Displaying Real 3-D Object Images Using a Computer-Generated Hologram

A thesis presented in partial fulfillment of the requirements

for the degree of

Master of Science in Computer Science at Massey University

Jingzhou Yuan

1998

ACKNOWLEDGEMENTS

I am indebted to my supervisor, Dr. John Hudson of the Department of Computer Science, Massey university, for his invaluable advice, assistance throughout this study. His patience and encouragement are greatly appreciated.

I also wish to thank:

Professor R. C. O'Driscoll of the Department of Physics, Massey University, for providing the facilities for my experiments and helping me operate the instruments.

Miss Heather North, who was a PhD student of the Department of Physics, Massey University, for helping me choose the correct recording materials.

I would also like to express my sincere thanks to the staffs of the Department of Computer Science, Massey University for providing the facilities for this study.

Finally, I would like to express thanks to my wife, for her patience, help and encouragement during the course of this study.

CONTENTS

INDEX OF FIGURES

ABSTRACT

The magic of an optical hologram that produced by optical system offer us a never ending sense of wonderment. The images reconstructed from an optical hologram exhibit all of the three dimensional properties with full, rich perspective effects, enabling us to catch sight of an object behind another by mere tilt of the head.

Computer-generated holograms, synthetic holograms and computer holograms are terms used to refer to a class of holograms that are produced as graphical output from a digital computer. It has been reported that a computer-generated hologram can also yield a three dimensional image. The main advantage of the computer-generated hologram is that it can used to generate a three dimensional image of an object that may not physically exist. But can a computer-generated hologram be used as a three dimensional display device?

This thesis examines the ability of a computer-generated hologram as a three dimensional display device. Many techniques have been used to produce computer-generated holograms. Mathematical descriptions of computer-generated holograms are discussed. The quality of the images reconstructed from these computer-generated holograms are examined. The computation time for producing these computer-generated holograms are compared.

Chapter 1

Introduction

1.1 Computer-generated holograms

Computer-generated holograms, synthetic holograms and computer holograms are terms used to refer to a class of holograms that are produced as graphical output from a digital computer. Up to now, the digital computer has been widely used for producing realistic pictures, but these pictures like ordinary photographs are just two-dimensional representations of three-dimensional reality; the viewer must use subjective, prior knowledge to determine the true nature of the scene depicted in the photograph. A computer-generated hologram can inherit the most important properties from an optical hologram. It can really yield a three-dimensional image.

The true three-dimensional image reconstructed by a hologram exhibits all the properties that a viewer would see if he were looking at the real object through a window the size of the photograph: for example, if something in the foreground blocks his view of the background, the viewer can look at the scene from another angle and see around the foreground object. Shapes which appear to be in the same plane in a conventional photograph can be seen in their real planes, and the observer can examine those planes by looking at the hologram from several viewpoints. Examples of these properties are shown in Figure 1.1 (a)--(c).

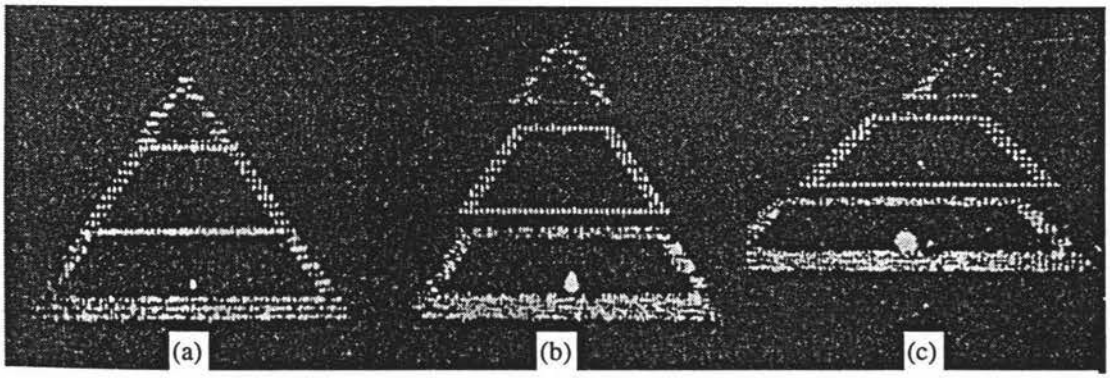


Figure 1.1 One image--three views. Notice the difference between (a), (b) and (c). An observer of the computer-generated hologram standing in front of it perceives the three images consecutively by shifting his/her head slightly [Lesem, Hirsch, and Jordan, Jr., 1969].

In general, for a synthesizing hologram, a computer is used to calculate the wave front propagation according to the object definition. The computation results is plotted

on a piece of paper, then it is photo-reduced to form a computer-generated hologram. The reduced size of the computer-generated hologram is typically about $5\times5mm^2$ depended on the output devices. Of course, this size is quite small comparing with the size of other display device such as an ordinary CRT. Then the hologram is illuminated by a laser beam in a dark room for image reconstruction. This is normal procedure for producing a computer-generated hologram.

1.2 Objective of this research

The purpose of this research is to investigate the abilities of a computer-generated hologram as a display device. Indeed, a computer-generated hologram can yield a three-dimensional image by illuminating it with a laser beam, but can we use white light to reconstruct an image from a computer-generated hologram? Could the image be reconstructed in normal conditions? What devices do we need to do so? How long it will take to produce a computer-generated hologram? How large a computer-generated hologram can be produced by existing device? This research tries to obtain the answers of these questions.

Since 1966, because of the interest in coherent optics and the availability of digital computers, this special field of using computer and graphic devices to make holograms has attracted the attention of many researchers and has resulted in many publications in the scientific journals. Techniques for producing computer-generated holograms have been described in these publications. Unfortunately, the computer-generated holograms described by these techniques were recorded on very high resolution film, such as Kodak 649F which has greater than 2000 [cyc./mm] resolution, laser light source was used for image reconstruction. Some of them required very large number of computations. Most objects reconstructed from the computer-generated holograms are simple objects such as letters made up of lines, or simply a frame box. There are no descriptions about how to generate a very complicated object such as an object consisting of many curve surfaces.

In this research, in order to find a better way for producing a computer-generated hologram from which a three-dimensional image can be reconstructed, different techniques have been used for this purpose. Some of these techniques have been modified for reconstructing a complicated image. The results have been compared according to my experiments.

1.3 Overview of the Thesis

Chapter 2 previews optical holography, optical holograms and computer-generated holograms. The mathematical description of holograms and their DFT (Discrete Fourier Transform) forms are presented including the representation of Fourier holograms in terms of DFTs and the representation of Fresnel holograms in terms of DFTs. The FFT (Fast Fourier Transform) is also described.

Chapter 3 describes the techniques for making computer generated holograms including the Fresnel zone plate method, the Detour Phase hologram, Burch's method, and Houng's technique.

Chapter 4 describes how to use Constructive Solid Geometry to represent an object. How to use ray tracing to obtain the readings for producing a computer generate hologram is also discussed.

Chapter 5 shows experimental results with synthesized holograms. It also shows how the limitation of the dynamic range and dimension of the hologram effect the quality of the image reconstructed from a computer-generated hologram.

Chapter 6 Conclusions and remarks

Chapter 2

Literature Review and Mathematical Description of the Holograms

2.1 Review Of Optical Holography and Optical Hologram

Optical holography was conceived and initially developed by the greatest scientist Dr. Dennis Gabor in 1948, more than a decade before the laser. This achievement (for which he received the Nobel Prize in 1971) would have remained no more than a scientist's interesting contribution to the wave optics without the laser [Chaimowicz 1989].

Accurately speaking, optical holography is the first truly three-dimensional image-making process, since it recreates what light waves actually do after reflecting from a real object. The hologram itself is but a thin flat piece of film or glass. In optical holography, a hologram is formed by the interference caused by reflected light from a "object" called the object beam encountering a pure beam (the reference beam) from the laser on a piece of photographic plate. The processed photographic plate is called a hologram. Because these kinds of holograms are produced by optical system, e.g. lens and laser source, we also refer them as optical holograms.

The unique characteristic of optical holography is the idea of recording the complete wave field, that is to say, both the amplitude and the phase of the light waves scattered by the object. Since all recording media respond only to the intensity, it is necessary to convert the phase information into variations of intensity [Hariharan, 1987]. There are numerous ways to make optical holograms, but perhaps the simplest method involves the use of the same beam of light from the laser both to illuminate the object. and the photographic plate. This was the type of transmission hologram first made by Leith and Upatnieks at the university of Michigan in 1962 [Leith and Upatnieks, 1962]. The Leith and Upatnieks scheme is shown in figure 2.1.

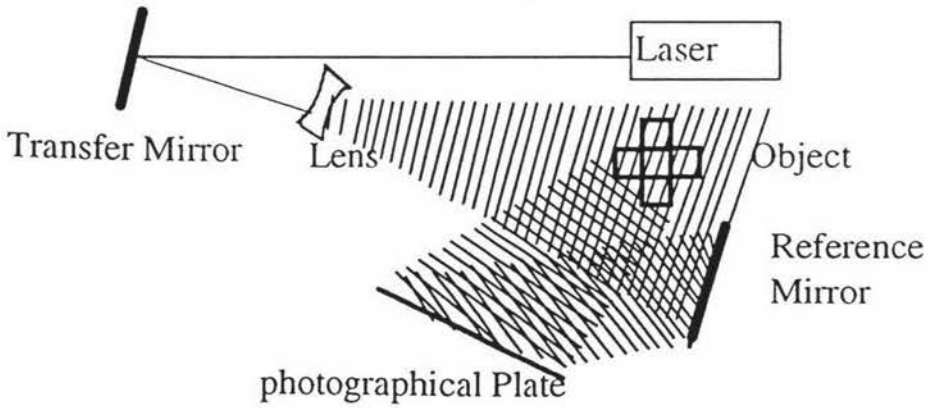


Figure 2.1. The Leith and Upatnieks scheme for making transmission hologram.

The interferometric recording of a wave field in a hologram is the first step of optical holography. This step is also called construction step. The second step is to reconstruct the wave field by diffraction (see appendix A). Without the reconstruction step, one can not observe any image from a optical hologram because it just contains interference fringes. The reconstruction step normally can be done by illuminating the hologram with the same laser beam, or a similar one. From the reconstruction of an optical hologram, one can observe a three-dimensional image. Some types of optical holograms can be reconstructed by ordinary sunlight, e.g. a rainbow hologram (See for example, reference Unterseher, Hansen and Schlesinger, 1982).

As shown in figure. 2.1, optical holography like ordinary photography, projects the real three-dimensional object onto the photographic plate, therefore , the object must really exist in the real world. Without the real objects, we could not make optical holograms.

2.2 The Distinctions Between Optical Holograms and Computer-Generated Holograms

The main difference between optical hologram and computer-generated hologram is that the construction step of a computer-generated hologram is performed synthetically supported by a digital computer. In optical holography, in order to make an optical hologram, the object must exist in the real world. Without it we could not make an optical hologram as mentioned in section 2.1. This restricts the applications of optical holograms in some areas. For example, we can use a optical hologram to test the visual effect of a car in space. But in order to make the optical hologram, first we have to build a model of the car. This is not convenient for some people particularly for parts designer! This shortcoming has no effects in computer-generated holograms because the object of a computer-generated hologram is just a mathematical representation. Given a mathematical description of a wave front or an object represented by an array of points, the computer can calculate the wave field scattered from the object and display the results on a computer screen or plot it on paper. The graphical output from the computer is photo-reduced to form the computer-generated hologram. Because the object is described mathematically in computer, therefore, it can be changed easily. This is where the strengths and possibilities of computer-generated holograms can be found.

Another distinction between a computer-generated hologram and an optical hologram is that in the off-axis reference beam transmission hologram as shown in figure 2.1, the amplitude transmittance of a transmission hologram recorded under ideal condition is proportional to

$$T(x,y) = \left| R e^{i2\pi\alpha x} + O(x,y) e^{i\varphi(x,y)} \right|^2$$
$$= R^2 + O^2(x,y) + 2RO(x,y)\cos[2\pi\alpha x - \varphi(x,y)]. \qquad (2.1)$$

In Eq. (2.1), $Re^{(i2\pi\alpha x)}$ represents the reference wave and $O(x,y)e^{i\varphi(x,y)}$ the object wave. $T(x,y)$ is the resulting intensity variation of the interference pattern between the two waves (for more details see section 3.7). In computer-generated holograms, the transmittance of the hologram and the object wave is not restricted to the relationship specified by Eq. (2.1) [Lee, 1980]. In fact, most of the work in computer-generated holograms has dealt with the problem of coding the complex object wave-front for convenient production on computer graphic devices. Coding as used here means the conversion of a complex valued function which represents the object into a real, non-negative function so that it can be recorded on a display device such as a CRT or a piece of paper, enable the displayed results to be photo-reduced to form a final computer-generated hologram which can be interacted with illuminating light for reconstruction of the original object.

In order to produce a computer-generated hologram the following steps should be considered:

(1) Defining the object The object can be in a sampled or continuous form in two or three dimensions.

(2) Converting the wave field scattered from the object to the hologram plane by a transformation (Fourier, Fresnel), which can be realized by inverse wave propagation methods, and display the result on a CRT or plot it on paper. Practical and physical constraints will guide the choice of procedure for realization. A coding scheme is applied to make the computer-generated hologram conform to requirements of the material and recording devices.

(3) Photo-reducing the result obtained from step 2 to the proper size. The processed photograph forms the computer-generated hologram.

(4) Proper illumination of the computer-generated hologram results in a reconstruction of the object (data) as a light distribution. The first and second steps will be discussed in more detail in Chapter 3 and 4. The specification of the object and hologram type leads to possible coding schemes from which to choose in the actual application.

In the far future, with a proper display device, the most obvious application of these computer-generated holograms is as a display device. A three-dimensional image of any mathematically defined object, whether it exists in real form or not, can be displayed visually: for example, mathematical models may be shown; designs and patterns may be tried out visually without the need for laborious model building.

patterns may be tried out visually without the need for laborious model building. Another potentially important application is to convert a record of a hologram made with non-visible illumination into a hologram which may be illuminated with visible light. It has also been suggested that a hologram could be used as a read-only storage device [Chamber, and Counttney-Pratt, 1966].

2.3 Review Of Computer-Generated Holograms

The first technique for making computer-generated holograms was described by Brown and Lohmann with their "detour phase hologram" in 1966 [Brown, and Lohmann, 1966]. Motivation included optical spatial filters, which then could be used for image enhancement and pattern recognition. They experimentally demonstrated two-dimensional objects. An interesting aspect of their technique is that the computer-generated hologram is made without explicit use of a reference wave or a bias (see next page Burch's method). Another interesting aspect of their hologram is that it was made by hand. The diffraction patterns from the four bars of the letter "E" were calculated separately and added, indicating that the computer-generated holograms can be produced inexpensively for the research purpose. The idea was to sample the Fourier transform of the wanted reconstruction equidistantly and divide the hologram into as many cells as sampling points. The complex number of each sampling point was coded into a pattern configuration within each cell. Its modulus was coded as the area of an aperture and its argument as the location of the aperture within the cell.

In their subsequent papers [Lohmann and Paris ,1967, Brown and Lohmann, 1969] a computer-driven plotter were used to produce the artwork for the computer-generated holograms. Their holograms first plotted on paper and then photo-reduced to form the final computer-generated hologram which can be viewed by illuminating it with a laser beam. Because their approach was prompted by the desirability of making binary spatial filter, only two dimensional object were considered.

Subsequent to Brown and Lohmann's work, a process for constructing a computer-generated hologram of three-dimensional objects by theoretically calculating and plotting a synthetic hologram was reported by Waters [Waters, 1966]. He observed the similarity between optical holograms and Fresnel zone plates, whose imaging properties are well known (see chapter 3, Waters's method), and used the Fresnel zone plate approach to calculate his hologram. The calculated result was plotted, using a mechanical plotting table. This plot was photo-reduced and illuminated with a coherent light to form reconstructed image. Waters succeeded in forming three-dimensional images of the edges of a tetrahedron. This computer-generated hologram had the

An alternative technique for making computer-generated holograms was described by Burch [Burch 1967]. Burch noticed that the terms R^2 and $O^2(x,y)$ in Eq. (2.1) do not contribute to the reconstruction of the object wave-front, he suggested the use of a constant bias, yielding the hologram

$$T_1(x,y) = K + 2RO(x,y)\cos[2\pi\alpha x - \varphi(x,y)] \qquad (2.2)$$

where K is a constant called bias function which is just large enough to make $T_1(x,y)$ positive for all (x,y), and R, $O(x,y)$, α and $\varphi(x,y)$ are the same as Eq. (2.1). Because the $T_1(x,y)$ can be sampled at regular intervals and therefore it is quite suitable for CRT display. Huang and Prasada [Huang, and Prasada, 1966] suggested a similar modification of Eq. (2.1) for making computer-generated holograms.

A more practical approach to the 3-D image display problem was suggested by King, Noll and Berry [King, Noll and Berry, 1970]. They used the computer to calculate the two-dimensional perspective view of the objects as it would appear on a projection screen at a particular viewing angle. many of these perspective projection images are recorded in a sequence on film. The film is then used in the optical system in Figure 2.3.1 to produce a composite hologram. Each frame in the film is sequentially imaged onto a diffuse screen. A narrow hologram about 3mm in width is recorded at a distance from the screen. After each exposure, the input film is advanced to the next frame and the hologram is incremented to the next position. In reconstruction because each eye observes a different perspective of the original object, this produces a stereoscopic image. As the composite hologram is moved before the viewer, he sees a 3-D object rotate in front of him. Because the hologram is composited by many vertical strips, this hologram has no vertical parallax. In applications where vertical parallax is desired, the computer can generate an array of n×n views of the object, and the vertical slit mask is replaced with a mask containing one square aperture, where n is equal to the width of the composite hologram divided by the width of the square aperture of the mask (assume that the composite hologram is a square). The resultant hologram will then contain an array of n×n square holograms instead of n vertical strips. No modifications are needed for the reconstruction. Figure 2.3.2 shows the setup of the composite hologram reconstruction.

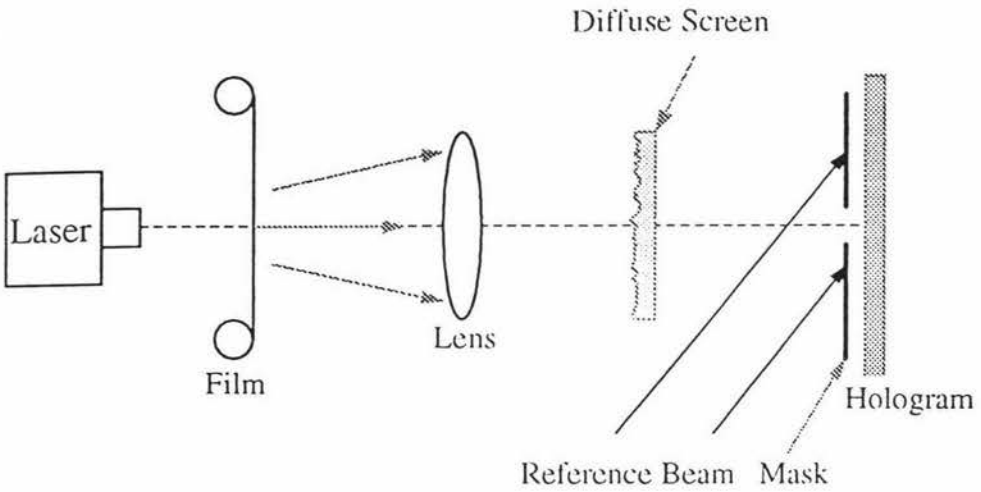


Figure 2.3.1 Optical setup for making a composite hologram.

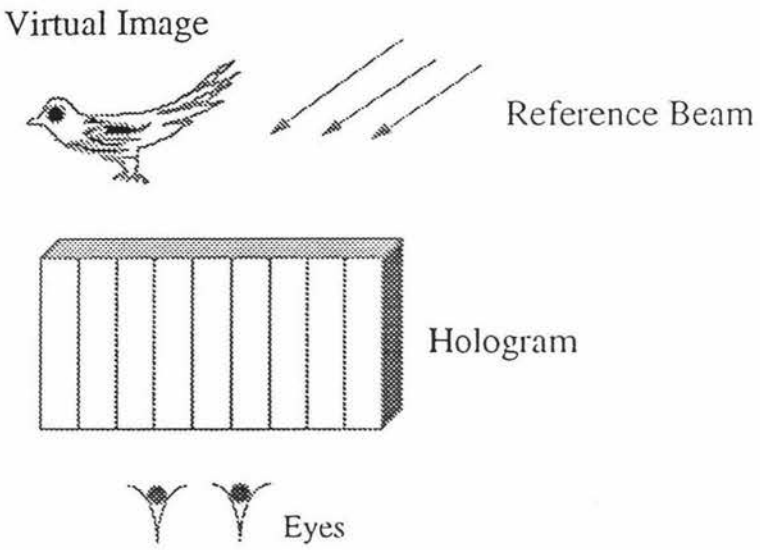


Figure 2.3.2 The composite hologram reconstructs as if it were a single hologram of a real object

Following the works of King, and *et al*, Yatagai [Yatagai, 1974,1976] has adopted this method to make computer-generated holograms to display 3-D images. Instead of recording the perspective projection images on the film, he converts the projection images into Fourier transform holograms (see section 2.4) which are then plotted on a large scale on a plotter. These holograms are then arranged in the same order as the projection images are obtained. The holograms along the horizontal direction correspond to the different perspective projection images. To increase the viewing aperture of the hologram each of the holograms is duplicated six times along the vertical direction. Therefore, the hologram will not have the parallax effect in the vertical direction. If vertical parallax is required, the hologram in the vertical direction have to be obtained from their corresponding perspective projection images. In

laser point source. The viewer looks through the holograms. Since the hologram are Fourier transform holograms, each eye of the viewer will see a different image of the object occurring at the same location. This also produces a stereoscopic image. By using holograms, the use of special viewing aids for fusing the two images to form the stereoscopic image is not needed. Yatagai's approach does not require further optical process to produce a composite hologram comparing with King, and *et al*'s approach. Some of the perspective projection images from Yatagai's holograms are shown in figure 2.3.3.

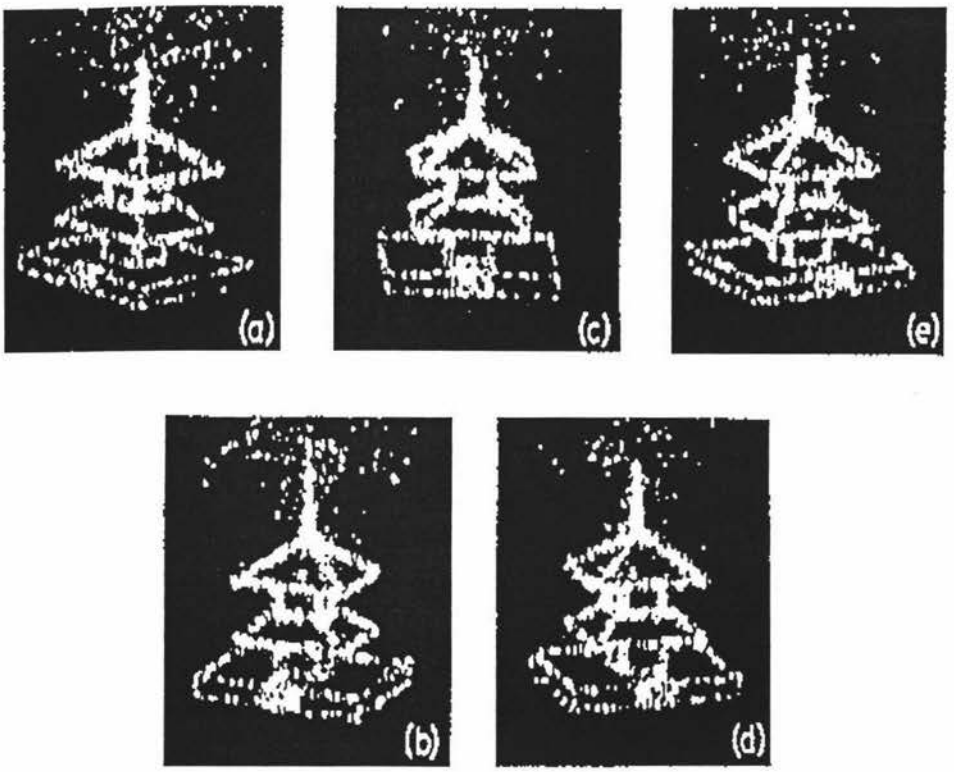


Figure 2.3.3 Perspective projection of a 3-D object reconstructed from the composited hologram [Yatagai, 1974].

2.4 Mathematical description of a computer-generated hologram

In order to choose a formal apparatus to solve the problem of synthesizing holograms for visualising objects which are specified by some mathematical description, and to understanding which is required of this description we will examine the scheme arrangement in figure 2.4. This is the general arrangement for a Leith-Upatnieks transmission type hologram. But first we introduce the basic notation and the coordinate systems to be used throughout the thesis. We then discuss the basic mathematics of making computer-generated holograms.

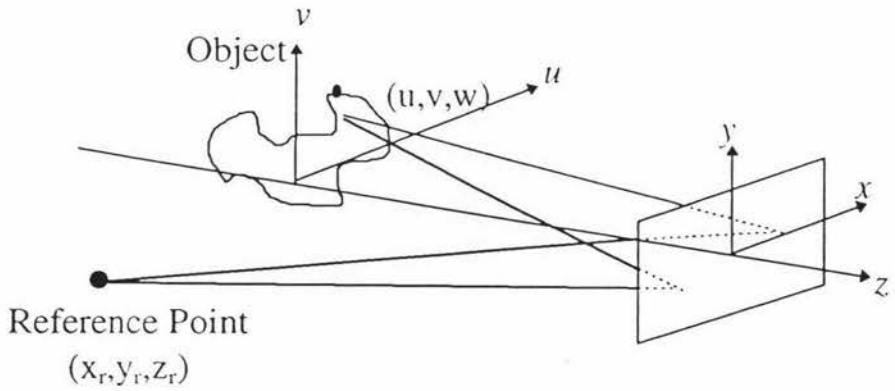


Figure 2.4 Generalized arrangement for recording a Leith-Upatnieks transmission hologram.

2.4.1 Notation

The origin of a Cartesian (x,y,z) coordinate system is centred in the recording medium. (Figure 2.4.1) Two-dimensional recording medium defines the x-y plane. An object space coordinate system is defined by (u,v,w). The u-v plane is parallel to the x-y plane and the distance between the two planes is d. the w-axis is the same with z-axis in the Cartesian coordinate system. A typical point in the object space coordinate system is defined by (u,v,w). If the reference beam is derived from a point source, the coordinate of this point will be denoted by (x_r,y_r,z_r),. If $z_r = \infty$, the reference beam is a plane wave and α_r will be the angle between the projection of the propagation vector **k** onto the x-z and the z-axis. Points in the image formed with the reconstructed wave front will be denoted by the image coordinate (x_i, y_i, z_i). The amplitude distribution of the light at the object will be denoted by $U(u,v,w)$. This field gives rise to a disturbance $O(x,y,z)$ at the hologram. The reference wave at the hologram can be described by $R(x,y,z)$. The time dependence of the fields has been omitted, and O and R are considered to be complex amplitudes. The final processed hologram will have a complex amplitude transmission $t(x,y,z)$ as a result of exposure with irradiance $I(x,y,z)=|H(x,y,z)|^2$, where

$$H(x,y,z)=O(x,y,z)+R(x,y,z) \tag{2.3}$$

is the total field at the hologram.

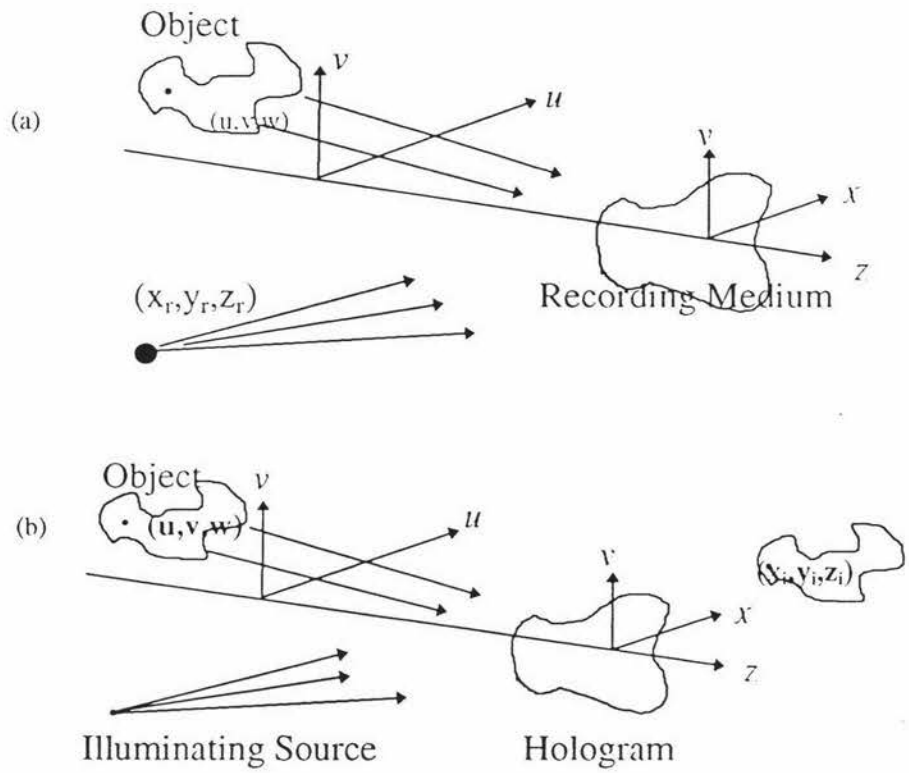


Figure 2.4.1 The coordinate system and notation.(a) recording the wavefront.
(b) Reconstructing the wavefront.

2.4.2 Mathematical description

Having a suitable notation, we can proceed to describe mathematically the recording of the hologram. Consider the arrangement of shown in figure 2.4. The reference point at (x_r,y_r,z_r) is the origin of a spherical reference wave. The object is defined by the surface S generally not including the point (x_r,y_r,z_r). No point of which lies on the line joining the reference point and any point on the recording medium. A typical point (u,v,w), lying on the surface *S*, gives rise to a spherical wave. The disturbance at the hologram caused by the object is the sum of the spherical wave from each point of the object:

$$O(x,y,z)=\frac{-i}{2\lambda}\iint_S U(u,v,w)(1+\cos\theta)\frac{exp(ikr)}{r}dudvdw \qquad (2.4)$$

where the integration is carried out over the surface S. The exponential expresses the change in phase of the wave travelling from (u, v, w) to (x, y, z) ; $1/r$ is the reduction in amplitude due to spreading of the spherical wave, and r is the distance between the two points (u,v,w) and (x,y,z). The $k=2\pi/\lambda$ is the magnitude of the wave vector with λ the wave length of the light. The factor $(1+\cos\theta)/2$ is the obliquity factor, θ being the angle between the normal to the surface at (u,v,w) and the line joining (u,v,w) with (x,y,z). The factor $-i$ with $i=\sqrt{-1}$ appears because the Huygen's wavelets are assumed advanced in phase by $\pi/2$ [Smith 1975].

Equation (2.4) is the basic mathematical description for synthesizing holograms. In principle, if we know the function $U(u,v,w)$ and the surface S, then we can calculate the function $O(x,y,z)$ according to Eq. (2.4) and record the result in a form which allow an interaction with light for reconstruction of the object.

However in general case, the evaluation of the integral in Eq. (2.4). is a very complicated problem. The kernel of this equation is $exp(ikr)/r$ where r is function of both the space coordinate system and the Cartesian coordinate system.

$$r=\sqrt{(x-u)^2+(y-v)^2+(z-w)^2} \qquad (2.5)$$

If we consider that the hologram is a plane and centred at the origin of the Cartesian coordinate system , then $z = 0$ and

$$r=\sqrt{(x-u)^2+(y-v)^2+w^2} \qquad (2.6)$$

when $w^2 >> u^2, v^2$ we may expand Eq. (2.6) as

$$r \equiv w+\frac{(u^2+v^2)}{2w}+\frac{(x^2+y^2)}{2w}-\frac{xu}{w}-\frac{yv}{w}+\ldots \qquad (2.7)$$

and Eq. (2.4) can be approximately written as

$$O_1(x,y)\approx\iint_S U(u,v,w)\frac{exp(ikr)}{r}dudvdw \qquad (2.8)$$

Where $r = w + \frac{(u^2+v^2)}{2w} + \frac{(x^2+y^2)}{2w} - \frac{xu}{w} - \frac{yv}{w}$.

The obliquity factor $(1+\cos\theta)/2$ is omitted, because at this condition it nearly equals to one. Anyhow the evaluation of this equation is still a difficult problem. If the object is defined in a plane and the geometry dimensions of the object are small in comparison with d which is the distance from the object plane to the hologram plane, then Eq. (2.8) can be simplified further:

$$O_1(x,y) \approx \iint_{(u,v)} U_1(u,v)\frac{\exp(ikr)}{d}dudv \tag{2.9}$$

where $r = d + \frac{(x-u)^2+(y-v)^2}{2d}$,

for $\frac{(x-u)^2+(y-v)^2|_{\max}}{d^2} < \sqrt{\frac{4K\lambda}{d}}$. (2.10)

Where K is the coefficient of the permissible phase error (the error is $K\pi$) in the propagation of the argument of the exponential function in Eq. (2.4) [Yaroslarskii, and Merlyakov, 1980].

In this case the integral in Eq. (2.9) converts into a Fresnel integral:

$$O_1(x,y) \approx \frac{exp(ikd)}{d}\iint_{(u,v)} U_1(u,v)exp\left\{\frac{ik}{2d}\left[(x-u)^2+(y-v)^2\right]\right\}dudv \tag{2.11}$$

The constant factor $\frac{exp(ikd)}{d}$ is the same for the entire hologram, and has no effect for synthesizing holograms, thus Eq. (2.11) can be simplified as

$$O_b(x,y) \approx \iint_{(u,v)} U_1(u,v)exp\left\{\frac{ik}{2d}\left[(x-u)^2+(y-v)^2\right]\right\}dudv \tag{2.12}$$

Those computer-generated holograms which are synthesized in accordance with this relation will be called "synthesized Fresnel holograms".

A further simplification is possible if

$$\pi(x^2+y^2)/(\lambda d) << 1,\ \pi(u^2+v^2)/(\lambda d) << 1, \tag{2.13}$$

so that these phase components can be ignored in the integral in Eq. (2.12). Then (2.12) converts into a Fourier integral:

$$O_c(x,y) \approx \iint_{(u,v)} U_1(u,v)\exp\left\{\frac{-ik}{d}(xu+yv)\right\}dudv. \tag{2.14}$$

We will label those holograms which are synthesized in accordance with (2.14) "Fourier holograms". It follows from (2.14) that a Fourier hologram is, within unimportant amplitude factor, the spatial Fourier spectrum of the function $U_1(u,v)$

$$O_o(v_x,v_y) = \iint_{(u,v)} U_1(u,v)exp\left[-i2\pi\left(v_x u + v_y v\right)\right]dudv \tag{2.15}$$

taken along the coordinate v_x, v_y in the scale $v_x = x/\lambda d$, $v_y = y/\lambda d$,

$$O(x,y) = O_o(v_x = x/\lambda d, v_y = y/\lambda d). \tag{2.16}$$

From the computation standpoint, however, it is convenient to express Eq. (2.12) in terms of a Fourier transformation:

$$O_b(x,y) = exp\left[\frac{i\pi}{\lambda d}\left(x^2+y^2\right)\right]\iint_{(u,v)} U_1(u,v)exp\left[\frac{i\pi}{\lambda d}\left(u^2+v^2\right)\right]\times$$
$$\times\exp\left[\frac{-i2\pi}{\lambda d}(xu+yv)\right]dudv. \tag{2.17}$$

Finally , we note that by transforming from a three-dimensional problem to a two-dimensional problem we are, strictly speaking, losing the capability of accurately determining the depth of the surface relief of the object. Even the Fresnel hologram contains only the distance from the object to the hologram plane, not the depth of the relief. Nevertheless, it is possible to synthesize a field which can reconstruct the object under certain conditions. Then the most important property of the holographic visualisation, the natural way in which the object is observed, is retained . Certain artificial methods can be suggested for conveying the relief.

2.5 Representation of a Fourier Hologram in terms of a DFT (Discrete Fourier Transformation)

Equation (2.15) and (2.17) are the start points to synthesize the computer-generated Fourier hologram and Fresnel hologram. For digital computation, the transformation of the function $O_b(x,y)$ and $O_o(x,y)$ must be discretized, because a computer can only be used to calculate discrete values.

