Appendix

Examples of Digital Hologram Reconstruction Programs

This appendix offers some programs for the calculation of diffraction and the reconstruction of digital holograms. These programs were written in the MATLAB language. This software allows scientific calculations to be carried out in a very straightforward way, considerably simplifying coding compared to other languages (such as VC++, VB, or DELPHI). The programs are organized into ".m" files. The programs have been validated in MATLAB 7.0. If a subsequent version is used, the reader will have to verify that the code is compatible.

The selected programs are the following:

– LIM1.m: diffraction calculation using the S-FFT algorithm;

– LIM2.m: diffraction calculation using the D-FFT algorithm;

– LIM3.m: theoretical simulation of a digital hologram;

– LIM4.m: reconstruction of a hologram using the S-FFT algorithm;

– LIM5.m: reconstruction of a hologram using the D-FFT algorithm;

A1.1. Diffraction calculation using the S-FFT algorithm

A1.1.1. *Code for the program: LIM1.m*

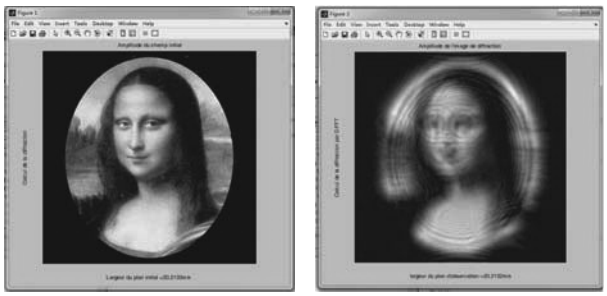
```
%-------------------------LIM1-------------------------
% Function: Diffraction calculation using the S-FFT algorithm
```

```
%
% Procedure: consider an image file as the amplitude distribution of
% the initial plane, then calculate the amplitude of the diffracted field,
% giving a wavelength and a diffraction distance.
%
% Variables:
% h : wavelength (mm);
% z0 : diffraction distance (mm);
% U0 : complex amplitude of the initial field;
% L0 : width of the initial field (mm);
% Uf : complex amplitude of the field in the observation plane;
% L : width of the observation plane (mm);
% In case of a rectangular image, it is padded with zeros to its larger dimension
%-----------------------------------------------------------------
clear;close all;
chemin='C:\';
[nom,chemin]=uigetfile([chemin,'*.*'],['initial image'],100,100);
[XRGB,MAP]=imread([chemin,nom]);
X=XRGB(:,:,1); % conserve the first channel if the image is RGB
h=input('Wavelength (mm) : ');
L0=input('Maximum width of the initial plane L0 (mm) : ');
k=2*pi/h;
[M,N]=size(X);
K=max(M,N);
% Zeros-padding to get KxK image
Z1=zeros(K,(K-N)/2);
Z2=zeros((K-M)/2,N);
Xp=[Z1,[Z2;X;Z2],Z1];
zmin=L0^2/K/h;
disp(['Minimum distance to fullfill sampling theorem : ',num2str(zmin),' mm']);
```

```
z0=input(['Diffraction distance z0 (mm) (>',num2str(zmin),'mm) : ']);
U0=double(Xp);
figure(1),imagesc(Xp),colormap(gray);axis equal;axis tight;ylabel('pixels');
xlabel(['Width of the initial plane: ',num2str(L0),' mm']);
title('Amplitude of the initial field');
%---------------
n=1:K;m=1:K;
x=-L0/2+L0/K*(n-1);
y=-L0/2+L0/K*(m-1);
[xx,yy]=meshgrid(x,y);
Fresnel=exp(i*k/2/z0*(xx.^2+yy.^2));
f2=U0.*Fresnel;
Uf=fft2(f2,K,K);
Uf=fftshift(Uf);
L=h*abs(z0)*N/L0;
x=-L/2+L/K*(n-1);
y=-L/2+L/K*(m-1);
[xx,yy]=meshgrid(x,y);
phase=exp(i*k*z0)/(i*h*z0)*exp(i*k/2/z0*(xx.^2+yy.^2));
Uf=Uf.*phase;
%---------------
If=abs(Uf);% amplitude of the diffracted field
figure(2),imagesc(abs(Uf)),colormap(gray);axis equal;axis tight;ylabel('pixels');
xlabel(['Width of the observation plane: ',num2str(L),'x',num2str(L),' mm']);
title('Amplitude of the image diffracted by S-FFT');
```

A1.1.2. *Examples of diffraction calculations using LIM1.m*

The studied object is an image of $M \times N = 512 \times 512$ pixels that represents a section of the famous painting the “Mona Lisa” (see Figure A1.1(a)).

