explore. And lastly, immeasurable gratitude will always go to my parents, David and Karen, for their constant and unconditional support.

Contents

1	Introduction	8
1.1	Motivation	8
1.2	History	9
1.3	Implementation: Discretizing the Problem	9
1.4	Application	11
2	Choosing a Lens	12
2.1	Integral Imaging	12
2.2	Elliptical Lens	13
2.3	Lüneburg’s Lens	15
2.4	Sphere: Uniform Index of Refraction	16
2.5	Tightness	18
3	Mapping the lens to a surface	24
3.1	Fiber Optics	25
3.2	Ulexite	26
3.3	Folding the plane to the surface	28
4	Encoding graphics	30

4.1	The planar solution	30
4.2	Out of the frame	32
4.3	Front/Back sorting algorithm	32
5	Experimental Results	33
6	Conclusion	37
A	Optimal index of refraction calculation	39
B	Multidimensional image encoding	42
B.1	Main Script	42
B.2	Function array4: Mapping Data to Imaging Plane	44
B.3	Function pixH: Hexagonal Grid Layout	45
C	Encoded Arrays	46

List of Figures

1-1	Depiction of multiple dimensions on a single display	10
2-1	Integral Imaging Array	13
2-2	An elliptical lens with in-axis incident light	14
2-3	An elliptical lens with off-axis incident light	14
2-4	Lüneburg’s Lens ray trace	16
2-5	Ray paths for uniform spheres of different indices of refraction	17
2-6	Geometric representation of angular error ϵ	18
2-7	Angular error ϵ profiles with respect to location of incidence θ	19
2-8	Weighted squared angular error ω profiles with respect to location of incidence θ	20
2-9	Tightness area τ with respect to index of refraction n	21
2-10	A spherical pixel viewed in the direction of light incidence with circles representing magnification and the limit of negative angular error	23
3-1	Fiber optic methods	25
3-2	Ulexite	27
3-3	Ulexite Array	27
3-4	An example folded structure to approximate the sphere.	28

4-1	Geometry of graphics projection calculations.	31
5-1	Rectangular spherical pixel array	34
5-2	Small hexagonal spherical pixel array	35
5-3	Large hexagonal spherical pixel array viewed from four different angles	36
C-1	Encoded image of inset sphere for rectangular grid with center one radius from the viewing plane.	47
C-2	Encoded image of inset sphere for rectangular grid with center one and a half radii from the viewing plane.	47
C-3	Encoded image of inset sphere for a small hexagonal grid with center one and a half radii from the viewing plane.	47
C-4	Encoded image of inset sphere for a large hexagonal grid with center one and a half radii from the viewing plane.	48

Chapter 1

Introduction

1.1 Motivation

Throughout the course of history, human beings have attempted to capture and preserve the world around them. Cave drawings, oil paintings, photographs, videos; all provide an abstraction of the world that might seek to represent and reproduce a moment or moments in time. However, even the most advanced of these current methods of image archival have served only to represent the world as singularly projected onto a two dimensional plane.

The limitation of these methods of image display is that the image portrayed projects the same light information, regardless of the location of the viewer. If you see a photograph hanging on a wall depicting the front of a house, and you move ten degrees to the left, you still see the same image (albeit a distorted image) of the front of the house. This thesis furthers the development of an image display system that will be able to adapt to such angular perturbations from the viewing surface, providing the viewer with the same information as if they were looking at a real three

dimensional object through a two dimensional window.

1.2 History

Multidimensional display is not a new concept. Holograms and 3D glasses are technologies which achieve parallax that have been in use for decades. However, holograms are by nature passive, static devices, while the implementation of 3D glasses is restrictive and limited. Other solutions to displaying three dimensional forms have involved projecting onto rotating mirrors to simulate a three dimensional object [3], though require large three dimensional volumes to display. Modern integral imaging techniques come close to achieving true multidimensional display [1]. 3D televisions implementing integral imaging technology have even become commercially available [Philips WOW]. However these integral image displays have substantial flaws, and it is solutions to these flaws which this paper will attempt to analyze.

1.3 Implementation: Discretizing the Problem

To implement a pixelated, multidimensional display, we must discretize the problem into different linear and angular dimensions. Imagine a video screen. However, instead of a regular two dimensional display, this screen is composed of many multi-pixels that emit different light information (color and/or intensity) with respect to the angle from which it is viewed. The information emitted for each individual pixel now depends on four variables: the pixel's rectangular position (x, y) in space (in the plane of the screen), and the angle (θ, α) from which the pixel is viewed (see Figure 1-1). Creating such a four dimensional screen would be able to hold enough display information to be able to sufficiently and fully represent a three dimensional space

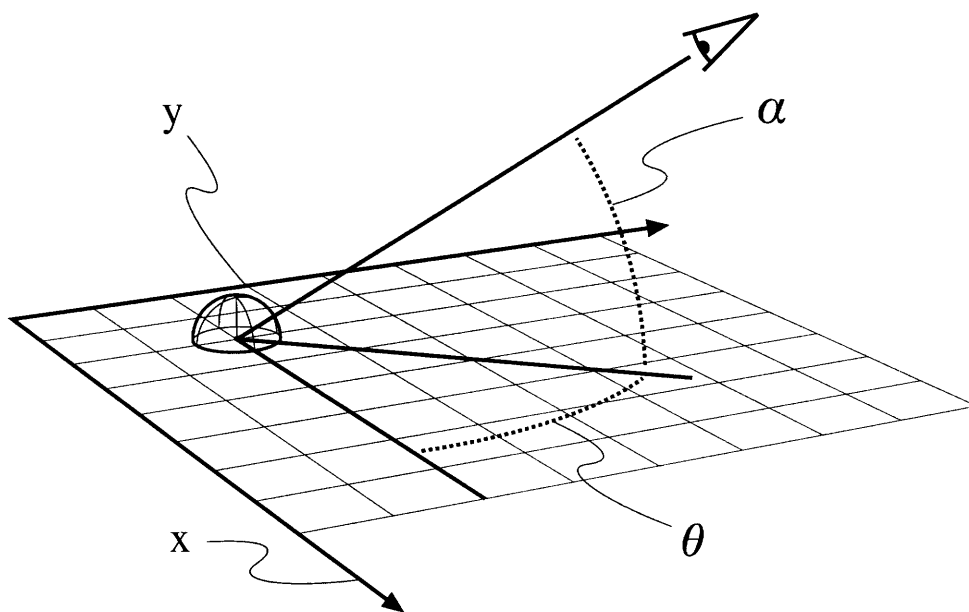


Figure 1-1: In the proposed display, each pixel displays information in four dimensions: the pixel's position (x, y) in the plane of the screen providing two dimensions, and the angle (θ, α) from which the pixel is viewed (in spherical coordinates).

viewed through a two dimensional window. This data structure is what integral imaging strive to obtain but are unable not fully achieve.

1.4 Application

The seemingly simple idea of a single, multidimensional pixel or multi-pixel is intriguingly broad in application. A static multi-pixel array of light emitters could become a window into another world. A multi-pixel array of lights sensors could function as a three dimensional camera. Time-varying emission of light from an array of multi-pixels would give rise to video, but in three dimensions. Combining sensing and emitting capability into the same multi-pixel, where light sensed at a given angle and location is emitted back from that same angle and location, would emulate a mirror.

This idea of a combined sensing and emitting multi-pixel has much more interesting uses than the implementation of a digital mirror, or a multidimensional entertainment screen. There is the potential for military application in the field of camouflage. Surrounding a body with such combined multi-pixels could conceivably be programmed to emit the same stream of light that struck the body on the opposite side, thus rendering the object invisible.

Chapter 2

Choosing a Lens

2.1 Integral Imaging

In order for a single pixel to show different information with respect to different viewing angles, it is necessary to find some way of uniquely mapping each element of an information array to a corresponding viewing angle. Ideally, light information must be produced, then modified, to project from the pixel in only the desired direction.

Methods of focusing different angles of incident light to a discrete array exist. Modern integral imaging devices are lenses made up of an array of curved surfaces on one side and a flat plane on the other with some constant index of refraction [5]. Incident light refracts with respect to the curved upper surface and is ‘semi-focused’ onto the planar surface below. This image is ‘semi-focused’ because the resulting projection is not focused precisely to a point for all angles and contains spherical aberrations.

Another distinct problem with modern integral imaging setups is that their func-

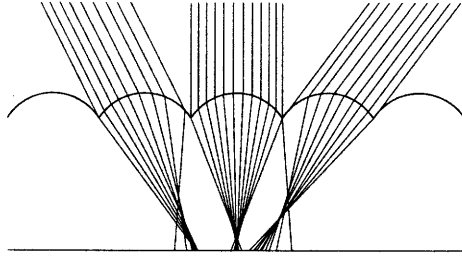


Figure 2-1: Integral imaging arrays can successfully map incident angles to areas on a planar surface, but have significant spherical aberration and limited viewing angle.

tional viewing angle is confined to a tight cone. This is due to two properties of modern integral imaging arrays. First, the incident surface is not symmetric with respect to the angle of incidence causing an increase of spherical aberration with large deviation from the normal direction. Multiple different lens types and shapes have been proposed to address the problem of spherical aberration [4]. This paper address a few specific shapes.