The function $U_1(u,v)$ which denotes the amplitude distribution of the light at the object can be simply thought of as defining the object. When the object $U_1(u,v)$ is defined by a series of points separated by distances $(\delta u,\delta v)$ in the u and v direction respectively, we can express Eq. (2.14) as the discrete Fourier transformation (DFT). For convenience, we now represent the object as $U(u,v)$ instead of $U_1(u,v)$. Assume that the object is bounded by a rectangle box or is non-vanishing in the rectangle area [-U/2,U/2,-V/2,V/2] as shown in figure 2.5, then we have:

$$U_{mn}=U(m\delta u, n\delta v), \qquad (2.18)$$
$$-(M/2) \le m \le (M/2)-1,$$
$$-(N/2) \le n \le (N/2)-1,$$

where M, N are even integer numbers and

$$M=U/\delta u \text{ and } N=V/\delta u. \qquad (2.19)$$

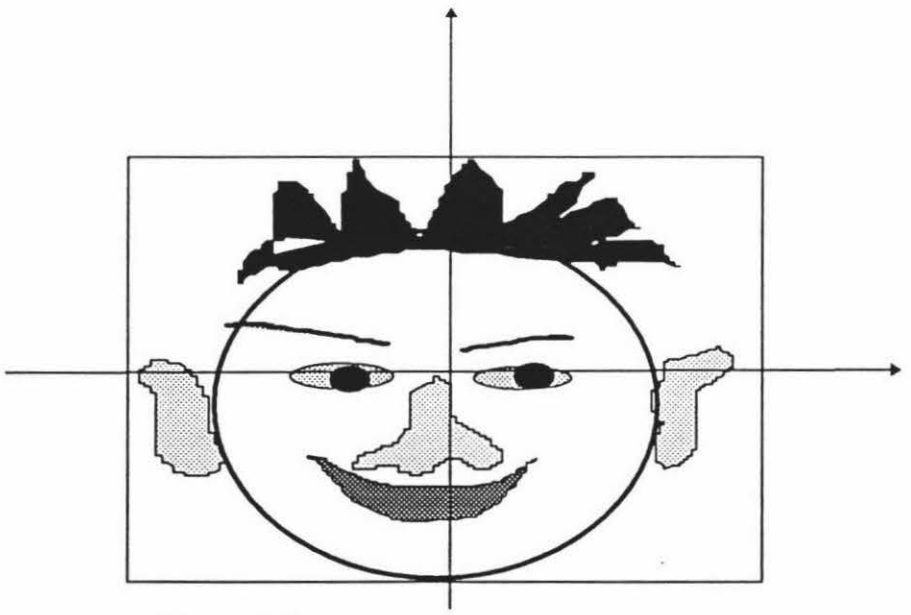


Figure 2.5.1 An object bounded by a box

Let us assume again that the hologram is bounded by the rectangle area say, [-X/2,X/2,-Y/2,Y/2], if we sample in the hologram at distances $(\delta x, \delta y)$, where

$$\delta x=1/U\ ,\ \delta y=1/V, \tag{2.20}$$

then the sampling is sufficient according to Nyquist's sampling theorem[†] . The sampling numbers in the hologram are J and K in the X and Y direction of the hologram plane respectively, where

$$J=X/\delta x,\ K=Y/\delta y, \tag{2.21}$$

If we define the distance between two points of the object as

$$\delta u=1/X, \qquad \delta v=1/Y, \tag{2.22}$$

then we have $M=J$, $N=K$, thus the function $O_o(v_x,v_y)$ in Eq. (2.15), which is the wave distribution at the hologram plane by the object defined by $U_{mn}=U(m\delta u, n\delta v)$, can be rewritten as the discrete Fourier transform (DFT).

$$O_{jk}=\sum_{m=-M/2}^{(M/2)-1}\sum_{n=-N/2}^{(N/2)-1}U_{mn}exp\left[-i2\pi\left(mj/M+nk/N\right)\right], \tag{2.23a}$$

$$O_{jk}=O_o(j\delta x,k\delta y), \tag{2.23b}$$
$$-M/2\le j\le M/2-1, \tag{2.23c}$$
$$-N/2\le k\le N/2-1. \tag{2.23d}$$

It is not necessary to restrict ourselves to that $\delta u=1/X$, and $\delta v=1/Y$ for an object which consists only points. In fact as long as $\delta u=$ constant, $\delta v=$ constant, and Eq. (2.20) is satisfied then Eq. (2.23a) is correct.

If the object does not consist of points, but a continuous description, then the object must be sampled since only discrete Fourier transform can be performed by a

[†] Nyquist sampling theorem: Sampling theory tells us that a signal can be properly reconstructed from its samples if the original signal is sampled at a frequency that is greater than twice f_h, the highest frequency component in its spectrum. This lower bound on the sampling rate is known as Nyquist rate.

computer. In the object plane, a sampled version of the $U(u,v)$ is introduced as followings:

$$U(u,v) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} U(m\delta u, n\delta v)\delta(u - m\delta u, v - n\delta v) \tag{2.24}$$

where M and N are sample numbers in the u, v direction respectively; $(\delta u, \delta v)$ are sample distances and $\delta(u,v)$ is the **impulse function**, also called the Dirac delta function (see for example, reference Briggs and Henson, p45-47, 1995). As long as

$$\delta u = 1/\mathrm{X} \quad \text{and} \quad \delta v = 1/\mathrm{Y} \tag{2.25}$$

(assume that the reduced dimensions of the hologram is bounded by the rectangular area [-X/2,X/2,-Y/2,Y/2]) the sample is sufficient according to the Nyquist's sampling theorem and we have

$$M=\mathrm{UX} \text{ and } N=\mathrm{VY}. \tag{2.26}$$

Substitution of (2.24) into (2.15) shows that the Fourier hologram of the discrete object
$U_{mn} = U(m\delta u, n\delta v)$ can be calculated as the finite sum

$$O_o(v_x, v_y) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} U_{mn} \exp\left[-i2\pi\left(\frac{mv_x}{\mathrm{X}} + \frac{nv_y}{\mathrm{Y}}\right)\right]. \tag{2.27}$$

Extend the digitisation in (2.27) to v_x and v_y in the same fashion and in analogy to Eq. (2.25),

$$O_o(v_x, v_y) = \sum_{j=-J/2}^{J/2-1} \sum_{k=-N/2}^{K/2-1} O_o(j\delta v_x, k\delta v_y)\delta(v_x - j\delta v_x, v_y - k\delta v_y) \tag{2.28}$$

where $O_o(j\delta v_x, k\delta v_y)$ are the readings of $O_o(v_x, v_y)$ taken on a rectangular raster with steps δv_x, δv_y along the coordinate v_x, v_y, and

$$O_{ojk} = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} U_{mn} \exp\left[-i2\pi\left(\frac{mj\delta v_x}{\mathrm{X}} + \frac{nk\delta v_y}{\mathrm{Y}}\right)\right], \tag{2.29}$$

with $\delta v_x = 1/\mathrm{U}$ and $\delta v_y = 1/\mathrm{V}$, thus the numbers of samples in the X and Y directions are J=XU and K=YV, respectively. From (2.26) we have

$M=J,\ N=K,$ (2.30)

substitute (2.30) into (2.29) we get

$$O_{ojk} = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} U_{mn} \exp\left[-i2\pi\left(\frac{mj}{M} + \frac{nk}{N}\right)\right]. \tag{2.31}$$

Equation (2.23a) is equal to (2.31) and both of them are two dimensional discrete Fourier transformations. A discrete Fourier transformation(DFT) is related to an integral Fourier transformation, but it has certain distinct features which stem from the fact that the metric of the argument space is an integer[Yaroslarskii, and Merlyakov, 1980]. A two dimensional DFT can be calculated as two successive one-dimensional Fourier transformations:

$$O^0{}_{jl} = \sum_{m=-M/2}^{M/2-1} U_{ml} \exp[-i2\pi(mj / M)], \tag{2.32}$$

$$O_{jk} = \sum_{l=-N/2}^{N/2-1} O^o{}_{jl} \exp[-i2\pi(lk / N)]. \tag{2.33}$$

From the computation stand point, the calculation of a computer-generated hologram from Eq.(2.23a) or from Eq.(2.32) and (2.33) will needs a large number of computations. But there are algorithms that can be used to calculate the DFT fast. In section 2.7, we will describe one of these algorithms.

2.6 Representation of a Fresnel Hologram in terms of a DFT

Now we consider a discrete representation of a Fresnel hologram of Eq. (2.17). In general, the digitization of Eq. (2.17) is a very hard problem, therefore, we restrict ourselves that the object is only defined by a series of points and the hologram is also defined by a series of points.

Assume that points separated by distances $(\delta u, \delta v)$ in the u and v direction respectively, in the object plane, since the object is bounded by [-U/2,U/2,-V/2,V/2], then we can rewrite Eq. (2.17) as

$$O(x,y)=\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}U_{mn}\exp\left[\frac{i\pi}{\lambda d}((x-m\delta u)^2+(y-n\delta v)^2)\right], \quad (2.34)$$

where

$$U_{mn}=U(m\delta u, n\delta v), \quad (2.35)$$

$-(M/2) \le m \le (M/2)-1,$

$-(N/2) \le n \le (N/2)-1,$

and with M, N are even integer numbers and

$$M=U/\delta u,\ N=V/\delta u, \quad (2.36)$$

Now consider the function

$$\varphi(x,y)=\pi((x-u)^2+(y-v)^2)/(\lambda d), \quad (2.37)$$

This function is a spherical wave front and has phase variation given by Eq. (2.37). This wavefront has spatial frequencies along the X-direction and Y-direction given by:

$$V_x=(x-u)/(\lambda d), V_y=(y-v)/(\lambda d) \quad (2.38)$$

The spatial frequencies of this wavefront are dependent on the location of the wavefront in a linear way. The maximum frequencies as can be seen, occur at the boundaries of the wavefront. Therefore, the spatial frequencies of the wavefront along the x-direction are bounded between $-(X+U)/(2\lambda d)$ and $(X+U)/(2\lambda d)$ and along the y-direction are bounded between $-(Y+V)/(2\lambda d)$ and $(Y+V)/(2\lambda d)$.

In this case, the maximum frequencies in the x-direction and y-direction are :

$$f_{maxx}=(X+U)/(2\lambda d), f_{maxy}=(Y+V)/(2\lambda d). \quad (2.39)$$

if we sample in the hologram at distances $(\delta x, \delta y)$, where

$$\delta x=1/(2f_{maxx}),\ \delta y=1/(2f_{maxy}), \quad (2.40)$$

then the sampling is sufficient according to Nyquist's sampling theorem. The sampling numbers in the hologram are J and K in the X and Y direction of the hologram plane respectively, where

$J=X/\delta x$, and $K=Y/\delta y$, (2.41)

thus we can rewrite Eq. (2.34) as:

$$O_{jk}=\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}U_{mn}\exp\left[\frac{i\pi}{\lambda d}((j\delta x-m\delta u)^2+(k\delta y-n\delta v)^2)\right], \quad (2.42a)$$

where

$$O_{jk}=O(j\delta x,k\delta y), \quad (2.42b)$$

$$-J/2\le j\le J/2-1, \quad (2.42c)$$

$$-K/2\le k\le K/2-1. \quad (2.42d)$$

Assume that

X=U, Y=V, $\delta u=2\delta x$, $\delta v=2\delta y$ and

let $\delta x/\sqrt{\lambda d}=1/Jx$, $\delta y/\sqrt{\lambda d}=1/Ky$ then we rewrite (2.42a) as

$$\begin{aligned}O_{jk}&=\exp\left[i\pi\left((j/Jx)^2+(k/Ky)^2\right)\right]\times\\&\times\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}\{U_{mn}\exp\left[\frac{i\pi}{\lambda d}((m\delta u)^2+(n\delta v)^2)\right]\times\\&\times\exp\left[-2i\pi(jm/J+kn/K)\right]\},\end{aligned} \quad (2.43)$$

where $M=U/\delta u$ and $N=V/\delta v$. The sum in Eq. (2.43) is a Fourier transform but can not be calculated by the fast Fourier transform because $M=J/2$ and $N=K/2$. In order to calculate Eq.(2.43) by the fast Fourier transform, we can redefine the object as:

$$U'_{mn}=\begin{cases}U_{m/2,n/2}, & \text{if } m \text{ and } n \text{ both are even number.}\\0, & \text{otherwise.}\end{cases}$$

and $M=J, N=K$, $-M/2\le m\le M/2-1$, $-N/2\le n\le N/2-1$, thus we can rewrite Eq. (2.43) as

$$\begin{aligned}O_{jk}&=\exp\left[i\pi\left((j/Jx)^2+(k/Ky)^2\right)\right]\times\\&\times\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}\left\{U'_{mn}\exp\left[i\pi((m/Mu)^2+(n/Nv)^2)\right]\times\right.\\&\left.\times\exp\left[-2i\pi(jm/M+kn/N)\right]\right\},\end{aligned} \quad (2.44)$$

where $\delta u/\sqrt{\lambda d}=1/Mu$ and $\delta v/\sqrt{\lambda d}=1/Nv$.

Eq. (2.44) is a discrete Fresnel hologram represented by a Fourier transformation. To calculate O_{jk} on the computer, we can follow Eq. (2.44), we first multiple U'_{mn} by a quadratic phase factor, then take the Fourier transform using the FFT algorithm and finally multiply the transform by another phase factor.

2.7 The Fast Algorithm For Discrete Fourier Transformation

In this section, we will consider the Radix-2 fast algorithm for calculating the DFT. The fast Fourier transform (FFT) is a useful tool for computer-generated holograms, because the kernel of the compute-generated hologram is Fourier transform, no matter whether it is a Fourier hologram or a Fresnel hologram. The algorithms for fast Fourier transform have been studied by many people [Cooley and Tukey 1966, Bergland 1968, Good 1971, and Radar 1968]. First we give the definitions of the DFT and its inverse [Briggis and Henson, 1995].

The one-dimensional DFT is defined by[†]

$$A_{[k]} = \sum_{n=0}^{N-1} a_{[n]} \exp[-i2\pi nk / N], \quad k=0,1,...N-1, \tag{2.7.1}$$

where $A_{[k]}$ is the kth coefficient of the DFT and $a_{[n]}$ denotes the nth sample of the spatial series which consists of N samples, and $i = \sqrt{-1}$. The $a_{[n]}$'s can be complex numbers and the $A_{[k]}$'s are almost always complex. For notation convenience Eq. (2.7.1) is often written as

$$A_{[k]} = \sum_{n=0}^{N-1} a_{[n]} W^{nk}, \quad k=0,1,...N-1 \tag{2.7.2}$$

where $W = \exp[-2i\pi/N]$. (2.7.3)

There exists the usual inverse of the DFT, and because the form is very similar to that of the DFT. The fast Fourier transformation (FFT) which will be given later can be used to compute it.

The inverse of Eq. (2.7.2) is

[†] The definition of the DFT is not uniform in the literature. Some authors use $A_{[k]}/N$ as the coefficients, other use $A_{[k]}/\sqrt{N}$.

$$a_{[m]} = \sum_{n=0}^{N-1} A_{[k]} W^{-kn}, \quad m=0,1,...N-1. \qquad (2.7.4)$$

This relationship is called the inverse discrete Fourier transform (IDFT).

The aim of the DFT is to calculate the sequence $A_{[k]}$, where k=0,1,...N-1. The calculations of the sequence of $A_{[k]}$ can be carried out on the basis of Eq. (2.7.1) directly. But in order to find all N coefficients $A_{[k]}$, the computer must perform about N^2 complex operations [N^2 complex multiplications and N(N-1) complex additions]. This computation can be very large even for the modest values of N, therefore, the DFT was useless for digital signal processing until the invention of the fast Fourier transformation algorithms [Cooley and Tukey, 1965].

2.7.1 The Radix-2 FFT Algorithm [Cooley and Tukey, 1965]

We consider the DFT $A_{[k]}$ of N-points sequence $a_{[n]}$, with $N=2^t$, where t is natural number. We can split the N-points input sequence $a_{[n]}$ into two (N/2)-points sequence $a_{[2n]}$ and $a_{[2n+1]}$ corresponding, respectively, to the even and odd samples of $a_{[n]}$. Let $x_n = a_{[n]}$, and $y_n = a_{[2n+1]}$, and then substitute these sub-sequences into Eq. (2.7.2), we obtain that

$$A_{[k]} = \sum_{n=0}^{N/2-1} (x_{[n]} W^{2nk} + y_{[n]} W^{(2n+1)k}), \qquad k=0,1,...N-1. \qquad (2.7.5)$$

If we expand the right side of expression (2.7.5), then it can be written as

$$A_{[k]} = \underbrace{\sum_{n=0}^{N/2-1} x_{[n]} W^{2nk}}_{\text{a DFT of length N/2}} + W^k \underbrace{\sum_{n=0}^{N/2-1} y_{[n]} W^{2nk}}_{\text{a DFT of length N/2}}, \quad k=0,1,...N-1. \qquad (2.7.6)$$

From Eq.(2.7.6) we see that the original DFT has been expressed as a simple combination of the DFT of the sequence x_n, and the DFT of the sequence, and y_n, both of length N/2. We will call these half-length DFTs X_k and Y_k respectively. By letting k = 0,1,...N/2-1, we can write

$$A_{[k]} = X_k + W^k Y_k, \quad k=0,1,...N/2-1 \tag{2.7.8}$$

and we have

$$A_{[k+N/2]} = X_{(k+N/2)} + W^{(k+N/2)} Y_{(k+N/2)}. \tag{2.7.9}$$

We note that $W^{N/2} = -1$ and that the sequences X_k and Y_k have a period of N/2, therefore we can rewrite Eq. (2.7.8) and Eq. (2.7.9) as

$$A_{[k]} = X_k + W^k Y_k, \tag{2.7.10}$$

$$A_{[k+N/2]} = X_k - W^k Y_k. \tag{2.7.11}$$

for k=0...N/2-1, which accounts for all N DFT coefficients $A_{[k]}$ Eq. (2.7.10) and Eq. (2.7.11) are often called **butterfly relations**. They give a recipe for combining two DFTs of length N/2 (corresponding to the even and odd sub-sequences of the original sequence) to form the DFT of the original sequence.

With this approach, ***called decimation in time***, the computation of a N-point DFT is replaced by that of two DFTs of length N/2 plus N additions and N/2 multiplications by W^k. But we are not stopped here. The same procedure can be applied again. Replace the two DFTs of length N/2 by 4 DFTs of length N/4 at the cost of N additions and N/2 multiplications. A systematic application of this method computes the DFT of length 2^t in the $t = \log_2 N$ stages, each stage converting 2^i DFTs of length 2^{t-i} into 2^{i+1} DFTs of length 2^{t-i-1} at the cost of N additions and N/2 multiplications. Consequently, the number of complex multiplications *M* and complex additions *A* required to computer a DFT of length N by the radix-2 FFT algorithm is

$M=(N/2)\log_2 N$

$A=N\log_2 N$ (2.7.12)

The procedure we have described above is shown schematically in figure 2.7.1 and is one of the fundamental FFT algorithms. The method consists of two stages:

- a reordering stage (often called bit-reversal), in which the input sequence is successively split into even and odd sub-sequences; and

a combine stage, in which the butterfly relations (2.7.10) and (2.7.11) are used to combine sequences of length 1 into sequences of length 2, then sequences of length 2 into sequences of length 4, and so on, until the final transform sequence $A_{[k]}$ is formed from two sequences of length N/2.

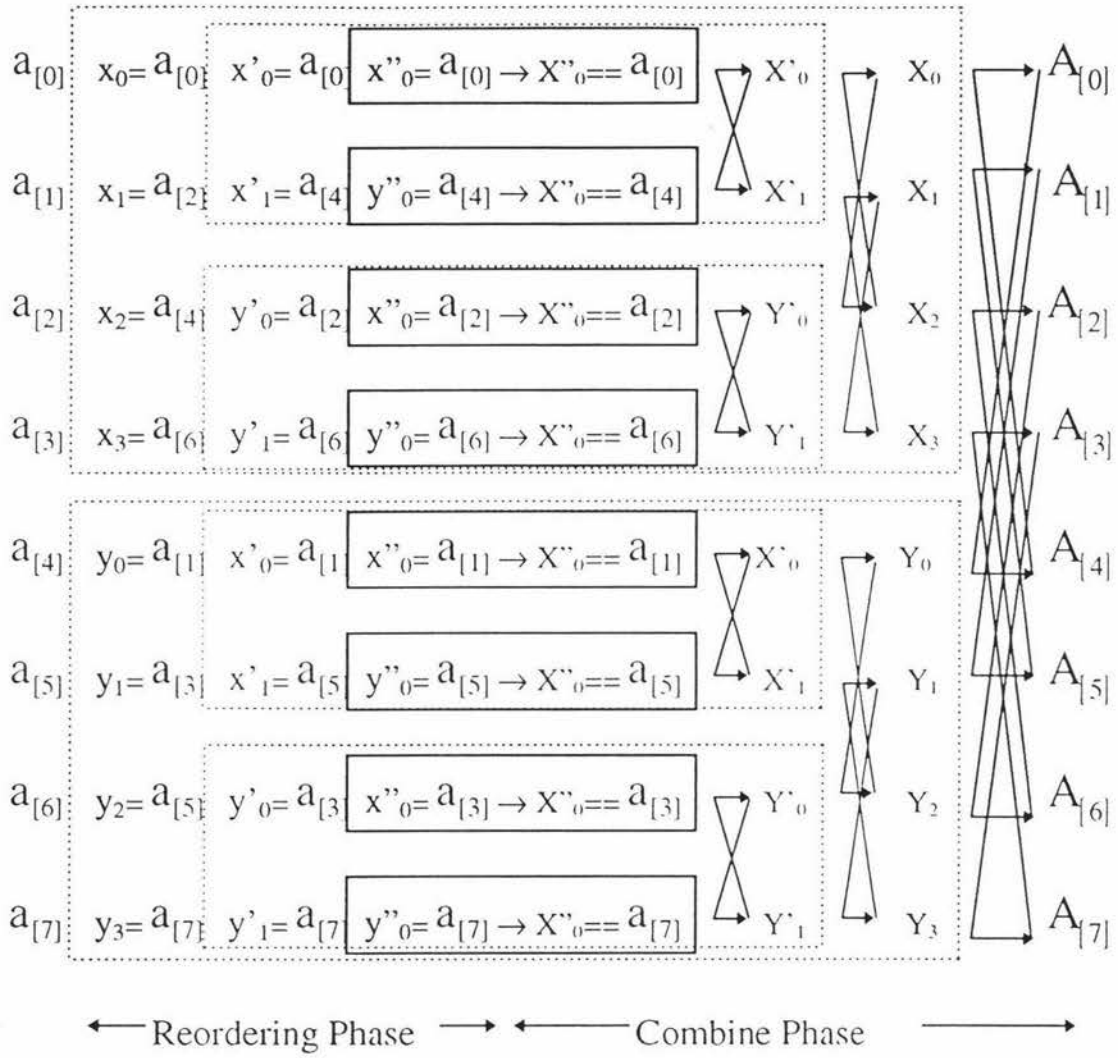


Figure 2.7.1 The Cooley_Tukey FFT for a sequence of length N=8. There are two stages for the algorithm. In the reordering stage, the input sequence **a**[n] is split into even/odd sequences recursively until N sequences of length 1 remain. The combine stage begins with N trivial DFTs of length 1. Then the DFT sequences of length 1 are combined in pairs of to form DFTs of length 2 and so on until the desired DFT A[k], of length 8 is formed. All of the combine steps use the Cooley-Tukey butterfly relations and are shown as webs of arrows. [Briggs, and Hensen 1995]

2.7.2 The Bit Reverse Procedures

The first stage of the FFT algorithm we have discussed above is to reorder the input sequence $a_{[n]}$. This stage is equivalent to perform the bit-reversing operation. An example is given to show the bit-reversing operation in figure 2.7.2.1. This procedure forms the basis of a bit-reversing computer function given by figure 2.7.2.2.

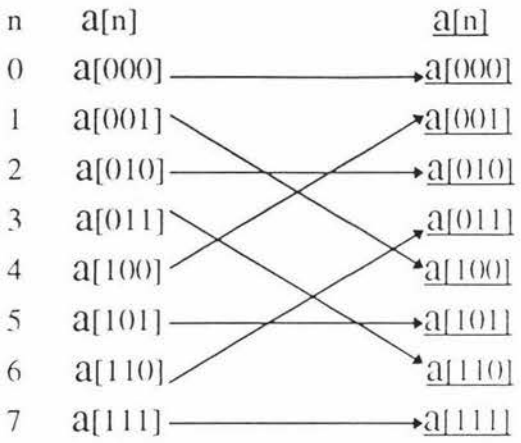


Figure 2.7.2.1 An example of bit-reversing operation for N=8.

```
int bitrev(int in,int M)
/* in  is the input number which will be bit-reversed. M is a positive integer, for a
sequence of N, we have 2^M = N */
{ int  k,n,r=0;
   n=in;
   for (k=0; k<M; k++)
   {r=(r<<1)+(n&1);
    n>>=1;
    }
 return r;
}
```

Figure 2.7.2.2. A "C" Function for calculating bit-reversing.

By calling the function "bitrev", we can easily reorder the sequence of $a_{[n]}$. There is another algorithm dealing with the reorder procedure. We will discuss that later.

2.7.3 Two Dimensional FFTs

Consider a two-dimensional DFT of size $N_1 \times N_2$, with

$$A_{[k_1][k_2]} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} a_{[n_1][n_2]} W_1^{k_1 n_1} W_2^{k_2 n_2} \qquad (2.7.13)$$

where $W_1 = \exp(-2i\pi / N_1)$, $W_2 = \exp(-2i\pi / N_2)$, and $k_1=0,\ldots,N_1-1$, $k_2=0,\ldots,N_2-1$.

In order to evaluate this DFT, we can rewrite Eq. (2.7.13) as

$$A_{[k_1][k_2]} = \sum_{n_1=0}^{N_1-1} W_1^{k_1 n_1} \sum_{n_2=0}^{N_2-1} a_{[n_1][n_2]} W_2^{k_2 n_2} \qquad (2.7.14)$$

As the first step, we can evaluate the N_1 DFTs X_{n_1,k_2} of the length N_2 which correspond to the N_1 distinct values of n_1

$$X_{[n_1][k_2]} = \sum_{n_2=0}^{N_2-1} a_{[n_1][n_2]} W_2^{k_2 n_2} \qquad (2.7.15)$$

$A_{[k_1][k_2]}$ is then obtained by calculating N_2 DFTs of length N_1 on the N_2 sequences X_{n_1,k_2} corresponding to the N_2 distinct values of k_2

$$A_{[k_1][k_2]} = \sum_{n1=0}^{N_1-1} X_{[n_1][k_2]} W_1^{k_1 n_1} \qquad (2.7.16)$$

This approach is called the row-column method because it can be viewed as equivalent to organising the input data into sets of row and column vectors in an array of size $N_1 \times N_2$ and computing, in sequence, first the DFTs of the columns and then the DFTs of the rows. When N_1 and N_2 are powers of two, the one-dimensional DFTs of the lengths N_1 and N_2 can be calculated by the FFT algorithms. In the case of a simple radix-2 decomposition, the number of the computations M is

$$M=N_1N_2[\log_2(N_1N_2)]/2 \qquad (2.7.17)$$

and , for a DFT of size N×N,

$$M=2NM_1, \qquad (2.7.18)$$

where M_1 is the number of multiplications required to compute a DFT of length N [Nussbaumer,1982].

Applying one-dimensional FFT transform to each row and then column of a two-dimensional sequence is not the absolute fastest way to perform the calculation. But it is far the simplest and is actually used in many programs. Somewhat faster approaches can be found in reference Dudgeon and Wersereau.

At section 2.7.1, we mentioned that by calling the function "bitrev", we can easily reorder the sequence of $a_{[n]}$. This is the "auto-sort" procedure as shown in figure 2.7.3.1.

```
for (k=1; k < N-1 ; k++)
 { i=bitrev(k,M);
   if (i > k)
     {T=a[k];
      a[k]=a[i];
      a[i]=T;
      }
 }
```

Figure 2.7.3.1 A fragment of "C" program used for reordering procedure.

If we keep the reordered index in an array, say index[n], then this procedure can be modified as

```
for (k=0; k < N ; k++)
  if (index[k] > k)
     {T=a[k];
      a[k]=a[index[k]];
      a[index[k]]=T;
      }
```

Figure 2.7.3.2 A modified fragment of "C" program used for reordering procedure.

This modification has no effect on the FFT algorithm when it is just used for one-dimensional sequence transformation. But it will take effect for multiple dimensional sequence transformation if we are going to use the row-column algorithm. For example, if we are trying to transform a two-dimensional sequence, say $a_{[N][N]}$, the reordering procedure in figure 2.7.3.1 will call the function "bitrev" 2N times. In figure 2.7.3.2, because the assignment statement i=bitrev(k,M) has be dropped, therefore, the computation time for 2N calling the function "bitrev" can be saved, but at the cost of N memory locations for storing the index[k] and the time to generate the index[k], k=0,...,N-1.

Certainly, we can use the "bitrev" function to generate the index[k]. The following fragment of "C" code can be used for this purpose.

```
for (k=0; k <N; k++)
  index[k]=bitrev(k,M);
```

where $M=\log_2 N$.

But we have found a more efficient way to do this. This method can be called "recursive index" algorithm. Consider the index k of a sequence $a_{[k]}$, where k=0...N-1, and $N=2^m$. We represent the sequence of the index as index[i], i=0,...,N-1. When N=4, the reordered index sequence should be

0, 2, 1, 3. N=4

If N=8, then the reordered index sequence is

0, 4, 2, 6, 1, 5, 3, 7. N=8

If N=16, then the reordered index sequence is

0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15. N=16. (2.7.19)

and so on. Therefore, we have

index[i+N/2]=index[i]+1; i=0,...,N/2-1. (2.7.20)

Note that the first number of the reordered sequence is always zero, and the second is N/2, that is index[0]=0;index[1]=N/2. This give us a hint that

index[0]=0,
index[1]=index[0]+N/2.

If we continue to analysis the sequence of (2.7.19), we found that

index[2]=index[0]+$N/2^2$,
index[3]=index[1]+$N/2^2$.

Note also that

index[4]=index[0]+$N/2^3$,
index[5]=index[1]+$N/2^3$,
index[6]=index[2]+$N/2^3$,
index[7]=index[3]+N/2

Thus, we obtained a recursive expression for the reordered index sequence that can be illustrated by figure 2.7.3.

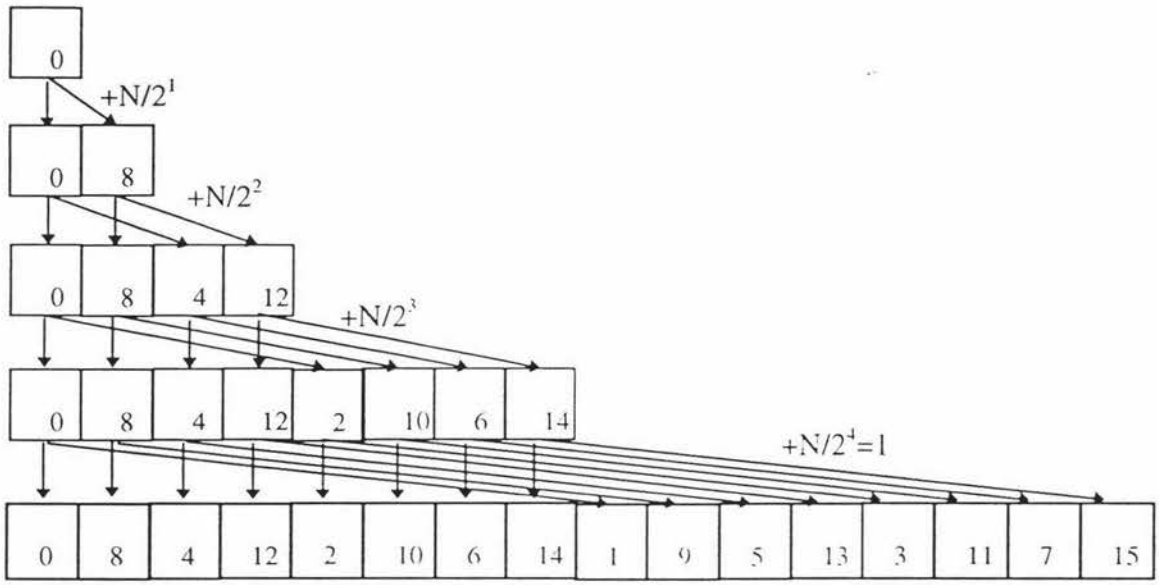


Figure 2.7.3.3. Recursively obtain the reordered subscript sequence for $N = 2^4 = 16$

With the illustration of figure 2.7.3.3, now we can easily write a "C" function to calculate the reordered subscript sequence as following

```
void reorder_index(int N, int M ) /* 2^M=N*/
{ int k=1, i, j;
  index[0]=0;
  for (j=0;j<M;j++)
    {for (i=0;i<k;i++)
       index[i+k]=index[i]+N/(k*2);
     k<<=1;
    }
}
```

Figure 2.7.3.4. The "C" function to calculate the reordered subscript sequence.

The calculation of the reordered sequence by this algorithm is faster than the bit-reversing algorithms. The reason is that in the bit-reversing algorithm, the bit reversal on 2^M numbers is repetitively doing the same work many times. But in the recursive index algorithm, the index sequence is only calculated once. Figure 2.7.3.5 shows the computation time for the bit-reversing algorithm and the recursive index algorithm on the SUN-3/80 workstation.

Number of M	Dimensions	Algorithm	Computation time (seconds)
20	1	bit-reversing	9
20	1	recursive-index	2
24	1	bit-reversing	220
24	1	recursive-index	34
10	2	bit-reversing	63
10	2	recursive-index	48
11	2	bit-reversing	309
11	2	recursive-index	213

Figure 2.7.3.5. The comparison of the computation time for the two algorithms.

As summary of section 2.7, figure 2.7.3.6 shows the relationship between the number of multiplications required by radix-2 FFT algorithm compared with the number of multiplications using the direct method.

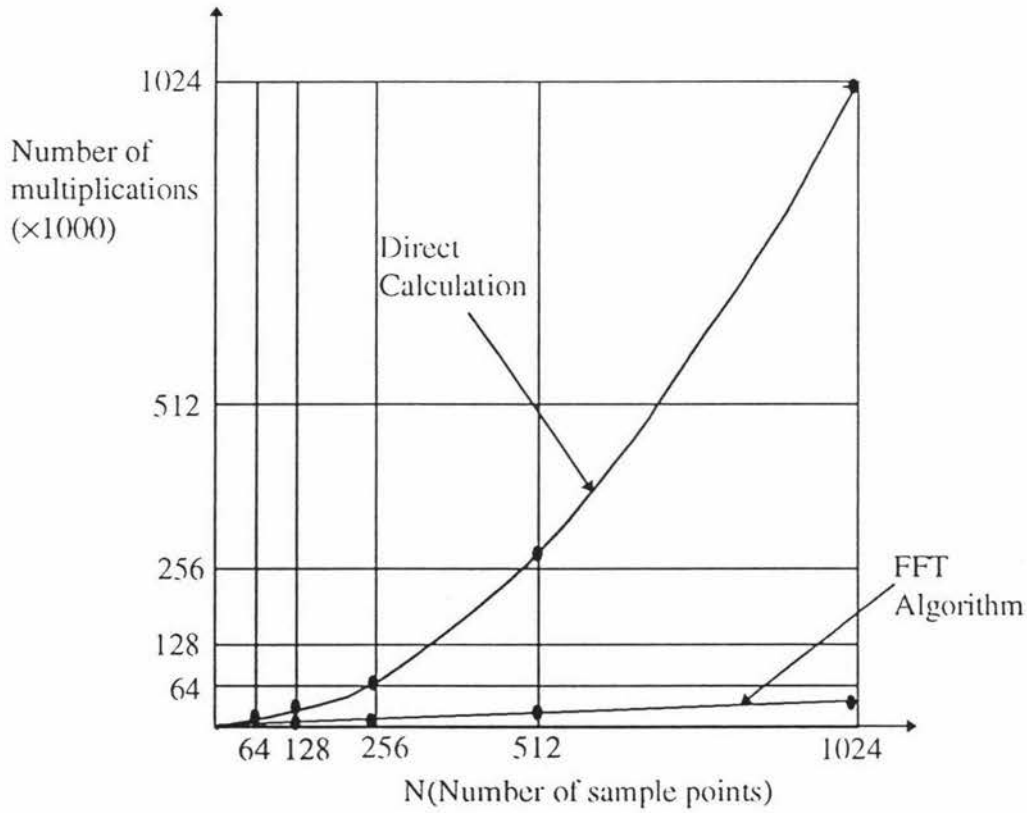


Figure 2.7.3.6 Comparison of multiplications required by direct calculation and FFT algorithm [Brigham,1974].