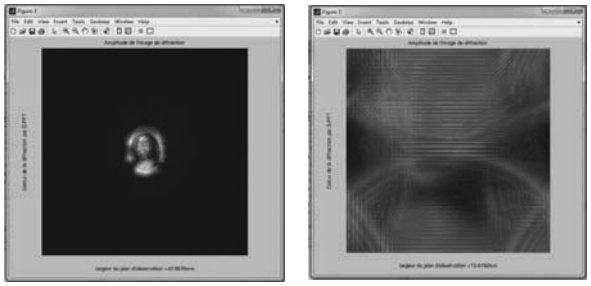


a) Initial amplitude b) Diffracted image

$L0 = L = 20.2133$ mm, $N \times N = 512 \times 512$
$h = 532$ nm, $z0 = 1500$ mm

Figure A1.1. *Comparison between the initial and diffracted fields*

Considering the parameters of the program to be $h = 0.000532$ mm (wavelength), and L0 = 20.2133 mm (width of the initial plane), Figure A1.1(b) shows the distribution of the modulus of the diffracted field in the observation plane. The minimum distance is given by zmin = $L0^2/N/h$ = 1500 mm; with a diffraction distance of z0 = 1,500 mm, the width of the observation plane L is equal to L0, satisfying relation [3.15]. The calculated field satisfies the sampling theorem. We note that after diffraction across a distance of 1,500 mm, the amplitude is strongly modified.



a) $L0 = 10$ mm, $L = 40.8576$ mm b) $L0 = 30$ mm, $L = 13.6192$ mm
$N \times N = 512 \times 512$, $h = 532$ nm, $z0 = 1500$ mm

Figure A1.2. *Influence of L_0 on the diffraction calculation by S-FFT*

From the calculation of diffraction using S-FFT (see section 3.2.1), the physical width of the diffracted field varies with the wavelength, the diffraction distance and the number of samples. The numerical result must satisfy the sampling theorem. As an example, we consider L0 = 10 mm and L0 = 30 mm. Figures A1.2(a) and (b)

show the respective results, calculated by LIM1.m. The result in Figure A1.2(a) satisfies the sampling condition. The width of the initial field is relatively small, and the width L calculated by L = h*z0*N/L0 is large (see section 3.2.1). In this case, the diffracted image is situated in a small zone at the center of the observation plane. This result is therefore not well optimized. In the case of Figure A1.2(b), the sampling condition is no longer satisfied. The width of the initial field is relatively large, and because of the phenomenon of spectral aliasing (see section 3.1.2), the result is incorrect and unusable.

The reader may exhaustively study the influence of the variation in the physical parameters on the obtained result using the program LIM1.m.

A1.2. Diffraction calculation by D-FFT

A1.2.1. *Code for the program: LIM2.m*

```
%--------------------------LIM2--------------------------
% Function: Diffraction calculation using the D-FFT algorithm
% Procedure: consider an image file as the optical amplitude
% distribution of the initial plane, the calculate the amplitude of the
% diffracted field, giving a wavelength and a diffraction distance
%
% Variables :
% h : wavelength (mm);
% z0 : diffraction distance (mm);
% U0 : complex amplitude of the initial optical field;
% L0 : width of the initial optical field and observation plane (mm);
% In case of a rectangular image, it is padded with zeros to its larger dimension
%-------------------------------------------------------------
clear;close all;
chemin='C:\';
[nom,chemin]=uigetfile([chemin,'*.*'],['initial image'],100,100);
[XRGB,MAP]=imread([chemin,nom]);
X=XRGB(:,:,1);
```