Second, modern integral imaging is not able to fully encode an entire hemisphere of viewing angles. At steep viewing angles, the lens will map not to the image below the specified lens, but instead to the image of an adjacent pixel. This paper will also address some possible solutions to mapping a full hemisphere of light information to a two dimensional data structure.

2.2 Elliptical Lens

Optical solutions exists which perfectly focus light from a given direction down to some unique location, thus mapping an incident angle to a specific point in space. A lens bends light rays depending on the proportional change in index of refraction over an interface. For a single direction and uniform index of refraction, the solution

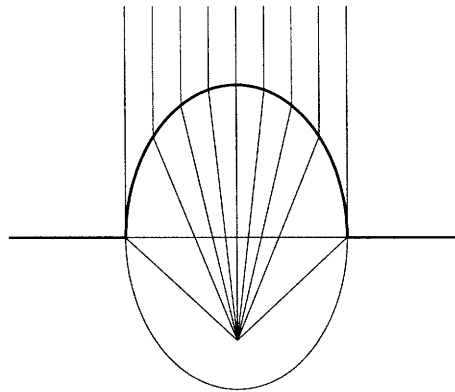


Figure 2-2: Creating an elliptical lens of eccentricity compatible with the material's index of refraction, all light incident from the normal direction will focus directly onto the opposite focus of the ellipse. The ellipse shown has an eccentricity compatible with the index of refraction of acrylic glass (1.490).

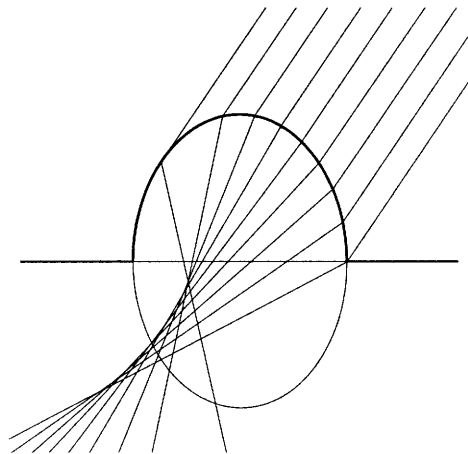


Figure 2-3: Creating a elliptical lens of eccentricity compatible with the material's index of refraction, all light incident from the normal direction will focus directly onto the opposite focus of the ellipse. However, if the incident light is angularly offset the light will no longer converge to a single point.

is fairly straight forward. Creating an elliptical lens, as shown in Figure 2-2, of eccentricity compatible with the material's index of refraction, all light incident from the normal direction will focus directly onto the opposite focus of the ellipse.

However, such focusing precision only holds true for this particular incident direction. If the incident light is angularly offset as in Figure 2-3, the light will no longer converge to a single point and will thus be of little use to the proposed application. Surely, a desirable property of a suitable lens would be to look geometrically similar to incident light, regardless of the viewing angle. A spherical boundary surface would be a natural selection considering its perfect symmetry. All calculations can be made with respect to a single direction, and may then be guaranteed to apply to any other incident angle due to symmetry.

2.3 Lüneburg's Lens

A solution to the focusing problem actually exists for the spherical case. Lüneburg's Lens [2] is a well known gradient-index lens which in theory focuses planar incident light directly to a point on the opposite side. Such a lens with a radius of r_0 would have a varying index of refraction as a function of radius r given by Equation 2.1. As can be seen from the equation, the sphere has an index of refraction of 1 at the surface of the sphere and an index of refraction of $\sqrt{2}$ at the center.

$$n = \sqrt{2 - \frac{r^2}{r_0^2}} \quad (2.1)$$

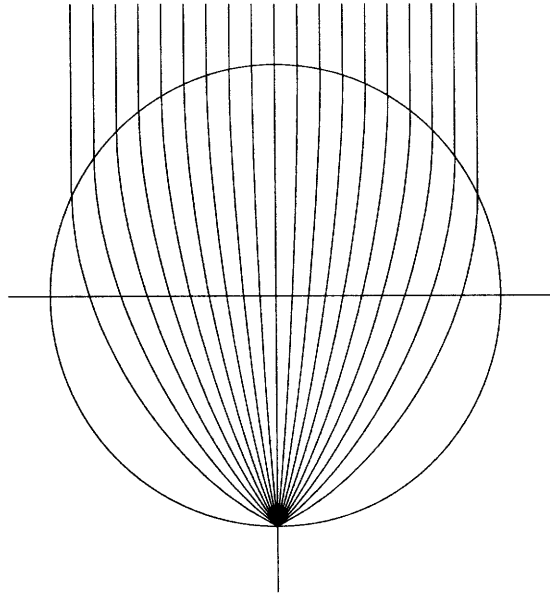


Figure 2-4: Lüneburg's Lens

2.4 Sphere: Uniform Index of Refraction

While Lüneburg's solution is a mathematical possibility, the task of making a true Lüneburg lens has remained an elusive task and is beyond the scope of this paper. However, one can approximate Lüneburg's lens by focusing light using a sphere of some uniform optimal index of refraction. Figure 2-5 depicts the refraction of incident light through a spherical lens for different indices of refraction (air has an index of refraction of 1.00). The incident light rays refract through the surface according to Snell's law:

$$\frac{\sin \theta_i}{\sin \theta_r} = \frac{n_r}{n_i} \quad (2.2)$$

The ratio of the sines of the incident to the refracted angles is equal to the inverted ratio of their indices of refraction. Note that symmetry maintains that the center of

the focused area will reside on the opposite side of the sphere at the point marked by the ray that runs through the center of the sphere. This ray does not change direction as it is normal to the lens surface upon incidence. The refracted rays are focused differently with respect to different indices of refraction.

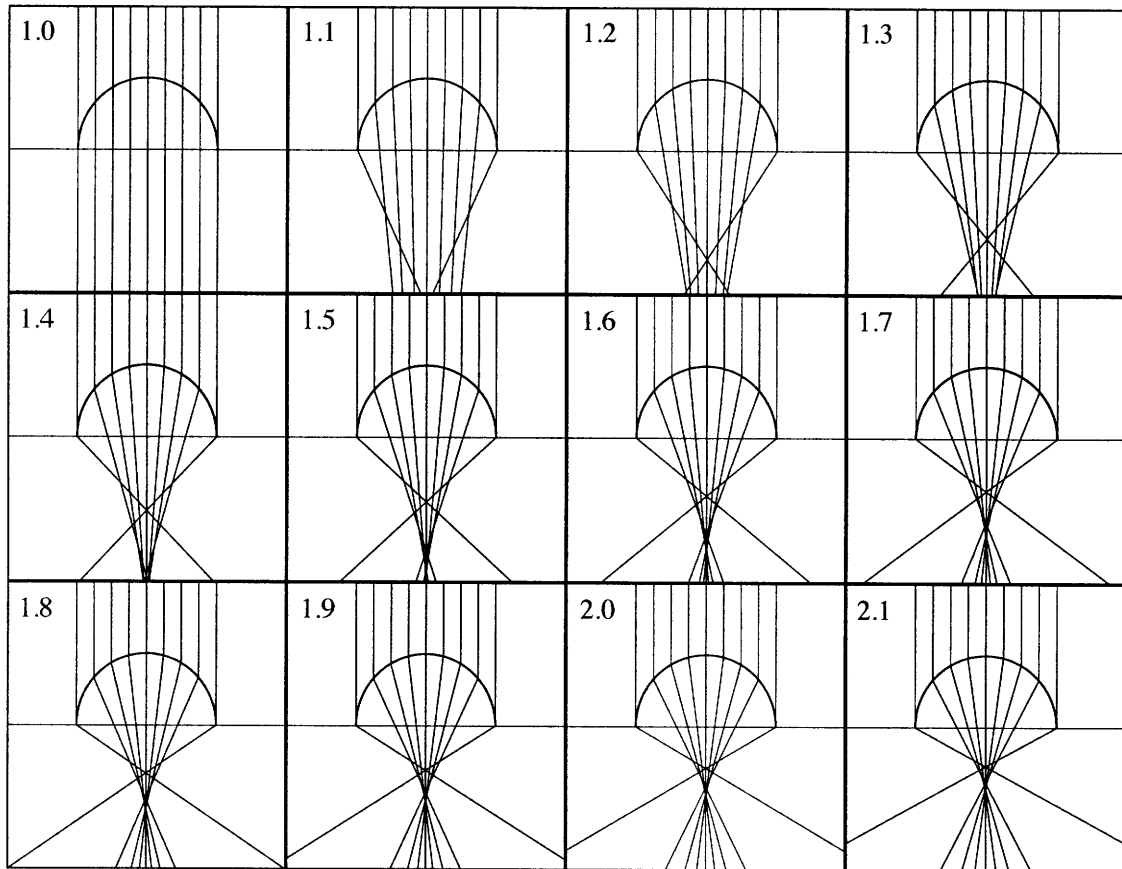


Figure 2-5: The refraction of incident light for different lens indices of refraction. The refracted rays are focused into a tighter area as the index of refraction increases.