Chapter 3

Coding Techniques For Computer-Generated Holograms

3.1 Introduction

In chapter 2, we have discussed the mathematical descriptions of computer-generated holograms, and the radix-2 FFT algorithm which is the fundamental tools that can be used in the computation of computer-generated holograms. In this chapter, we will discuss some of the techniques that can be used to really "draw " the interference patterns on the output device of the computer.

In order to produce a computer-generated hologram, we must transform the array of numbers which are found through calculations and which describe the hologram, into a physical object ("the hologram") which is capable of interacting with light to restore the image for observer. In principle, various physical carriers can be used to record the computer-generated holograms: photographic materials, thermoplastics, and liquid crystals, among others. But unfortunately, these materials have certain limitations.

- It is not possible to record both parts of a complex number which correspond to the amplitude and phase of the light, in the same region on the carrier. The recorded value must be non-negative.

- The recording media are non-linear; this is, the physical parameter which is used for the recording (a transmission coefficient, a reflection coefficient, etc.) varies in a non-linear way with the quantity being recorded.

- The recording systems have a finite resolution, only a finite number of linearly independent quantities can be recording on a unit area of the carrier.
  [Yaroslavskii, and Merzlyakov, 1980].

Because of the limitations of the physical recording media, therefore, how to record complex quantities became the first problem we must deal with if we wish to synthesise holograms. Anyhow, techniques have been discovered to solve the problem, we will discuss these techniques in the next several sections.

3.2 The Binary Detour-Phase Hologram (refer to section 5.2)

While it is possible to use an output device with grey scale capabilities to produce a computer-generated hologram, a considerable simplification results if the amplitude

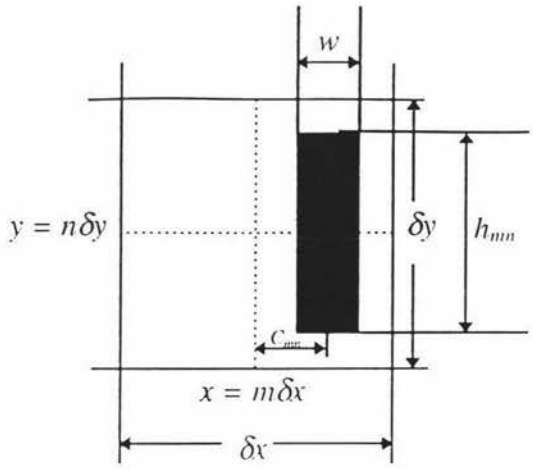


Figure 3.2.1 A typical sampling cell in a detour phase hologram. The height and the position of the aperture inside the cell is used to encode the amplitude and the phase of the complex wavefront.

A_{mn} and φ_{mn} are the amplitude and phase of the Fourier coefficient, δx and δy are the dimensions of each cell. The maximum value of $\{A_{mn}\}$ is normalized to 1. The parameter W normally is selected as $\delta x/2$. This coding procedure is carried out for all the cells in the hologram. Thus the hologram contains many small apertures. If we plot the binary pattern on a piece of paper or a screen, then photo-reduce it to proper size, a binary detour-phase hologram has been made. That the binary detour-phase hologram works can be verified by putting it in the optical system in Figure 3.2.2. The hologram is illuminated by a parallel laser beam. The Fourier transform of the desired wavefront occurs in an off-axis region on the Fourier transform (back focal) plane of the lens L1. The aperture mask shown in figure 3.2.2 passes only one diffracted wave from the hologram through the optical system. Lens L2 performs the inverse Fourier transformation to produce the original object wavefront at the back focal plane of L2 (see next page footnote ↑). The aperture mask in the optical system converts the hologram to a bandpass system. Without it the optical system images the hologram one to one at the back focal plane of the lens L2. The wavefront at the back focal plane of the lens L2 is the bandpassed output of the diffracted waves from the hologram [Lee,1978].

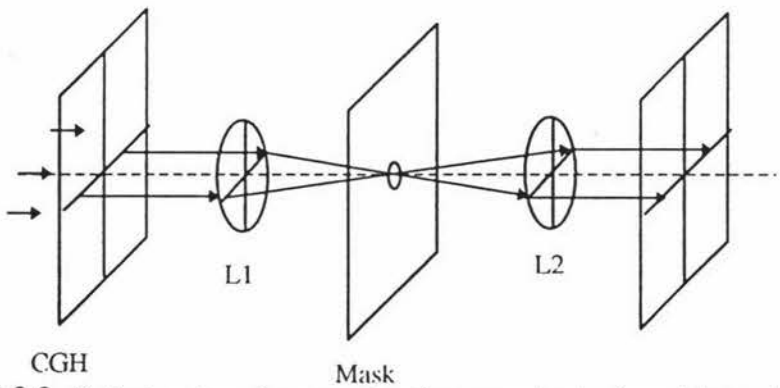


Fig. 3.2.2 Optical system for reconstructing wavefronts from binary detour-phase holograms [Lee,1978].

The binary detour-phase method derives its name from the fact that a shift of the aperture in each cell results in the light transmitted by it travelling by a longer or shorter path to the reconstructed image. Figure 3.2.3(a) shows a typical binary detour-phase hologram of the letters ICO, while figure 3.2.3 (b) shows the image produced by it.

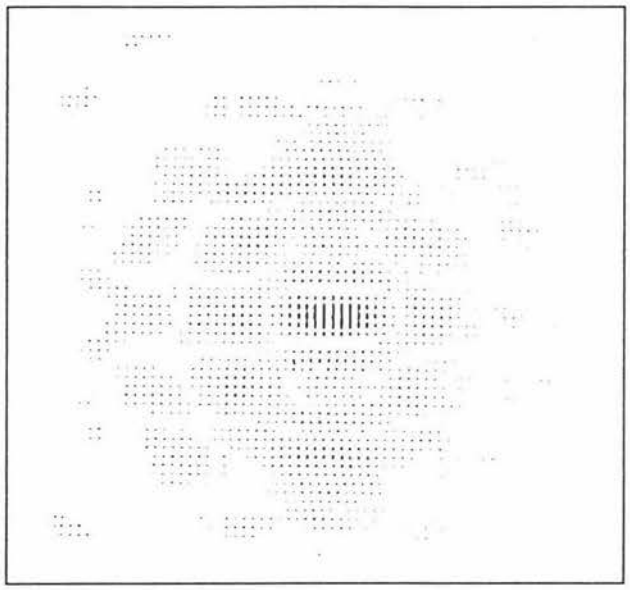


Figure 3.2.3 (a) A typical detour-phase hologram [Loghman, and Paris, 1969].

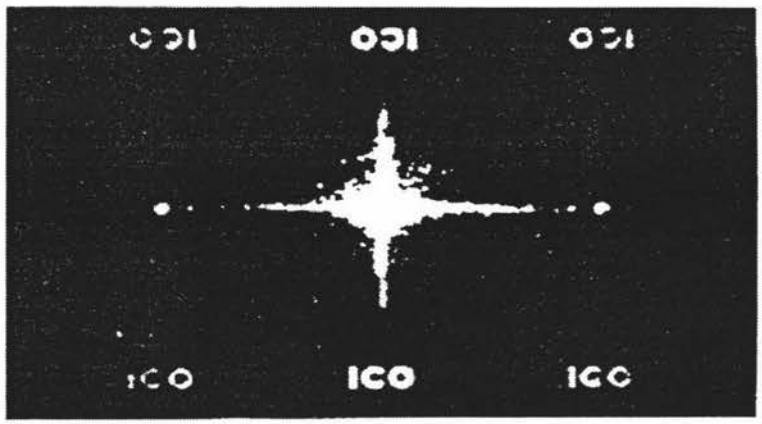


Figure 3.2.3 (b) Image reconstructed from (a). [Loghman and Paris, 1969]

↑: Assume that a transparency, $U_1(x,y)$, in the x, y plane is illuminated by a uniform plane wave of unit amplitude as shown in figure A, the wave distribution at the u,v plane can be calculated by

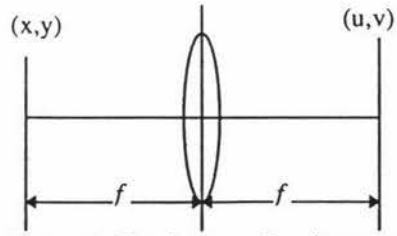


Figure A. Fourier-transforming system.

$$U_2(u,v) = \frac{1}{i\lambda\pi}\iint U_1(x,y)\exp\left\{-i2\pi\left[\frac{xu+yv}{\lambda f}\right]\right\}dxdy.$$

where f is the focal length of the lens. This equation describes a Fourier transform of U_1, and shows us how a lens can be used to perform the Fourier transformation [Cathey, 1974].

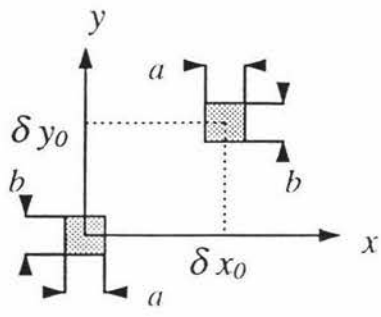


Figure 3.2.4. Diffraction at a rectangular aperture. [Hariharan, 1984].

To understand how this method of encoding the phase works, consider a rectangular aperture $(a \times b)$ in an opaque sheet (the hologram) which is the hologram (if we use negative film to reduce the hologram, the black rectangular apertures will be reversed, that is, they will become transparent apertures), centred on the origin of the coordinates, as shown in figure 3.2.4, and illuminated with a uniform laser beam of unit amplitude. The complex amplitude $O(x_i, y_i)$ at a point (x_i, y_i) in the diffraction pattern formed in the far field is given by the Fourier transform of the transmitted amplitude and is

$$O(x_i, y_i) = ab\,\mathrm{sinc}\left(\frac{a x_i}{\lambda z}\right)\mathrm{sinc}\left(\frac{b y_j}{\lambda z}\right) \tag{3.2}$$

where $\mathrm{sinc}(x) = \dfrac{\sin(\pi x)}{\pi x}$.

Now assume that the centre of the rectangular aperture is shifted to a point $(\delta x_0, \delta y_0)$ and the sheet is illuminated by a plane wave incident at an angle. If the complex amplitude of the incident wave at the sheet is $\exp[i(\alpha\delta u_0 + \beta\delta v_0)]$, the complex amplitude in the diffraction pattern becomes

$$\begin{aligned} O(x_i, y_i) &= ab\,\mathrm{sinc}\left(\frac{a x_i}{\lambda z}\right)\mathrm{sinc}\left(\frac{b y_j}{\lambda z}\right) \\ &\quad \times \exp\left[i\left(\alpha + \frac{2\pi x_i}{\lambda z}\right)\delta x_0 + i\left(\beta + \frac{2\pi y_i}{\lambda \pi}\right)\delta y_0 \right] \\ &= ab\,\mathrm{sinc}\left(\frac{a x_i}{\lambda z}\right)\mathrm{sinc}\left(\frac{b y_j}{\lambda z}\right) \\ &\quad \times \exp\left[i(\alpha\delta x_0 + \beta\delta y_0)\right]\exp\left[2\pi i\left(\frac{u_i \delta x_0}{\lambda z} + \frac{v_i \delta y_0}{\lambda z} \right)\right]. \end{aligned} \tag{3.3}$$

if $a x_i << \lambda z$, $b y_i << \lambda z$, Eq. (3.3) reduces to

$$O(x_i, y_i) = ab\exp[i(\alpha\delta x_0 + \beta\delta y_0)]\exp\left[2\pi i\left(\frac{x_i\delta x_0}{\lambda z} + \frac{y_i\delta y_0}{\lambda z}\right)\right]. \quad (3.4)$$

If the computed complex amplitude of the object wave at a point ($m\delta x_0$, $n\delta y_0$) in the hologram plane is

$$U(m\delta x_0, n\delta y_0) = A_{mn}\varphi_{mn},$$

where $A_{mn} = |U(m\delta x_0, n\delta y_0)|$ and $\varphi_{mn} = \exp[i\varphi(m\delta x_0, n\delta y_0)]$, (3.5)

then its modulus and phase at this point can be encoded, as shown in figure 3.2.1, by making the area of the aperture located in this cell equal to the modulus so that

$$ab = A_{mn} \quad (3.6)$$

and by displacing the centre of the aperture from the centre of the cell by an amount given by Eq. (3.1), thus the complex amplitude in the far field due to this aperture is

$$O_{mn}(x_i, y_i) = A_{mn}\exp[i(\alpha m\delta x_0 + C_{mn}) + i\beta\delta y_0] \times \exp\left[\frac{2\pi i}{\lambda z}(mx_i\delta x_0 + ny_i\delta y_o + C_{mn})\right]. \quad (3.7)$$

The total diffracted amplitude in the far field then is

$$O(x_i, y_i) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} O_{mn}(x_i, y_i). \quad (3.8)$$

If the dimension of the cells and the angle of illumination are chosen so that $\alpha\delta x_0 = 2\pi$, $\beta\delta y_0 = 2\pi$ and $C_{mn} << \lambda z$, then Eq. (3.8) reduces to

$$O(x_i, y_i) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\{A_{mn}\exp[i\varphi(m\delta x_0, n\delta y_0)]\times \exp\left[\frac{2\pi i}{\lambda z}(mx_i\delta x_0 + ny_i\delta y_0)\right]\}. \quad (3.9)$$

This is the discrete Fourier transform of the computed complex amplitude in the hologram plane, in other words, the desired reconstructed image [Hariharan,1984].

The binary detour-phase method is a cost-saving approximation which allows off-axis reconstruction. Because the method has a transmittance which is, at any points on the hologram either clear or opaque, therefore, photographic processing can be

simplified since the necessity to obtain precise grey scale values is eliminated. Photographic materials may be used in the saturation regions of their characteristic curves. Film grain noise is minimised [Burckhardt, 1970]. The chief disadvantage of this method is they are very wasteful of plotter resolution since the number of the addressable plotter points in the cell must be large to minimise the noise due to quantization of the modulus and the phase of Fourier coefficients. When the number of quantization levels used is fairly large, this noise is effectively spread over the whole image field independent of the form of the signal. However, when the number of the phase quantization levels is small, the noise terms become shifted and self-convoluted versions of the signal, which are much more annoying [Goodman & Silvestri, 1970; Dallas,1971a,1971b].

3.3 Lee's sampling method

An alternative method of recording a computer-generated hologram is due to Lee [Lee, 1970]. His technique is somewhat similar to the detour phase technique. The phase information of the complex wavefront is encoded by the positions of the samples in the hologram.

The main idea of Lee's method is to separate transmission of the real and imaginary parts of the complex quantity and to separate positive and negative values. Each complex valued sample O_{mn}, which is calculated by Fourier transform, is represented by four positive real samples on the hologram. Let O_r and O_i be the real and imaginary parts of the sample O_{mn}, then it can be written as

$$O_{mn} = O_r + O_i. \tag{3.10}$$

We determine the non-negative real quantities as

$$O_1 = \begin{cases} O_r, & \text{if } O_r \geq 0 \\ 0, & \text{if } O_r < 0 \end{cases}$$

$$O_3 = \begin{cases} -O_r, & \text{if } O_r < 0 \\ 0, & \text{if } O_r \geq 0 \end{cases}$$

$$O_2 = \begin{cases} O_i, & \text{if } O_i \geq 0 \\ 0, & \text{if } O_i < 0 \end{cases}$$

$$O_4 = \begin{cases} -O_i, & \text{if } O_i < 0 \\ 0, & \text{if } O_i \geq 0 \end{cases} \tag{3.11}$$

Then $O_{mn} = O_1 - O_3 + iO_2 - iO_4$. (3.12)
Where O_1, O_2, O_3, and O_4 are all real and positive.

In plotting the hologram, each cell on the hologram is divided into four equal-sizes sub-cells arranged in the succession in one direction as shown in Figure 3.3.1a. The four sub-cells then contribute phase components of the complex valued O_{mn} with relative phases of 0, $\pi/2$, π, $3\pi/2$, that is, the phase of the complex valued O_{mn} is quantized into four levels. To control the amplitude of O_{mn}, the four non-negative values, in actual use, two of them are always zero, are also quantized into four levels. The maximum value of is normalized to one, that is , $\max\{O_{mn}\}=1$, and then multiplied by the dimension of the cell, so that the height of the narrow bars are always within the cell. By this procedure, two of the subcells in any cell in the hologram are totally opaque, while other two are partially transmitting. This gives complete control of the resultant complex amplitude [Hariharan, 1980].

A simplification of this technique has been proposed by Burckhardt [Burckhardt, 1970], who pointed out that the complex number can be represented by non-negative components in a two-dimensional simplex (Figure 3.3.1(c)), in contrast with Lee's method in which the complex number is represented as a vector with non-negative components in "biorthogonal" coordinates (Figure 3.3.1(d)). That is $O_{mn} = O_1 \alpha + O_2 \beta + O_3 \gamma$, where O_1, O_2 and O_3 are all real and positive, and at most two of them are non-zero. By employing Burckhardt's solution, we just only need three positive real values to represent one complex valued sample and therefore, one cell in the hologram is only divided into three sub-cells, as is shown in figure 3.3.1(b).

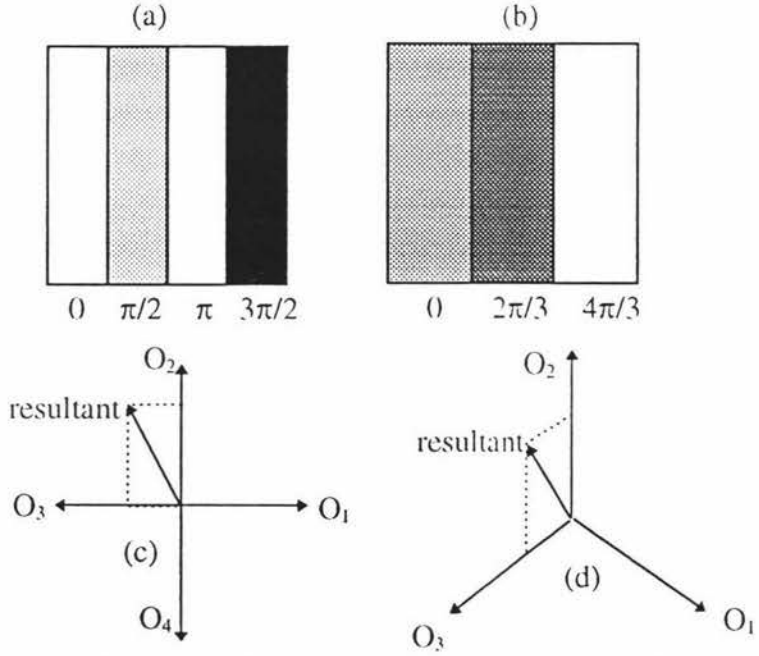


Figure 3.3.1. An example of sub-cell arrangements and phasor diagrams for detour-phase hologram developed by (a) Lee [Lee,1970] ; and (b) Burckhardt [Burckhardt, 1970].

3.4 The Kinoform Hologram

The kinoform is a different form of the computer-generated hologram. It is not recorded as amplitude transmittance on the film as are the previous types of the computer-generated holograms but as relief patterns on film.

In the case where the object is diffusely (see chapter 5 for more detail) illuminated, the magnitude of the Fourier coefficients are relatively unimportant, and the object can be reconstructed using only the values of their phases. The kinoform made by Lesem, Hirsch and Jordan [Lesem, Hirsch and Jordan, 1974], were Fourier transform type holograms. To make such a kinoform, the discrete Fourier transform of an object is first calculated by the computer. The phase angles of the complex samples of the Fourier transform are determined and the amplitude of the transform is set to equal 1. Thus, the amplitude transmittance t_{mn} of the cell corresponding to a Fourier coefficient with modulus $|o_{mn}|$ and phase φ_{mn} would be

$$t_{mn} = \exp\{i\varphi_{mn}\} \tag{3.13}$$

In order to simplify recording, integral multiples of 2π radians are subtracted from the computed phases, so that they vary only between 0 and 2π over the entire kinoform.

In plotting the kinoform, the computed values of the phase φ_{mn} are quantized into a multilevel grey scale and used to control a photographic plotter which exposes a photosensitive surface to code the pattern.. The resulting record is next photo-reduced to a size suitable for illumination, and finally the photograph is bleached with a tanning bleach (see reference Collier, Burckhardt and Lin, 1971) to convert the grey levels to corresponding changes in optical thickness. With proper control of exposure and processing, the amplitude transmittance of the final kinoform can be made to conform closely to Eq. (3.13) [Hariharan, 1980]. In principle, it is not necessary to plot the kinoform directly on film then photo-reduce it again, instead, we can first plot the kinoform on a colour (grey scale) CRT and then photo-reduce it to the proper size.

Kinoforms are on-axis devices which have the ability to diffract all the incident light into the final image, therefore, only one image is reconstructed. However, a multilevel grey scale is required and great care in bleaching must be exercised. Any error in the recorded phase shift results in light diffracted into the zero order (see the spot in the centre of figure 3.4.1(a) and 3.4.1(c) which can spoil the image.

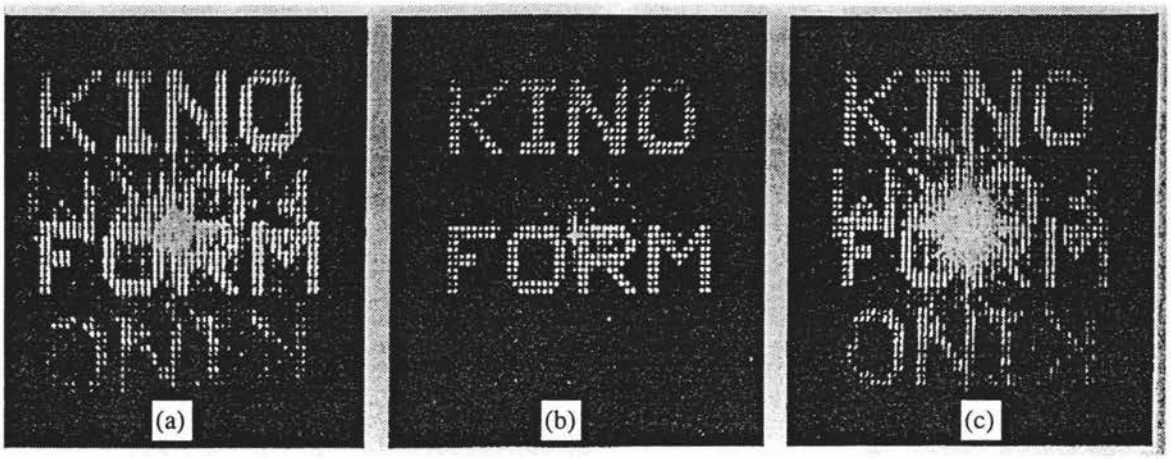


Figure 3.4.1 Photographs of images produced by kinoforms: (a) under-modulation (too short an exposure time); (b) correct exposure achieving good phase matching; (c) over-modulation (too long an exposure time) [Lesem, Hirsch and Jordan, 1969].

3.5 The ROACH

The ROACH is an acronym for Referenceless On-Axis Complex Hologram [Chu, Fienup and Goodman,1973]. It is a multi-emulsion kinoform. The kinoform has a main disadvantage that is when photo-reducing the plotted pattern, care must be given to ensure that the phase matching condition expressed by Eq. (3.13) is satisfied accurately. The ROACH makes uses of multi-emulsion colour film, usually Kodachrome II, as a recording medium to obtain most of the advantages of the kinoform without its major disadvantages. Different layers of the film are exposed selectively by light of different colours, When illuminated with light of a given colour, one layer of the film absorbs, while the other layers, which are effectively transparent, can cause phase shifts due to variations in the film thickness and refractive index. Thus, both the amplitude and phase of the transmitted beam can be controlled by a single element.

Kodachrome II is colour positive film which has three emulsion layers. After processing the layers are red absorbing, green absorbing and blue absorbing. Besides the colour-specific absorption, there is also an exposure-dependent phase shift in each layer. The phase shift is due to index-of -refraction changes within each layer and thickness variations of each layer. By careful attention to exposure, the complex amplitude transmittance of the film, for monochromatic light, can be controlled [Dallas, 1980].

To record a ROACH which is to be illuminated finally with red light, the computed values of the moduli $|O_{mn}|$ of the Fourier coefficients are used, in the first step, to

control the brightness of a black and white CRT display. This is then photographed on to the Kodachrome II film, by using red filter. In the next step, the computed values of the phases φ_{mn} of the complex Fourier coefficients are displayed and exposured to the film through a blue-green filter to generate a relief pattern.

After processing, the red absorbing layer of the film controls the amplitude of the transmitted red light so that it is proportional to the moduli $|O_{mn}|$ of the Fourier coefficients. The blue and green absorbing layers are transparent to red light, but they introduce phase shifts corresponding to the phases of φ_{mn} of the Fourier coefficients due to the thickness variations introduced by the blue-green exposure. In practice, the red absorbing layer also introduces a phase shift proportional to the attenuation due to it. It is possible to compensate for this by subtracting from the blue-green exposure a component proportional to $|O_{mn}|$.

Since all light is diffracted into a single image, the diffraction efficiency of the ROACH is very high. In addition, because both the amplitude and the phase of the object wave are encoded, the quality of the reconstructed image by a ROACH is superior to that of the kinoform. The ROACH is also superior to the binary detour-phase hologram, because only one display spot is required for each Fourier coefficient, and quantization noise is negligible [Hariharan, 1980].

3.6 The Waters's Fresnel Zone Plate Hologram (refer to section 5.1)

Another technique for synthetic hologram was reported by Waters [Waters, 1966]. He observed the similarity between optical holograms and Fresnel zone plates, whose imaging properties which we will discuss later, and used the Fresnel zone plate approach to calculate his hologram. The calculated result was plotted, using a mechanical plotting table.

Many of the earlier papers in optical holography consider the optical hologram to be a collection of Fresnel zone plates, each reconstructing a point of the image, that is, the object is assumed to be made up of point sources. Each illuminated point gives rise to a spherical wave which when recorded in conjunction with a plane reference wave produces a distribution similar to the far field amplitude pattern due to a pinhole. This recording is equivalent to the formation of a Fresnel zone plate on the hologram. The hologram, then, is a collection of Fresnel zone plates, one for each point [Cattey, 1974].

The Fresnel zone plate is made of black and white rings (Figure 3.6.1). A spherical wave is produced by a point source and incidents on the hologram plane. The ring pattern produces phase relations between the light beams passing the hologram plane. The diffraction pattern from a point aperture has annular rings, or zone, of positive or negative phase. Fresnel zone plates are plates in which the zone of positive phase are transparent, while those of negative phase are opaque. When a Fresnel zone plate is illuminated by laser light from a point source, it does image that light. One way to analyse the holographic process is to consider the object as consisting of discrete point apertures, so that the hologram may be thought of as the superposition of many Fresnel zone plates.

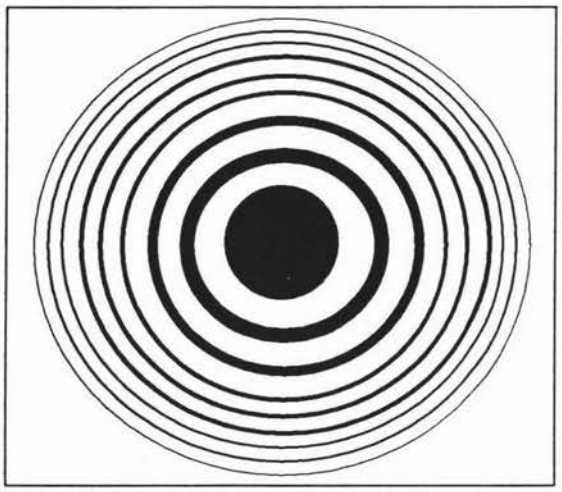


Figure 3.6.1 A Fresnel zone plate.

Waters considered a point diffracting site located in a distance z from the plate where the interference of the diffracted wavelets and a reference beam is displayed. He assumed that the non-disturbed portion of the beam and diffracted light are in phase at a given point ($x=0, y=0$) on the display plane, the intensity, I, at any point on the plane then can be calculated by

$$I = I_0 \left\{ 1 - \frac{k}{\pi z} \sin\left[\frac{k(x^2 + y^2)}{2z} \right] + \frac{k^2}{4\pi^2 z^2} \right\} \qquad (3.14)$$

where I_0 is the intensity of the non-disturbed beam and $k = \frac{2\pi}{\lambda}$ the wave propagation number with λ the wave length. The first and third terms within the braces are constants, representing respectively the intensity of the non-disturbed beam and the diffracted light. The second term is the interference term and , for any given position on the plane, can be positive or negative depending upon the phase of the diffracted wavelets.

When the second term goes through zero

$$x^2 + y^2 = \frac{2mz\pi}{k} \quad \text{for } m=0, 1, 2,... \tag{3.15}$$

it represents the boundaries of the positive and the negative phase zones. Thus Eq. (3.15) can be used to establish the boundaries of the alternating, completely opaque and transparent portions of a Fresnel zone plate [Waters, 1966].

Waters pointed out that Eq. (3.14) can be regarded as the intensity distribution for a hologram of a point located at (0,0,z); and the zone plate specified by Eq. (3.15) is then a special case of this type of hologram. In order to reduce the complexities of theoretical synthesis of holograms of non-existent objects, a binary coding system consisting only of opaque and transparent portions could be used, as in Eq. (3.14) gives rise to grey tones in the hologram. The decision as to whether a portion of the hologram should be opaque or transparent can be made by investigating the sign of the function:

$$f(x,y)= \sin\left[\frac{k(x^2 + y^2)}{2z}\right]. \tag{3.16}$$

If the object consists only points, all that is necessary for synthesizing a hologram is to sample different portions of the hologram plane using Eq. (3.16) and making the portions either opaque or transparent depending upon the sign of the function.

The theory thus far represented is adequate only for a single point. Now let us to explain how to really produce the hologram. Assuming that the object is made up N points, one typical point has the coordinates in space denoting by (u_i, v_i, z_i), i = 0,...,N-1. If there are M×M portions in the hologram plane, then the intensity of each portion in the hologram can be determined by

$$I(x,y) = \sum_{i=0}^{N-1} \sin\left[\frac{k((x-u_i)^2 + (y-v_i)^2)}{2z_i}\right] \quad x = 0,...,\text{M-1},\ y = 0,...,\text{M-1}. \tag{3.17}$$

where $k = \frac{2\pi}{\lambda}$ is the wave propagation number.

After the intensity of each portion in the hologram is computed by a computer, the results can be plotted by a automatic drafting table or displayed on a computer screen

on an enlarged scale, then photo-reduced to the proper size, and illuminated by a laser beam for observation.

Waters's zone plate hologram has several properties: (a) the reconstructed image displays all the properties associated with three-dimensionality; (b) any fraction of the hologram can be used to reconstruct the entire image with an attendant loss in the resolution and decrease in the intensity because of the smaller aperture; (c) either an on-axis or off-axis synthesis can be formulated. Some examples of holograms made by Waters's method are shown in chapter 5.

3.7 Decomposition Method (refer to section 5.6)

As we mentioned above, a three dimensional image can be reconstructed by a Waters's Fresnel zone plate hologram. The three dimensional image reconstruction is one potential advantage of computer-generated holograms for display purpose. Of course, optical holograms could reconstruct three dimensional images, but computer-generated holograms can be used to reconstruct images which may not physically exist. This potential was recognised early in the computer-generated hologram development and three-dimensional computer-generated holograms were fabricated [Waters 1968; Lesem, Hirsch and Jordan, 1969; Brown and Lohnmann 1969].

In above sections, we have discussed the many techniques for recording wavefronts in computer-generated holograms. These methods can also record the wavefront from a three dimensional object. However, the exact calculation of the wavefront of a three dimensional object at a hologram is rather complicated, therefore, some simplification must be made. One way to simplify the computation of the wavefront of a three dimensional object is to assume that the object consists of many independent scatters. Each scatters is considered as a point source. If we put the object in the optical system as shown in figure 3.7.1, then we can calculate the wavefront distributions of the object at the hologram plane. The three dimensional object is located at the front focal plane of the lens L. The hologram is assumed to be at the back focal plane. The three dimensional object is first decomposed into J planes in depth, as shown in figure 3.7.2.

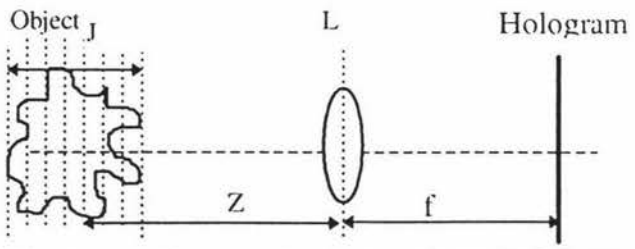


Figure 3.7.1 Optical system used in computing the wavefront of a 3-D object [Lee, 1980].

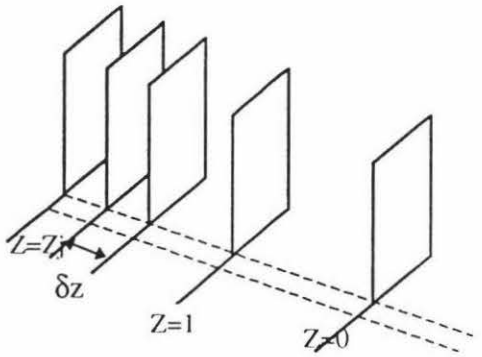


Figure 3.7.2. Illustration of how a 3-D set of date is decomposed into planes, δz apart, parallel to the hologram [Bryngdahl and Wyrowski, 1990].

Each plane may contain several points which are parts of the object. For an object point at (u,v,z_n) the wavefront at the hologram plane cold be calculated by

$$\sigma(u,v,z_n)=U(u,v,z_n)\exp\left\{i\pi(f-z_n)\frac{x^2+y^2}{\lambda f z_n}\right\}\exp\left\{-i2\pi\frac{xu+yv}{\lambda f}\right\}, \qquad (3.18)$$

where $U(u,v,z_n)$ is the amplitude of the point source and f is the focal length of lens L. when there are more than one points, at the distance z_n from the lens, the collective wavefront at the hologram from these points is the sum of each point distribution and can be given by

$$O(x,y,z_n)=\exp\left\{i\pi\frac{x^2+y^2}{\lambda f z_n}\right\}\sum_x\sum_y U(u,v,z_n)\exp\left\{-i2\pi\frac{xu+yv}{\lambda f}\right\}, \qquad (3.19)$$

The total wavefront at the hologram for the entire three dimensional object is the superposition of the function $O(x,y,z_n)$:

$$O(x,y)=\sum_{n=0}^{J}O(x,y,z_n). \qquad (3.20)$$

The summation in Eq. (3.19) is a discrete two-dimensional Fourier transform of the points at a distance z_n from the lens and can be calculated according to Eq. (2.23a), that is, take the discrete Fourier transform and then multiply by the phase factor $\exp\{i\pi(x^2+y^2)/\lambda f z_n\}$ element by element in the Fourier spectrum. [Lee,1980].

$O(x,y)$is a paraxial approximation of the wavefront of the three dimensional object at the hologram plane. That means the object is located at the front focal plane of the lens and the hologram plane is located at the back focal plane of the lens. For

$|f - z_n| << f$ the amount of the phase variation in the quadratic phase term is small and can easily be incorporated into the computer-generated hologram. When the calculation is complete, the detour phase method (refer to section 3.1) can be used to construct. Figure 3.7.3 shows images reconstructed in three dimensions from such a hologram. The planes in depth may be focused on individually as shown in the figure.

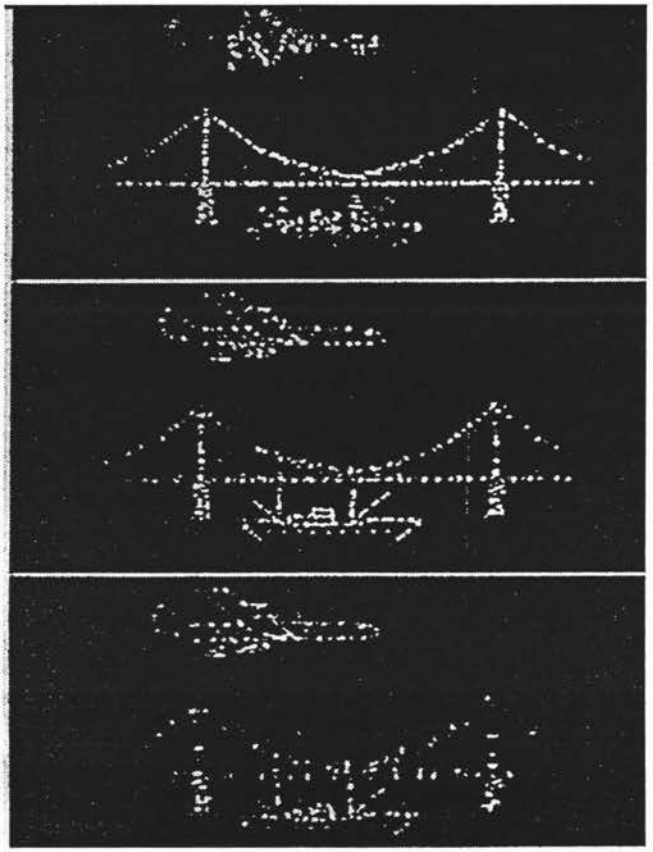


Figure 3.7.3. Three- dimensional reconstructed image consisting of three planes spaced in depth [Brown And Lohmann ,1969].