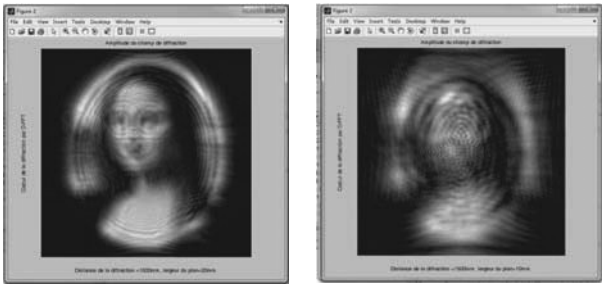
```
h=input('Wavelength (mm) : ');
L0=input('Maximum width of the initial field L0 (mm) : ');
k=2*pi/h;
[M,N]=size(X);
X=double(X);
K=max(M,N);
% Zeros-padding to get KxK image
Z1=zeros(K,(K-N)/2);
Z2=zeros((K-M)/2,N);
Xp=[Z1,[Z2;X;Z2],Z1];
zmax=L0^2/K/h;
disp(['Maximum distance to fullfill sampling theorem : ',num2str(zmax),' mm']);
z0=input(['Diffraction distance z0 (mm) (<',num2str(zmax),'mm) : ']);
U0=Xp;
figure(1),imagesc(Xp),colormap(gray);ylabel('pixels');
axis equal;axis tight;
xlabel(['Width of the initial field =',num2str(L0),' mm']);title('Initial amplitude ');
%---------------Diffraction calculation by D-FFT
Uf=fft2(U0,K,K);
Uf=fftshift(Uf); % Spectrum of the initial field
fex=K/L0;fey=fex;% sampling of frequency plane
fx=[-fex/2:fex/K:fex/2-fex/K];
fy=[-fey/2:fey/K:fey/2-fey/K];
[FX,FY]=meshgrid(fx,fy);
G=exp(i*k*z0*sqrt(1-(h*FX).^2-(h*FY).^2)); % Angular spectrum transfer function
% Diffraction
result=Uf.*G;
Uf=ifft2(result,K,K);
%---------------End of D-FFT calculation
If=abs(Uf);
```

```
figure(2),imagesc(abs(Uf)),colormap(gray);ylabel('pixels');
axis equal;axis tight;
xlabel(['Diffraction distance = ',num2str(z0),' mm, largeur du plan= ',num2str(L0),'
mm']);title('Amplitude of field diffracted by D-FFT');
```

A1.2.2. *Examples of diffraction calculations using LIM2.m*

We again consider the "Mona Lisa" from Figure A1.1(a). With h = 0.532e-3 mm (wavelength), z0 = 1,500 mm (diffraction distance) and L0 = 20.2133 mm (width of the initial plane), Figure A1.3(a) shows the modulus of the amplitude of the diffracted field in the observation plane. From section 3.4.2, the width of the observation plane L is always equal to L0 and the calculated result satisfies the sampling theorem (see section 3.2.3).

If we now consider the width of the initial plane to be L0 = 10.10665 mm and keep the other parameters the same as in Figure A1.3(a), the obtained result is shown in Figure A1.3(b). In this case, the angular spectrum transfer function does not satisfy the sampling conditions. However, the spectrum of the image being very narrow and localized at the center of the spectrum, the result does not suffer from aliasing.



a) $L0$ = 20.2133 mm = L b) $L0$ = 10.10665 mm = L

$N \times N = 512 \times 512$, h = 532 nm, $z0$ = 1500 mm

Figure A1.3. *Diffraction calculations using the D-FFT algorithm*

To solve a practical diffraction problem, we must therefore make a suitable choice between the S-FFT and D-FFT methods.

A1.3. Simulation of a digital hologram

A1.3.1. *Code for the program: LIM3.m*

```
%-------------------------LIM3-------------------------
% Function: simulate a hologram using an image whose width
% is less than a quarter of the initial plane.
% The simulated hologram will be saved with the name "Ih.tif".
%
% Procedure: (l) Read an image file
%            (2) Give the parameters asked for on the screen
% Variables :
%  h : wavelength (mm);
%  Ih : hologram;
%  L : width of the hologram (mm);
%  L0 : width of the diffracted object field (mm);
%  z0 : recording distance (mm);
% In case of a rectangular image, it is padded with zeros to its larger dimension
%-------------------------------------------------------------
clear;close all;
chemin='C:\';
[nom,chemin]=uigetfile([chemin,'*.*'],['Choose an image file'],100,100);
[XRGB,MAP]=imread([chemin,nom]);
X=double(XRGB(:,:,1));% We recover the image of the red RGB
% band (channel 1)
[M,N]=size(X);
% Extended size to two times
K=2*max(N,M);
% Zeros-padding to get N×N image
```

```
Z1=zeros(K,(K-N)/2);
Z2=zeros((K-M)/2,N);
Obj=[Z1,[Z2;X;Z2],Z1];
% Parameters
h=input('Wavelength (mm) : ');
k=2*pi/h;
L=input('Maximum width of the object (mm) : ');
z0=input(['Recording distance z0 (mm) : ']);
pix=abs(z0)*h/L;
Lx=K*pix;
Ly=K*pix;
disp(['Pixel pitch to fullfill sampling conditions : ',num2str(pix),'mm']);
disp(['Width of the object field = ',num2str(Lx),'mm x ',num2str(Ly),'mm']);
% Object field
psi=2*pi*(rand(K,K)-0.5);% Random phase
Ao=Obj.*exp(i.*psi); % Complex field in the object plane
figure;imagesc(Obj);colormap(gray);
colormap(gray);ylabel('pixels');
axis equal;axis tight;
xlabel(['Width of the object field = ',num2str(Lx),'mm x ',num2str(Ly),'mm']);
title('Initial Object');
%---------------Calculation using S-FFT
% Complex factor in the integral
n=-K/2:K/2-1;m=-K/2:K/2-1;
x=n*pix;y=m*pix;
[xx,yy]=meshgrid(x,y);
Fresnel=exp(i*k/2/z0*(xx.^2+yy.^2));
```