2.5 Tightness

In order to choose the optimal index of refraction, a measure of tightness must be established. Consider a sphere of some uniform index of refraction n . Light is incident on the surface at some angle θ from the vertical. The light refracts through the sphere and terminates at some angle ϵ from the desired location. ϵ is given by Equation 2.3.

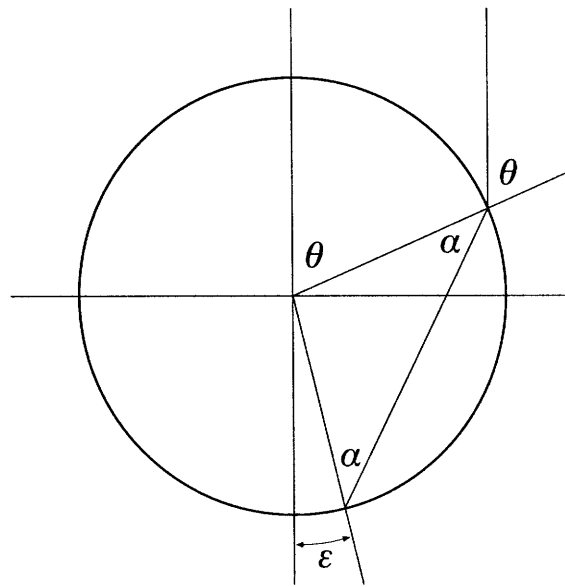


Figure 2-6: Incident light hits a spherical pixel, refracting through, and terminates at some angle ϵ from the desired location. Here, α is the refracted angle through the material interface given by Snell's Law, $\alpha = \arcsin\left(\frac{\sin\theta}{n}\right)$

$$\epsilon = 2 \arcsin\left(\frac{\sin\theta}{n}\right) - \theta \quad (2.3)$$

The spherical aberration that occurs due to the imperfect focusing of the lens can be seen in Figure 2-7 which shows the angular error ϵ profiles as a function of

location of incidence θ at different indices of refraction n . For small θ , the spherical aberration is positive, while at higher θ , the spherical aberration is negative. Positive angular error corresponds to a distorted magnified image while negative error results in a distorted and inverted magnified image. The indices of refraction shown include those for Crown Glass ($n = 1.55$), Bromine ($n = 1.65$), Sapphire ($n = 1.77$), and LaSFN9 Glass ($n = 1.88$).

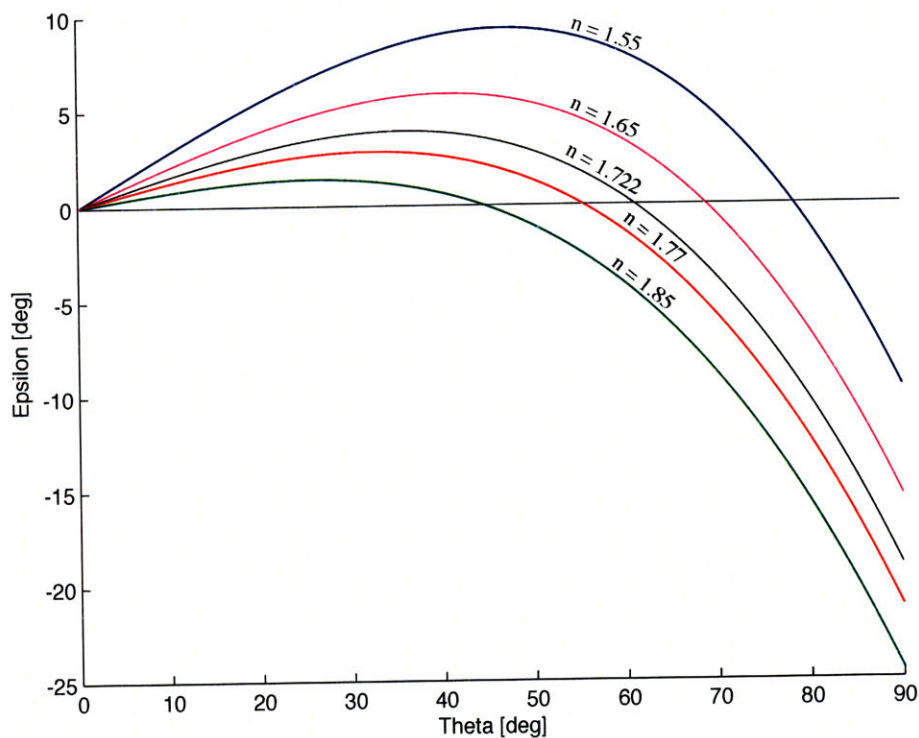


Figure 2-7: Angular error ϵ profiles with respect to location of incidence θ at different indices of refraction n .

Ideally, a tightness parameter would be a measure of the total error flux that the viewer would observe. In other words, angular error should be weighted less if the angle between the sphere's surface normal and the viewing direction is large. We

define a tightness parameter τ to be the sum over the viewing surface of the sphere of the square of the error in the direction of incidence. The surface normal in the direction of incidence is proportional to $\cos \theta$.

$$\tau = \oint_S \epsilon^2 \cos \theta dS = \int_0^{2\pi} \int_0^{\pi/2} \epsilon^2 \cos \theta (\sin \theta r^2 d\theta d\psi) \quad (2.4)$$

$$\omega = \epsilon^2 \cos \theta \sin \theta \quad (2.5)$$

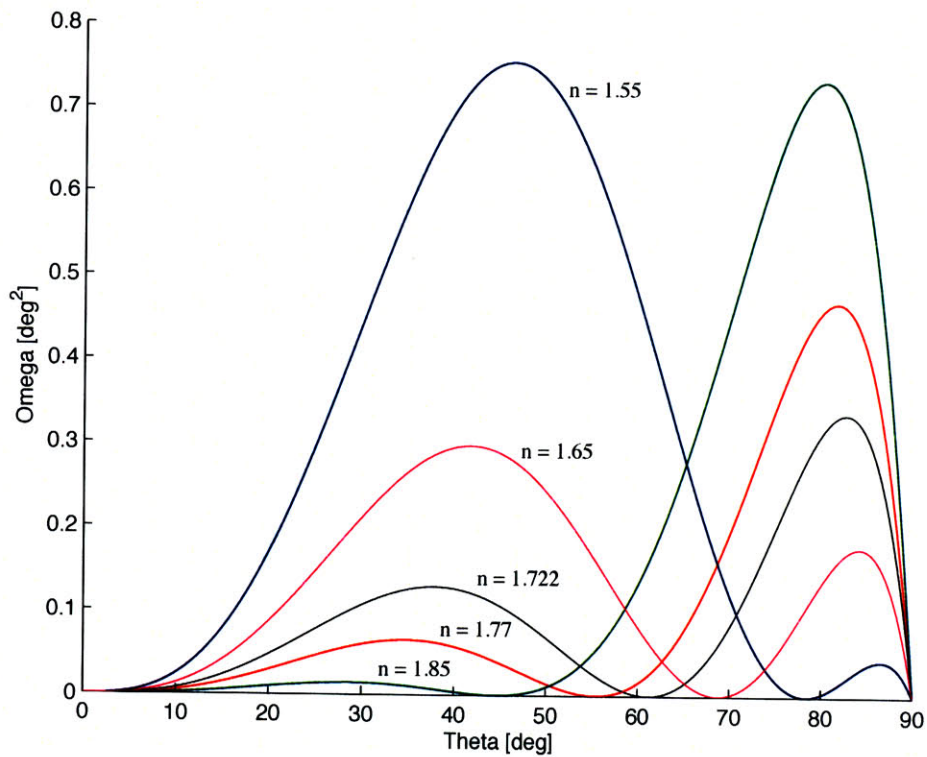


Figure 2-8: Weighted squared angular error ω profiles with respect to location of incidence θ at different indices of refraction n .

This normalized error can be represented by a parameter ω defined by Equation 2.5. A plot of weighted squared angular error ω profiles with respect to location of incidence θ at different indices of refraction n can be found in Figure 2-8. Tightness τ would then be the integral of ω over the surface of the sphere.