3.8 Off-Axis Reference Beam Hologram (refer to section 5.4)

In the off-axis reference beam transmission hologram described in section 2.2, the amplitude transmittance of a hologram record is proportional to

$$\begin{aligned} T(x,y) &= |Re^{i\psi(x,y)} + O(x,y)e^{i\varphi(x,y)}|^2 \\ &= R^2(x,y) + O^2(x,y) + 2RO\cos[\psi(x,y) - \varphi(x,y)] \end{aligned} \tag{3.21}$$

where $Re^{i\psi(x,y)}$ represents the reference wave. $R(x,y)$ and $e^{i\psi(x,y)}$ are the amplitude and the phase of the reference wave respectively. $O(x,y)e^{i\varphi(x,y)}$ represents the amplitude and phase of the object distributions on the hologram plane. If the reference

wave is a plane wave, then the reference wave can be written as $Re^{i2\pi\alpha x}$, where α is the angle between the plane wave propagation vector **k** onto the x-z plane and the z-axis, as shown in figure 3.8.1.

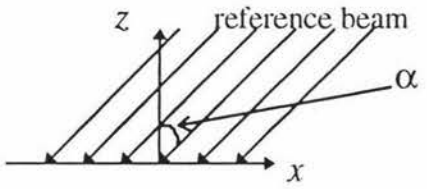


Figure 3.8.1 Illustration of the relationship between the reference wave propagation and the coordinates system.

The function $T(x,y)$ describes the relationship between the reference wave and the object wave. Thus computer-generated holograms could be made from Eq. (3.21). Because the function $T(x,y)$ is already real and non-negative, the computer can simply calculates its values at the discrete intervals and produces a graphical output of the function $T(x,y)$ on a screen or other device. In practice, the object wave can be calculated first according to Eq. (2.23a) and the reference wave can be added.

It is not necessary to calculate the function $T(x,y)$ according to Eq. (3.21), because some terms in it are not important for the reconstruction of the desired wave-front $O(x,y)e^{i\varphi(x,y)}$. Burch [Burch,1967] reported that the term $O^2(x,y)$ could be omitted and the term $R^2(x,y)$ could be equal to one, if we use a plane beam as reference wave, thus the amplitude transmittance of the hologram should be

$$T(x,y)=\boldsymbol{K}+2O(x,y)\cos[2\pi\alpha x-\varphi(x,y)] \tag{3.22}$$

where $\boldsymbol{K}$ is a constant which is just large enough to make $T(x,y)$ positive for all (x,y). Moreover, if the object wavefront is normalised so that the amplitude of the object wavefront has the values between 0 and 1, then selecting $\boldsymbol{K} = 2$ can guarantee that $T(x,y)$ is always great than zero for all (x,y) pairs. Further, we can also normalise the function $T(x,y)$ so that it has values between zero and one, therefore, Eq. (3.22) becomes

$$T(x,y)=0.5\{1+O(x,y)\cos[2\pi\alpha x-\varphi(x,y)]\}. \tag{3.23}$$

A small fragment of a computer-generated hologram made according to Eq. (3.23) is shown in figure 3.8.1(a) ; the image obtained from the hologram is show in figure 3.7.1(b).

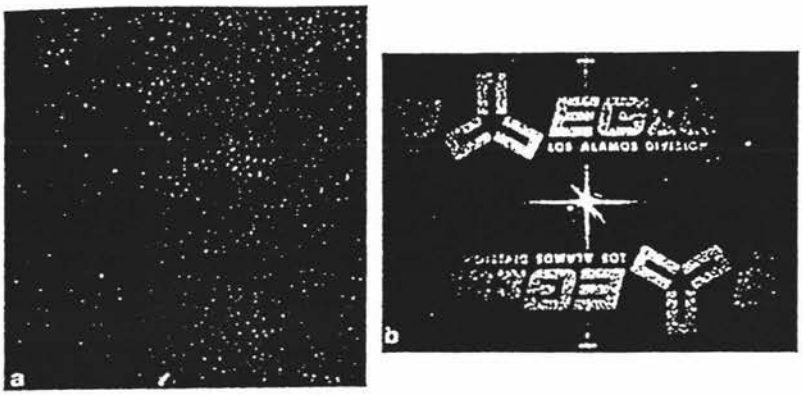


Figure 3.8.1 (a) A fragment of a computer-generated hologram made by Burch's method.
(b) The image reconstructed from the hologram [Campbell, Wecksung, and Mansfield, 1974].

Huang and Prasada [Huang and Prasada, 1967] have proposed another method for making off-axis computer-generated holograms. They suggested that use of the following amplitude transmittance for the hologram

$$T(x,y)=0.5*O(x,y)\{1+\cos[2\pi\alpha x-\varphi(x,y)]\} \qquad (3.24)$$

This method has the advantage of that the contrast in the hologram is very high and independent of the fluctuation in $O(x,y)$.

3.9 Composite Computer-Generated Holograms

In section 2.3, we have introduced a more practical approach to the 3-D image display problem suggested by King, Noll and Berry [King, Noll and Berry, 1970]. They used the computer to calculate the two-dimensional perspective view of the objects as it would appear on a projection screen at a particular viewing angle. Many of these perspective projection images are recorded in a sequence on film. The film is then used in the optical system in figure 2.3.1 to produce a composite hologram. Yatagai [Yatagai, 1974,1976] used another technique similar to King, and *et al's* to make composite computer-generated holograms. But instead of recording the perspective projection images on the film, he converts the projection images into Fourier transform holograms which are then used to form the composite hologram.

Recently, K. Haines and D. Haines [Haines and Haines , 1992] reported another technique to make composite computer-generated hologram. Their method, called the direct method, is similar in principle to King's, but use a lens instead of the mask in King's method, and they used a different method to calculate the object projections. They compute the various views of the object using the system shown in figure 3.9.1.

They used a system like that shown in figure 3.9.2 to form the contiguous image-plane hologram elements. The position of each element in the hologram plane corresponds to a viewpoint location in the primary object plane. The image used in the

construction of the element is the viewplane project from the viewpoint. As a result, the laser light rays passing through every parts of the composite hologram simulate the computed rays passing through the primary object [Haines and Haines , 1992].

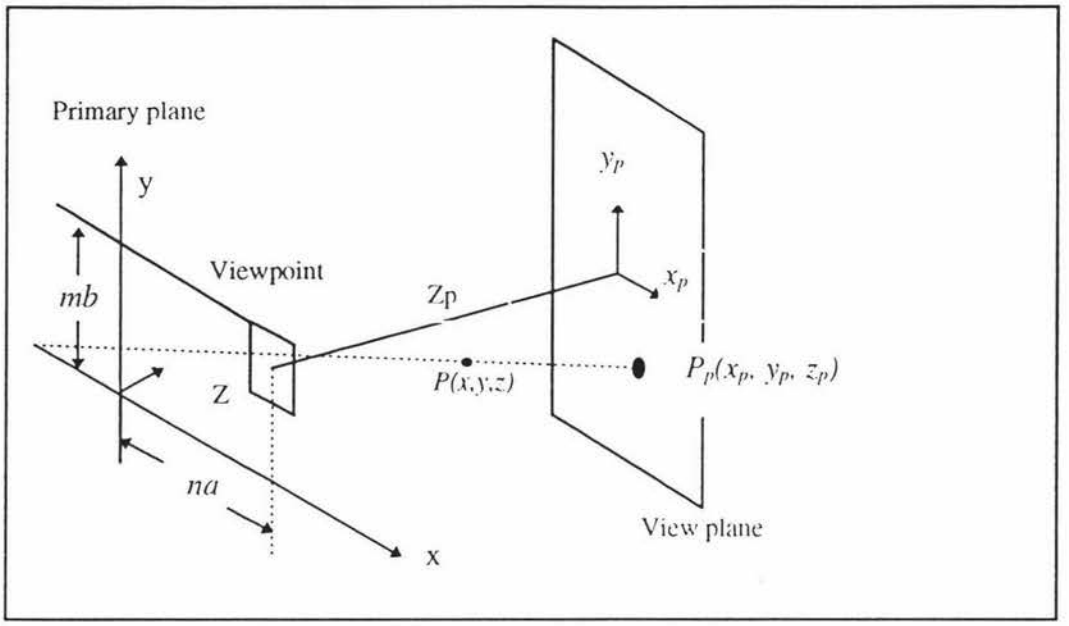


Figure 3.9.1 Perspective projection for the direct method [Haines and Haines, 1992].

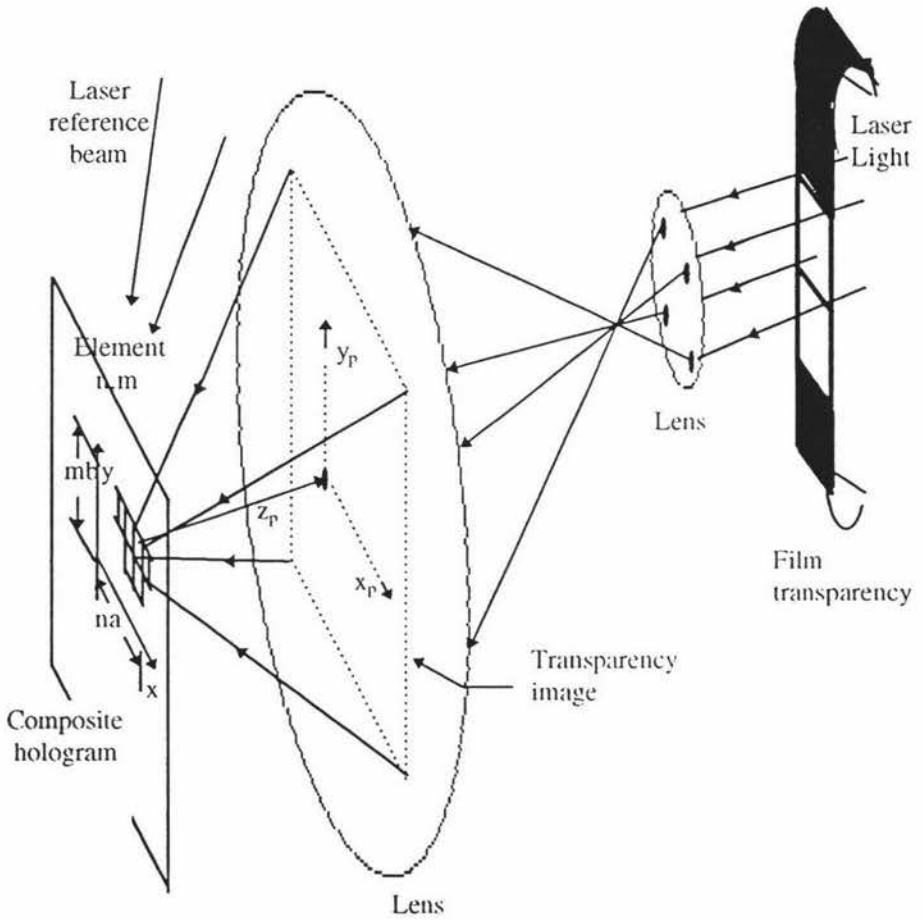


Figure 3.9.2 Hologram formation for the direct method [Haines and Haines, 1992].

Chapter 4

Representation of the Object

4.1 Introduction

In section 2.2, we mentioned that the first step to produce a computer-generated hologram is defining the object. This is the fundamental step in synthesizing holograms, because we need the sampled amplitudes and the phase information about the object to calculate the wave field distributed by the object in the hologram plane with a transformation (Fourier, Fresnel). Of course, for simple object, it can be defined as a array of points, and this can be done by hand. For example, in the two dimensional situation, assume that the object is a letter, say F, which is consisted of three lines and has the dimension $\Delta\times\Delta$ (say, Δ=10mm), assume also that we have sampled the object at high enough frequency and we got 128×128 sampled points, then we can use a two-dimensional array to define the object. Here we assumed that the all the amplitudes of the object are constant. The phase information we will discuss later. There is no difficulty for us to represent this simple object, but if the object is very complicated, for example, a helicopter, then how can we represent the object and sample it to get the necessary information for our synthesizing hologram purpose ?

The representation of a three dimensional object in the computer is a really tricky task, and there is no general perfect representation for all objects so that we can examine the object for various purposes. However, many sophisticated techniques have been developed for representation of the object in some special area, for example, uniform cubic B-splines can be used to represent a curved surface of an object for computer simulation. One of the sophisticated techniques in three dimensional object representation is Constructive Solid Geometry. It can be used to represent a new object which is produced by performing some operations on some well known objects. This is exactly what we want in the first step of our synthesizing hologram procedure, that is, defining a three dimensional object which may not exist in the world.

4.2 Constructive Solid Geometry (CSG)

In CSG, objects are modeled by a description of how they may be constructed by combining simple solids together. More formally, a solid object is represented by a structure in which so called 'primitive objects', such, as rectangular blocks, cylinders, cones, or spheres, are combined using Boolean set operations [Requicha 1980]. These operations, union, intersection and difference are illustrated in figure 4.2.1.

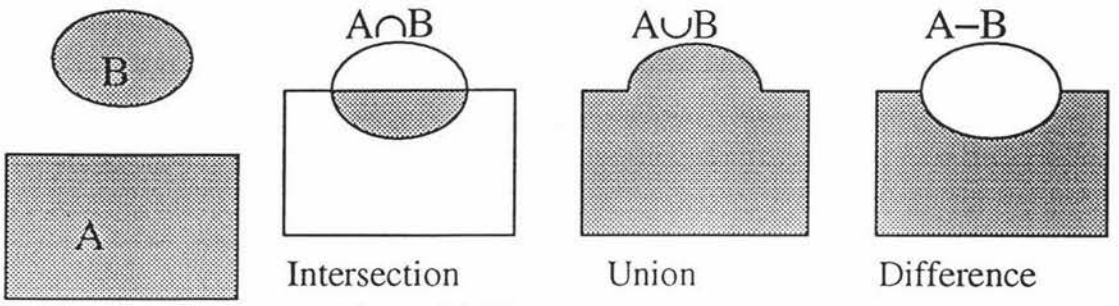


Figure 4.2.1 Boolean set operations.

The internal data structure of a CSG representation is a tree (CSG tree). At a terminal node (primitive node), information about a primitive is stored, for example, we need four parameters to represent a rectangle, that is the length, the width, the thickness and the center of the rectangle. At an internal node (composite node) the set operation to be applied to the objects defined by the left and right children of the node, and pointers to these children are stored. Figure 4.2.2 shows a CSG model and its CSG tree.

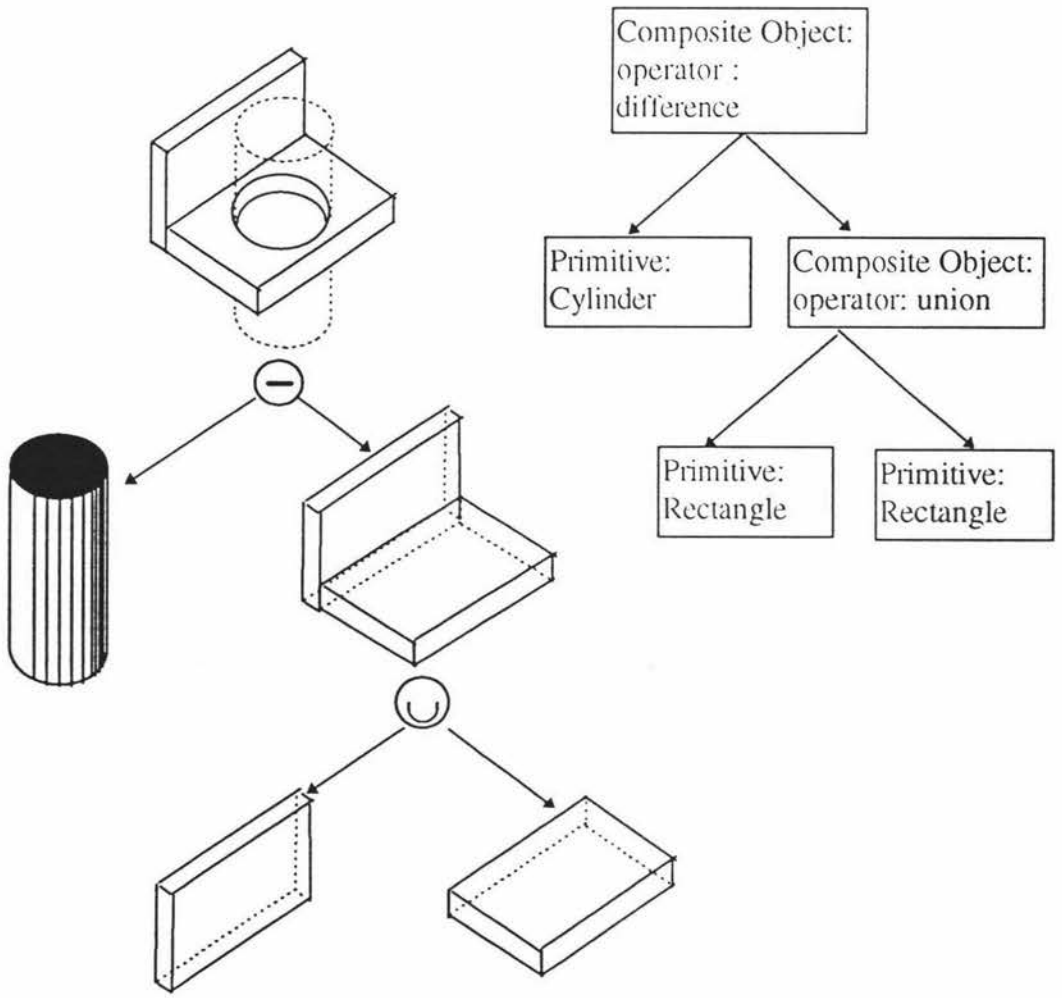


Figure 4.2.2 An solid object defined by CSG and its internal data tree.

An object does not have a unique representation in the CSG, since many combinations of different primitives in differing relationships can be used to construct the same end result. As a trivial example, two mutually annihilating primitives could be added to the appropriate part of the description without affecting the result. The domain of CSG representation is relatively large. With a suitable variety of primitives a wide range of objects can be described exactly.

Directly applying the Boolean set operations to the set of points defined by a 3- D solid can lead to anomalous results, as illustrated in figure 4.2.3(a), where 'dangling edges' can be produced. The regularized Boolean set operations [Requicha and Voelcker, 1977] can be used to void this problem, as shown in figure 4.2.3(b).

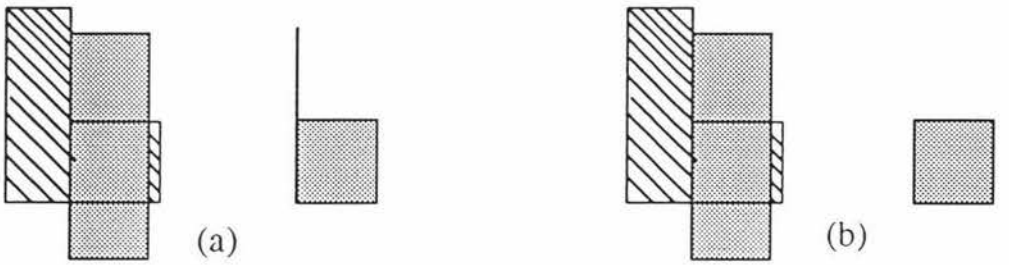


Figure 4.2.3. (a) Non-regularized Boolean operation; (b) Regularized Boolean operation [Jared, 1987].

4.3 Primitive Object Representation

In order to employ the Boolean set operations to construct object models, we must start with some simple object elements to combine. These are called primitives. The primitives only provide templates for the various objects and in making an actual model it is necessary to obtain copies of the primitives of an appropriate size, position and orientation in space. If the primitives are considered as unit solids, then transformations, such as scaling, translation and rotation are used to generate appropriately sized and located copies.

Some simple object primitives are implicit surfaces. Mathematically an implicit surface can be defined by an equation of the form $F(x,y,z) = 0$, in which F is a "well-behaved" function.

An implicit surface divides 3-D space into two parts, the part where $F(x,y,z) \geq 0$ and the part where $F(x,y,z) < 0$. These parts are termed half-spaces. If F is of degree two a quadric surface is defined. For example: $x^2 + y^2 - r^2 = 0$, which defines an unbounded cylinder surface in 3-D space. The plane $ax + by + cz + d = 0$ is a first-

degree implicit surface. A completely general quadric surface can be expressed as following:

$$ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxy + 2gx + 2hy + 2jz + k = 0, \qquad (4.1)$$

For convenience, Eq. (4.1) can be expressed in the form of a matrix equation:

$$[x, y, z, 1]\begin{bmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0. \qquad (4.2)$$

By employing matrix representation, an arbitrary 4×4 transformation matrix, such as scaling matrix, translation matrix and rotation matrix can be applied to it easily.

Many familiar objects in the world are quadric surfaces, such as spheres, ellipsoids, cylinders, cones, and paraboloids. Figure 4.3.1 depicts a number of quadric surfaces. By using quadric surfaces, the simple objects (primitives) can be defined by half-spaces. For example, a cylinder with radius r and height h can be defined as the intersection of two planes and a unbounded cylinder of the half-spaces:

$$\begin{aligned} & x^2 + y^2 - r^2 \leq 0, \\ & z \geq 0, \\ & z \leq . \end{aligned} \qquad (4.3)$$

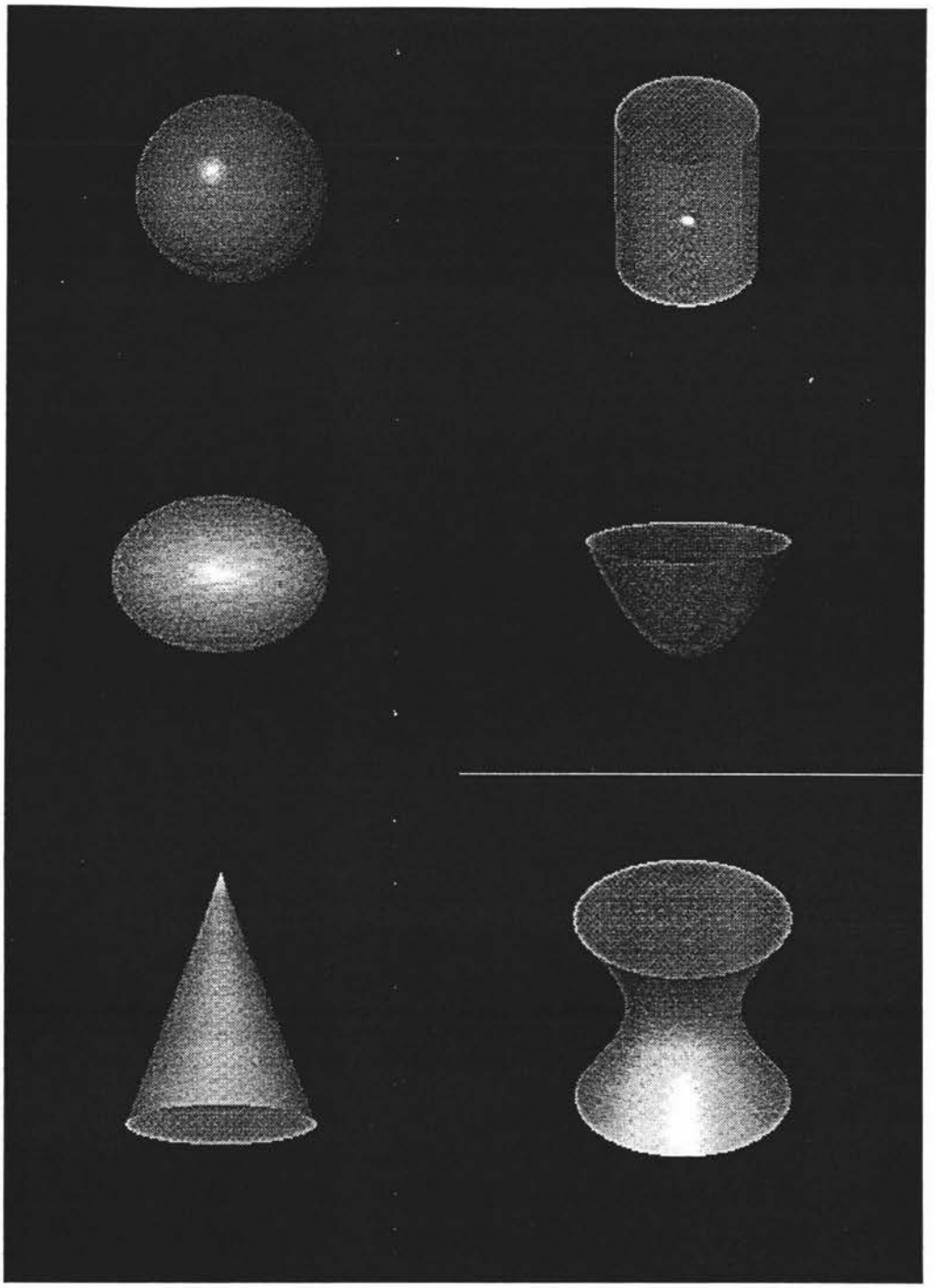


Figure 4.3.1 A number of Quadric surfaces (refer to Figure 4.9.9 footnote).

4.4 CSG Ray Tracing - Sampling the object

In section 4.2 and 4.3, we described that the CSG model is used to represent an object by means of Boolean set operations applied to some primitives. These primitives are in the continuous form. Because a digital computer can only process discrete values, we have to sample the object to get a finite set of points of the object. For synthesizing a hologram, we need the amplitude and the phase information of

each sampled point. We know that for a certain moment and certain observed point in the space, the intensity of the observed point is proportional to the square of the amplitude of the light when it travels through the point [Moller, 1979], that is

$$I_{mn} = |U_{mn}|^2 \qquad (4.4)$$

where I_{mn} is the intensity of the point in the m row and n column. U_{mn} is the amplitude of that point. $|U_{mn}|$ is the absolute value of the amplitude. Therefore, if we obtain the intensity of a point then we can get the amplitude of that point.

One popular technique to get the intensity of a point is called ray tracing [Roth, 1982], which is a display algorithm and could be used in CSG. On the CRT ray tracing directly display an object which is represented by a CSG model from the CSG tree. In order to display the object, ray tracing should perform CSG classification as a part of the display process, restricted to the visible parts of the model. This is equivalent to determine the intensity of each pixel of the screen if we restrict ourselves on that the object is illuminated by monochrome light.

4.5 Basic Principles of Ray Tracing

The basic assumption of ray tracing is that the light travels as a straight line. This is in conflict with our assumption that the light travels as wave in the synthesizing hologram domain. This conflict does not affect our goal of synthesizing holograms, because at this moment what we need is to get the amplitude information of one point (about the phase information we will discuss in chapter 5).

Ray tracing is originally a technique for generating high-quality images [Whitted, 1980]. It is conceptually simple and elegant. Figure 4.5.1 illustrates the basic principle of ray tracing. The three dimensional object to be sampled resides in the object space, and is observed through an imaginary rectangular window which may be thought of as being divided into a regular grid, for example M×N small elements. These elements correspond to pixels or resolution elements at desired resolution. The window is located between the eye of the observer and the object space. We sample the object by sending rays from the eye point through the center of the resolution element of the window into object space, and determining where the ray hits the surface of the object. In terms of geometry, these points are the intersections of the rays with the object surface.

If the ray hits the object, then the point that the ray meets with the object surface can be calculated, and a lighting calculation is performed at the intersection point to

determine the intensity contributed to the resolution element or pixel by the surface hit. This calculation uses the surface normal at the intersection point and other elements that we will discuss in section 4.6. Ray tracing offers us a simple and elegant way to sample the object. Assume that the size of the resolution element of the window is small enough, then we can get a finite set of points which is exactly we want for our synthesizing hologram problem.

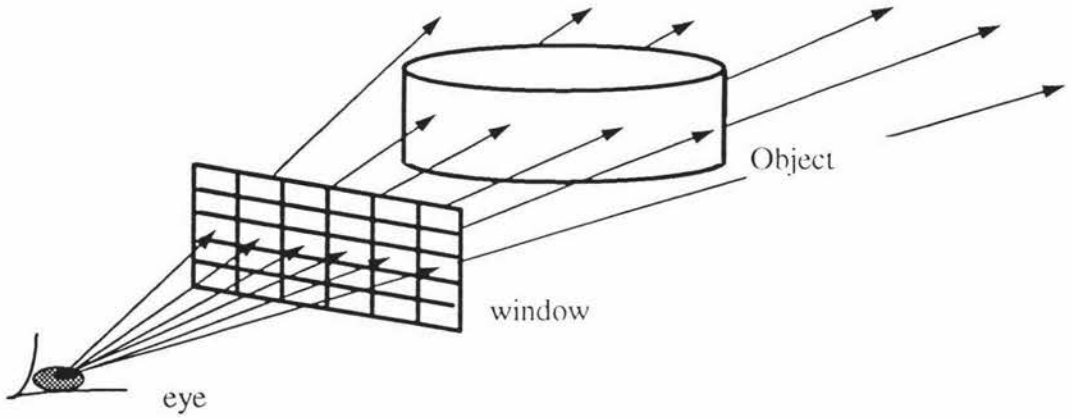


Figure 4.5.1 Basic principle of ray tracing [van Wijk, 1986].

4.6 Intersection Calculation

Intersection calculation is the heart of CSG ray tracing. It is a computation that determines whether a given ray hits the object, and if so, computes the location of the intersection To do this, first we represent a ray as two vectors: the location of the ray origin and direction vector where the ray is headed. Every point P on the ray can be described parametrically, in terms of a variable t :

$$P = loc + t\ dir \tag{4.5}$$

where $loc = (x_0, y_0, z_0)$ is the origin of the eye point; $dir = (x_1 - x_0, y_1 - y_0, z_1 - z_0)$, the point $P_1 = (x_1, y_1, z_1)$ is the center of resolution element on the window. For $t = 0$, P lies on the origin (**loc**). As t increases P moves long the ray, getting farther from the origin. If $t < 0$. the point is on the same line as the ray, but behind the ray origin. All objects that intersect the ray satisfy that equation for some t, Figure 4.6.1 depicts a ray and a few points along it.

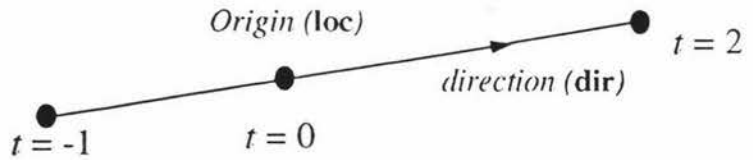


Figure 4.6.1. Anatomy of a ray. The dots signify points with ray parameters $t = -1$, $t = 0$ (the origin) and $t = 2$.

The direction of vector of a ray is normalized. The reason is that if the ray direction is normalized, the ray parameter for a given point indicates that point's distance from the ray origin, and it is convenient for writing ray-object intersection routines (see the description of the sphere intersection routines as an example).

Having a suitable representation of the ray, now we can solve the problem of the intersection calculation. To do this , we can separate the parameterized equation of a ray (Eq. (4.5)) in terms of a single variable t again as following:

$$\begin{aligned} x &= \text{loc}.x + t\ \text{dir}.x \\ y &= \text{loc}.y + t\ \text{dir}.y \\ z &= \text{loc}.z + t\ \text{dir}.z. \end{aligned} \tag{4.6}$$

Now, what we need is to find a representation for each kind of object that enables us to determine t at the object's intersection with the ray. One of the easiest objects for which to do this is the sphere. The sphere with center $C_0=(a, b, c)$ and radius r can be represented by the equation

$$(x-a)^2+(y-b)^2+(z-c)^2-r^2=0 \tag{4.7}$$

Substituting the values of x, y, and z from Eq. (4.6) into Eq. (4.7) yields

$$(\text{loc}.x + t\ \text{dir}.x - a)^2+(\text{loc}.y + t\ \text{dir}.y - b)^2+(\text{loc}.z + t\ \text{dir}.z - c)^2-r^2=0 \tag{4.8}$$

Collecting terms gives

$$A t^2 + B t + C = 0, \tag{4.9}$$

where

$$\begin{aligned} A &= \text{dir} \bullet \text{dir}, \\ B &= \text{dir} \bullet (\text{loc} - C_0), \\ C &= (\text{loc} - C_0) \bullet (\text{loc} - C_0) - r^2, \end{aligned} \tag{4.10}$$

where the '•' denotes dot product and the '–' denotes vector subtraction. If the ray direction vector is normalized, A is guaranteed to be 1, thus Eq. (4.9) becomes

$$t^2 + B t + C = 0. \tag{4.11}$$

Eq. (4.11) is a quadratic in t, with coefficients expressed entirely in constants derived from the sphere and ray equations, so it can be solved using the quadratic formula. If

there are no real roots, then, the ray and the sphere do not intersect; if there is one real root, then the ray grazes the sphere. Otherwise, the two roots are the points of the intersection with the sphere; the one that yields the smallest positive t is the closest which is the one we need. The intersection of a ray with the general quadric surfaces introduces in section 4.3 can be obtained in a similar fashion.

Finding the intersection of a ray with a plane is quite easy. Since the equation of a plane is

$$Ax + By + Cz + D = 0, \tag{4.12}$$

substitution from Eq. (4.6) yields

$$A(\text{loc}.x + t\ \text{dir}.x) + B(\text{loc}.y + t\ \text{dir}.y) + C(\text{loc}.z + t\ \text{dir}.z) + D = 0, \tag{4.13}$$

and

$$t = -\frac{D + A\,\text{loc}.x + B\,\text{loc}.y + C\,\text{loc}.z}{A\,\text{dir}.x + B\,\text{dir}.y + C\,\text{dir}.z}. \tag{4.14}$$

If the denominator of Eq. (4.14) is zero, then the ray and plane are parallel and do not intersect. Otherwise, there is a intersection point.

4.7 Find the Normal

The normal of a surface at a point is used to find the shade at that point, we will discuss the shade in section 4.9. The normal to a quadric surface (indeed, the normal to any implicit surface) is parallel to the *gradient* at the point given. The *gradient* of a function $F(x,y,z)$ is a vector

$$\nabla F(x, y, z) = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right). \tag{4.15}$$

For a quadric surface the *gradient* is

$$(2ax+dy+fz+g,\ \ 2by+dx+ez+h,\ \ 2cz+ey+fx+j) \text{(refer to Eq. (4.1))} \tag{4.16}$$

For a plane, its gradient is (A,B,C) (refer to Eq. (4.12)).

4.8 CSG Ray Classification

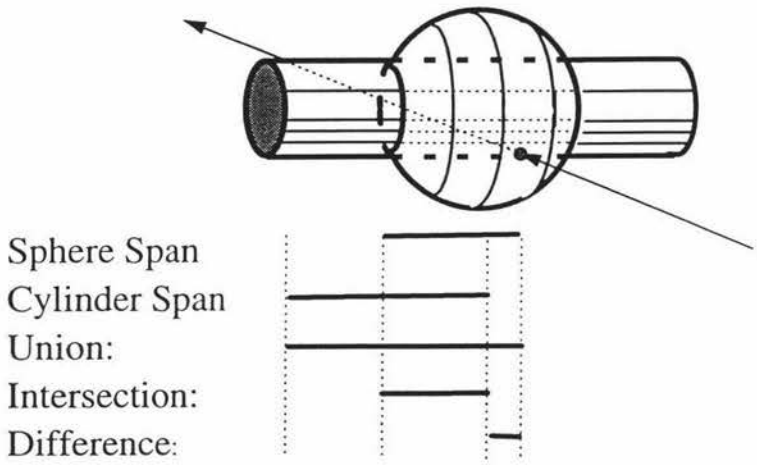
Assume that we already have a ray tracing system that can determine the intensity of each resolution element on the window for all types of primitive objects available in a solid modeling system, it can be adapted for determining the intensity of each resolution element on the window for the composite object by performing the CSG classification for each ray. The aim of this is to work out which parts of the primitive objects found along the ray path belong to the composite object.

As we described before, CSG is based on a collection of primitive solids. Instances of these are scaled, rotated and translated in 3D space, and then combined with the set operation to form more complex, composite objects. In determining the 3D set operation (union, difference, and intersection) of two solids is very difficult when it must be done by direct comparison of one solid with another if we use other representations such as boundary representation [Foley, van Dam, Feiner and Hughes, 1990]. In contrast, ray tracing allows the 3D problem to be reduced to a set of simple 1D calculations. The intersections of each ray and primitive object yield a set of t values, each of which specifies a point at which the ray enters or exits the object. Each t value thus defines the beginning of a span in which the ray is either in or out of the object. Boolean set operations are calculated one ray at a time by determining the 1D union, intersection , or difference of the span from the two objects along the same ray. In this way, the spans belonging to the evaluated CSG model can be found by traversal of the CSG tree. Figure 4.8.1 (next page) shows the spans defined by a ray passing through two objects, and the combinations of the spans that result when the set operations are performed .