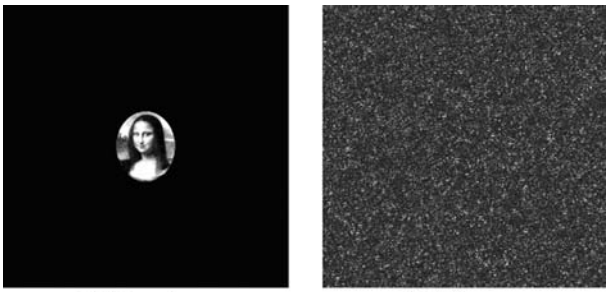
```
f2=Ao.*Fresnel;
Uf=fft2(f2,K,K);% Zero padding at KxK
Uf=fftshift(Uf);
% Complex factor in front of the integral
% Pitch in sensor plane
ipix=h*abs(z0)/K/pix;
xi=n*ipix;
yi=m*ipix;
L0x=K*ipix;
L0y=K*ipix;
[xxi,yyi]=meshgrid(xi,yi);
phase=exp(i*k*z0)/(i*h*z0)*exp(i*k/2/z0*(xxi.^2+yyi.^2));
Uf=Uf.*phase;
%--------------End of S-FFT calculation
disp(['Width of the diffracted field = ',num2str(L0x),'mm x ',num2str(L0y),'mm']);
figure,imagesc(abs(Uf)),colormap(gray);ylabel('pixels');
axis equal;axis tight;
xlabel(['Width of the diffracted field = ',num2str(L0x),'mm x ',num2str(L0y),'mm']);
title('Diffracted field in the detector plane (modulus)');
% Reference wave
ur=Lx/8/h/z0; % Spatial frequencies
vr=ur;
Ar=max(max(abs(Uf)));% Amplitude of the reference wave
Ur=Ar*exp(2*i*pi*(ur*xx+vr*yy));% Reference wave
%---------------Calculation of the hologram
H=abs(Ur+Uf).^2;
% 8-bit digitization
Imax=max(max(H));
Ih=uint8(255*H/Imax);
```

```
nom='Ih.tif';
imwrite(Ih,nom);% Recording the hologram
disp(['Pixel pitch = ',num2str(ipix),' mm avec ',num2str(K),'X',num2str(K),' pixels']);
figure,imagesc(Ih),colormap(gray);ylabel('pixels');
xlabel(['Pixel pitch = ',num2str(ipix),' mm avec ',num2str(K),'X',num2str(K),'
pixels']);
title(['Digital hologram with the name : ',nom]);
```

A1.3.2. *Example of the calculation of a hologram with LIM3.m*

In this program, the object is an image file, and the diffraction is calculated using the S-FFT method. From Chapter 5 (see section 5.1.2), the width of the figure must be less than a quarter of the object plane so that the image reconstructed by S-FFT may be separated from the parasitic orders. The inclination of the reference wave is simulated so that the reconstructed image lies in the second quadrant of the reconstruction plane.

As an example, the image of the "Mona Lisa" is the object, and it is padded with zeros to form the object plane with $N \times N$ = 1,024 × 1,024 pixels (see Figure A1.4(a)). Letting the wavelength be h = 0.000532 mm, the pixel width of the hologram pix = 0.00465 mm and the recording distance z0 = 1,500 mm. The hologram of 1,024 × 1,024 pixels calculated by LIM3.m is displayed in Figure A1.4(b). This hologram may be reconstructed by the different methods described in Chapter 5.