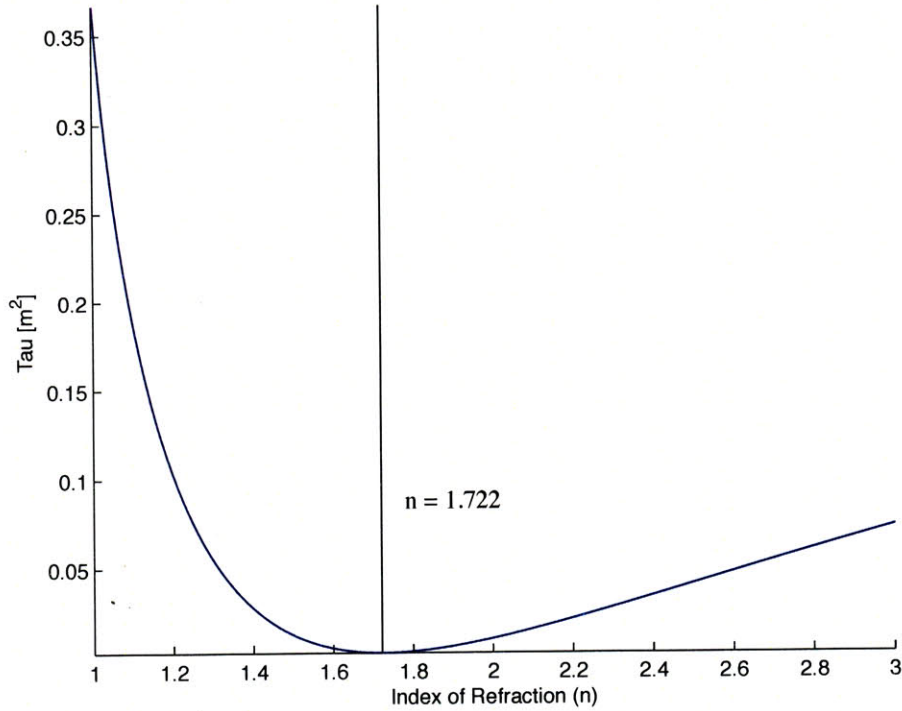


Figure 2-9: Tightness area τ with respect to index of refraction n . Tightness area is minimized for $n = 1.786$.

$$0 = \frac{d\tau}{dn} = \frac{d}{dn} \int_0^{\pi/2} \left[\theta - 2 \arcsin \left(\frac{\sin \theta}{n} \right) \right]^2 \cos \theta \sin \theta d\theta \quad (2.6)$$

By taking the derivative of τ with respect to the index of refraction n , we can minimize the total weighted squared angular error to achieve the optimal index of refraction for uniform spherical balls. This index of refraction is $\boxed{n=1.722}$.

Figure 2-10 can give some reference as to what this optimization achieves. Figure 2-10 shows one spherical pixel viewed in the direction of light incidence. The inner set of circles represent the area on the opposite sphere that is magnified by a spherical lens with respective uniform indices of refraction. The outer set of circles represent the perceived location of the boundary between positive and negative angular error, i.e. the point at which the spherical aberration of the magnified area becomes so great as to invert the image. The circles at $n = 1.722$ minimize the total weighted squared angular error. This index of refraction essentially balances the maximization of magnification ratio while still keeping inverted spherical aberration to a minimum.

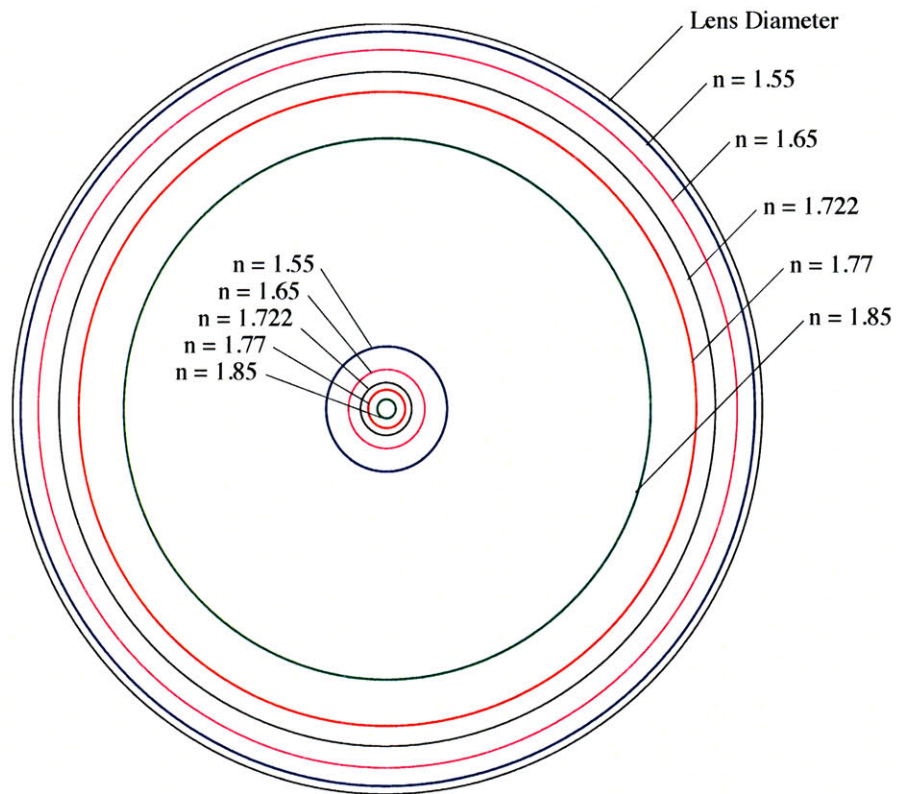


Figure 2-10: A spherical pixel viewed in the direction of light incidence with circles representing magnification and the limit of negative angular error. Inner set of circles represent the area on the opposite sphere that is magnified by a spherical lens with respective uniform indices of refraction. The outer set of circles represent the in plane location of the boundary between positive and negative angular error.

Chapter 3

Mapping the lens to a surface

Assuming that a spherical lens can be obtained that can focus incident rays from a specific angle to a singular point, it is then necessary to map the back of the spherical lens to a surface that can accurately store the complete hemisphere light information. This chapter proposes two possible solutions to this problem.

First, the back surface of the spherical lens could be mapped to a trivial and known surface that is easily manipulated. For instance, mapping the sphere down to a plane would yield a convenient solution. To make a static multidimensional image, one would only need to place behind the lenses a sheet of paper with a correctly encoded image pattern. For a dynamic multidimensional image, a time variant screen such as an LCD display could be utilized. While this solution is desirable from a image encoding standpoint, it encounters some difficult optical and physical limitations.

Alternatively, the back surface of the spherical lens could be mapped to a surface approximating the spherical curvature of the lens. Folding multiple planes containing the correctly encoded light information around the spherical lens could yield a solu-

tion where no interface between the lens and light information would be necessary. While this solution is optically and physically conceivable, implementing the folded surface and image encoding is nontrivial.

3.1 Fiber Optics

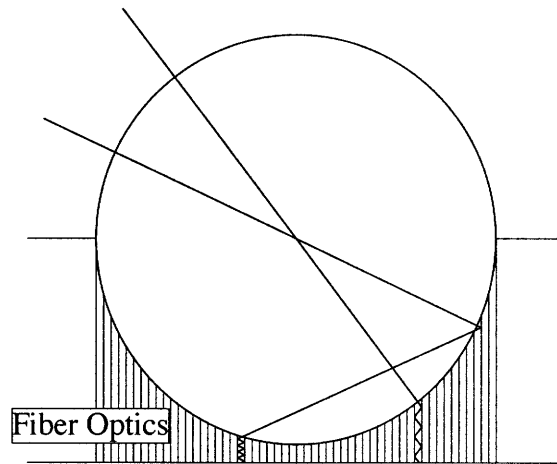


Figure 3-1: Using fiber optics could map the back surface of the spherical surface to a plane, but would not solve the problem of limited viewing angle.

One possible solution to mapping the back surface of the spherical lens to a plane would involve the use of fiber optics. Making a fiber optic faceplate of high resolution could potentially map the light focused upon each point along the spherical surface down to a common plane. However, fiber optics unfortunately have a limited acceptance angle for incoming light.

For a light ray to successfully propagate through the fiber, it must enter within a specific acceptance cone related to the difference in refractive index between the core and sleeve materials of the fiber. Thus, while packing vertical optical fibers into a

faceplate and cutting out spherical sockets might map most of the center area of each lens correctly to the plane, the outer edges of the sphere, and thus larger viewing angles will not be mapped by the fiber optic array because the incident light will fall outside of the fiber's acceptance cone. This solution fails to solve the problem of a restricted viewing angle.

Potentially, optical fiber faceplates might be able to be constructed with a very high numerical aperture using materials of vastly different index of refraction. Such construction may yield an acceptance cone wide enough for practical purposes. However, such high numerical aperture faceplates do not yet exist and would be prohibitively expensive to custom build. Alternatively, one might be able to devise a way of packing fiber optics non-vertically such that each fiber is normal to both the surface of the sphere on one end and the surface of the plan on the other. This solution also yields tremendous problems for fabrication thus will be discarded for the moment.