The pseudo-code of an algorithm to perform ray classification by recursive, bottom-up tree traversal is shown in figure 4.8.2.

```
/*buttom-up ray classification algorithm */
raysegments *Rayclass( objectdef * object, Ray ray)
{ Raysegments * leftsegments, *rightsegments;
 if (object ->type == primitive)
   {return computed segment of ray inside primitive;}
 else {leftsegments=Rayclass(object->left);
       rightsegments=Rayclass(object->right);
       return combine(leftsegments,rightsegments,object->op);
  }
}
```

Figure 4.8.2 Bottom-Up Ray Classification.



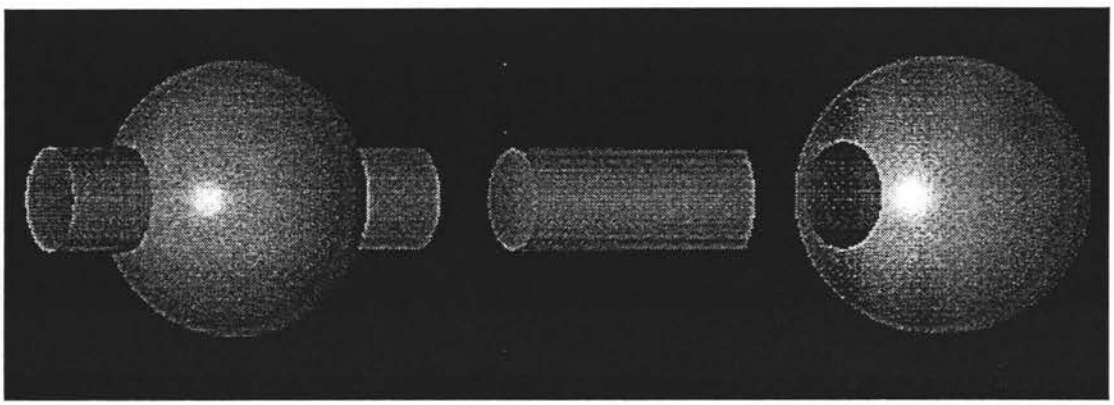


Figure 4.8.1 Performing CSG classification with a ray (refer to figure 4.9.9 footnote).

4.9 Illumination and Shading

In section 4.8, we assumed that we already have a ray tracing system that can determine the intensity of each resolution element on the window for all types of primitive objects available in a solid modeling system. In this section we will concentrate on where the light that falls on our objects come from, its qualities, and how the object interacts with it.

To get the intensity of each resolution element is equivalent to shading the object surfaces base on the position, orientation, and the characteristics of the surfaces and the light sources illuminating them. A model for the interaction of light with a surface normally is called an illumination model. Many illumination models exist, such as the Phong model, diffuse reflection, and ambient light. We concentrate on the diffuse reflection model and ambient light model, because these models are used in this research project.

4.9.1 Light Sources

Every object we see in the real world emits light. This light has different origins. One possibility is that the object itself produces the light and emits it, for example, the sun or a lamp. Another more common case is that the light comes from somewhere else and it reflected by the object. The second case is the only one we will study in following sections.

When we see an object, we actually see the light reflected from its surfaces. The light that is reflected originates basically from two different types of sources. One type we call a light-emitting source. The other type we will call a light reflecting source because the light coming from it is not produced by it.

light emitting sources, can be neither point source or distributed sources. If the light emitting surface is small compared to the surface onto which it shines, we have a point source, otherwise we have a distributed source. The laser beam light in the optical holography system is light emitting source and can be considered as a point source.

Light reflecting source are all non-emitting objects in the scene, including objects that are not shown in the scene, in other words, the whole environment. When we try to produce a realistic display, we must recognize that reflected light is coming from practically all directions: the wall of the room, all illuminated objects, the sky, the landscape, whatever. It follows that a surface is not exposed directly to a light emitting source will still be visible because of the multitude of reflecting source around it. For example, something that is outside but shaded from the sun is still visible because the sky itself and many other things are light reflecting sources. Light coming from such sources is called ***ambient light*** [Pokorny and Gerald,1989].

4.9.2 Diffuse Reflection

When light falls on a surface, it can be reflected, absorbed, or transmitted through the surface. These three effects are not exclusive; usually all of them take place. The precise physical process of reflection is very complicated; we consider only a simple model of diffuse reflection to get the necessary information for our synthesizing hologram purpose.

In diffuse reflection, incoming light is not reflected in a single direction but is scattered almost randomly in all possible directions. In addition, the incoming light is influenced by the surface. A surface will hardly reflect all the incoming light; part will be absorbed by the surface. The part that is not absorbed will be reflected randomly in

all directions. Therefore, the direction from which the incoming light comes is unimportant. If the object is illuminated by ambient light, and we assume that the ambient light impinges equally on all surfaces from all directions. The illumination equation for this model can be written as

$$I=I_a k_a \tag{4.17}$$

where I is the intensity of the object, I_a is the intensity of the ambient light, assumed to be constant for all objects. The amount of ambient light reflected from an object's surface is determined by k_a, the ***ambient reflection coefficient***, which ranges from 0 to 1. The ambient reflection coefficient is a material property, it may thought of as characterizing the material from which the surface is made. The ambient reflection coefficient is an empirical convenience and does not correspond directly to any physical property of real materials. Because in Eq. (4.17) I is a constant, this means that the light coming from the ambient light will always be the same, producing a uniform illumination of the surface at any viewing position. The illumination will always be the same no matter whether the surface is curved or not. Figure 4.9.1 shows an example of an object illuminated by ambient light. Therefore, ambient light by itself is not of much interest. As we see later, it is used to account for all the complex ways in which light can reach an object that are not otherwise addressed by the illumination equation.

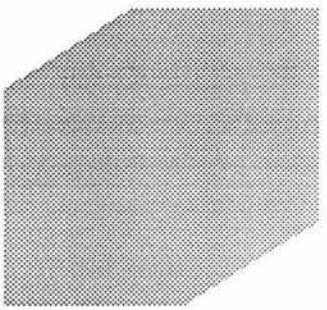


Figure 4.9.1. A cube illuminated by ambient light. What we can see in this figure is just a polygon.

Now we consider diffuse reflection from a point source. Point light source differs from ambient light in that we must consider the angle from which it comes in our reflection model. This angle called the incident angle that is the angle between the vector that points to the light source, L, and the surface normal at the point N as shown in figure 4.9.2.

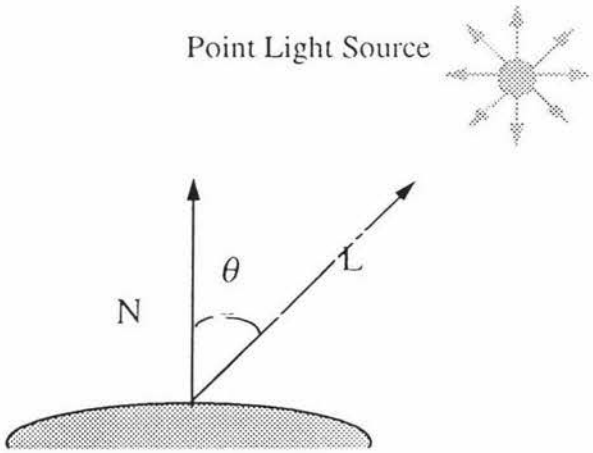


Figure 4.9.2. The incident angle.

Diffuse reflection from a point source also known as Lambertian reflection. For Lambertian surface (the surface reflects light as diffuse reflection), the amount of light seen by viewer is independent of the viewer's direction and is proportional only to $\cos\theta$, the angle of the incidence of the light. This is the *Lambert's cosine law.* The diffuse illumination equation can be written as

$$I=I_p k_d \cos\theta. \tag{4.18}$$

where I_p is the point light source's intensity; the material's diffuse reflection coefficient k_d is a constant between 0 and 1 and varies from one material to another. The angle θ must be between 0 and $\pi/2$ if the light source is to have any direction effect on the point being shaded. This means that we are treating the surface as self-occluding, so that light cast from behind a point on the surface does not illuminate it [Foley *et al*, 1990].

Lambert's cosine law also told us that the intensity of the impinging light is proportional to the density of light rays that hit an area of constant size. Therefore, a factor that influences the density of the point light is the distance of the point light source from the surface. In many application, the distance from the point source can be ignored because the distances involved are so large compared to the size of the illuminated surface. But sometimes, it is important to allow for this, so Eq. (4.18) can be modified as

$$I=(I_p k_d \cos\theta)/d^2. \tag{4.19}$$

If the normal N of the surface and the incident vector L are normalized, then the cosine of the incident angle θ can be computed as the **dot product** of the vector L pointing to the light source and the vector N normal to the surface:

$$\cos\theta = \mathrm{N}\bullet \mathrm{L}. \tag{4.20}$$

Figure 4.9.3 shows a series of simulations of a sphere illuminated by a single point source. These simulations are displayed on a CRT. The shading model calculated the intensity at each pixel at which the sphere was visible using the illumination Eq. (4.19). Objects illuminated in this way look harsh, as when a flash gun illuminates an object in an otherwise darkroom. Therefore, an ambient component is commonly added to produce a more realistic illumination equation:

$$I=I_a k_a + I_p k_d \cos\theta \qquad (4.21)$$

where $k_a+k_d <= 1$.

Figure 4.9.4 shows the pictures produced by Eq. (4.21).

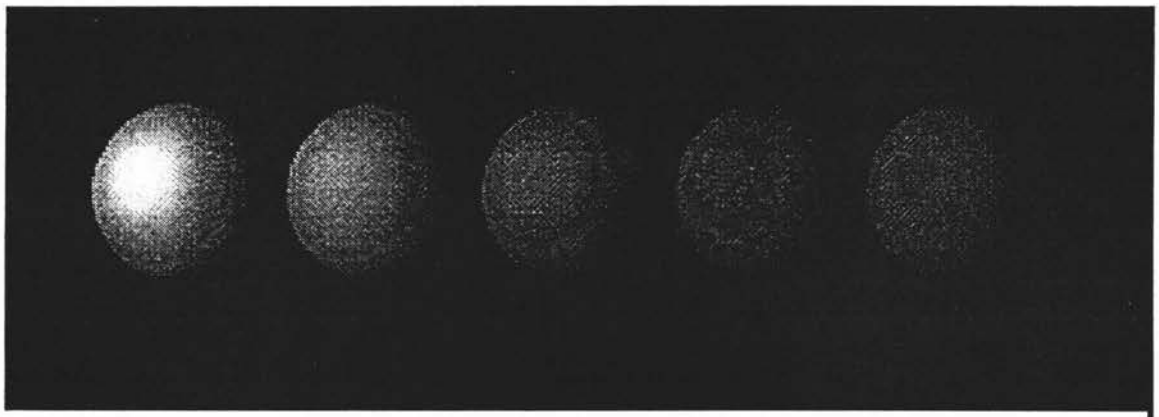


Figure 4.9.3 Spheres shaded using a diffuse reflection model Eq.(4.19). From left to right, k_d = 1.00, 0.85, 0.70, 0.55, 0.45.

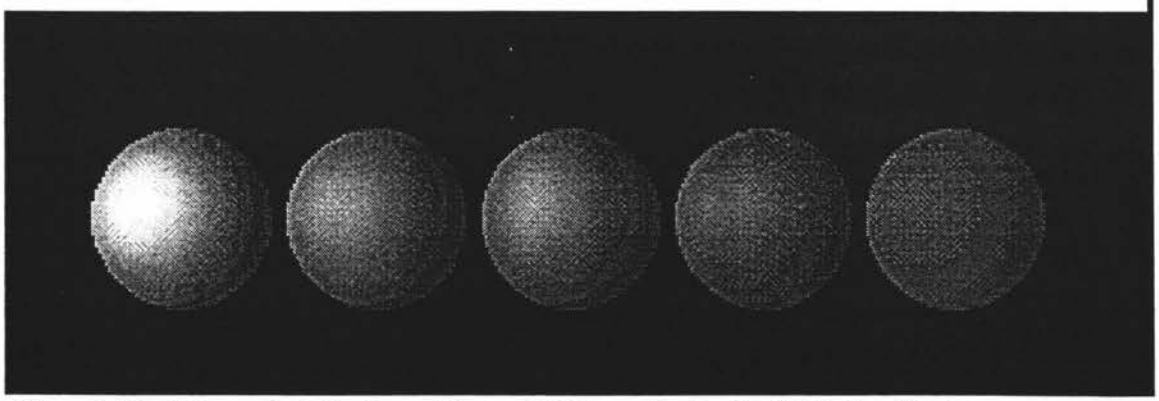


Figure 4.9.4. Spheres shaded using ambient and diffuse reflection Eq.(4.22). For all spheres, k_a = 0.30. From left to right k_d =0.7 , 0.6, 0.55, 0.45, 0.30. We used 16 intensities interpolated with $D^{(4)}$ (refer to next section and figure 4.9.9 footnote) to draw these pictures .

4.10 Dynamic Range And Intensity Level

In above sections, we have discussed how to represent an object and the methods to obtain the intensities of the object. In principles, we now should have the ability to display an object on a CRT according to the sampled intensity values of the object. The maximum intensity we obtained has been assumed to be 1.0 and the minimum is 0.0. In computer graphics, to display the object we can map these values to integers by multiplying these values by a constant which is the number of the intensity levels. The number of intensity levels is the range of intensities can be really displayed on the screen by a computer. The number of the intensity levels is determined by ***dynamic range*** which is the ratio between the maximum and the minimum intensities. The dynamic range in synthesizing hologram is a very tricky problem we will discuss it again later. In the real word, there is no upper limits for the maximum intensity, but this is not the case in computer graphics. Normally the maximum intensity is specified as 1.0. The minimum attainable intensity I_0 for a CRT is anywhere from about 0.005 up to 0.025 of the maximum intensity of 1.0, thus the dynamic ranges correspond to these values are from 200 to 40. The dynamic range typically is quantized into 256 intensity levels and identified with integers from 0 to 255 in the computer.

This number of intensity levels corresponding to the dynamic range from 40 to 200 is good enough for us to produce continuous-tone black and white images such that the production appears to be continuous, because practical tests show us that 64 intensity levels is the absolute minimum needed for contour-free printing of continuous-tone black and white images on paper [Foley *et al.* 1990]. But many computer screens can only display few intensities and some of them can only display two intensities, that is, black and white. If we use bilevel intensities to display an object, the object will appear as illuminated by ambient light as shown in figure 4.9.5. If we use 16 intensity levels to display an object, contouring appears as shown in figure 4.9.6.

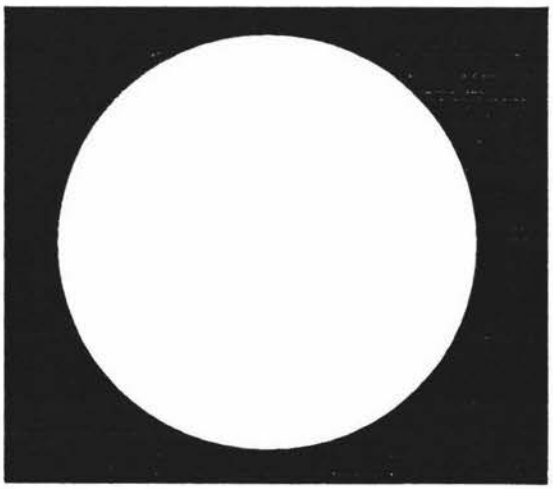


Figure 4.9.5 Bilevel display of a sphere (refer to figure 4.9.9 footnote).

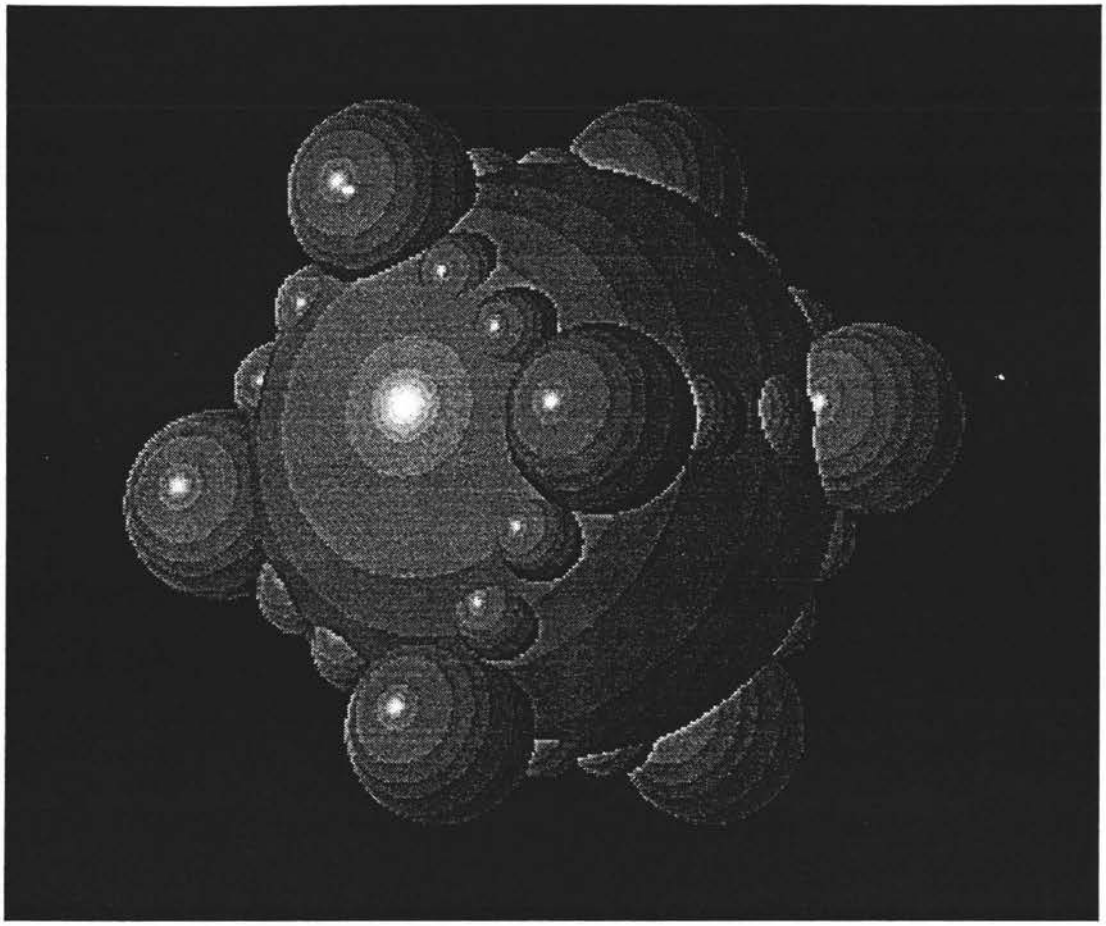


Figure 4.9.6 Spheres displayed using 16 intensity levels. (refer to figure 4. 9. 9 footnote.)

One solution of this problem is to use dither patterns to approximate the image to be displayed. The idea of dither pattern is that using several resolution units of the display device to present one resolution unit of the image. For example, assume that we want to display an image with 4 intensity levels on a CRT, we can use a 2×2 pixel area of bilevel to produce 5 intensity levels at the cost of halving the spatial resolution along each axis. The patterns shown in figure 4.9.7 can be used to fill the 2×2 areas with the number of 'on' pixels that is proportional to the desired intensity.

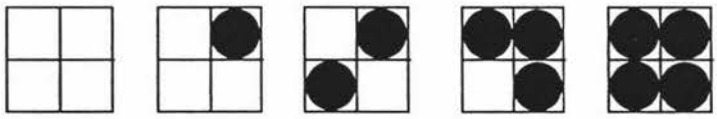


Figure 4.9.7 A 2×2 dither pattern.

A dither pattern can be represented by dither matrix. There are many possible dither matrices. Bayer [Bayer, 1973] has developed dither matrices that minimize the texture introduced into the displayed images. For the case 2×2, the dither matrix, represented by $D^{(2)}$, is

$$D^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}. \qquad 4.22)$$

To display an intensity I, we turn on all pixels whose values are less than I. In general an $n \times n$ group of bilevel pixels can provide n^2+1 intensity levels but it will reduce the spatial resolutions of the display device to one-nth on each axis.

Large dither matrices can be found using a recurrence relation [Judice, Jarvis and Ninke, 1974] to computer $D^{(2n)}$ from $D^{(n)}$, with $U^{(n)}$ defined as an $n \times n$ matrix of 1s, that is,

$$U^{(n)} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ . & . & . & . \\ 1 & 1 & \dots & 1 \end{bmatrix}. \qquad (4.23)$$

the recurrence relation is

$$D^{(n)} = \begin{bmatrix} 4 * D^{(n/2)} + 0 * U^{(n/2)} & 4 * D^{(n/2)} + 2 * U^{(n/2)} \\ 4 * D^{(n/2)} + 3 * U^{(n/2)} & 4 * D^{(n/2)} + 1 * U^{(n/2)} \end{bmatrix}. \qquad (4.24)$$

Here * denotes scaling the matrix. Applying this relation to $D^{(2)}$ yields

$$D^{(4)} = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}. \qquad (4.25)$$

As we motioned before, in order to produce a continuous-tone black and white image, the intensity levels must be great than 64. If we use dither pattern to produce these intensity levels we need a 8×8 dither pattern, this means we have to use 64 pixels to represent one resolution unit of the image, this wastes a lot of the resolution units of the display device and if the number of the resolution units of the image is great than the number of resolution units of the display device, then we could not produce these intensity levels. One approach to solve the problem is a simple adaptation of the dither pattern. Assume that we want to display an image which has 512×512 samples with 256 intensity levels, we have to use a $D^{(16)}$ dither pattern to represent these intensity levels, but instead of using 256 pixels to present one sample, we determine whether or

not turn on the pixel at (x,y) by this way: if the image at point (x,y) has intensity $I(x,y)$ then we can compute

$$\begin{aligned} i&=x \text{ modulo } 16, \\ j&=y \text{ modulo } 16. \end{aligned} \tag{4.26}$$

Then, if

$$I(x,y) > D_{ij}^{(16)}. \tag{4.27}$$

we turn on the pixel at point (x,y), otherwise we turn off the pixel. Figure 4.9.8 shows an example of this approach.

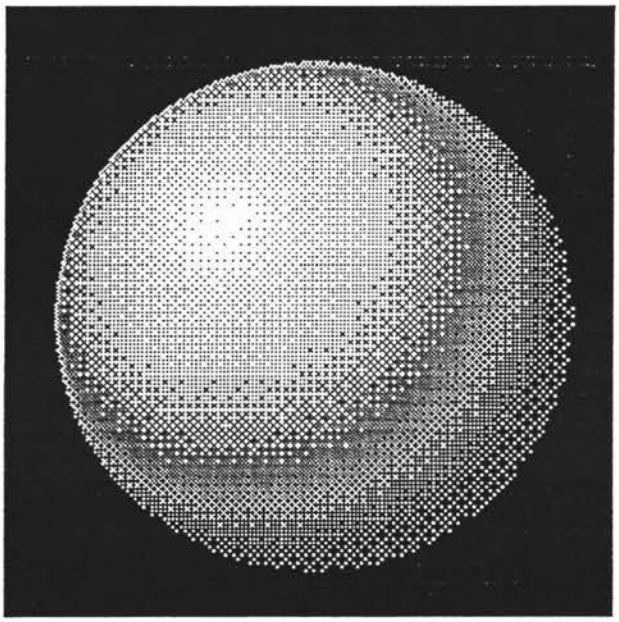


Figure 4.9.8 A continuous-tone picture produced with $D^{(16)}$. We made this picture on a SUN-3/80 with a black and white screen. The used illumination equation is Eq. (4.21). Parameters: $I_a=I_p=256$, $k_a=0.25$, $k_d=0.75$.

Comparing this picture with figure 4.9.5, shows that the visual effect of this picture is far better than the visual effect of figure 4.9.5 shows. This approach is not only restricted to bilevel display devices but also can be used for the case that the display device can display only a few intensities. For example, if a display device can display 16 intensities, we can use a 2×2 dither pattern to increase the display intensities of the device to 64. To do this, assume that we want to display an image with 64 intensity levels, we determine the intensity at point (x,y) as follows

$$\begin{aligned} i&=x \text{ modulo } 4, \\ j&=y \text{ modulo } 4. \\ I_r &= \text{round } \{I(x,y)/16\}, \end{aligned} \tag{4.28}$$

then, if $I(x,y)$ modulo $4 > D_{ij}^{(2)}$, where i and j denote the i row and j column of the 2×2 dither matrix respectively, we display the point (x,y) with intensity I_r, otherwise we display the point with I_r -1. This method may be called intensity interpolation. Figure 4.9.9 shows an image produced by this method. The method described by Eq. (4.26) and Eq. (4.27) is a special case of this method, that is it has only two intensity levels.

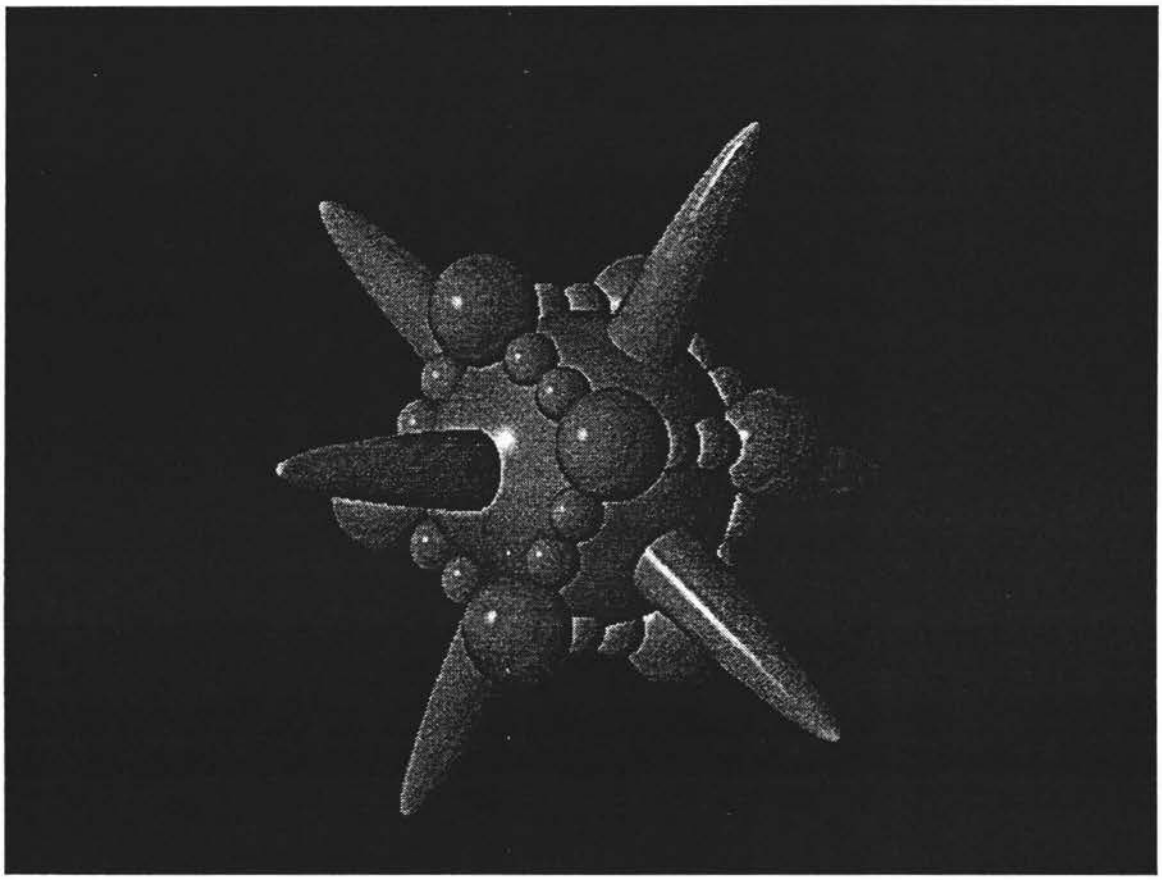


Figure 4.9.9 An object displayed by 16 intensity levels and interpolated with $D^{(4)}\uparrow$.

Comparing figure 4.9.9 with figure 4.9.6, shows that the picture's visual effects have been improved. This method eliminates the contouring effects in figure 4.9.6, but does not reduce any spatial resolutions of the display device.

↑: We used a PC to produce figure 4.3.1, 4.8.1, 4.9.3, 4.9.4, 4.9.5, 4.9.6, and 4.9.9. These pictures were drawn by ray tracing. The illumination equation for these pictures (figure 4.9.3 and 4.9.4 are not included) is

$$I = I_a k_a + I_d(k_d \cos\theta + k_s \cos^n \theta)$$ (refer to Eq. (4.17), (4.18) and (4.21))

where k_s is a constant which ranges between 0 and 1, depending upon k_a and k_d, k_a+ k_d + k_s <= *1*. The item $k_s \cos^n \theta$ determines the high light spot size in these pictures, where n is an integer. The bigger the n is, the smaller the high light spot size will be. The parameters for these pictures were

I_a=I_p=64,
k_a=0.2,
k_d=0.55,
k_s=0.25,
n=80.

Because the PC we used can only display 16 intensity levels, so figure 4.3.1, 4.8.1 and 4.9.9 were displayed by 16 intensity levels and interpolated with $D^{(4)}$.

Chapter 5

Experiments With Synthesizing Holograms

In above chapters we have discussed the techniques which can be used to draw the interference patterns calculated by a computer to form a computer-generated hologram, the techniques to represent an object and how to obtain the samples of the object. In this chapter we will show some examples of the computer-generated holograms and discuss how the limitation of the dynamic range of the display device and dimension of the hologram affect the quality of the image reconstructed from a computer-generated hologram.

In our research laboratory, there are some SUN-3/80 computers, several personal computers and a laser printer. The display device of the SUN-3/80 computer is black and white with 1024×1024 pixels. The memory of the SUN-3/80 computer is quite large. The laser printer is a HP laserjet IIID printer which can only print out black and white images. The display devices of the PC computers are colour screens, but the memories of these computers are limited, they are not helpful for our synthesizing holograms, because we need a lot of memory. Since CRTs and printers are common devices in many laboratories, we will show some experimental results of computer-generated holograms depending on these devices.

5.1 Experiments with Synthesizing Waters's Hologram

Perhaps the simplest computer-generated hologram is Waters's hologram. We have discussed this type of computer-generated hologram in chapter four. To produce Waters's hologram we should follow the steps as shown in figure 5.1.

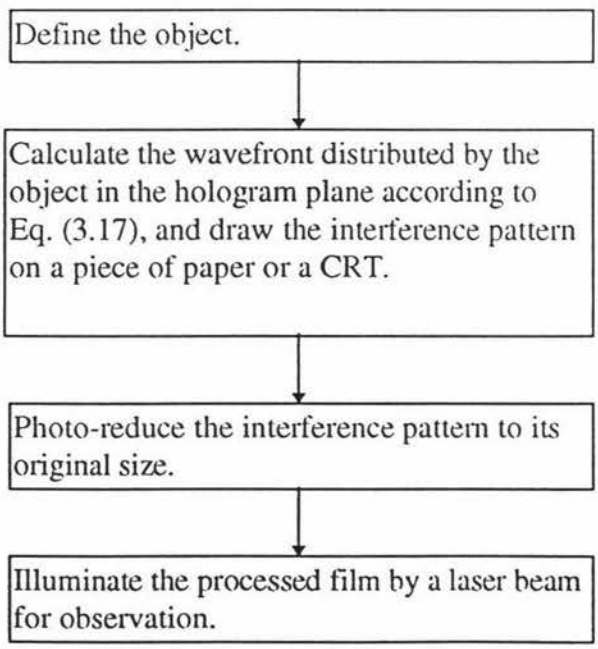


Figure 5.0 The procedure for making Waters's hologram.

First we consider the definition of the object in the two dimensional situation.

5.1.1 Define the object

The object we considered is the letter 'F', which is consisted of three lines. The object is bounded by a $10\times10\,mm^2$ square area. We used a array of three dimensional point to represent the object. The point in the array is defined by spatial cooardinates, and the interval between two points are δu=10/32 (mm) and δv=10/32 (mm) in the direction of v and v axis respectively. Thus the object is defined by 82 points. We set all of the z values of the points as a constant, this means all of the points are on the same plane z = 800. We will discuss the situation when z is not a constant, but at present, we restrict ourselves on z is a constant.

5.1.2 Calculate the wavefront at the hologram plane

The hologram plane is at of z = 0, and is also bounded by a $10\times10\,mm^2$ square area. Figure 5.1 illustrates the coordinate system we used and the relationship between the object plane and the hologram plane

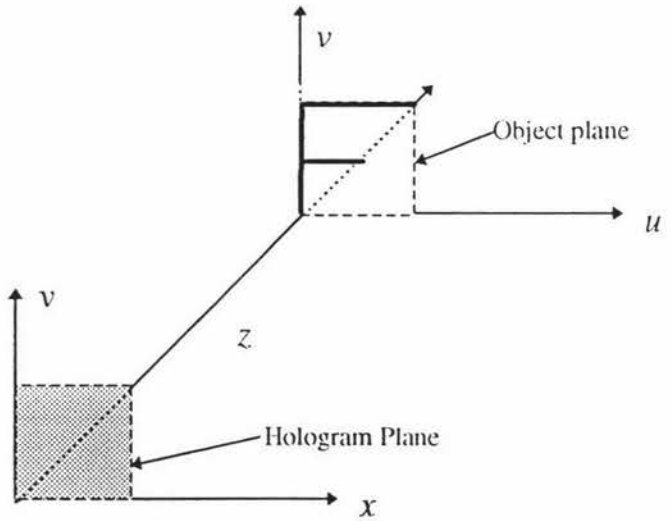


Figure 5.1 The relationship of the hologram plane and the object plane.

Now, let us consider how to calculate the wavefront distributed by the object at the hologram plane. Following Waters's work , we know that the wavefront distributed by the object at any point on the hologram plane can be calculated by Eq. (3.17) approximately, that is

$$F(x,y)=\sum_{i=0}^{N-1} sin\left[k\left(\frac{(x-points[i].u)^2+(y-points[i].v)^2}{2points[i].z}\right)\right] \quad (5.1)$$

where $k = 2\pi/\lambda$ with λ the wave length, for red light λ=0.6238×0.001 (mm), and $0 \le x \le 10$, $0 \le y \le 10$. In order to compute $F(x,y)$ and draw the interference pattern according to Eq.(5.1), we have to digitise $F(x,y)$. Consider the phase function

$$\varphi(x,y)=k\frac{(x-points[i].u)^2+(y-points[i].v)^2}{2points[i].z} \quad (5.2)$$

in the square braces in Eq. (5.1), it is the spherical wavefront phase variation at far field condition scattered by point $points[i]$. The maximum frequencies of this phase variation in the x-direction and y-direction are :

$$f_{maxx}=\frac{\max(x)-\min(u)}{\lambda points[i].z}=\frac{10.0}{0.6328\times 800\times 0.001}=19.75,$$

$$f_{maxy}=\frac{\max(y)-\min(v)}{\lambda points[i].z}=\frac{10.0}{0.6328\times 800\times 0.001}=19.75. \quad (5.3)$$

Thus, if we sample the hologram plane at distances $(\delta x, \delta y)$, where

$$\delta x = \delta y = 1.0 / (2 f_{maxx}) = 1.0 / (19.75 * 2), \quad (5.4)$$

then the sampling is sufficient because Nyquist's sampling theorem requires that the sampling numbers must be greater than twice the highest frequency component in its spectrum in order to avoid aliasing. For convenience, we take $\delta x=\delta y=1.0/51.2$, therefore, we obtained 512×512 sampled values of the function $F(x,y)$. If we use one pixel to represent one sampled value of the function $F(x,y)$, then we can draw the interference pattern according to these sampled values. To really draw it, we follow Waters's approach, that is, if $F(x,y) > 0$ then set pixel (x,y) to black, otherwise set it to white. A "C" function used to compute and draw the interference pattern according to Eq.(5.1) is given in appendix B. Figure 5.2 depicts a interference pattern we produced by using Waters's approach.