a) Object plane b) Hologram

Figure A1.4. *Simulation of the digital hologram*

A1.4. Reconstruction of a hologram by S-FFT

A1.4.1. *Code for the program: LIM4.m*

```
%-------------------------LIM4-------------------------
% Function: the reconstruction of a hologram using the S-FFT
% algorithm
%
% Procedure: (l) Read an image file
%            (2) Give the parameters asked for on the screen
%
% Variables:
%  Ih : hologram;
%  h : wavelength (mm);
%  L : width of the hologram (mm);
%  L0 : width of the diffracted object field (mm);
%  z0 : reconstruction distance (mm);
%  U0 : complex amplitude in the reconstruction plane;
%------------------------------------------------------------
clear;clc;close all;
chemin=’C:\';
[nom,chemin]=uigetfile([chemin,'*.*'],['Choose a hologram'],100,100);
I1=imread([chemin,nom]);
figure;imagesc(I1);colormap(gray);axis equal;axis tight;title('Digital hologram');
Ih1=double(I1)-mean2(double(I1));
[N1,N2]=size(Ih1);
N=min(N1,N2); % Restriction to NxN
Ih=Ih1(1:N,1:N);
pix=input('Pixel pitch (mm) : ');
h=input('Wavelength (mm) : ');
```

```
z0=input('Reconstruction distance z0 (+ for a real image, - for a virtual image)
(mm) : ');
L=pix*N;
%------------------------Reconstruction by S-FFT
n=-N/2:N/2-1;
x=n*pix;y=x;
[xx,yy]=meshgrid(x,y);
k=2*pi/h;
Fresnel=exp(i*k/2/z0*(xx.^2+yy.^2));
f2=Ih.*Fresnel;
Uf=fft2(f2,N,N);
Uf=fftshift(Uf);
ipix=h*abs(z0)/N/pix;
x=n*ipix;
y=x;
[xx,yy]=meshgrid(x,y);
phase=exp(i*k*z0)/(i*h*z0)*exp(i*k/2/z0*(xx.^2+yy.^2));
U0=Uf.*phase;
%-------------------------End of S-FFT
If=abs(U0).^0.75;
Gmax=max(max(If));
Gmin=min(min(If));
L0=abs(h*z0*N/L);
disp(['Width of the reconstruction plane =',num2str(L0),' mm']);
figure;imagesc(If,[Gmin,Gmax]),colormap(gray);axis equal;axis
tight;ylabel('pixels');
xlabel(['Width of the reconstruction plane =',num2str(L),' mm']);
title('Image reconstructed by S-FFT');
p=input('Display parameter (>1) : ');
```

```
while isempty(p) == 0
  imagesc(If,[Gmin Gmax/p]),colormap(gray);axis equal;axis tight;ylabel('pixels');
  xlabel(['Width of the reconstruction plane =',num2str(L),' mm']);
  title(' Image reconstructed by S-FFT ');
  p=input('Display parameter (>1) (0=end) : ');
  if p==0,
    break
  end
end
```

A1.4.2. *Example of reconstruction with LIM4.m*

Using the hologram calculated previously by LIM3.m, Figure A1.5 shows the result of the reconstruction. We observe in Figure A1.5(a) a very intense central spot which corresponds to the Fresnel transform of the mean value of the hologram. It is therefore difficult to observe the virtual image. Capping the grayscale at 1/20 of the maximum reconstructed amplitude (consider p = 20 in the dialog box at the end of the program's execution), Figure A1.5(b) shows the reconstruction plane. We now clearly observe the image in the upper left part of the plane, the expanded zero order, and the twin image at the bottom right.

This program may be used for any experimental hologram.

a) Display parameter $p = 1$ b) Display parameter $p = 20$

$L = 4.76$ mm, $L0 = 171.6129$ mm, $N \times N = 1024 \times 1024$
$h = 0.000532$ mm, $z0 = 1500$ mm