3.2 Ulexite

Ulexite is a naturally occurring mineral of interesting optical properties. It is composed of long, parallel, tightly-packed crystalline structures that act like a fiber optic baseplate with fibers on the molecular scale. Just like a faceplate made from fiber optics, this material transmits light from one surface to the other along its fibrous structure. However, just like optical fibers, it will not transmit light when observed from steep angles. None the less, this material is readily available and is of higher resolution than fiber optic baseplates. A ulexite baseplate was machined with spherical cavities to project light information up from the lower plane to an array of hemispherical surfaces.

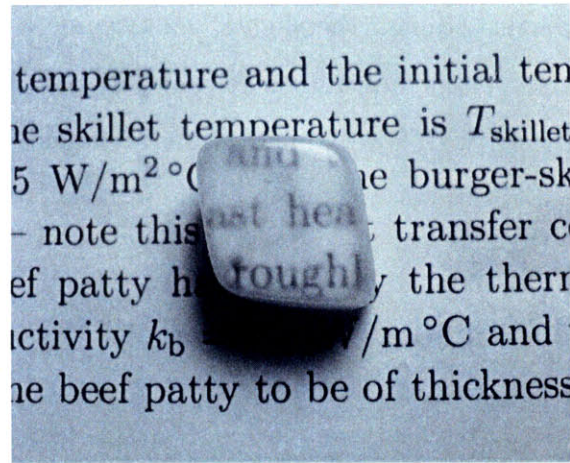


Figure 3-2: Ulexite

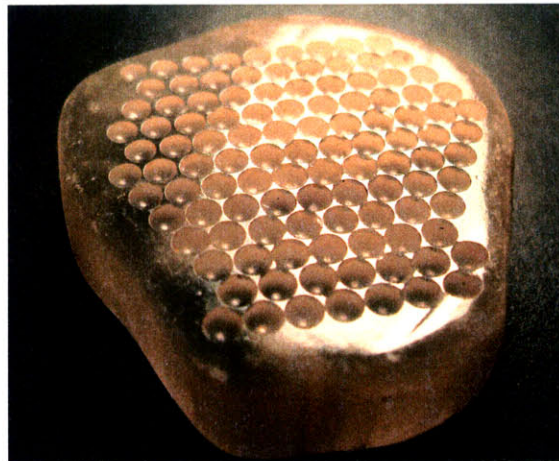


Figure 3-3: A hemispherical array machined from Ulexite. The same limitations for the fiber optical acceptance cone was experimentally observed in this ulexite sample.

However, due to a combination of large imperfections in the ulexite and the natural attenuation of light through the slightly milky mineral, the test plate was able to project only a very faint and unrecognizable image. Potentially, if ulexite could be grown synthetically with fewer imperfections and greater transparency, such a material might be more useful as a solution to the mapping problem.

3.3 Folding the plane to the surface

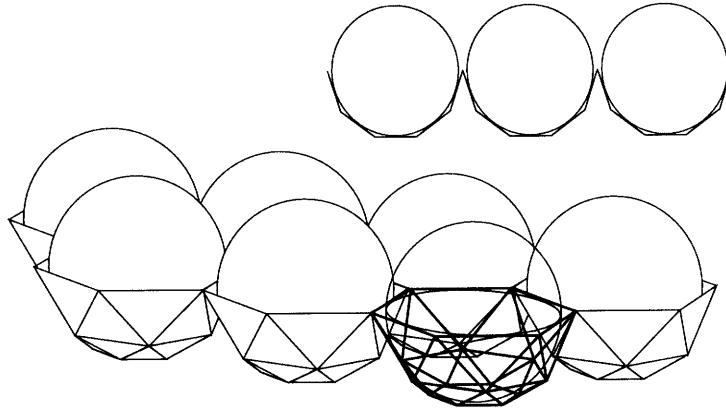


Figure 3-4: An example folded structure to approximate the sphere.

Instead of mapping the surface of the sphere down to a plane, it might be possible to instead wrap the imaging surface around the sphere. While it would be very difficult to print or transmit to a spherical surface, it would be plausible to approximate the sphere with a number of folded, planar surfaces. Triangulating the hemisphere could yield quite a good approximation, even with only some small number of surfaces. As long as the normal distance between the surface of the sphere and the imaging plane is small, the spherical aberration observed will remain negli-

gible. However, encoding graphics for such a surface proves nontrivial and was not pursued. While this paper will not continue to examine this specific implementation, it is presented here as the most reasonable solution to the mapping problem.

Chapter 4

Encoding graphics

Suppose a multidimensional display of the proposed specifications were successfully built. This chapter attempts to propose an algorithm for encoding multidimensional graphics for the display. Encoding methods for a spherical lens system mapped to a plane will be considered, while the proposed spherical lens system mapped to the folded approximated spherical surface will not. For examples of fully encoded images for the fabricated displays, refer to Appendix C.

4.1 The planar solution

For the planar solution, each pixel can be considered separately. The algorithm given essentially acts like a fisheye lens to incoming light, mapping the entire hemisphere of light information to a circle on the plane. Assuming an ideal mapping from the back surface of a spherical lens to a plane, the encoding of multidimensional display information can be readily calculated. Consider a single point $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$ in three dimensional space that is to be projected onto a single spherical pixel located at

$(x_p, y_p, 0)$ (let the pixel array lie in the plane $z = 0$). The projection of this three dimensional point can be expressed in terms of two angular parameters θ and α along with the known radius of the sphere and coordinates of the pixel.

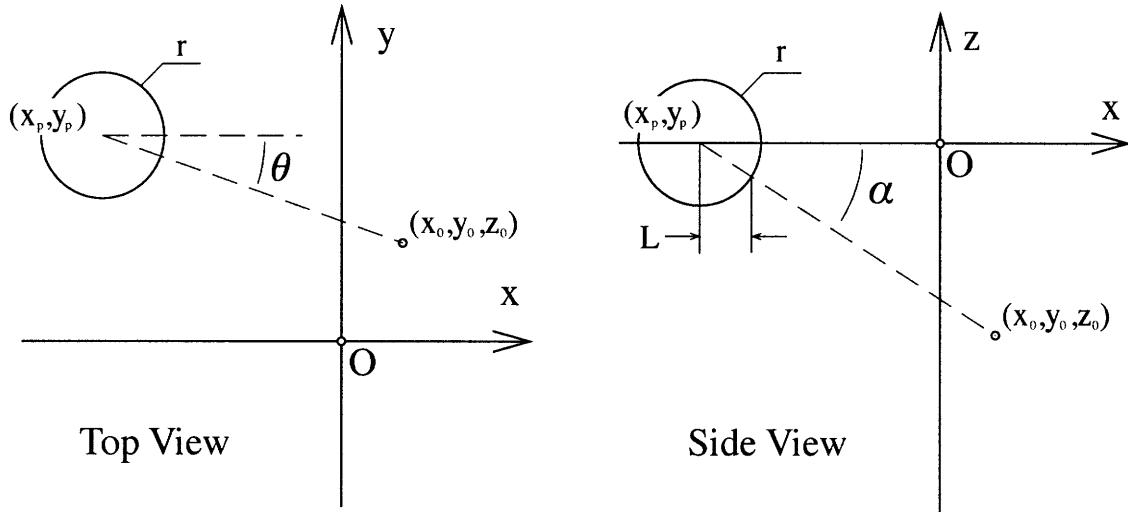


Figure 4-1: Geometry of graphics projection calculations.

$$\theta = \text{atan2}(y_p - y_0, x_p - x_0) \quad (4.1)$$

$$\alpha = \arctan\left(\frac{z_0}{\sqrt{(x_p - x_0)^2 + (y_p - y_0)^2}}\right) \quad (4.2)$$

In this case, θ is calculated using the function atan2 which is the two argument version of the inverse tangent which returns the angle in the correct quadrant. The projected off-axis distance L is the radius of the pixel projected onto the plane from an offset angle α , thus,

$$L = r \cos \alpha \tag{4.3}$$

and the projected point corresponds to the x and y coordinates:

$$\boxed{(x', y') = (x_p + L \cos \theta, y_p + L \sin \theta)} \tag{4.4}$$

4.2 Out of the frame

Such a multidimensional display does not have to be confined to projection into the frame. Instead, one might just as easily create the illusion of a three dimensional object extending out of the frame. In order to accomplish this task for three dimensional data points with $\mathbf{z}_0 > \mathbf{0}$, it must be recognized that the data point is closer to the upper hemisphere of the pixel, thus the point on the imaging surface is directly opposite on the sphere. This is equivalent to saying that when $\mathbf{z}_0 > \mathbf{0}$, θ increases by π , or in other words, L becomes negative.

4.3 Front/Back sorting algorithm

When inputting large amounts of three dimensional data points connected to make complex surfaces, it then becomes necessary to adopt some sort of algorithm to throw out data points that reside behind other data points. A brute force algorithm can be adopted. First calculate the vector $\tilde{\mathbf{v}}_i$ between each data point and the pixel. Compare each pair of data points. If the dot product between two data vectors is less than some small finite tolerance ν , then throw out the data point with larger magnitude $|\tilde{\mathbf{v}}_i|$. In this way, only the data points in direct line of sight to the pixel in question will be considered.