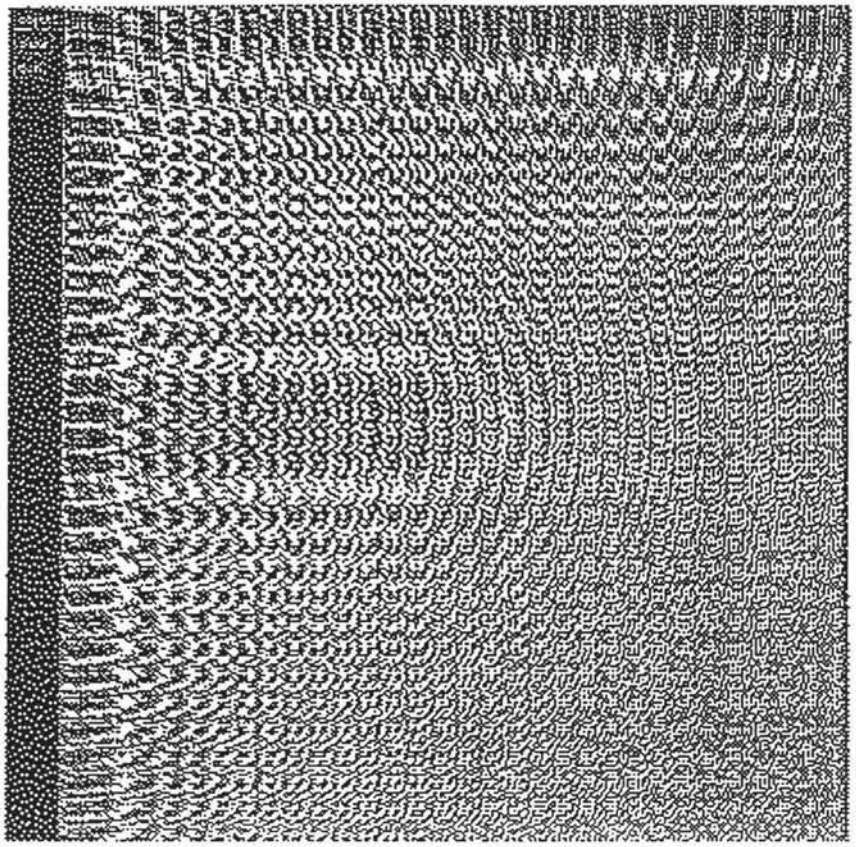


Figure 5.2 The interference pattern of the object "F" generated by the computer according to Waters's approach.

Figure 5.2 is first drawn on the CRT of the SUN-3/80 computer, then printed out on a laser printer. The pattern is photo-reduced to $10\times10\,mm^2$, thus we obtained a $10\times10\,mm^2$ square hologram.

In reconstruction, a red laser beam is used to illuminate it, Figure 5.3 shows the image reconstructed from the hologram.

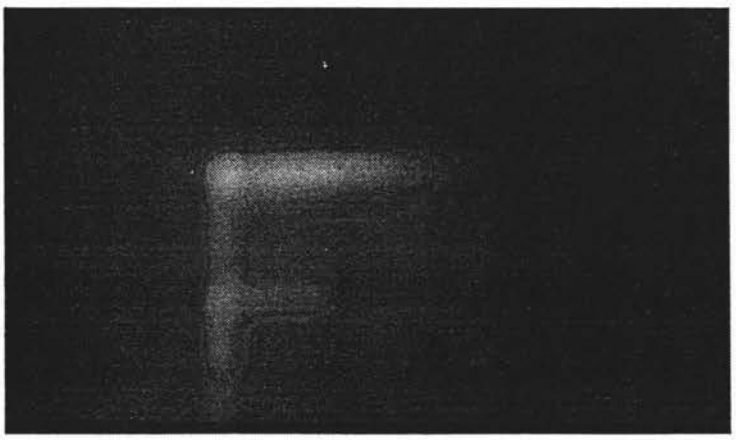


Figure 5.3 The image reconstructed from the hologram by red laser beam.

We surprisedly found that the hologram can be reconstructed by white light in normal condition environment. In a clear day, we used a car rear mirror to reflect the sun light as the light source illuminating the hologram, as a result, an image is also reconstructed from it. Figure 5.4 shows the image reconstructed by the white light reflected from the sun.

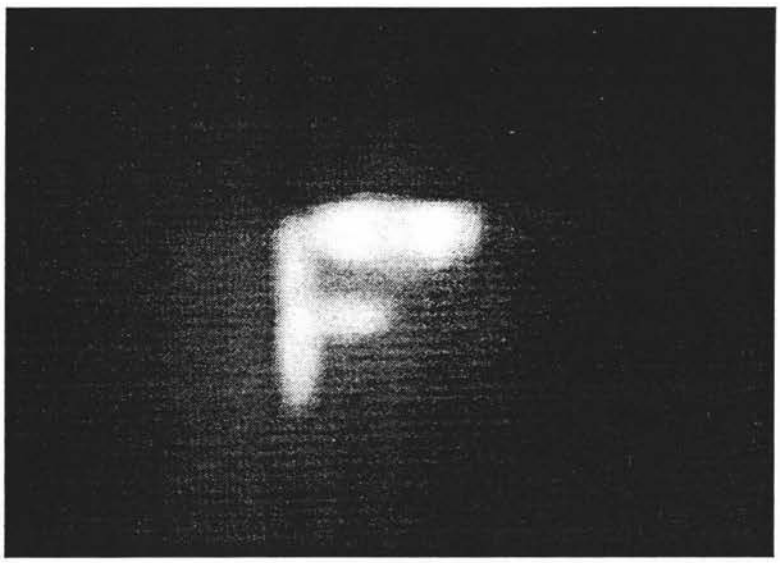


Figure 5.4 The image reconstructed from the hologram by the sun light.

The image reconstructed from the hologram we made is two dimensional because the object is defined on the object plane. In order to examine whether a three dimensional image can be reconstructed from this type of computer-generated hologram, we rotated the object by $\pi/4$ along the u-axis about the origin of the object plane, therefore the z value of each point of the object is not a constant, but the highest frequency of the wavefront distributed by the object did not change. Following the procedure we described above, another computer-generated hologram was made. The interference pattern is shown in figure 5.5.

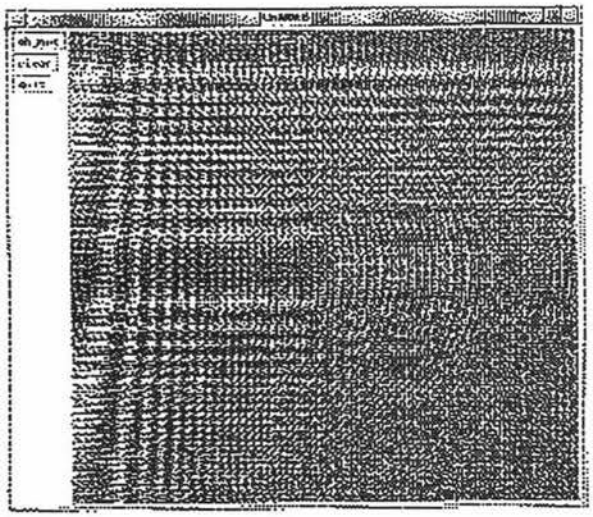


Figure 5.5 The interference pattern generated by the computer. The object is the "F" rotated by $\pi/4$ along the u-axis about the origin of the object plane.

Comparing figure 5.5 with figure 5.2, we can see that the object is really rotated, but the reconstructed image from the hologram did not exhibit the three dimensional properties. It was still two dimensional as shown in figure 5.4, only a slight difference could be seen by eye. The reason is that the largest **z** value of the object is 800+7.07, comparing with the smallest **z** value of the object , that is 800 , the difference is too small to be observed. Therefore we rotated the object by $\pi/2$ and translated it so that the object laid on the v=5 plane. In order to reconstruct an image with depth effect, we scaled the object so that the largest z value of the object became 1600, as shown in figure 5.6.

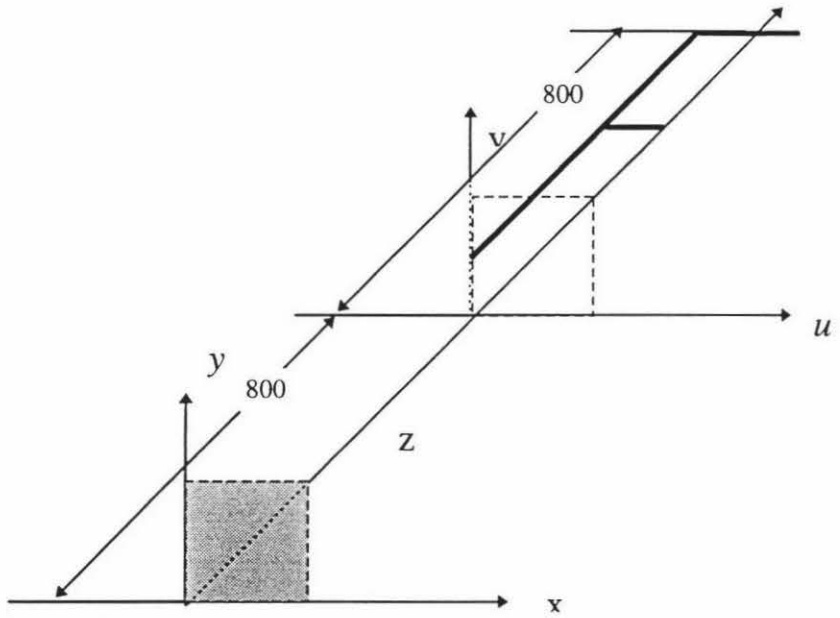
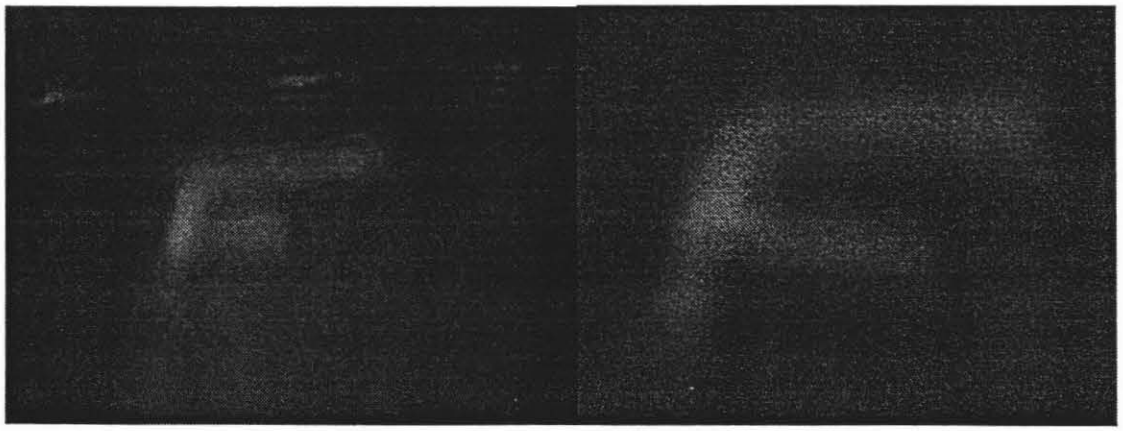


Figure 5.6 The relationship between the hologram plane and the scaled object.

The image reconstructed from this hologram (figure 5.8) exhibits all of the properties of a three dimensional object. As we move our eyes, we can see different views of the object. Figure 5.7 shows different views of the images reconstructed from the hologram. The interference pattern we produced according to this object is shown in figure 5.8.



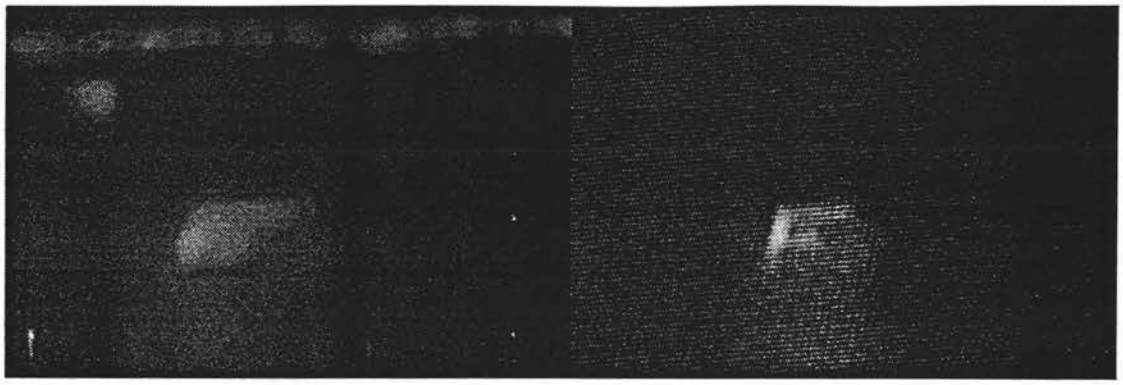


Figure 5.7. Images reconstructed from the hologram made by photo-reducing figure 5.8 to $10\times10\ mm^2$.

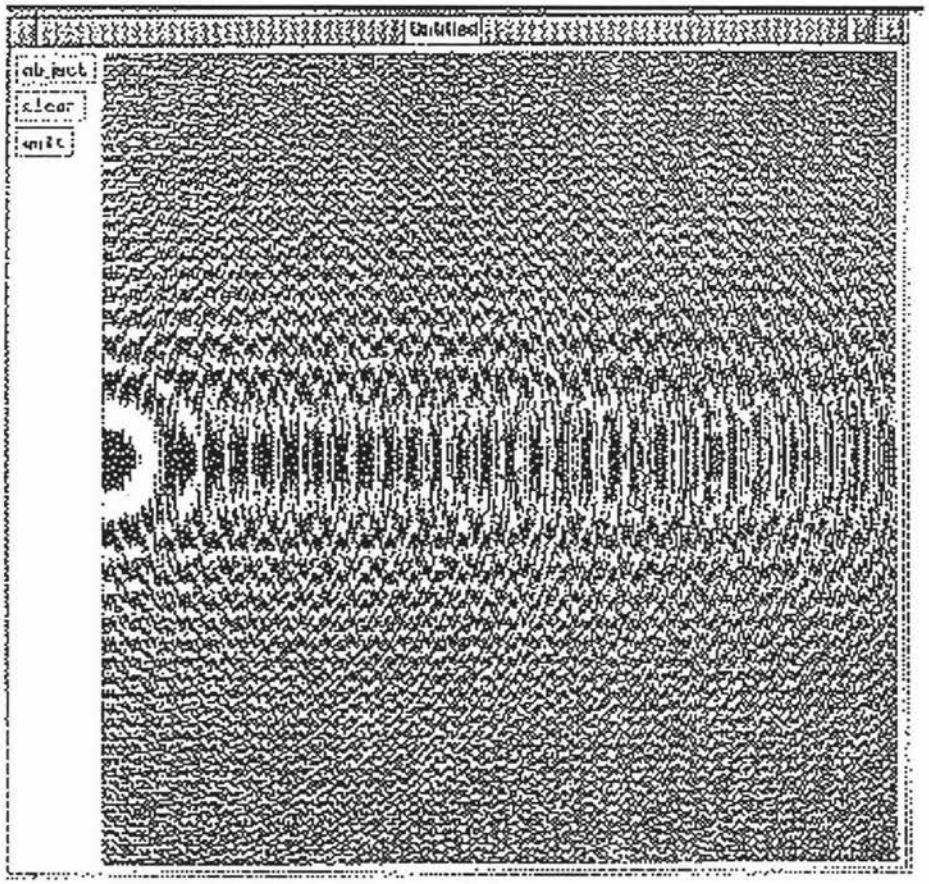


Figure 5.8 The three-dimensional object interference pattern generated by the computer. The object is the letter "F". All three bars of the object laid on the v=5 plane. In order to reconstruct an image with depth effect, we scaled the object so that the largest z value of the object became 1600(mm).

5.1.3 The properties of Waters's hologram

Waters pointed out that this type of computer computer-generated hologram has several properties: (a) the reconstructed image displays all the properties associated with three-dimensionality; (b) any fraction of the hologram can be used to reconstruct the entire image with an attendant loss in the resolution and decrease in the intensity

because of the smaller aperture; (c) either an on-axis or off-axis synthesis can be formulated. Our experimental results have shown these properties.

But Waters's hologram suffers from some problems. First, as we see the image reconstructed from figure 5.9, the quality of this image is quite poor. One of the reason is that the object is sampled very coarsely. In principle, the more sampled values of the object we get, the better the reconstructed image will be. Unfortunately, if we do not change the sample rate of the hologram plane, and just sample the object more, the reconstructed image will be blurred. Figure 5.11 shows example of this case. Figure 5.9 is produced by sampling the hologram at distances (δx=10.0/512.0 (mm), δy=10.0/512.0 (m)), the object is still the letter "F", but the interval between two points are δu=10.0/64 (mm) and δv=10.0/64 (mm) in the direction of u and v axis respectively. Thus the object is defined by 160 points. The reconstructed image is shown in figure 5.10. As a result, we could not clearly see what the reconstructed images represent. As we increase the sample rate in the hologram plane, the reconstructed image becomes better.

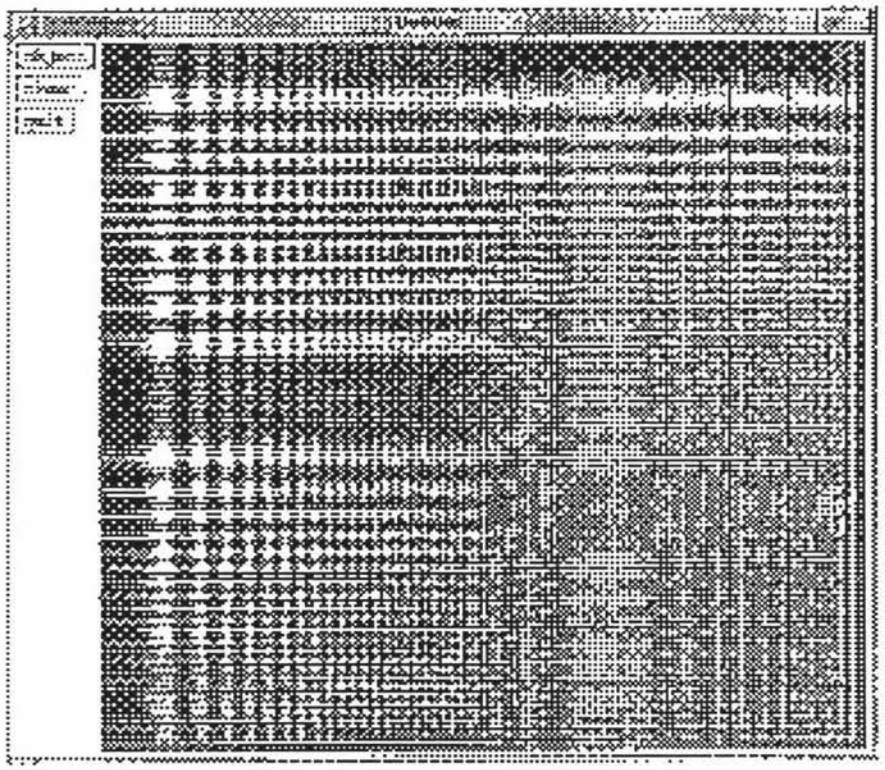


Figure 5.9. The interference pattern generated by the computer. The sampling distances of the object are δu=10.0/64.0(mm) and δv=10.0/64.0(mm). The object is the letter "F".

Figure 5.10. The image reconstructed from the hologram produced by photo-reducing figure 5.9 to $10\times10\ mm^2$.

Secondly, the computation time for producing a Waters's hologram is very long. It took about 2 minutes to draw the interference pattern of figure 5.7. The computation time to produce a Waters's hologram can be calculated by examining Eq. (5.1). For each sample of the hologram plane and for each point of the object, the computer needs two subtractions, six multiplications and one call of the "sin". Assume that the object consists of N points, then for each sample of the hologram, the total time needed is

[3(subtractions or additions)+6(multiply)+1(sin operation)]×N, (5.5)

assume that N=128×128 and the computer can complete the operations of Eq.(5.5) in S second (in our computer system S ≅0.06 second), if the sample number of the hologram is M×L =1024×1024, then it will take 17.5 hours to draw the interference pattern. This is only when the object is bounded by a small square. If we want to produce a bigger hologram, then the computation time will be extremely high.

5.2 Experiments with Synthesizing Fourier Holograms and Fresnel Holograms

In synthesizing Waters's holograms, as we discussed above, for a fairly simple object, the amount of calculation, and the computer time involved, is not large. But for complex objects this amount and time grows dramatically, and the approach is no longer applicable.

Computer-generated holograms can be made according to the mathematical description in chapter two. Now let consider synthesizing Fourier holograms. In synthesizing Fourier holograms, we should follow Eq. (2.23a) which describes the relationship between an object and its hologram plane. Thus the procedure can be described as following:

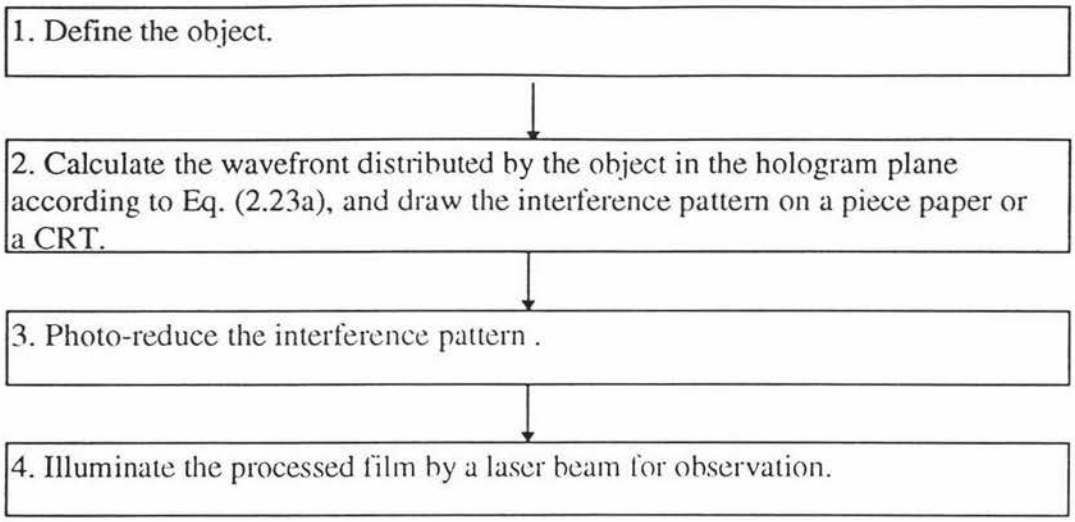


Figure 5.2.1 The procedure for making a Fourier hologram.

5.2.1 Define the object

To make a Fourier hologram, we need a two dimensional array of samples of the object to perform the Fourier transformation. In order to avoid aliasing, we have to obtain enough samples of the object. If the object is bounded by the rectangular area, say, [-U/2,U/2,-V/2,V/2], and the hologram is bounded by the rectangular area, say, [-X/2,X/2,-Y/2,Y/2], then we must sample the object at distance $(\delta u, \delta v)$, where δu=1/X and δv=1/Y. Thus the number of the samples of the object will be $M \times N$, where M=U/δu=U×X and N=V/δv=V×Y. This number is termed the space-bandwidth product (SBWP) [Lohmann, 1967].

SBWP=M×N. (5.6)

The SBWP of the Fourier spectrum is equal to the SBWP of the object (see Eq.(2.30)), so the complexity of the hologram is directly related to the complexity of the object. Eq. (2.23a) describes the relationship between the object $U(u,v)$ and its spectrum $O(x,y)$ in the discrete Fourier transform. The function $U(u,v)$ is the amplitude distribution of the light at the object and can be thought of as defining the object as we described in section 2.5. It is a complex function which can be expressed as

$$U(u,v) = |U(u,v)| exp\{i\varphi(u,v)\}. \quad (5.7)$$

where $|U(u,v)|$ denotes the amplitude of the function and $\varphi(u,v)$ denotes the phase of the function. Our eye is only sensitive to the intensity. We do not see the phase, but the phase does influence the structure of the computer-generated hologram. The most strongly felt of this influences is the Fourier power spectrum peak-to-average-values ratio "R" (R=max(modulus of the Fourier spectrum)/average(modulus of a Fourier spectrum)). This ratio determines the relation between computer-

generated hologram diffraction efficiency[†] and what information will be lost in plotting [Dallas, 1980].

The dynamic range D_m of a display device is the ratio between the maximum and minimum intensities that can be display by the device. The minimum attainable intensity I_0 for a CRT is anywhere from about 0.005 to 0.025 of the maximum intensity of 1.0. If I_0=0.002, then its corresponding dynamic range will be 50. In order to produce a continuous tone black and white image such that the production appears to continuous, then the appropriate value for n, the number of intensity levels, is found by [Foley et al,1990]

$$n = \log_{1.01}(1 / I_0). \tag{5.8}$$

where $1/I_0$ denotes the dynamic range of the display device. The dynamic range is very important since it determines how a Fourier spectrum will actually 'fit' into the computer-generated hologram. For example, consider the square root of the Fourier spectrum peak-to-average ratio to be significantly greater than the dynamic range of a CRT, or other display devices, i.e.,

$$D_m << \sqrt{R}. \tag{5.9}$$

If the Fourier spectrum modulus is simply scaled so that the maximum modulus is one, then most of the hologram will be quantized to zero. Even if the results are not so severe, the diffraction efficiency will be very low [Dallas, 1980]. Figure 5.2.2 shows the situation when most of the Fourier spectrum moduli are quantized to zero.

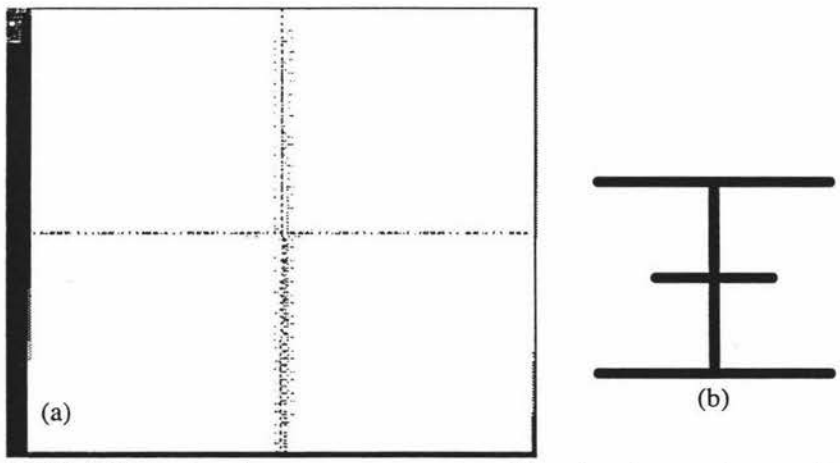


Figure 5.2.2 (a) The interference pattern produced by using detour phase method. The object is Chinese character "king" (figure 5.2.2 (b)) which is represented by a non-zero elements in a 128×128 matrix. The maximum mudulus of the Fourier spectrum of this object is normalized to one, and quantized to 8 levels. Notice that most of the Fourier spectrum moduli are quantized to zero, so image could not be reconstructed from it. We produced this picture on a SUN -3/80 with a black and white screen.

[†] Diffraction efficiency: In a hologram, the percentage of the incident illumination light which is diffracted into forming the image. The greater the diffraction efficiency, the brighter the image will appear in a given light.

5.2.2 Top-clipping [Dallas , 1980]

One way to solve the problem of that the limited dynamic range of the display devices and the large peak-to-average ratio is top-clipping: multiply the Fourier spectrum by a positive constant so that the maximum modulus is greater then one, then set all modulus values greater than one to one. But the question is what constant should we choose? Unfortunately, there is no right answer this question. In practice, we used the average value of the maximum and minimum modulus as the constant to clip the Fourier spectrum, that is K=(1+min(modulus))/2, but the result is not good. Figure 5.2.3 shows a computer-generated hologram produced by detour phase method after clipping

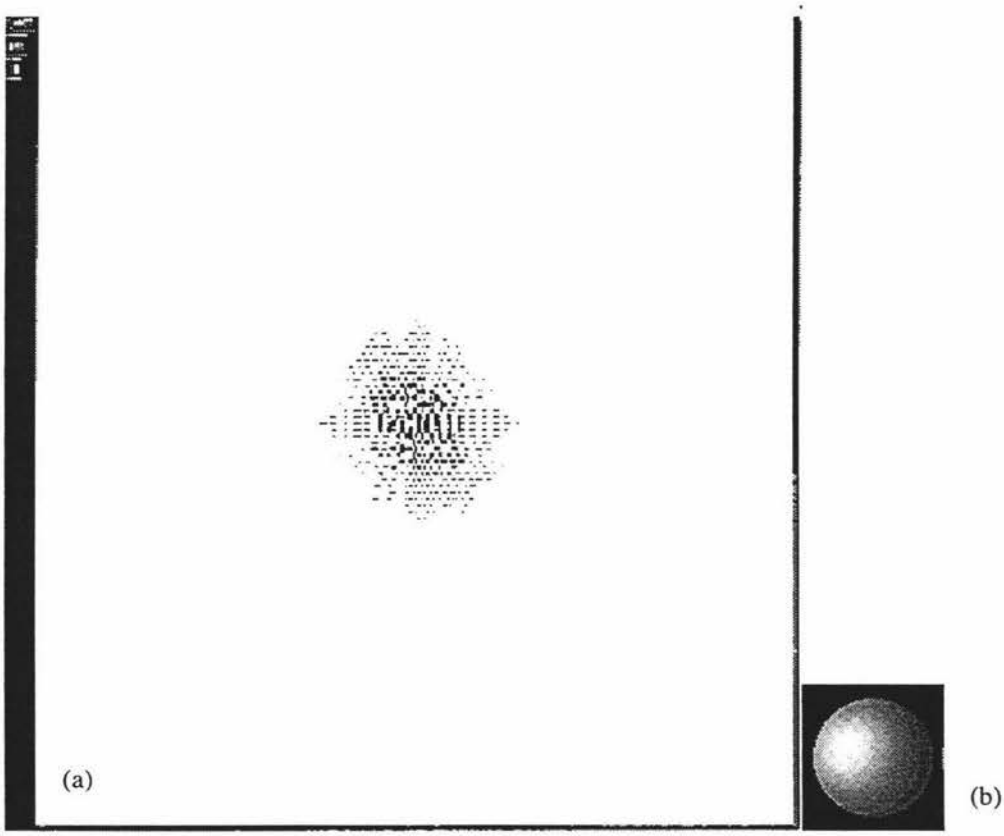


Figure 5.2.3 (a) A detour phase computer-generated hologram after the object Fourier spectrum clipped by its average value of modulus. The object is a sphere (figure 5.2.3 (b)) which presented by a matrix of 128×128 elements. These elements represent the intensity of the object obtained by ray tracing with illumination equation Eq.(4.21). The phase of the object is assumed to be a constant. We generated both of the pictures. (a) was generated on a SUN-3/80 with a black and white screen; (b) was drawn on a PC.

Surely the image could not reconstructed from this hologram. Thus, we repeated the clipping procedure several times, each time we calculated the clip constant again, so

that most of the modulus in the Fourier spectrum can be drawn. Figure 5.2.4 shows such a computer-generated hologram by clipping the Fourier spectrum of the object shown in figure 5.2.3 (b) 5 times and Figure 5.2.5 shows the image reconstructed from it.

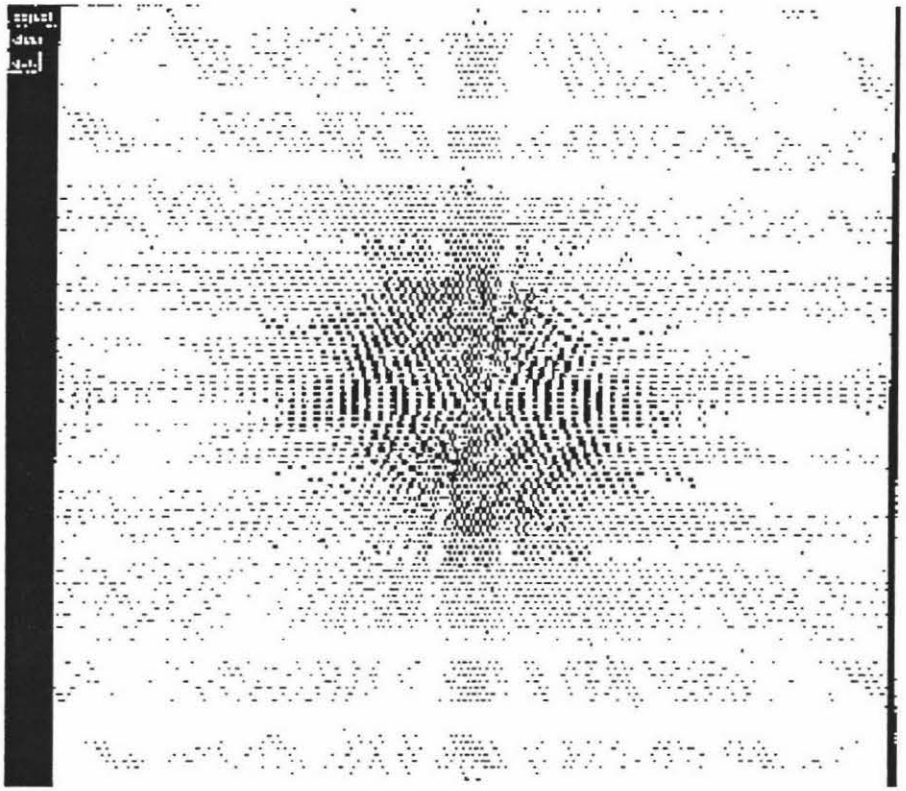


Figure 5.2.4. The computer generated interference pattern by top-clipping the Fourier spectrum of the object shown in figure 5.2.3 (b). The pattern is drawn by detour phase hologram method. We produced this picture on a SUN -3/80 with a black and white screen and printed it out on a piece of paper for photo-reducing.

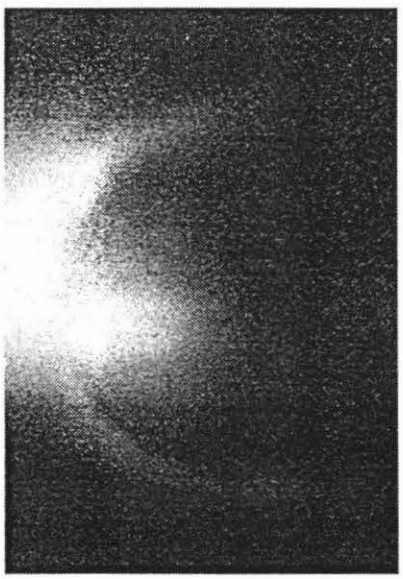


Figure 5.2.5. The image reconstructed from the hologram made by photo-reducing figure 5.2.4 to 10×10 mm^2.

Top-clipping, decreases $\sqrt{R}$ but affects the image. Edge enlargement is the usual result as we see the image reconstructed from the hologram made by photo-reducing figure 5.2.4 to $10\times10\,mm^2$. A desired image with slowly varying modulus and constant phase will have a Fourier spectrum which sharply peaked at u=0,v=0. Top-clipping diminishes the low spatial frequencies relative to the high.

5.2.3 The Diffuser

Another way to solve the problem of that the limited dynamic range of the display devices and the large peak-to-average ratio is to provide the desired image with a phase function, or diffuser, which will bring $\sqrt{R}$ and D_m closer together.

5.2.3.1 *Random phase diffuser*

In optical holography, one may place a diffuse between the beam splitter and the object as shown in figure 5.2.6 (The diffuser physically is frequently a ground glass plate). If a diffuser is present, it yields an important result. The diffuser "spreads" out the illuminating wavefront and randomises the phases of the light incident upon the various part of the object [Lesem, Hirsch and Jordan, 1968]. In a computer-generated hologram, a diffuser can be used to smooth the Fourier spectrum of the object. Furthermore, the object information is distributed over the hologram. Blocking off a part of the hologram will introduce noise (reduction of resolution) but no macroscopic signal damage [Wyrowski, Hauck, and Brynhdahl, 1986].

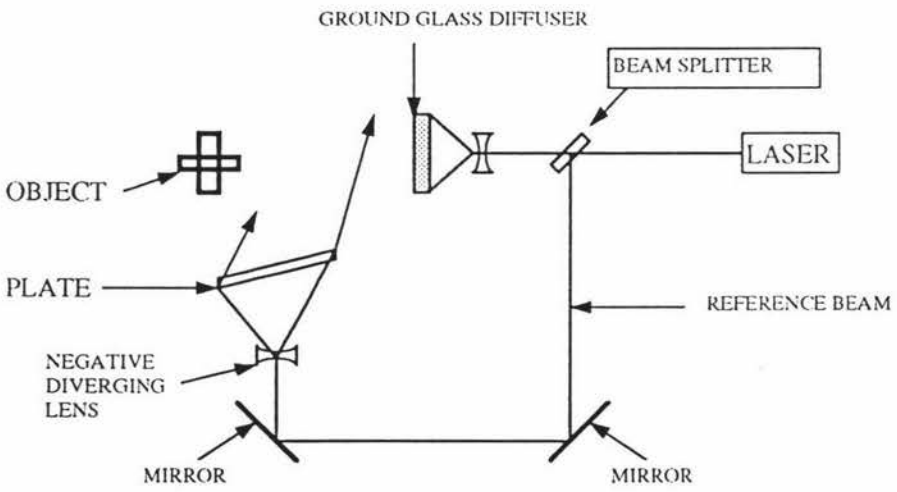


Figure 5.2.6 Optical configuration for making reflection holograms [Koechner, 1979].

