L-10 Computer-Generated Holography

OBJECTIVES:

To understand the process of image reconstruction by producing simple holograms on a laser printer and reading them out with an optical lens system.

APPARATUS:

HeNe laser; microscope objective; achromat; mirror; hologram mount; screen; mirror; tilt and translation stages for lenses; mounting rail; computer with programs ”cgholo”, ”PAINT”, and ”LVIEW”; laser printer.

INTRODUCTION:

A hologram is basically a complicated diffraction grating; the diffraction pattern from the hologram ”reconstructs” an image that the diffraction grating represents. We can understand how a hologram works by remembering that a diffraction grating produces intensity maxima at angles θ according to the formula $n\lambda = d\sin\theta \approx d\theta$ for $\theta \ll 1$.

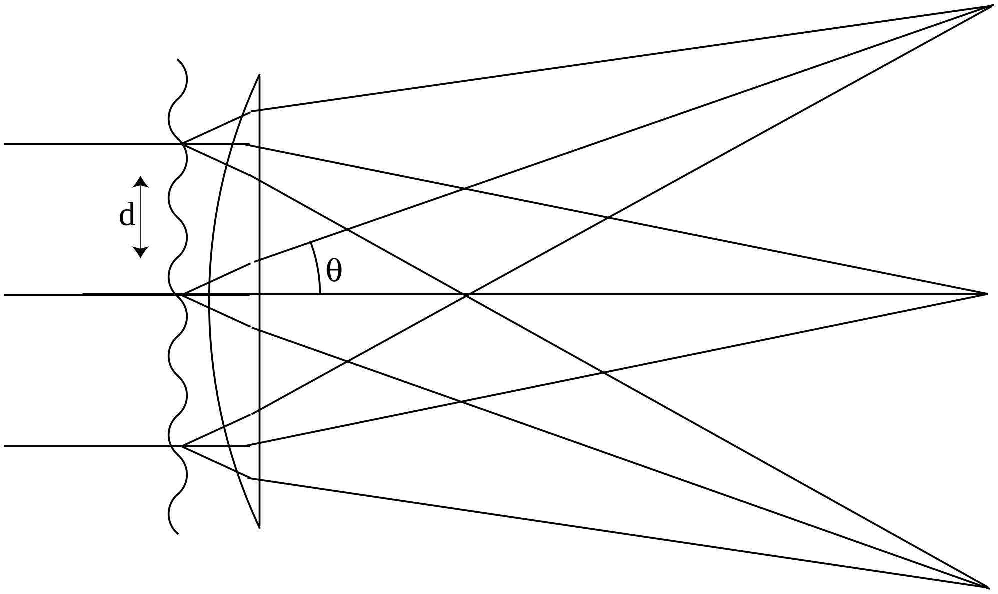

Figure 1: When collimated light strikes a grating that repeats every distance d, the diffracted angles are $\theta = n\lambda/d$. A lens is often used to focus light at each angle to a point as shown. Only $n = 1, 0, -1$ are shown here.

By controlling d, therefore, we can control the diffraction angles and therefore the position where the lens focusses the light down. In this laboratory you will construct diffraction gratings of increasing complexity using a simple computer program and a laser printer. By the end, you will understand how to make a diffraction grating that can write your name or make a simple picture on a screen.

EXPERIMENT:

Assemble the optical system shown below. Take care that the laser beam is centered on each lens; this will give you the smallest possible spot and therefore the best image when you are done.

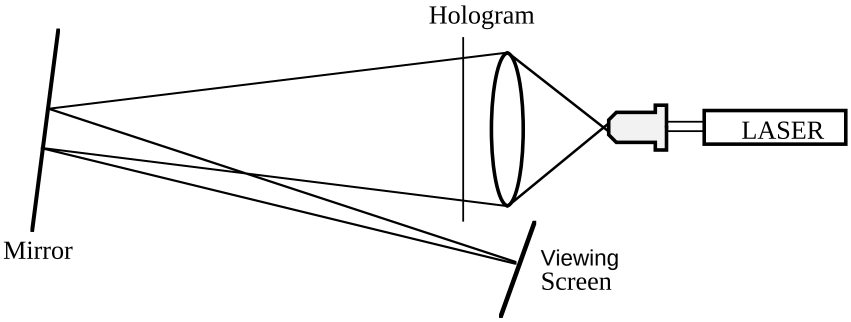

Figure 2: Layout for computer-generated holography.

When you get a nicely focussed spot on the viewing screen, your instructor will give you a hologram to view. Carefully adjust your lenses to get the sharpest image. Note: during the course of the laboratory you may find it necessary to periodically redo this.