Chapter 5

Experimental Results

For a bench level confirmation of the preceding theory, cursory multidimensional displays were made using 1/8" acrylic spheres. Since acrylic has an index of refraction less than the ideal calculated in Chapter 2, it should be expected that the spherical aberration will be larger and magnification of the encoded image to be smaller than in the ideal case. The first fabricated array was rectangular composed of a 13 x 13 grid of spheres. An outline stencil for the balls was laser printed from clear 1/8" acrylic in addition to 1/16" front and back casings to contain the balls.

Acrylic spheres were aligned within the acrylic stencil. In actuality, this packing proved difficult as spheres do not naturally pack into a rectangular grid. The encoded image was aligned in direct contact with the spherical lenses so as to minimize the spherical aberration due to air between the lens and encoded image. This was a first order model to test the viability of using spherical lenses, so no method was used to map the surface of the spherical lenses down to the plane. This approximation proved extremely accurate for angles within 15°, though deviated from the calculated at higher angles.

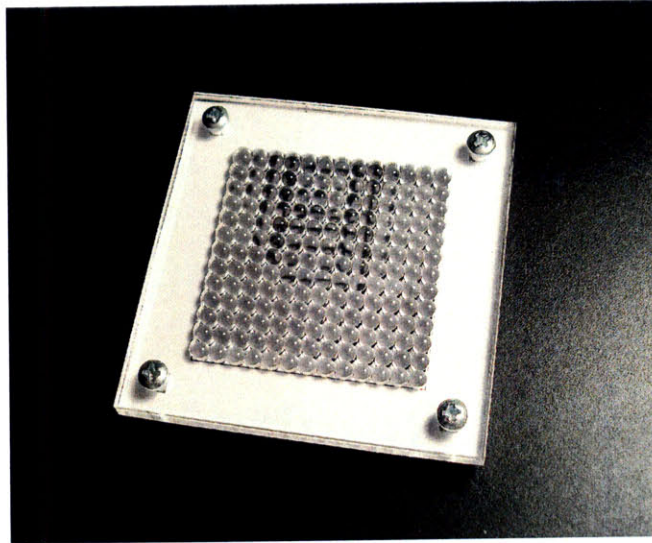


Figure 5-1: 13x13 Rectangular spherical pixel array

Due to the unnecessary difficulty in packing spheres along a rectangular grid, a hexagonal grid was adopted for subsequent displays. A hexagonal grid also yields the largest possible spatial density of in plane spheres thereby increasing our perceived two dimensional planar resolution. For the first hexagonal array, a side length of 8 pixels was used to yield the same number of pixels used for the 13 x 13 rectangular array. The number of pixels h_n in a hexagonal array with side length n can be given by the recursive formula,

$$\left\{ \begin{array}{l} h_1 = 1 \\ h_n = h_{n-1} + 6(n - 1) \end{array} \right\} \quad (5.1)$$

For the small hexagonal display, 1/8" glass spheres were used with an index of refraction slightly higher than that of acrylic. In addition to the image being much crisper than the acrylic, the higher index of refraction yielded a higher overall

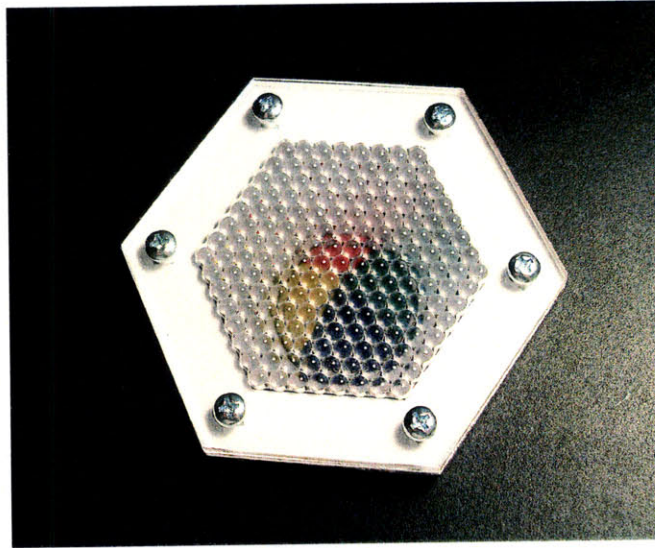


Figure 5-2: Small hexagonal spherical pixel array, 8 pixels on a side

magnification, translating a smaller area on the imaging sphere surface to the upper projection surface.

Lastly, a much larger hexagonal array was built using 16 spherical pixels on a side. While the two previous displays only required 169 each, this larger display required 721. Here, the projected four color ball image was made twice as large as the previous images. As such, equivalent points on this image were set twice as far back as in the smaller hexagonal display.

Here, the desirability of having a very tight focal point as discussed in Chapter 2 was observed; for larger imaging distances from the pixel plane, larger blur was experienced by the viewer. This is consistent with our theory, as some measure of blur β must be proportional to $\sin \epsilon_{\max}$ where ϵ_{\max} is the maximum positive angular error over the surface as defined in Equation 2.3. For the case of Lüneburg's Lens, $\epsilon_{\max} = 0$, thus no blurring effect would be observed.

As was mentioned in Chapter 5, for the produced displays, the lower surface of the spherical lens was not mapped to the plane. The error for this approximation can be readily seen in Figure 5-3. As the angle of viewing incidence increases, the projected image breaks down. As can be seen in c), the observer begins to observe the black line boundary between pixel images, while at even higher viewing angles shown in d), the observer begins to see the image of the adjacent sphere image creating a kind of reflection of the original image.

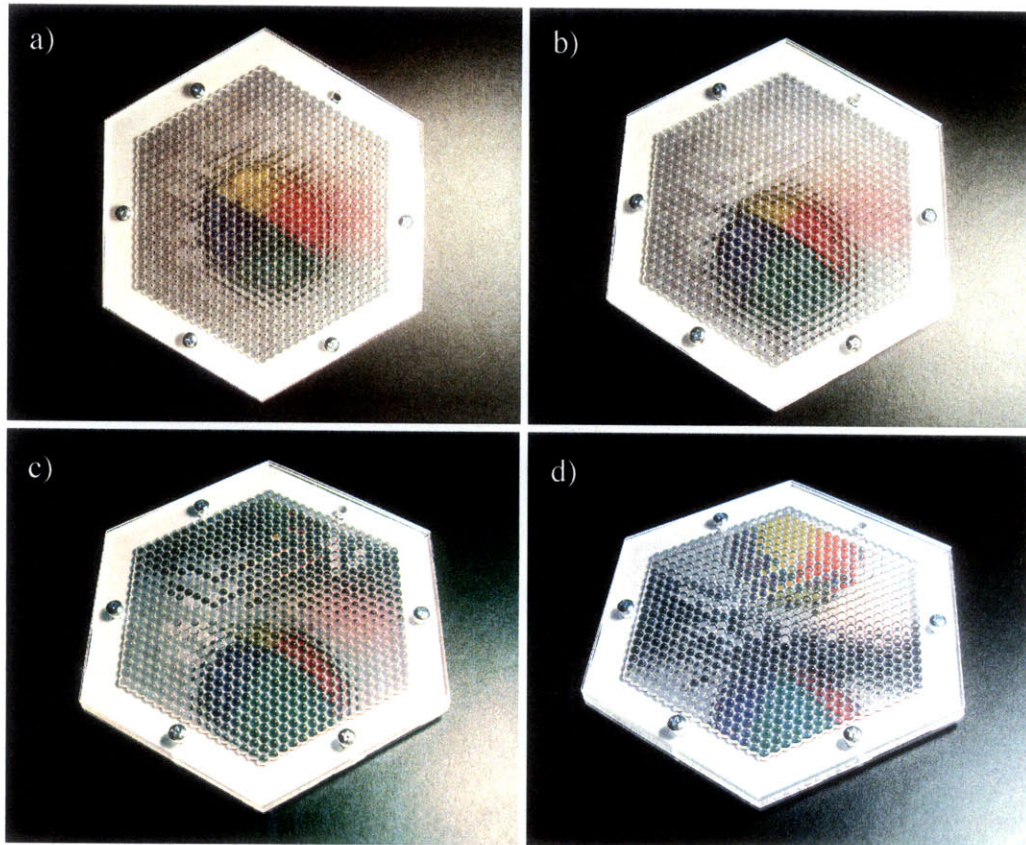


Figure 5-3: Large hexagonal spherical pixel array viewed from a) directly above, b) approximately 15° off axis, c) 30° off axis, and d) 45° off axis.

Chapter 6

Conclusion

Theoretically, a multidimensional display could be constructed from ideal multi-pixels. These multi-pixels would be composed of two components: a Lüneburg's lens to focus incident light to a point on the opposite side of the sphere, and an encoded image printed on a hemispherical surface. In reality both of these two components are very difficult to make. Thus approximations can be made.