The simplest way to pick a diffuser phase function for the object is to use a random number generator to generate pseudo-random phases distributed uniformly between 0 and 2π radian. These phases may be quantized, i.e., restricted to take on only a finite

number of the values. It has been shown [Burchhardt, 1970] that as few as three randomly chosen phases, equally spaced about the unit circle, are sufficient to closely approximate behaviour of a continuous-phase random diffuser (refer to section 3.3, figure 3.3.1 (b)).

A computer-generated hologram with random diffuser is shown in figure 5.2.7. The object we defined is the Chinese character "king" which is represented by a non-zero elements in a 128×128 matrix. Before computing the discrete Fourier transform of the matrix, each of the elements in the matrix is multiplied by a random phase factor to help reduce the dynamic range in the transform. After the transform has been computed, the maximum modulus of the Fourier spectrum of the matrix is normalised to one, and quantized to 8 levels. Then the detour phase coding technique is applied to all the elements in the transform to produce the binary hologram. The image reconstructed from the hologram is shown in figure 5.2.8

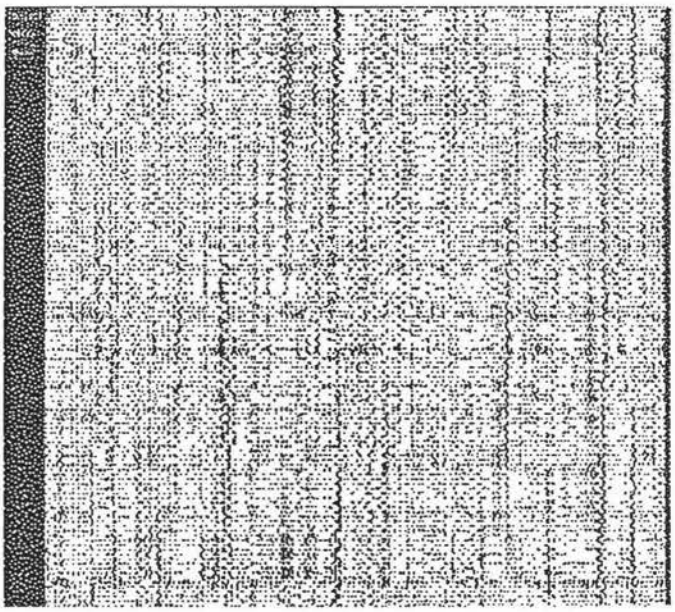


Figure 5.2.7. An example of detour phase hologram with random phase diffuser. The object is Chinese character "king" (figure 5.2.2 (b)) which is represented by a non-zero elements in a 128×128 matrix. This the original pattern produced by computer. The final hologram is $10\times10\,mm^2$. We produced this picture on a SUN -3/80 with a black and white screen and printed out on a piece of paper for photo-reducing.

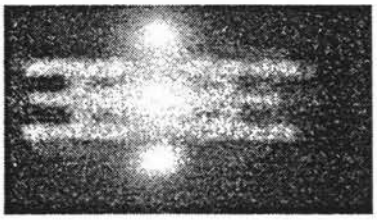


Figure 5.2.8. The image reconstructed from the hologram produced by photo-reducing figure 5.2.7 to $10\times10\,mm^2$.

5.2.3.2 *Deterministic diffuser*

Deterministic diffusers can be divided into the general-purpose diffusers and image-specific diffusers. General-purpose diffusers are designed to work reasonably well with a large class of objects; the diffuser functions are independent of the desired image moduli. Image-specific diffusers are designed for one image; the generating algorithm takes image modulus values into account.

There are some special phases $\varphi(u,v)$ that will lead to the discrete Fourier pair

$$U_{mn} = K\exp[i\varphi_{mn}] \xleftarrow{DFT} F(U_{mn}) = K\exp[i\phi_{mn}], \qquad (5.10)$$

where K denotes a constant, $F(U_{mn})$ denotes the Fourier transform of the discrete function U_{mn}, and *DFT* is the discrete Fourier transform. Eq.(5.13) states that the phase $\varphi(u,v)$ generates a discrete spectrum of constant amplitude after superposition onto a constant amplitude. Such a phase will completely eliminate the DC peak in the spectrum of an arbitrary distribution $|U_{mn}|$. Dallas [Dallas ,1973] introduced a whole class of them. In our experiments, we used the Schroeder code [Chu, and Goodman ,1972] as the phase function. The Schroeder code, for an object with uniform modulus, has an absolutely uniform discrete transform. It can be described as

$$\phi_{mn} = \pi\left[\frac{m^2}{M} + \frac{n^2}{N}\right], \qquad (5.11a)$$

where $-M/2 \le m \le (M/2)-1, \quad -N/2 \le n \le (N/2)-1.$ (5.11b)

This phase function does not bring the peak-to-average ratio $\sqrt{R}$ of the Fourier spectrum closer to the dynamic range D_m, but it does smooth the modulus of the Fourier spectrum. Figure 5.2.9 illustrates that the modulus distributions of the Fourier spectrum with different phases.

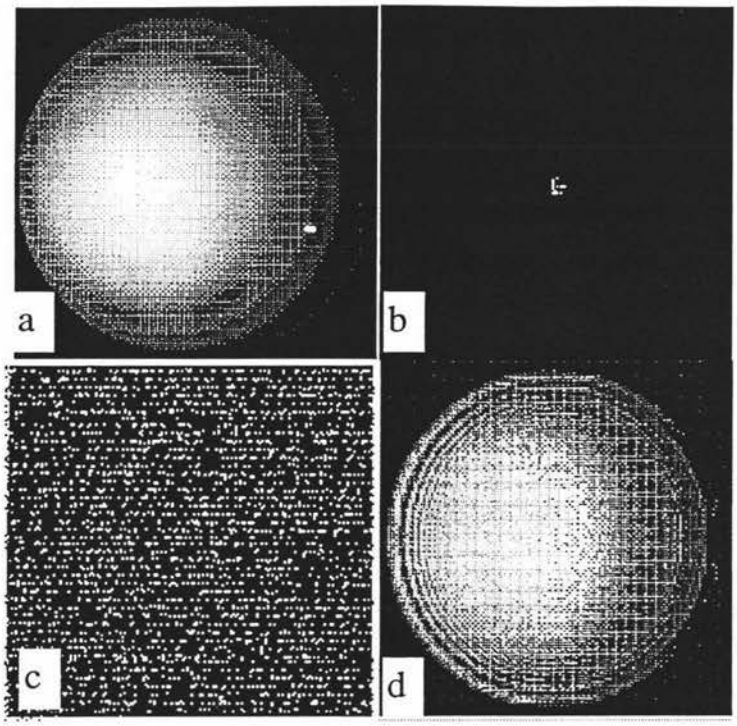


Figure 5.2.9. (a) The intensity distributions of the original object (a sphere). This picture is produced by ray tracing. The illuminate equation is Eq.(4.21); (b) Modulus of the Fourier spectrum of the original object introduced by a constant phase factor; (c) Modulus of the Fourier spectrum of the original object introduced by a random phase factor; (d) Modulus of the Fourier spectrum of the original object introduced by a Schroeder code phase factor. We produced these pictures on a SUN-3/80 with a black and white screen.

5.3 Calculate the wavefront at the hologram plane and draw the interference pattern

Once we have a suitable phase function, and enough intensity samples of the object, we can compute the distributions of the wavefront at the hologram plane by Eq. (2.19a). Before performing the discrete Fourier transformation, the intensity samples represented by a two-dimensional array can be multiplied by the phase factor element by element, then the discrete Fourier transformation can be performed.

To draw the interference pattern, we can use detour phase method. We have already shown an example of this type of computer-generated hologram and image reconstructed from it in figure 5.2.7 and 5.2.8. The chief disadvantage of this method is they are very wasteful of plotter resolution since the number of the addressable plotter points in the cell must be very large to minimise the noise due to quantization of the modulus and the phase of Fourier coefficients. Huang [Huang, 1971] pointed out that when the modulus and the phase of the Fourier coefficients are quantized into more than 64 levels, the quality of the reconstructed image is far better than the quality of the reconstructed image at less quantization levels. This suggests that 64

quantization levels is the minimum needed for better image reconstruction. But in other side, the more quantization levels, the more space we need. Assume that we are going to draw the interference pattern on a CRT which has the resolution of 1024×1024 elements, if we use one pixel to represent one quantization level, and the amplitude and the phase of the Fourier coefficients is quantized into 64 levels, then we need 64×64 pixels to represent one Fourier coefficient, thus only 16×16 Fourier coefficients can be represented by the CRT, that is to say, the object must be bounded in a 4×4 (mm^2) square, and the final hologram size should be 4×4 (mm^2). For our display purpose, this size of the object is too small to be observed. The Lee's sampling method also suffers from the same problem.

5.4 Burch's method

As we motioned in Chapter 2, in the off-axis reference beam transmission hologram the amplitude transmittance of a transmission hologram recorded under ideal condition is proportional to

$$\begin{aligned} T(x,y) &= \left| Re^{i2\pi ax} + O(x,y)e^{i\varphi(x,y)} \right|^2 \\ &= R^2 + O^2(x,y) + R|O(x,y)|e^{i[2\pi\alpha x-\varphi(x,y)]} + RO(x,y)e^{i[-2\pi\alpha x+\varphi(x,y)]} \\ &= R^2 + O^2(x,y) + 2R|O(x,y)|\cos[2\pi\alpha x-\varphi(x,y)], \end{aligned} \tag{5.12}$$

where $Re^{i2\pi\alpha x}$ represents the reference wave, $O(x,y)e^{i\varphi(x,y)}$ represents the object wave, and $\alpha=(\sin\theta)/\lambda$, θ is the angle between the direction of the plane reference wave propagation and the normal of the hologram plane, and the α is called the carrier frequency. The terms R^2 and $O^2(x,y)$ are often referred as bias term. $T(x,y)$ is the resulting intensity variation of the interference pattern between the two waves. A Computer-generated hologram could be made from the function $T(x,y)$. Because it is already real and positive, the computer could simply calculate its values at discrete intervals and produce a graphical output of the function on a CRT display. The sampling rate for the function $T(x,y)$ is dependent upon the total bandwidth of the function.

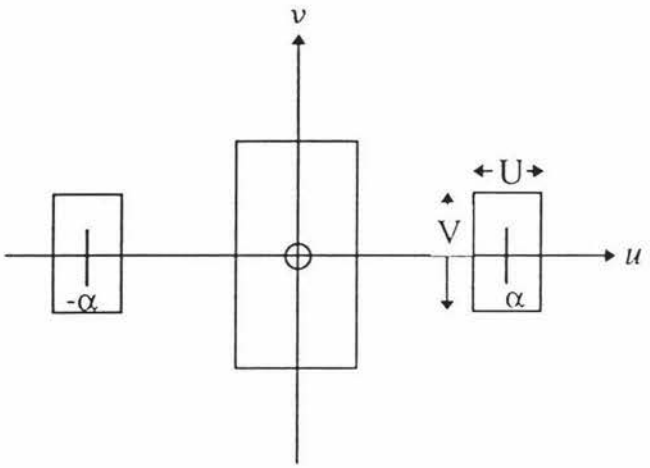


Figure 5.2.10. Spatial distribution in an off-axis reference beam hologram. The two rectangular regions centring at $\pm\alpha$ are the spatial frequencies from the two conjugate waves reconstructed from the hologram. [Lee, 1978].

Lee [Lee, 1978] explained that if a transparency with transmittance $T(x,y)$ is put in the optical system in figure 3.2.2, the Fourier transform of the transparency will have values in the region shown in figure 5.2.10. The two rectangular regions centring at $u=\pm\alpha$ contain the Fourier transform of the complex function $O(x,y)e^{i\varphi(x,y)}$ and its complex conjugate. Each occupies an area U×V in the u,v plane, where U and V are, respectively, the spatial bandwidths of the function $O(x,y)e^{i\varphi(x,y)}$ along the two coordinates. The circle in the centre in figure 5.2.10 is an impulse function resulting from the constant R^2 in Eq.(5.12). The rectangle in the centre represents the spatial frequencies from the term $O^2(x,y)$ in Eq.(5.12). To avoid overlapping of these terms in the hologram plane, the carrier frequency, α, must be equal to at least 1.5U. The highest frequencies of the function $T(x,y)$ along the two coordinates are, 2U and V, respectively. Therefore, the sampling intervals for the function $T(x,y)$ must be smaller than

$$\delta x = \frac{1}{4\mathrm{U}} \quad \text{and} \quad \delta y = \frac{1}{2\mathrm{V}}. \tag{5.13}$$

Burch [Burch,1967] noticed that the term $O^2(x,y)$ is not important to the reconstruction of the desired wavefront $O(x,y)e^{i\varphi(x,y)}$. Its presence increases the bandwidth requirement of the hologram along the U coordinates. Thus, Burch suggested that in making computer-generated holograms, the amplitude transmittance of the hologram should be

$$T(x,y) = 0.5\left\{1 + |O(x,y)|\cos\left[2\pi\alpha x - \varphi(x,y)\right]\right\}. \tag{5.14}$$

Because of the absence of the term $O^2(x, y)$ in Eq. (5.15), the carrier frequency α can be as low as U/2 and the sampling intervals become

$$\delta x = \frac{1}{2U} \quad \text{and} \quad \delta y = \frac{1}{V} . \tag{5.15}$$

Surely, this sampling interval is less than the sampling interval we discussed in the detour phase hologram (see Eq.(2.20)), that is to say, the SBWP of the hologram and the size of the DFT are increased. Thus if we use the narrow bar method (detour phase method) to draw the Burch's hologram, we have to reduce to half of the object size in the U direction compared with the detour phase method. But because the function $T(x,y)$ is already real and positive thus we can use grey-scale display to draw the function $T(x,y)$. Meyer and Hickling [Meyer and Hickling , 1967] exhibit such method. The hologram constructed by them exhibits 12 grey levels, obtained by "hitting" each CRT raster point as many as 12 times, depending on the scaled values of the computed hologram. The introduction of grey scale in general saves a lot of space in the display device compared with the detour phase method. In principle, by using Burch method, a CRT with resolution 1024×1024 and grey scale can be used to produce a size of 22×22(mm^2) hologram. But when photo-reducing the interference pattern displayed on the CRT, care must be taken, because the exposure of the film is used to control the modulus of the hologram transmittance. A knowledge of CRT-film photo processing is necessary [Dallas, 1980].

Two of computer-generated holograms we made according to Eq.(5.14) are shown in figure 5.2.11(a) and (b). The images obtained from the holograms are shown in figure 5.2.12(a) and (b). The holograms are Fourier transform holograms with 512×512 samples. In Figure 5.2.11(a) the original data samples have been multiplied by a pseudo-random phase factor. This is done by multiplying , element by element, the original distribution of the specified field amplitude on the object by the equiprobable values ±1. The object is sampled by ray tracing. The illumination equation we used is Eq.(4.21). Because the phase function has only two values, only the contour of the object is reconstructed as figure 5.2.12(a) shows. In figure 5.2.11(b), all the samples are the same with the samples used in the production of figure 5.2.11(a), but the phase function is Schroeder code. From these figures, we can see that the quality of the images in figure 5.2.12(b) is better than the quality of the image in figure 5.2.12(a). In normal conditions, no plane reference beam can be obtained. So Figure 5.2.11 (b) is synthesized with a point which scatters a spherical wave as the reference beam. Figure 5.2.11(a) is synthesized with a plane wave as the reference beam. These holograms are drawn by the intensity interpolating method we

described in section 4.10, because our CRT is black and white. The final computer-generated holograms are $10\times10(\text{mm}^2)$.

Another modification on Eq. (5.12) for making off-axis holograms was suggested by Huang and Prasada [Huang and Prasada, 1966]. They proposed the use of the following amplitude transmittance for the hologram

$$T(x, y) = 0.5|O(x, y)|\{1 + cos[2\pi\alpha x - \varphi(x, y)]\}. \qquad (5.16)$$

This method does not have the bandwidth reduction advantage of Burch's hologram because the bias term in Eq.(5.16) is not a constant. Also, since the average transmittance of this hologram is proportional to $|O(x,y)|$, this hologram is very sensitive to the non-linearity of the recording medium. The advantage of this method is that the contrast in the hologram is high and independent of the fluctuation in $|O(x,y)|$.

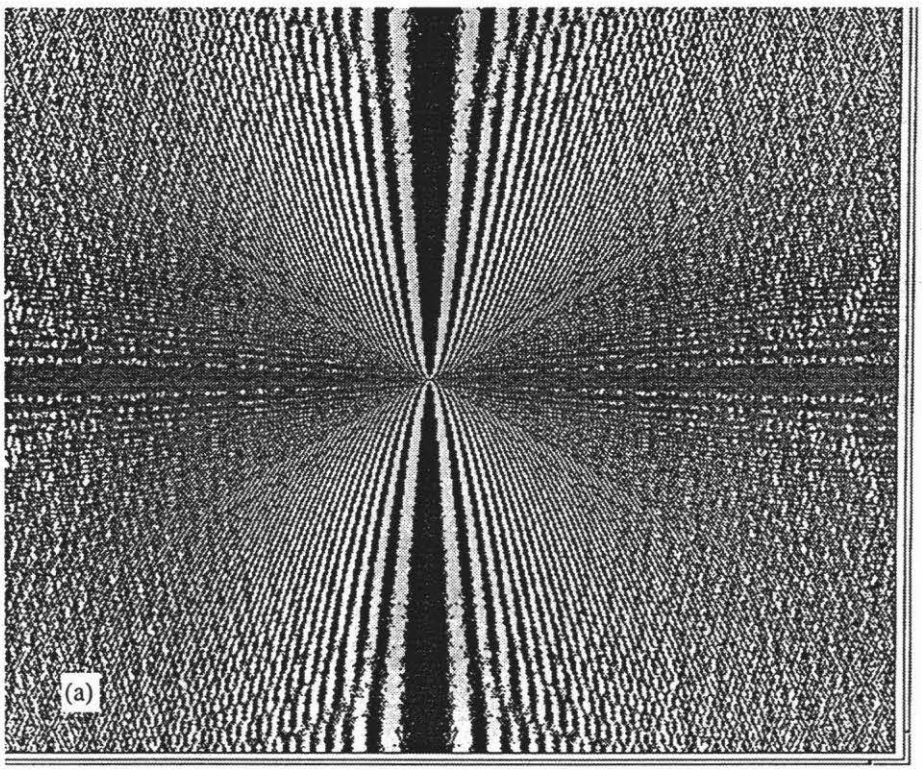


Figure 5.11. The interference patterns produced by Burch method.

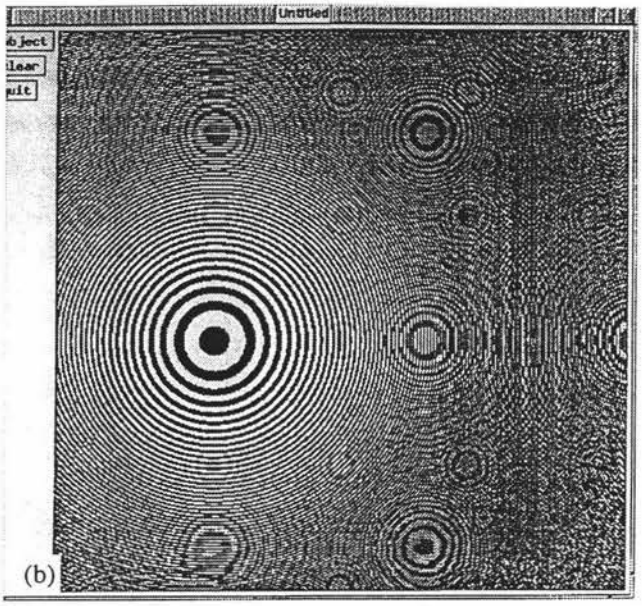
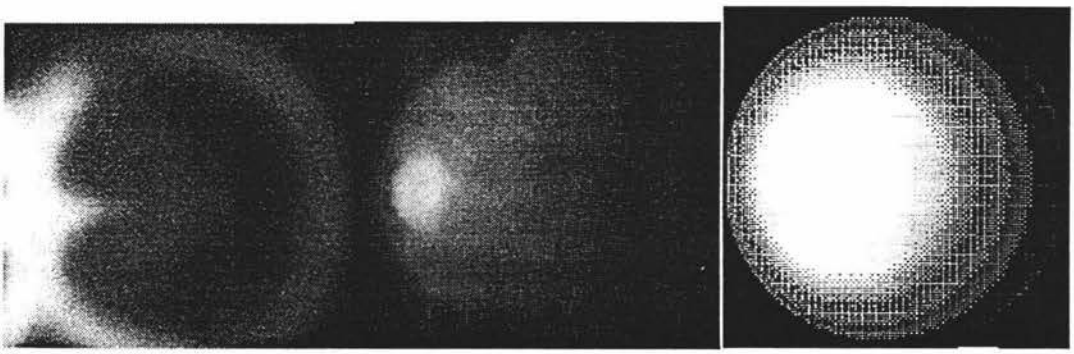


Figure 5.11 continue.



(a) (b) (c)

Figure 5.2.12. (a) and (b) are the images reconstructed from the holograms produced by figure 5.2.11(a) and (b) respectively, (c) is the original object. We made figure 5.2.11(a), (b) and (c) on a SUN-3/80 with a black and white screen and printed it on a piece paper for photo-reducing.

5.5 Experiment with Synthesizing Fresnel Holograms

Eq. (2.17) is the starting point for synthesizing a Fresnel hologram and is the general description of the relationship between an object and wavefront distribution by the object at the hologram plane. For convenient, this relationship can be expressed as Fourier transform

$$F_{RT}(x,y)=exp\left[i\frac{k}{d}\left(x^2+y^2\right)\right]FT\left\{U(u,v)exp\left[i\frac{k}{d}\left(u^2+v^2\right)\right]\right\}, \quad (5.17)$$

where $F_{RT}(x,y)$ denotes the Fresnel transform , FT denotes the Fourier transform. and $k=\frac{\pi}{\lambda}$ with λ the wave length. Eq.(5.17) is equivalent to the following steps:

superposition of a quadratic phase onto the $U(u,v)$, calculate the Fourier transform, and once more, superposition of another quadratic phase onto the calculated spectrum. However, it is necessary to sample the object in order to computer the *FT* by a computer; that is, to perform the discrete Fourier transform (DFT). Eq. (2.44) describes the discrete relationship between the sampled object and the sampled wavefront distribution at the object at the hologram plane, thus we should be able to synthesizing the Fresnel hologram according to Eq. (2.44). But Bryngdahl and Wyrowski [1990] pointed out that in general, Eq. (2.44) could not be calculated correctly. Aliasing errors will appear in the hologram plane. The smaller d is, the worse the errors will be, because Eq.(2.44) is not band-limited[†].

Bryngdahl and Wyrowski also pointed out that for aliasing errors can be avoided in the hologram plane, if we reduce the calculation of Eq. (5.17) to

$$F_{RT}(x,y) = exp\left[i\frac{k}{d}\left(x^2+y^2\right)\right] FT\left\{\sqrt{I(u,v)}exp\{i\varphi(u,v)\}\right\}. \qquad (5.18)$$

where $\sqrt{I(u,v)}$ denotes the intensity of the object. Specifically this means that the Fourier transform of $U(u,v) = \sqrt{I(u,v)}exp\{i\varphi(u,v)\}$ should be computed according to Eq. (2.23a) and subjected to the conditions in Eq.(2.19) and Eq.(2.21), and then a quadratic phase is superposed onto the resulting spectrum. The result is the complex amplitude $F_{RT}(x,y)$ in the hologram plane. Therefore, Eq.(2.44) must be modified to

$$O_{jk} = \exp\left[i\pi\left(j/J\right)^2 + (k/K)^2\right)\Big] \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} U_{mn} \exp[-2i\pi(jm/M + kn/N)], \qquad (5.19)$$

where $J = \mathrm{UX}/(\lambda d)^2$, $K = \mathrm{VY}/(\lambda d)^2$ and $M = \mathrm{UX}$, $N = \mathrm{VY}$ and

$-M/2 \le j \le (M/2)-1,$

$-N/2 \le k \le (N/2)-1.$

This is the desired equation for synthesizing Fresnel holograms.

An example of the computer-generated Fresnel hologram is shown in figure 5.2.13(a) and image reconstructed from the hologram is shown in figure 5.2.13(c). The hologram has 512×512 samples. It is drawn on a CRT by Burth's method and then

---

[†] **Band-limited** function: if a periodic function f has the property that the Fourier coefficients c_k are zero for $|k| > M$, where M is some natural number, then f is said to be **Band-limited.**

printed on a piece of paper by a laser printer. The final size of the hologram is 10×10 mm^2.

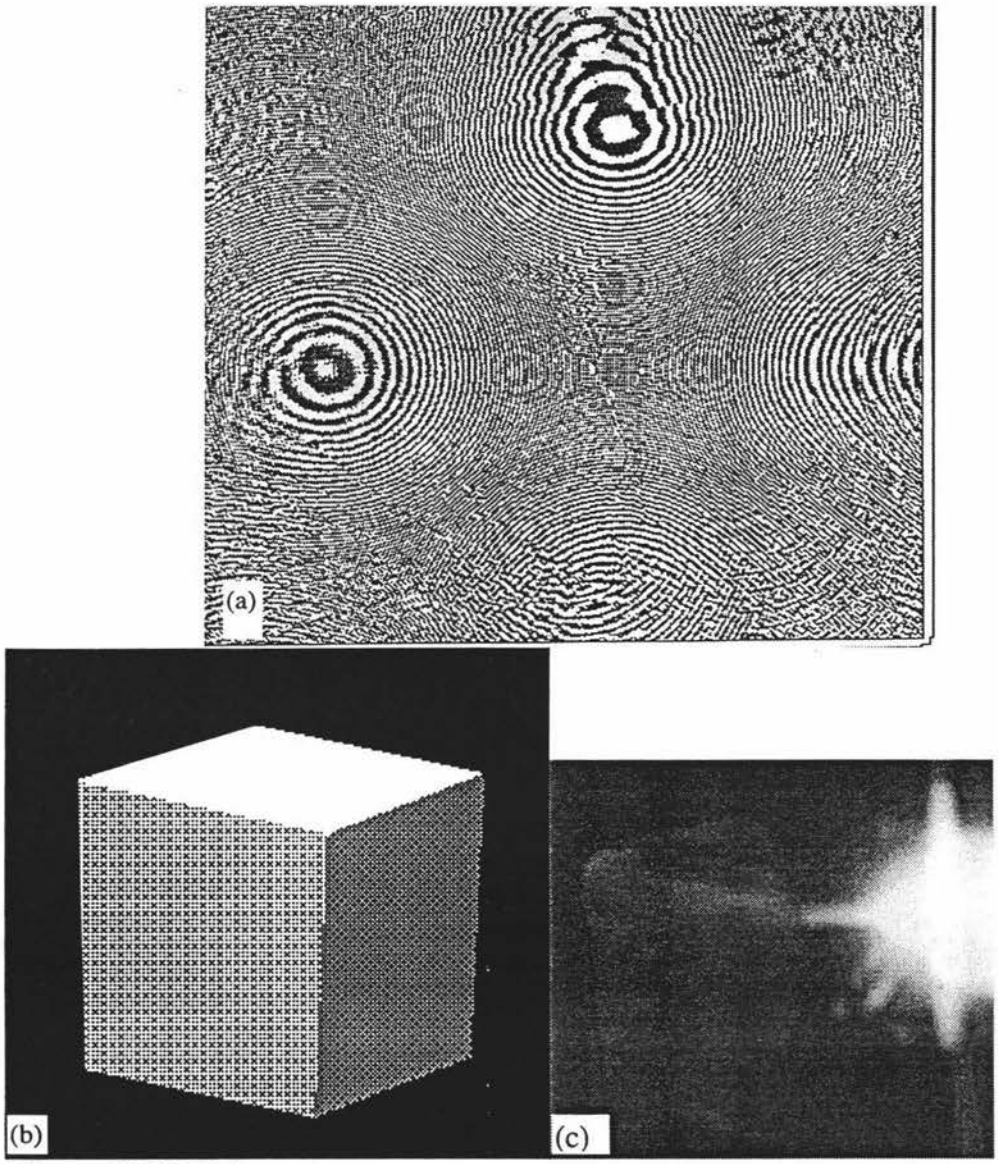


Figure 5.2.13. (a) The interference pattern generated by the computer; (b) The original object; (c) The image reconstructed from the hologram. We produced (a) and (b) on a SUN-3/80 with a black and white screen and printrd it on a piece of paper for photo-reducing.

5.6 Experiments with 3-D Objects

In above sections, we have shown some computer-generated holograms and the images reconstructed from them. Some of the objects we defined are three-dimensional and the objects are sampled by ray tracing, that is to say, the sampled points which are parts of the object are really in three-dimensional, but we did not observe the parallax effects in the images reconstructed from the Fourier or Fresnel holograms. The reason is that there is no depth information involved in the Fourier transform, thus we have to transform a three dimensional problem to a two

dimensional problem. As we mentioned before, One method that can be used to overcome this problem is to decompose the three-dimensional intensity distribution into J planes in depth, as shown in figure 3.6.2. For the simplified intensity object we have

$$I(u,v,z)=\sum_{j=1}^{J} I_j\left(u,v,z_j\right) \tag{5.20}$$

where $I_j\left(u,v,z_j\right)$ denotes intensity distributions of the points which are parts of the object in the jth plane. Because each of $I_j\left(u,v,z_j\right)$ is two dimensional, therefore can be treated separately.

The jth plane is located in z_j and the three dimensional problem is decomposed into J two dimensional situations with the intensity $I_j\left(u,v,z_j\right)$ in $z=z_j$. Because this decomposition involves the three-dimensional position of the object as parameters, a Fourier hologram will not suffice, but one is necessary which by nature comprises many planes in depth; that is, the Fresnel hologram for three-dimensions. The method of in section 5.5 could be applied (see Eq. (5.19)).

The Fresnel transformation for each of the two dimensional slices with the intensity $I_j(u,v,z_j)$ in $z=z_j$ can be expressed as

$$F_{RTj}(x,y,z_i)=exp\left[i\frac{\pi}{\lambda z_j}\left(x^2+y^2\right)\right]FT\left\{\sqrt{I_j(u,v,z_j)}exp\left\{i\varphi_j(u,v,z_j)\right\}\right\}. \tag{5.21}$$

The final complex amplitude in the hologram plane is

$$F_{RT}(x,y,z)=\sum_{j=1}^{j} F_{RTj}(x,y,z_j). \tag{5.22}$$

The objects we chose to synthesize the holograms according to this method are a sphere as shown in figure 5.2.14(a) and a cube as shown in figure 5.2.14(a). The objects are sampled by ray tracing with 512×512 sample points and the illumination equation Eq.(4.21) is used to obtain the intensity of each points of the objects. Each of the samples contains two parameters, the intensity of the sampled points and the spatial position of the object. After the objects have been sampled, the samples are decomposed into 10 planes according to their positions in space. In order to perform the Fourier transform, we represent each of the planes by a array with 512×512

sample values, but many of them are zero. Each of the arrays first is multiplied by the phase factor, that is, the Schroeder code, element by element, and then the Fourier transform is performed. After that another quadratic phase is superposed onto the resulting spectrum. After the amplitude of the holograms were calculated according to Eq.(5.22), we added a spherical wave as reference wave, and used the Burch's method to produced the final holograms. The interference patterns obtained from these objects are shown in figure 5.2.14(b) and figure 5.2.15(b). The holograms produced by photo-reducing figure 5.2.14(b) and figure 5.2.15(b) are illuminated by a white light point source. Figure 5.2.14(c), (d), (e), (f) and (g) shows images reconstructed from the hologram of the sphere. Figure 5.2.15 (c), (d), (e) and (f) shows the images reconstructed from the hologram of the surface of cube.

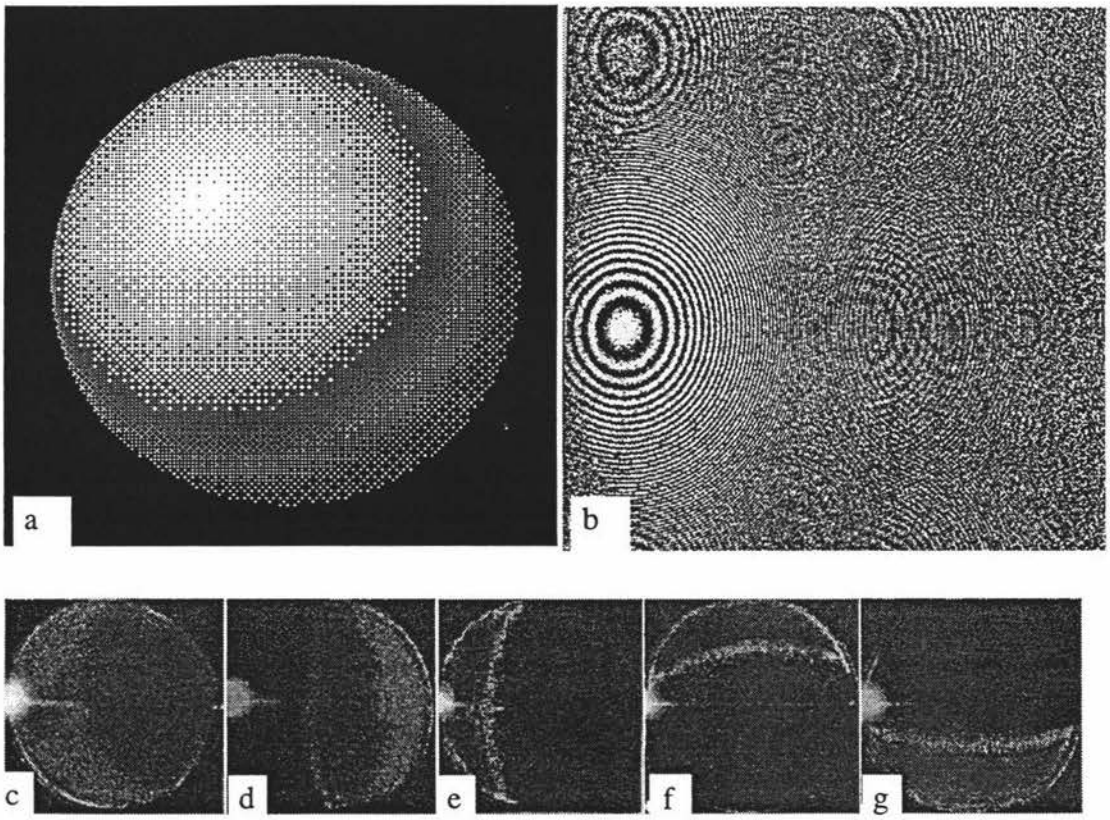


Figure 5.2.14. (a) The original object; (b) The interference pattern produced by the computer; (c),(d),(e),(f) and (g) are the reconstructed images from the hologram produced by photo-reducing (a) to $10\times10\,\mathrm{mm}^2$. The reconstructed images are obtained by illuminating the hologram with the same light source but different viewpoints. We produced (a) and (b) on a SUN-3/80 with a black and white screen.