Now let's take a look at the software we need to make diffraction gratings and holograms. A C program has been written for you; it is shown on the next page. The top portion of the program reads in a graphics file, the second portion calculates the hologram, and the third saves it to a file. Your instructor will give you an input file on which you can run the program. Then take the output file and open it with LVIEW and print the hologram on your printer. Check that it works.

For the first part of the lab you should now comment out the code that calculates the hologram and uncomment the code, just below it, that you will use to write a grating. This code consists of a nested loop, with variables X and Y, which are the coordinates on the transparency. Inside this loop you can make any function you like. Start by making a simple diffraction grating $\cos(2\pi X/\Lambda)$. Because the laser printer is limited to 300 dpi, Λ cannot be shorter than what value?

Print out your grating and look at the diffraction pattern on the screen. What do you see? Make measurements of the pattern and so determine the wavelength of the laser. You should be able to get the wavelength correct within a few percent.

Before we go further, we need to explain one other thing the program does. The laser printer is binary, only capable of writing a black dot or a lack of a black dot. Thus we must convert our cos function, which varies smoothly from -1 to 1, to binary. We do this by finding the maximum and minimum values of your hologram, then setting a threshold. Everything above this threshold becomes a 1; everything else a 0. Thus your cosine function becomes a rectangle of the same wavelength.

```
//Hologram program, kludged to work with LVIEW/PAINT
#include "stdafx.h"
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define twopi 6.283185307
#define N 300
#define thres 0.6
int main(int argc, char* argv[]){
    char pixel,obj,junkstring[30];
    int X,Y,row,col,junk,R;
    float pix[N][N],kdf=twopi/2.0,ranphi,max=0.0,min=0.0,res;
    FILE *out,*in;
    out=fopen("holo.pgm","w");
    in=fopen("syf.pbm","r");
    fprintf(out,"P5 %d %d 255\n",N,N);
    fscanf(in,"%1s %1s %d %d %d",&junk,&junk,&junk,&R,&junk);
    res=R;
    printf("res=%d,junk=%d\n",R,junk);
    srand(17);
    for(X=0;X<N;X++){for(Y=0;Y<N;Y++){pix[X][Y]=0.0;}}
    //--------------------------------------------------------------------
    //construct real hologram from input image-comment out this section
    // to make diffraction gratings
    for(row=0;row<R;row++){
        for(col=0;col<R;col++){
            fscanf(in,"%1c",&obj);
            fscanf(in,"%1c",&obj);
            fscanf(in,"%1c",&obj);
            if (obj>2 || obj<0){
                printf("row,col=%d %d\n",row,col);
                ranphi=((float) rand())/((float) RAND_MAX)*2.*3.1415926;
                for(X=0;X<N;X++){for(Y=0;Y<N;Y++){
                        pix[X][Y]+=cos(kdf*(X*(1.-row/res)+Y*col/res)+ranphi);
                }}
            }
        }
    }
    //---------------------------------------------------------------------
    //---------------------------------------------------------------------
    //example code for making diffraction gratings-comment out this section
    //to make holograms
    /*for(X=0;X<N;X++){for(Y=0;Y<N;Y++){
        pix[X][Y]=cos(kdf*(X/1.2+Y/2.4));
        pix[X][Y]=cos(kdf*(X/1.4+Y/2.7));
    }}*/
    //---------------------------------------------------------------------
    //find max min
    for(X=0;X<N;X++){for(Y=0;Y<N;Y++){
            if(pix[X][Y]>max)max=pix[X][Y];
            if(pix[X][Y]<min)min=pix[X][Y];

    }}
    printf("max,min=%f %f\n",max,min);
    for(X=0;X<N;X++){for(Y=0;Y<N;Y++){
            if((pix[X][Y]-min)/(max-min)>thres){
                pixel=255;
            } else {
                pixel=0;
            }
            fprintf(out,"%1c",pixel);
    }}
    return 0;
}
```

Now make another diffraction grating, but tilt it by 30° angle from the previous one.

Show that you can make diffraction gratings that produce dots at various positions.

Make a complicated grating with, say, 5 periods instead of one. How does it come out? Make the letter "T".

You now know how to make an arbitrarily complicated pattern. By simply superposing gratings that produce first order diffraction at various points, you can make any pattern you want on the screen. Now look at the code you commented out earlier to see how it accomplishes this.

You are now ready to make your own hologram. Comment out your grating code and uncomment the holography code. Prepare a graphics file with PAINT that has a picture, or your name, on it, and run it through the program.

FURTHER READING:

"Holography without photography", Thad G. Walker, Am. J. Phys. **67**, 783, September 1999.