For the first component, spheres of uniform index of refraction can be utilized instead of a gradient-index lens. The ideal index of refraction to minimize the total weighted squared angular error of the incident light is 1.722, very close to that of the material sapphire along with other specialty glasses. The limitations due to this approximation are that there will be an intrinsic blur associated with displaying three dimensional data at large distances from the lens plane. However, this error may become negligible when pixel sizes decrease and the angular blur approaches the possible printable resolution of the projected image.

For the second component, the hemispherical surface can be approximated using a folded set of planar surfaces that approximate the sphere. The encoded image could

be printed on a pattern of the unfolded surface, then subsequently folded up. Alternatively, further approximations could be made by mapping the back surface of the spherical lens to a plane using fiber optics of the mineral ulexite. This approximation would limit the viewable angle allowable for such a display.

Thus there is mathematical basis for both exact and approximated multi-pixels that can map the two angular parameters of incident angular light to a finite two dimensional image surface. The existence of such a pixel makes the applications of multidimensional dimensional display and invisibility optically feasible undertakings. Further research would be needed to verify the plausibility of the amount of data manipulation and real-time distribution of light information that would be involved for the case of invisibility.

Appendix A

Optimal index of refraction calculation

A MATLAB script to study index of refraction for spherical lenses.

```
close all; clear all
    %Epsilon function definition
Epsilon=@(theta,n) (2*asin(sin(theta)/n)-theta);
    %Omega function definition
Omega=@(theta,n) (((2*asin(sin(theta)/n)-theta).*cos(theta)).^2.*sin(theta));
    %Tau function definition
Tau=@(n) (quad(@(theta) (((2*asin(sin(theta)/n)-theta).*cos(theta)).^2.*sin(theta))),0,pi/2));
    %Counters
n=1000; index=1:1/n*2:3;
theta=0:1/n*pi/2:pi/2;

    % Figure 1: Tau vs. index of refraction
figure(1); hold on
for j=1:n+1
    out(j)=Tau(index(j));
end
plot(index,out,'-b')
[m,i]=min(out);
    % Optimal index of refraction
```

```

optIOR=index(i)
plot([optIOR,optIOR],[0,max(out)],'-k')
xlabel('Index of Refraction (n)'); ylabel('Tau [m^2]')

%Figure 2: Epsilon vs. Theta
figure(2); hold on
for j=1:n+1
    out(j)=Epsilon(theta(j),index(i));
end
plot(theta*180/pi,out*180/pi,'-k')
M(1)=max(out); plot([0,90],[0,0],'-k')
for j=1:n+1
    out(j)=Epsilon(theta(j),1.77);
end
M(2)=max(out); plot(theta*180/pi,out*180/pi,'-r')
for j=1:n+1
    out(j)=Epsilon(theta(j),1.85);
end
M(3)=max(out); plot(theta*180/pi,out*180/pi,'-g')
for j=1:n+1
    out(j)=Epsilon(theta(j),1.55);
end
M(4)=max(out); plot(theta*180/pi,out*180/pi,'-b')
for j=1:n+1
    out(j)=Epsilon(theta(j),1.65);
end
M(5)=max(out); plot(theta*180/pi,out*180/pi,'-m')
xlabel('Theta [deg]'); ylabel('Epsilon [deg]')
D(1)=fsolve(@(theta) Epsilon(theta,index(i)),pi/2);
D(2)=fsolve(@(theta) Epsilon(theta,1.77),pi/2);
D(3)=fsolve(@(theta) Epsilon(theta,1.85),pi/2);
D(4)=fsolve(@(theta) Epsilon(theta,1.55),pi/2);
D(5)=fsolve(@(theta) Epsilon(theta,1.65),pi/2);
%M(i) is the largest angular error
%D(i) is the theta at which the angular error becomes negative

%Figure 3: Omega vs. Theta
figure(3); hold on
for j=1:n+1

```



```

        out(j)=Omega(theta(j),index(i));
    end
    plot(theta*180/pi,out*180/pi,'-k')
    for j=1:n+1
        out(j)=Omega(theta(j),1.77);
    end
    plot(theta*180/pi,out*180/pi,'-r')
    for j=1:n+1
        out(j)=Omega(theta(j),1.85);
    end
    plot(theta*180/pi,out*180/pi,'-g')
    for j=1:n+1
        out(j)=Omega(theta(j),1.55);
    end
    plot(theta*180/pi,out*180/pi,'-b')
    for j=1:n+1
        out(j)=Omega(theta(j),1.65);
    end
    plot(theta*180/pi,out*180/pi,'-m')
    xlabel('Theta [deg]'); ylabel('Omega [deg^2]')

```

```

    % Figure 4: Projected Areas
    figure(4); hold on
    axis equal; axis off
    theta=0:1/n*2*pi:2*pi;
    r=1; plot(r*sin(theta),r*cos(theta),'-k')
    r=cos(pi/2-D(1)); plot(r*sin(theta),r*cos(theta),'-k')
    r=cos(pi/2-D(2)); plot(r*sin(theta),r*cos(theta),'-r')
    r=cos(pi/2-D(3)); plot(r*sin(theta),r*cos(theta),'-g')
    r=cos(pi/2-D(4)); plot(r*sin(theta),r*cos(theta),'-b')
    r=cos(pi/2-D(5)); plot(r*sin(theta),r*cos(theta),'-m')
    r=cos(pi/2+M(1)); plot(r*sin(theta),r*cos(theta),'-k')
    r=cos(pi/2+M(2)); plot(r*sin(theta),r*cos(theta),'-r')
    r=cos(pi/2+M(3)); plot(r*sin(theta),r*cos(theta),'-g')
    r=cos(pi/2+M(4)); plot(r*sin(theta),r*cos(theta),'-b')
    r=cos(pi/2+M(5)); plot(r*sin(theta),r*cos(theta),'-m')

```

Appendix B

Multidimensional image encoding

A set of MATLAB scripts and functions encode three dimensional images.

B.1 Main Script

```
close all; clear all; figure(1); axis equal; axis off; hold on
% -----

n=8;           % pixel array parameter
pD = 0.125;   % pixel diameter in inches
pR = pD/2;    % pixel radius in inches
bD = 1;       % ball diameter in inches
bR = bD/2;    % ball radius in inches
bH = -1.5*bR; % distance between the screen and the ball center
% -----

[xpts,ypts,k]=pixH(n,pD); % Calculate pixel locations (Hex)
% -----

p=1e2;        % number of points per object
q=(0:1:p)/p*2*pi; % map points between 0 and 2*pi
t=0.5;        % line thickness for graphs
grey=[0.3 0.3 0.3]; % grey color RGB
```

```

% graph bounding box with width of (width+pD) and height of (height+pD)
% centered on the origin (fill black)

% graph the circular pixel boundaries (fill black)

for i=1:k
    x=xpts(i)+pD/2*cos(q);
    y=ypts(i)+pD/2*sin(q);
    fill(x,y,'w');
end
% -----

% draw objects

%define surface (full sphere)

for i=1:k
    p=1e2;    % number of points per object
    [fillx,filly,fillz]=fullSphere(bD,bH,p);

    T=sqrt(xpts(i)^2+ypts(i)^2+bH^2-bR^2);    % define acceptance limit

    [xplot,yplot]=angles4(fillx,filly,fillz,xpts(i),ypts(i),pR,T);

    h=convhull(xplot,yplot);

    xplot=xplot(h);
    yplot=yplot(h);

    plot(xplot,yplot,'Color','k');
end
for count=1:4

    % define surface (quarter sphere)

    p=1e2/2;    % number of points per object
    [fillx,filly,fillz]=quartSphere(bD,bH,count,p);

    % map object to pixel and plot

```

```

for i=1:k
    T=sqrt(xpts(i)^2+ypts(i)^2+bH^2-bR^2);    % define acceptance limit

    % project object onto pixel

    [xplot,yplot]=angles4(fillx,filly,fillz,xpts(i),ypts(i),pR,T);

    % draw projection in different colors

    if count==1
        fill(xplot,yplot,[1 0 0],'LineStyle','none');
    else if count==2
        fill(xplot,yplot,[1 1 0],'LineStyle','none');
    else if count==3
        fill(xplot,yplot,[0 0 1],'LineStyle','none');
    else
        fill(xplot,yplot,[0 1 0],'LineStyle','none');
    end
    end
end
end
end
end

```

B.2 Function array4: Mapping Data to Imaging Plane

```

function [xnew,ynew] = angles4(xpts,ypts,zpts,x,y,pR,l)

k=[];
theta=atan((y-ypts)/(x-xpts));

for i=1:length(xpts)
    if xpts(i)-x<=0
        theta(i)=theta(i)+pi;
    end
    if sqrt((xpts(i)-x)^2+(ypts(i)-y)^2+zpts(i)^2)<=l && zpts(i)<=0