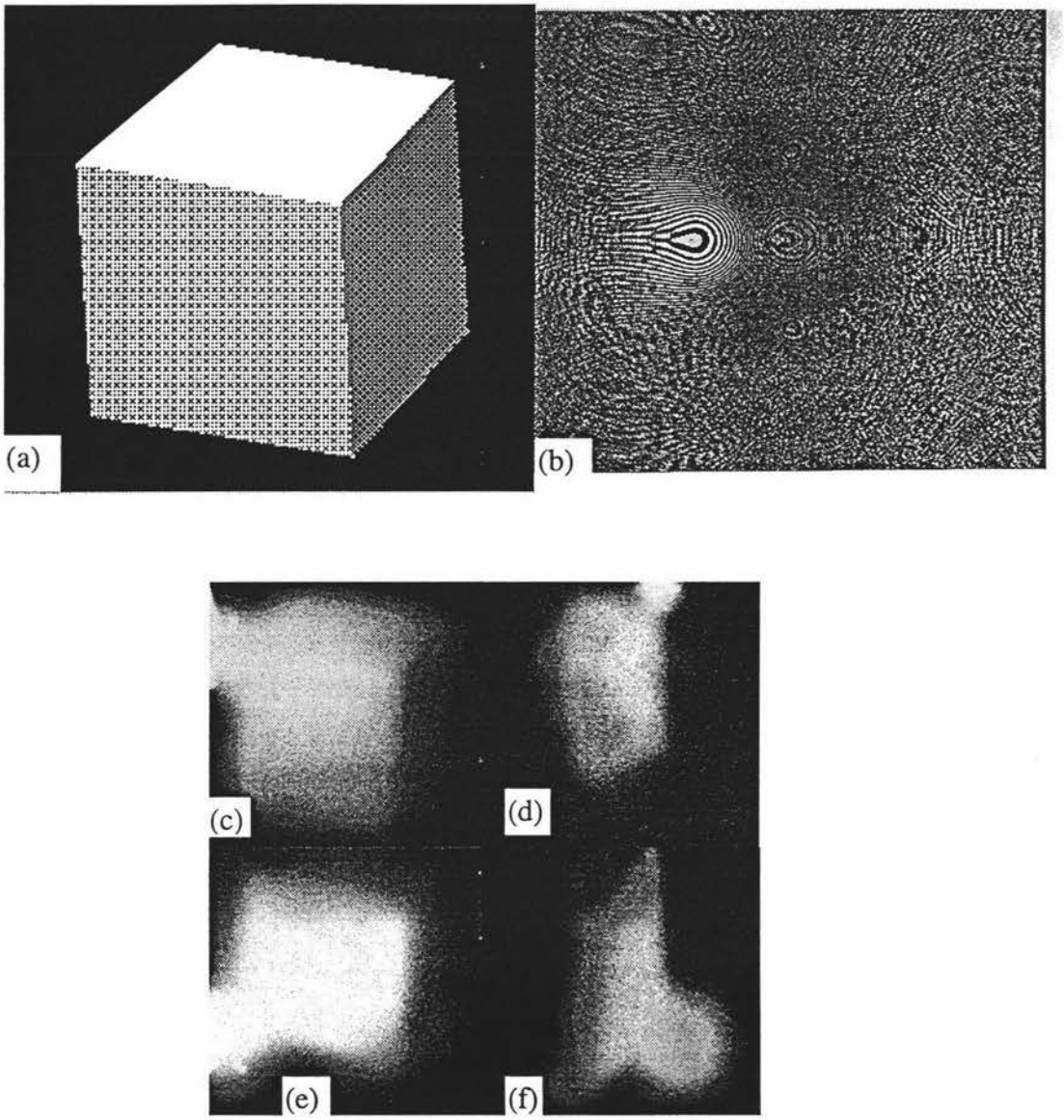


Figure 5.2.15. (a) The original object. (b) The interference pattern generated by the computer. (c), (d), (e) and (f) are reconstructed images from the hologram. They are obtained by illuminating the hologram with a point light source but the illuminating positions are different. We produced (a) and (b) on a SUN-3/80 with a black and white screen.

5.7 Synthesis of a stereo hologram

The synthesized Fourier holograms and be reproduce for observation a projection of an object on a plane perpendicular to the observation direction. These holograms can convey spatial relation only to the extent that these relations affect the projections, that is, through perspective distortions and through the masking of the some parts of the various planes by others. Fourier holograms of objects cannot convey the depth of various planes of the object. Anyhow , it is possible to synthesize holograms with different cross-sectional views of the object in different planes by Fresnel holograms, although the parallax effect is quite weak because of the small size of the hologram.

But it is impossible to convey the three-dimensional nature of the object in a convenient and natural way for visual observation. Because of our binocular vision, an important factor in the perception of three dimensions during observation under natural conditions is the stereo effect: the subjective perception that an object is three-dimensional if it is observed with the two eyes from different perspectives or with parallax. This effect can be reproduced with digital holograms if separate holograms are synthesized for different perspectives of the object and then observed simultaneously by the two eyes. Such holograms are referred to as synthesized stereo holograms.

Using computer to produce synthesized stereo holograms have reported by King, Noll and Berry [King, Noll and Berry, 1970], and Yatagai [Yatagai, 1974,1976] as described in chapter 2. The most advantage of King, and *et al's* method is that the quality of image reconstructed from their hologram is quite good, as figure 5.15 shows. Another advantage of their method is the computation time of their hologram is less because only different perspectives of the object are calculated. The main disadvantage of their method is that optical processing is needed. In fact their method is like producing an optical hologram. Thus a very high resolution recording medium is required. Another disadvantage of their method is that in order to produce a stereo hologram with 360° degree observation angle, a lot of film is required because many perspective projections of the object must be recorded on the film first.

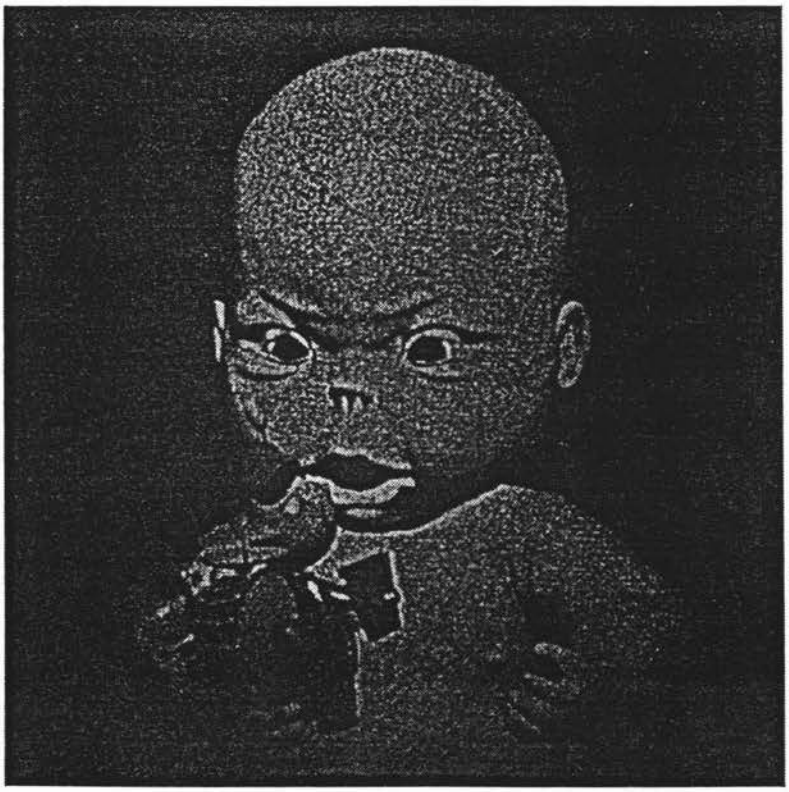


Figure 5.2.16. The image reconstructed from a King *et al's* hologram [Haines and Haines, 1992]. This kind of hologram is not a pure computer-generated hologram (refer to chapter 6 and section 2.3, figure 2.3.1).

Yatagai adopted this method to make computer-generated holograms to display 3-D images. Instead of recording the perspective projection images on the film, he converts the projection images into Fourier transform holograms which are then plotted on a large scale plotter by the detour phase hologram method. These holograms are then arranged in the same order as the projection images are obtained. The greatest advantage of his method is the optical processing procedure is not needed, and the computer generated hologram is photo-reduced only once. But the computation time for his hologram is greater than King, and *et al's* hologram. In order to observe the stereoscopic effect with computer generated holograms, we followed Yatagai's method to produce a stereo hologram. But instead of using the detour phase coding technique, Burch's method was used. The test object was generated by the computer; its geometric dimensions and coordinates on the image could be changed through programming. Figure 5.2.17(a) shows the object. The object is represented by CSG, and calculated by ray tracing. For each of the perspective projections of the object, we calculated its Fourier hologram, and added the same spherical wave as the reference wave. After all of the computer-generated holograms for each of the perspective projections of the object have been calculated, we arranged them in the order of calculation and photo-reduced to form the final stereo hologram. Figure 5.2.17(b) shows the interference pattern generated by the computer. When the hologram is illuminated by a white light point source, it does construct different images if we see it at the different viewpoints as shown in figure 5.2.17 (c), (d) and (e), but we did not see the parallax effects.

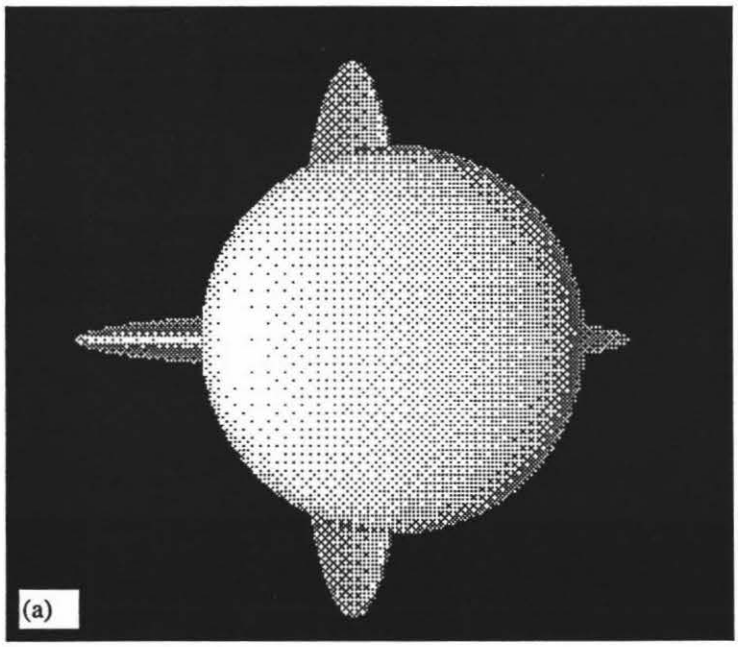
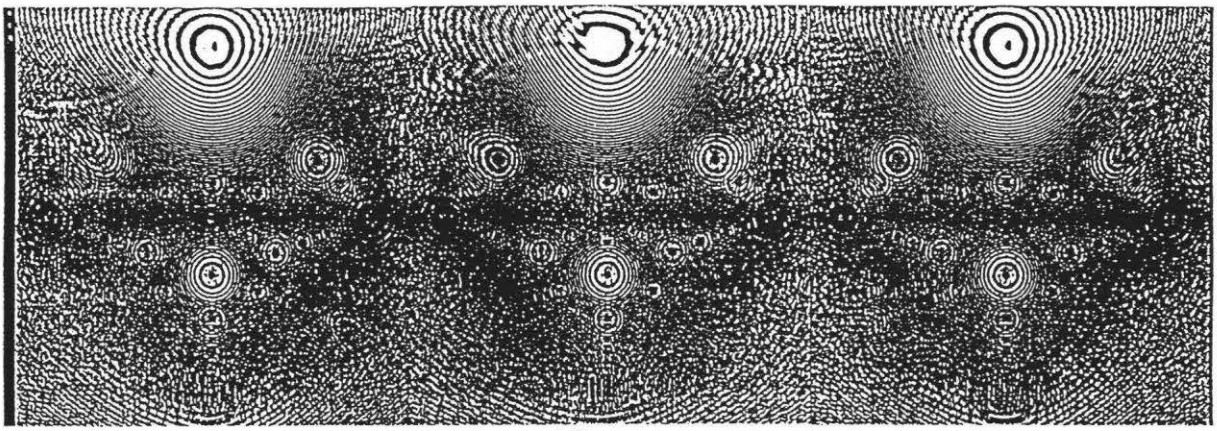


Figure 5.2.17 (a) The original object.



(b)

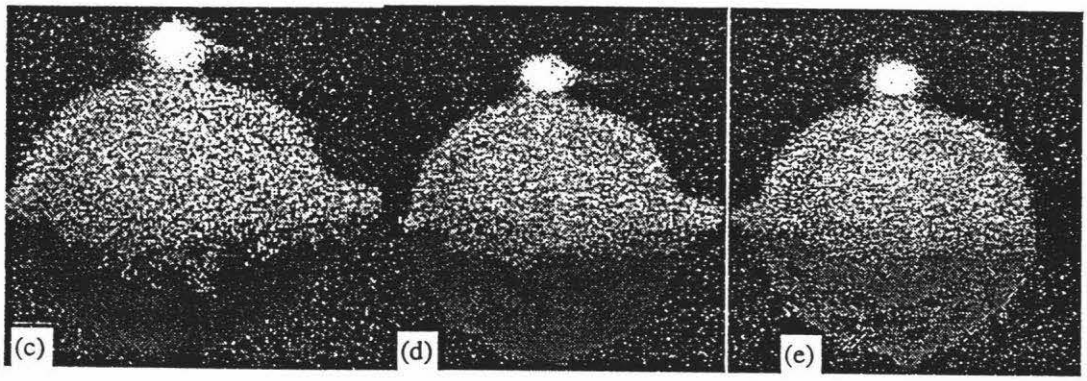


Figure 5.2.17 (continue). (b) The stereo interference pattern produced by the computer. (c), (d) and (e) are the reconstructed images from different viewpoints. We produced (a) and (b) on a SUN-3/80 with a black and white screen.

Chapter 6

Conclusions And Remarks

In the above chapters, we have discussed the procedures to make a computer-generated hologram, the mathematical descriptions of the computer-generated holograms, the different techniques for producing computer-generated holograms and how the limitations of the hardware devices affect the quality and the dimensions of the images reconstructed from a computer-generated hologram. From the experiments, we can see that a three dimensional image of an object which may not physically exist but just in mathematical form, can be reconstructed from a computer-generated hologram. It is possible to reconstruct images from a computer-generated hologram by white light source in normal environment conditions provided with a point light source. The reconstructed images from a Waters's hologram show us all properties of the true three dimensional imaging with perspective effects. We can catch sight of an object behind another by a mere tilt of the head. This is the potential advantage of computer-generated holograms for three dimensional display purposes.

Computer-generated holograms now have been used in the areas of ***diagnostic and testing, digital and optical interconnects, high-energy physics, inverse scattering, medicine diagnostics and therapy, memories synthesis and error rates, optical data processing , optical elements and imaging and display*** [Tricoles, 1987]. But in the area of three-dimensional display, computer-generated holograms are not used because the computer-generated holograms still suffer from various problems.

- ***The quality of the reconstructed images.*** The quality of the reconstructed images from a *"pure"* computer-generated hologram is far less than the quality of the reconstructed image from a optical hologram or the reconstructed images from a non-pure computer-generated holograms. *"Pure"* here means that the hologram is generated only by the computer and the plotter (Laser printer, Plotter, CRT *et al*) without any further optical processing. In general, the quality of the reconstructed images from a non-pure computer-generated hologram such as King *et al*'s holograms (see figure 5.16) is much better than the quality of reconstructed images from a pure computer-generated hologram. One of the important factors that affects the image quality of the computer-generated hologram is the limitation of the hardware used in the recording procedures. In optical holography, the commonly used recording media are silver halide emulsions, such as **Agfa-Gevaert** Scientia 8E75 and **Eastman kodak** Spectroscopic plate/film which can be used to record holograms with spatial frequencies great than 2000 [cyc./mm]. In pure computer-generated holograms, there is no device that can achieve this resolution.

Even if the high quality laser printer has only 1200 bpi, that is 47 [cyc./mm]. In fact, for a photo-reduced computer-generated hologram, the resolution is still far less the resolution of an optical hologram. Another factor that affects the image quality of the computer-generated hologram is the limitation of the dynamic range of the hardware device. For a photographic slide, its typical dynamic range is 1000, which corresponds to the number of intensities 700 (refer to Eq.(5.8)). In computer-generated holograms, the dynamic range is rather higher because of the DC peak of the Fourier spectrum. This means that more quantization levels are needed. But unfortunately, display devices with very large number of intensities are not common. In many laboratories or computer centres the typical display devices are CRT with 1024×1024 pixels provided with 256 intensity levels and laser printer with resolution 600 bpi and 16 intensities. With these devices, plus the necessary camera, we can make pure computer-generated holograms, but not with high quality reconstructed images.

- ***The dimensions of computer-generated holograms***. The dimensions of a pure computer-generated hologram strongly depend upon the limitation of hardware used in the recording stage of the computer-generated holograms (the dimensions, the resolution, the dynamic range of the output device, and the calculation speed of the computer). The typical dimensions of the photo-reduced computer-generated holograms are between $0.5\times0.5(\mathrm{mm}^2)$ to $2.5\times2.5(\mathrm{mm}^2)$. This size is useless for display purposes because the reconstructed images are too small to be observed. Non-pure computer-generated holograms (such as King *et al'* s holograms) can be made relatively large and could be used for display purposes. But these require further optical processing. They are expensive. The laser gun and other instruments for further optical processing are not common devices in many laboratories.

- ***The computation time and memory***. The computation time for making pure computer-generated holograms is determined by two factors. The first is the dimensional of the computer-generated hologram and the technique used to make it. The second one is the computer hardware, e.g. the memories in the computer and the calculation speed of the computer. A small size computer-generated hologram, for example, the detour phase hologram which has typical SBWP as 128×128 corresponding to a $10\times10(\mathrm{mm}^2)$ photo-reduced computer-generated hologram can be calculated fast. In our SUN-3/80 computer, it is required only 0.04 second to perform the fast Fourier transform, and about 30 seconds to draw the interference pattern with 8 quantization levels. For the Burch's hologram with SBWP=512×512, corresponding to a $10\times10(\mathrm{mm}^2)$ computer-generated hologram, it required about 180 seconds to draw the interference pattern on a CRT. When the parabolic method is used, the computation time is about 20 minutes for a

$10\times10(mm^2)$ computer-generated hologram. If we are going to make a $100\times100(mm^2)$ detour phase computer-generated hologram and the hardware device can display the interference pattern, then the computer must perform a Fourier transform for a two dimensional array which contains 10000×10000 elements. The computation time for this transformation will be extremely high even if the fast Fourier transform is used. For making pure computer-generated hologram, the memory to be used in the calculation of the Fourier transform is quite large, for example, to make a detour phase hologram with SBWP=128×128, we need a two dimensional array which contains 128×128 complex elements. For each of the complex elements, it requires two double precision numbers to represent the real part and the image part, if a double precision number is represented by 8 bytes, then the total memory needed to perform the Fourier transform is 128×128×2×8 = 256KB. This number of memory required is not too large. If a $100\times100(mm^2)$ computer-generated hologram is going to be produced, the memory needed for performing the Fourier transform will be 1.6GB. The huge memory required is impossible for some computers such as a PC. It is also a problem for some sophisticated computers and operating systems, such as the SUN workstation and the UNIX operating system.

In summary, the quality of the reconstructed images from the pure computer-generated holograms have not yet reached the level of conventional holograms because of the various problems discussed above. Therefore using a pure computer-generated hologram as a common three dimensional display device is not suitable at present. In this regard the method developed by King, Noll and Berry for displaying three-dimensional images obtained from computer calculations (refer to section 2.3 and figure 5.16) is far more effective than the pure computer-generated holograms approaches we discussed in above chapters. May be in the far future, with a suitable device, a pure computer-generated hologram could be a common three-dimensional display device.

Appendix A

Diffraction and Hologram Reconstruction

[Unterseher, and *et al*, 1982]

Normally we think of light travelling in straight lines. If the light encounters an object, a shadow is formed. The straight the light, the more distinct a shadow. If we look closely at the edge of the shadow, however, we will notice that it is blurred. This is because light will bend around the edge of an object. This bending of light around corners is known as *diffraction.*

When light passes through a hole, it will be diffracted around the edges. If we make this hole very small, on the order of the size of the wavelength of light, an interesting phenomenon occurs.

We would expect that, if light passed through a small opening, the waves would continue on through as shown in figure A.1.

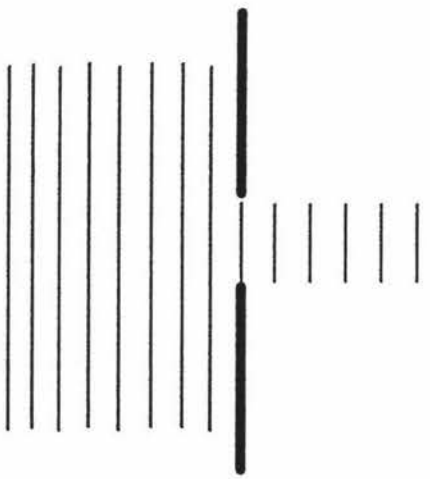


Figure A.1. Diffraction not taking place.

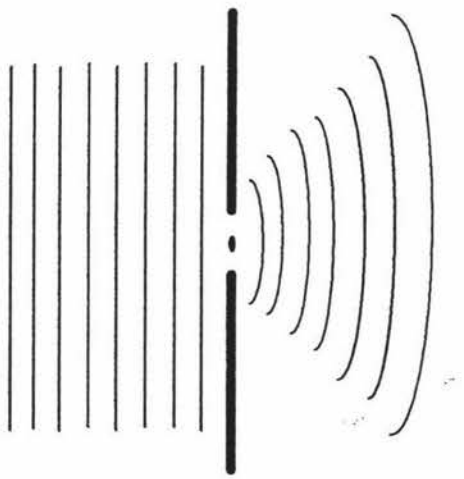


Figure A.2. Diffraction through a small aperture.

This, however, is not the case. What actually happens is as follows:

The small hole acts as a point source for a new set of waves which are bent toward the edges because of diffraction (figure A.2). The size of the opening will determine the amount of diffraction; the smaller the hole, the greater the bending of light. Large holes will bend it less.

If we now add more holes to the barrier, each individual hole will act as its own source for a new wave front. If we consider a medium that has a great number of

these holes, all of different size and configuration, it's easy to see that an extremely complex aggregate wave form cam be generated out of the effects of all of them together.

The hologram, of course, provides this kind of medium. The hologram is composed of millions of very tiny light and dark spots, representing the fringes of the original interference pattern. When we shine light onto the hologram, the dark spots act as barriers, yet light passes through the tiny clear holes, diffracting as it does. The interference pattern and the new wave fronts duplicate those originally present when the hologram was formed. Now when we see the hologram, we see the very same complex wave front that was originally reflected from the object, only now it is being produced by a lot of little holes. As far as we are concerned , it's hard to tell the difference. The new waves spread out to fill the room, just as they did when the object was there. Each of our eyes makes a separate image out of two perspectives of this wave form and three dimensions are synthesised in the brain just as they were for the real object.

Appendix B

Examples of Implementation

The following program implement the Radix-2 fast Fourier transform algorithm.

```
#include <stdio.h>
#include </usr/include/math.h>

#define M_PI        3.14159265358979323846
#define D_PI        M_PI*2

typedef struct co
{double r;
 double i;
}complex;

complex C_b(complex in1,complex in2);
complex C_a(complex in1,complex in2);
complex C_s(complex in1,complex in2);
complex C_d(complex in1,complex in2);
complex C_scale(complex in1,double in2);
complex C_c(complex in);
complex Exp(double in);
double  C_m(complex in);
double  Argu(complex in);



#include "complex.h"
#include <stdlib.h>
#include <stdio.h>
#include </usr/include/math.h>

complex C_a(complex in1,complex in2)
{complex out;
 out.r= in1.r + in2.r;
 out.i=in1.i + in2.i;
 return out;
 }

complex C_s(complex in1,complex in2)
{complex out;
 out.r= in1.r - in2.r;
 out.i=in1.i - in2.i;
 return out;
 }
```

```
complex C_b(complex in1,complex in2)
{complex out;
 out.r=in1.r*in2.r-in1.i*in2.i;
 out.i=in1.r*in2.i+in1.i*in2.r;
 return out;
}

//in1 is devided by in2. if in2 is equal tto zero, exception arise.
complex C_d(complex in1,complex in2)
{complex out;
 if ((in2.r*in2.r+in2.i*in2.i)==0.0)
   {printf("Divided By Zero! Program halts!\n");exit(0);
   }

 out=C_b(in1,C_c(in2));
 out.r/=(in2.r*in2.r+in2.i*in2.i);
 out.i/=(in2.r*in2.r+in2.i*in2.i);
 return out;
}

complex C_c(complex in)
{ complex out;
  out.r=in.r;
  out.i=-in.i;
  return out;
}

complex Exp(double in)
{complex out;
 out.r=cos(in);
 out.i=sin(in);
 return out;
}

complex C_scale(complex cin,double din)
{complex out;
 out.r=cin.r*din;
 out.i=cin.i*din;
 return out;
}

double  C_m(complex in)
{return sqrt(in.r*in.r+in.i*in.i);}

double  Argu(complex in)
{return atan2(in.i,in.r);}
```

```
#include "complex.h"
#include <stdlib.h>
#include <stdio.h>
#define  N1 512
#define  N2 512
#define  Ex1 9
#define  Ex2 9

int  index[N1];
void getindex(void);
void fft1d(complex *in,int N,int L);
void refft1d(complex *in,int N,int L);
void fft2d(complex (*in)[N1][N2]);
void refft2d(complex (*in)[N1][N2]);


#include "fft.h"
#include <math.h>

void getindex(void)
{register int i,j,k=1;
  index[0]=0;
  for (j=0;j<Ex1;j++)
  {for (i=0;i<k;i++)
     index[i+k]=index[i]+N1/(2*k);
   k<<=1;
  }
}


void fft1d(complex *in,int N,int L)
{long int i,j,le,le1,k,n,m;
 register int dex;
 complex U,W,T;

 complex *ou;
 ou=in;
// for (i=0;i<N;i++)
//  ou[i]=C_b(ou[i],Exp(M_PI*i));

 for (k=0;k<N;k++)
   {
    if (index[k]>k)
     {T=ou[k];
      ou[k]=ou[index[k]];
      ou[index[k]]=T;
     }
   }
```

```
   for (m=1;m<=L;m++ )
    { le=pow(2.0,m);
      le1=le/2;
      U.r=1.0;U.i=0.0;
      W=Exp(-M_PI/(le1*1));
      for (j=0;j<le1;j++)
       { for (i=j;i<N;i+=le)
          { k=i+le1;
           T=C_b(ou[k],U);
           ou[k]=C_s(ou[i],T);
           ou[i]=C_a(ou[i],T);
           }
         U=C_b(U,W);
        }
     }

   for (i=0;i<N;i++)
    {ou[i]=C_b(ou[i],Exp(M_PI*(i-N/2)));
     ou[i]=C_scale(ou[i],1.0/(double)N);
    }
}


void fft2d(complex (*in)[N1][N2])
{ long int k1,k2;
 complex *row,clm[N1];

 for (k2=0;k2<N2;k2++)
    { for (k1=0;k1<N1;k1++)
       clm[k1]=(*in)[k1][k2];
      fft1d(clm,N1,Ex1);

      for (k1=0;k1<N1;k1++)
        (*in)[k1][k2]=clm[k1];
    }

 for (k1=0;k1<N1;k1++)
    {
     row=&(*in)[k1][0];
     fft1d(row,N2,Ex2);
    }
 }
```

```
void refft1d(complex *in,int N,int L)
{long int i,j,le,le1,k,n,m;
 complex U,W,T;
 complex *ou;
 ou=in;
 for (k=0;k<N;k++)
   { if (index[k]>k)
     {T=ou[k];
      ou[k]=ou[index[k]];
      ou[index[k]]=T;
     }
   }
   for (m=1;m<=L;m++ )
    { le=pow(2.0,m);
      le1=le/2;
      U.r=1.0;U.i=0.0;
      W=Exp(M_PI/le1);
      for (j=0;j<le1;j++)
       { for (i=j;i<N;i+=le)
          { k=i+le1;
           T=C_b(ou[k],U);
           ou[k]=C_s(ou[i],T);
           ou[i]=C_a(ou[i],T);
           }
         U=C_b(U,W);
        }
     }
   for (i=0;i<N;i++)
     ou[i]=C_scale(ou[i],1.0*N);
}

void refft2d(complex (*in)[N1][N2])
{ long int k1,k2;
 complex *row,clm[N1];

 for (k2=0;k2<N2;k2++)
   { for (k1=0;k1<N1;k1++)
      clm[k1]=(*in)[k1][k2];
     fft1d(clm,N1,Ex1);
     for (k1=0;k1<N1;k1++)
      (*in)[k1][k2]=clm[k1];
   }

 for (k1=0;k1<N1;k1++)
   {
    row=&(*in)[k1][0];
    refft1d(row,N2,Ex2);
   }
 }
```

The following program demonstrate the procedure of making a Waters's hologram.

```
/*Computer-generated hologram  -  Waters's hologram program */
#include "libsx.h"
#include </usr/include/math.h>
#include <stdio.h>
#defineM_PI          3.14159265358979323846
#define lambda       0.0006328
#define No           32+32+16
typedef struct
 { double x,y,z;
  }3Dvector;
3Dvector points[No];

int width  = 512;
int height = 512;   /* size of draw area (with initial values) */

Widget  drawArea;
Widget objectButton, clearButton, quitButton;

void waters(void);

/* callbacks */
void dra(Widget,int wid,int hei,void*);  /* draw the diagram */
void doF();
void doClear();
void doQuit();
setup(){

 /* create Widgets*/

void dra(Widget drawArea,int wid,int hei,void *data)
{ ;}
 objectButton = MakeButton("object",doF,NULL);
 clearButton = MakeButton("clear",doClear,NULL);
 quitButton = MakeButton("quit",doQuit,NULL);
 drawArea = MakeDrawArea(width,height,NULL,0);

 /* position widgets */

 SetWidgetPos(objectlButton,NO_CARE,NULL,NO_CARE,0);
 SetWidgetPos(clearButton,PLACE_UNDER,objectButton,NO_CARE,0);
 SetWidgetPos(quitButton,PLACE_UNDER,clearButton,NO_CARE,0);
 SetWidgetPos(drawArea,PLACE_RIGHT,objectButton,NO_CARE, 0);

 /* install callback functions (for the default mode ) */

 SetButtonDownCB(drawArea,NULL);
```

```
  SetButtonUpCB(drawArea,NULL);
  SetMouseMotionCB(drawArea,NULL);
  /* start up the display */

  ShowDisplay();
  GetStandardColors();
  SetColor(BLACK);
}

void doQuit(){
  exit(0);
}
void doClear(){
   ClearDrawArea();
}
void doF(){
 waters();
}

void defineF(void)
{ double dx=10.0/32.0;
   int i, n=0;
  //define the up horizontal line of the letter "F".
    for (i=0; i<32; i++)
      { points[n].x= (double)(i)*dx;
        points[n].y= 10.0;
        points[n].z= 800.0;
        n++;
       }
  //define the middle horizontal line of the letter "F".
    for (i=0; i<16; i++)
      { points[n].x= (double)(i)*dx;
        points[n].y= 5.0;
        points[n].z= 800.0;
        n++;
       }
 //define the vertical line of the letter "F" .
    for (i=0; i<32; i++)
      { points[n].x=0.0;
        points[n].y=(double)(i)*dx;
        points[n].z= 800.0;
        n++;
       }
}
```

```
void waters(void)
{ double  P_l= M_PI/lambda;
  double x,y,i, j, sigma,dx=512/10.0;
  int n;
 defineF();
  for ( i = 0.0; i < 10.0; i+=dx)
    for ( j = 0.0; j < 10.0; j+=dx)
      { sigma=0.0;
        for ( n = 0; n < No; n++)
          { x = i - points[n].x;
            y = j - points[n].y;
           sigma+=sin (P_l*(x*x+y*y)/points[n].z);
           }
       if (sigma > 0.0)
         putpixel((int)(i*dx),(int)(j*dx),1);
      }
}

void main(void)
{
  setup();
  MainLoop();
}
```

References

Bayer, B.E., "An optimum method for two level radiation of continuous tone pictures", in Conference Record of the International Conference on Communications, 1973, 26-11-26-15.

Bergland, G. D., "A fast Fourier transform algorithm usinf base 8 iterations", Math.Comput. Vol. 22, p275-279, 1968.

Briggs,W. L., and V.E. Hensen, The DFT An Owner's Manual for the Discrete Fourier Transform. 1995.

Brigham, E.O., The Fast Fourier Transform, 1974.

Brown, B. R., and A.W. Lohmann, "Computer-generated Binary Holograms", IBM Journal of Research and Development. Vol. 13, p160, Mar. 1969.

Brown, B. R., and A. W. Lohmann, "Complex spatial filtering with binary masks", Applied Optics. Vol. 5, p967-969,1966.

Bryngdahl, O., and F. Wyrowski, "Digital Holography- Computer-Generated Holograms, Progress in Optics XXVIII, Edited by E. Wolf, 1990.

Burch, J. J., Proceedings of the IEEE, Vol. 55, p599, 1967.

Burckhardt, C. B., Applied optics. Vol. 9, p1949, 1970.

Cattey, W. T., Optical Information Processing and Holography, p55, 1974.

Cattey, W. T., Optical Information Processing and Holography, p108, 1974.

Chaimowicz, J. C. A., Light Wave Technology An Introduction, 1989.

Chamber, R. P., and T. S. Counttney-Pratt, "Bibliography on holograms", J. Soc. Motion Pic. and Telev. Eng. Vol. 75, p759, 1966.

Chu, D. C., and J. W. Goodman , Applied Optics, Vol. 11, p1716,1972.

Cooley, J. W., and J. W. Tukey, "An algorithm for machine computation of complex Fourier series", Math. Comput. Vol.19, p297-301, 1965.

Dallas, W. J., "Phase quantization - a compact derivation", Applied Optics, Vol. 10, p673-674, 1971a.

Dallas, W. J., "Phase quantization in holograms- a few illustration", Applied optics, Vol. 10, p674-6, 1971b.

Dudgeon, D. E., and R. M. Mersereau, Multidimensional digital signal processing, 1984.

Foley, J. D., A. van Dam, SK Feiner and J.F. Hughes, Computer Graphics principles and practice, Addison-Wesley Publishing Company, Inc. 1990.

Good, I. J., " The relationship between two fast Fourier transforms", IEEE transaction, C-20, p310-317, 1971.

Goodman, J,W. and A.M. Silvestri, "Some effects of Fourier domain phase quantization", IBM journal of research & development, Vol. 14, p478-84, 1970.

Haines, K., and D. Haines, 'Computer Graphics for Holography', IEEE Computer Graphics & Application, p37-46, 1992.

Hariharan, P., Optical Holography principles, Techniques and Applications, 1987.

Huang,T.S., "Digital Holography", Proceedings of the IEEE, Vol. 59, No.9, p1335, 1971.

Huang,T.S., and B. Prasada, MIT/REL Quar. Prog. Rep. Vol. 81, p199, 1966.

Jared, G.E.M., "Introduction to the solid modelling", Geometric Modeling and Computer Graphics techniques and applications, edited by R.A Earnshaw, R.D. Parslow and J.R. Woodwark, 1987.

King, M. C., A. M. Noll and D. H. Berry, Applied Optics, Vol. 9, p471, 1970.

Korchner, W., "Holographic Portraiture", Handbook of Optical Holography, edited by H. J. Caulfield, 1979.

Lee, W. H., "Binary synthetic hologram", Applied Optics, Vol.13, p1677, 1974.

Lee, W.H., "Computer-generated holograms: techniques and applications", In Progress in Optics, Vol. 16, edited by E. Wolf, p121-232. Amsterdam: North-Holland, 1978.

Leith, E.N., and J. Upatnieks, J.Opt. Soc.Am., Vol. 52, p1123,1962.

Lohmann, A.W., "The Space-Bandwidth Product Applied to Spatial Filtering and Holography"; Tech. Rept. IBM RJ438, IBM Research Laboratories, San Jose, 1967.

Lonmann, A. W., and D.P. Paris, Applied Optics, Vol. 6, p1739. 1967.

Meyer, A.J., and R. Hickling, "Hologram Synthesised on a Computer-Operated Cathode Ray Tube", Journal of the Optical Society of America, Vol. 57, No. 11, p1388, Nov. 1967.

Moller, K.D., Optics, p80, 1979.

Mortenson, M., Geometric Modeling, Wiley, New York, 1985.

Nussbaumer, H.J., Fast Fourier Transform and Convolution Algorithms. Edited by T.S. Huang, 1982.

Pokorny, C.K. and C.F. Gerald, Computer Graphics: The principles Behind The Art And Science, Franklin, Beadle and Associates. 1989.

Radar, C. M., "Discrete Fourier transforms when the number of data samples in prime", Proceedings of IEEE, Vol. 56, p1107-1108, 1968.

Requicha, A.A.G., "Representations of rigid solid: theory, methods and systems", ACM Computing Surveys, Vol. 12, No. 4, p437-464, Dec. 1980

Requicha, A.A.G., and H. B Voelcker, "Constructive Solid Geometry", University of Rechest Production Automation Project, Technical memorandum, TM-25, Nov. 1977.

Roth, S. D., "Ray casting for modeling solids", Computer Graphics and Image Process, Vol. 18, No.2, p109-144, 1982.

Smith, H. M., Principles of Holography, second edition, 1975.

Untersehr, F., J. Hansen and B. Schlesinger, Holography Handbook, 1982.

Van Wijk, J.J., " On new types of solid models and their visualisation with ray tracing", Doctoral thesis, Delft University of Technology, Delft University Press, Delft, 1986.

Waters, J. P., Journal of the Optical Society of America, Vol. 58, p1284, 1968.

Waters, J.P., Applied physics. letter, Vol. 9, p405, 1966.

Whitted, T., "An improved illumination model for shaded display", Communications of the ACM, Vol. 23, No.6, p343-349,1980.

Wyrowski, F., R. Hauck, and O. Bryngdahl, Journal Optical Society of American. A5, p1058, 1986.

Yatagai, T., " Stereoscopic approach to 3-D display using computer-generated holograms", Applied Optics. Vol. 15, No.11, p2722, 1976;

Yatagai, T., Optical communications. Vol. 12, p 43, 1974