```

```

        k=[k i];
    end
    if sqrt((xpts(i)-x)^2+(ypts(i)-y)^2+zpts(i)^2)>=1 && zpts(i)>=0
        k=[k i];
    end
end
end
alpha=atan(zpts./sqrt((xpts-x).^2+(ypts-y).^2));
for i=1:length(xpts)
    if zpts(i)>=0
        alpha(i)=alpha(i)+pi;
    end
end
end

l=pR.*cos(alpha);
xnew=x+l.*cos(theta);
ynew=y+l.*sin(theta);
xnew=xnew(k);
ynew=ynew(k);

```

B.3 Function pixH: Hexagonal Grid Layout

```

function [xpts,ypts,k]=pixH(n,pD)

k=hexn(n);
xpts=zeros(1,k);
ypts=zeros(1,k);
a=0;
for i=1:n
    for j=1:(n+i-1)
        ypts(1,a+j)=pD*(n-i)*sqrt(3)/2;
        ypts(1,end+1-a-j)=-ypts(1,a+j);
        xpts(1,a+j)=pD*(n+i-2)*(j-1)/(n+i-2)-pD*(n+i-2)/2;
        xpts(1,end+1-a-j)=-xpts(1,a+j);
    end
    a=a+(n+i-1);
end
end

```

Appendix C

Encoded Arrays

Test graphics encoding was done for three different arrays, a rectangular array 13 x 13 pixels and two hexagonal arrays with side length 8 and 16 respectively. An inset ball in four colors was used as a test model and was modeled using a variety of depths. Given a 1/8" pixel size, the ball was typically made to be 1" in diameter, 2" for the large hexagonal grid.

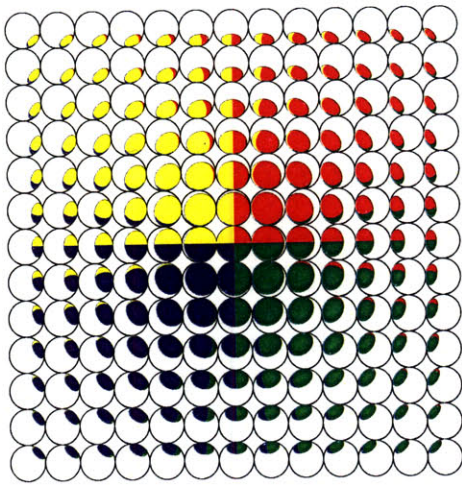


Figure C-1: Encoded image of inset sphere for rectangular grid, center one radius from the viewing plane.

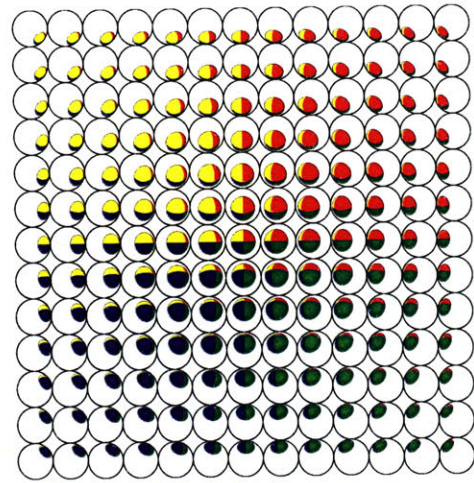


Figure C-2: Encoded image of inset sphere for rectangular grid, center one and a half radiuses from the viewing plane.

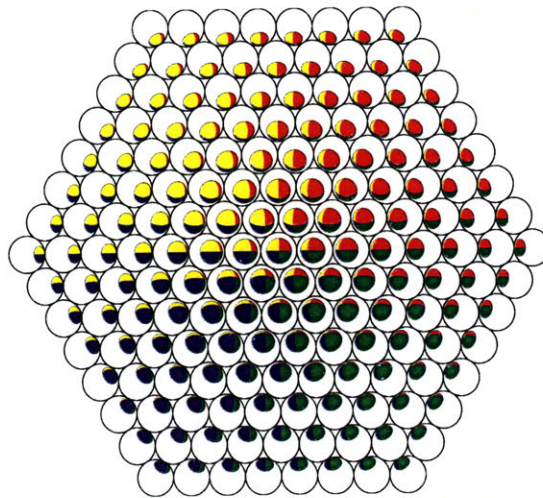


Figure C-3: Encoded image of inset sphere for a small hexagonal grid with center one and a half radii from the viewing plane.

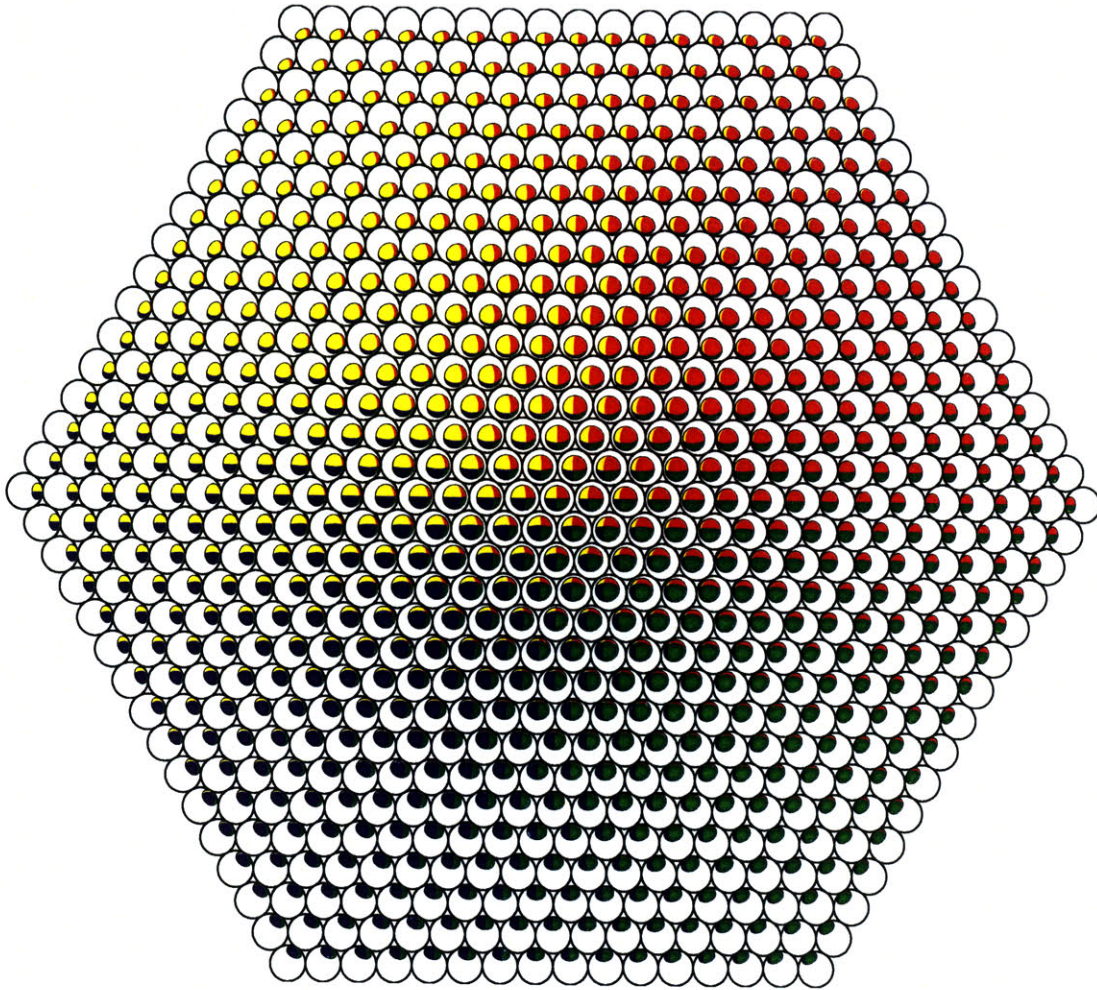


Figure C-4: Encoded image of inset sphere for a large hexagonal grid with center one and a half radii from the viewing plane.

Bibliography

- [1] Martin Fuchs, Ramesh Raskar, Hans-Peter Seidel, and Hendrik P.A. Lensch. Towards passive 6d reflectance field displays. *ACM Transactions on Graphics*, 27(3), August 2008.
- [2] Roman Ilinsky. Gradient-index meniscus lens free of spherical aberration. *Journal of Optics A: Pure and Applied Optics*, 2(5):449451, 2000.
- [3] Andrew Jones, Ian McDowall, Hideshi Yamada, Mark Bolas, and Paul Debevec. Rendering for an interactive 360deg light field display. *SIGGRAPH Papers Proceedings*, 2007.
- [4] Hwi Kim, Joonku Hahn, and ByoungHo Lee. The use of a negative index planoconcave lens array for wide-viewing angle integral imaging. *Optics Express*, 16(26), December 2008.
- [5] M. G. Lippmann. La photographie integrale. *French Academy of Sciences*, March 1908.