Figure A1.5. *Image reconstruction by S-FFT*

A1.5. Adjustable-magnification reconstruction by D-FFT

A1.5.1. *Code for the program: LIM5.m*

```
%-------------------------LIM5---------------------------------
% Function: reconstruct an image with a adjustable magnification using
% the D-FFT algorithm.
% We may decrease the perturbation by the zero order by suppressing the
% mean value of the hologram.
%
% Procedure: (l) Read an image file
%                (2) Give the parameters asked for on the screen
%                (3) Calculate by D-FFT with adjustable magnification
%
% Variables:
%  Ih : hologram;
%  h : wavelength (mm);
%  L : width of the hologram (mm);
%  L0 : width of the diffracted object field (mm);
%  z0 : recording distance of the hologram (mm);
%  zi : reconstruction distance (mm);
%  zc : radius of the reconstruction wave's wavefront (mm);
%  U0 : complex amplitude in the reconstruction plane;
%  pix : pixel pitch of the hologram (mm);
%  ipix : pixel pitch in the reconstructed field using S-FFT;
%------------------------------------------------------------
clear;close all;
chemin='C:\';
[nom,chemin]=uigetfile([chemin,'*.*'],['Choose the hologram to be
reconstructed'],100,100);
I1=imread([chemin,nom]);
```

```
Ih1=double(I1);
figure;imagesc(I1);colormap(gray);axis equal;axis tight;
title('Digital hologram');
pix=input('Pixel pitch (mm) : ');
h=input('Wavelength (mm) : ');
z0=input('Reconstruction distance z0 (+ for a real image, - for a virtual image) (mm)
: ');
k=2*pi/h;
[N1,N2]=size(Ih1);
N=min(N1,N2);
Ih=Ih1(1:N,1:N)-mean2(Ih1(1:N,1:N));% suppression of the mean value
L=pix*N;
disp(['Width of sensor : ',num2str(L),' mm']);
pg=input('Filter the 0 order of the hologram (1/0) ? ');
if pg==1,
   fm=filter2(fspecial('average',3),Ih); % see section 5.3.4.
   Ih=Ih-fm;
end
%--Reconstruction by S-FFT to find center/bandwidth of object
n=-N/2:N/2-1;
x=n*pix;y=x;
[xx,yy]=meshgrid(x,y);
Fresnel=exp(i*k/2/z0*(xx.^2+yy.^2));
f2=Ih.*Fresnel;
Uf=fft2(f2,N,N);
Uf=fftshift(Uf);
ipix=h*abs(z0)/N/pix;
xi=n*ipix;
yi=xi;
figure;imagesc(xi,yi,abs(Uf).^0.75);colormap(gray);axis equal;axis tight;
```

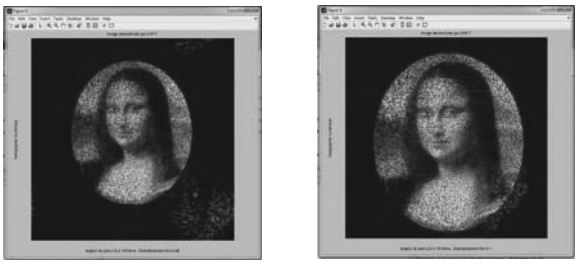
```
title('Click on the upper-left and lower-right corner of the object');
XY=ginput(2);
% Center and width of the object
xc=0.5*(XY(1,1)+XY(2,1));
yc=0.5*(XY(1,2)+XY(2,2));
DAX=abs(XY(1,1)-XY(2,1));
DAY=abs(XY(1,2)-XY(2,2));
%--Reconstruction with adjustable magnification
Gyi=min(L/DAX,L/DAY);
Gy=input(['Magnification factor for the reconstruction (ideal : ',num2str(Gyi),') : ']);
zi=-Gy*z0;
zc=1/(1/z0+1/zi);
% Spherical wave calculation
sphere=exp(i*k/2/zc*(xx.^2+yy.^2));
% Illumination of the hologram by a spherical wave
f=Ih.*sphere; % Spectrum of hologram multiplied by spherical wave
TFUf=fftshift(fft2(f,N,N));
% Fourier space
du=1/pix/N;dv=du;
fex=1/pix;fey=1/pix;
fx=[-fex/2:fex/N:fex/2-fex/N];
fy=[-fey/2:fey/N:fey/2-fey/N];
[FX,FY]=meshgrid(fx,fy);
% Spatial frequencies of reference wave
Ur=xc/h/abs(z0);
Vr=yc/h/abs(z0);
% Transfer function
Du=abs(Gy*DAX/h/zi);
Dv=abs(Gy*DAY/h/zi);
Gf=zeros(size(f));
```

```
Ir=find(abs(FX-Ur) < Du/2 & abs(FY-Vr) < Dv/2);
Gf(Ir)=exp(-i*k*zi*sqrt(1-(h*(FX(Ir)-Ur)).^2-(h*(FY(Ir)-Vr)).^2));
% Reconstruction
if sign(z0) == -1
  U0=fft2(TFUf.*Gf,N,N);
elseif sign(z0) == +1
  U0=ifft2(TFUf.*Gf,N,N);
end
Gmax=max(max(abs(U0).^0.75));
Gmin=min(min(abs(U0).^0.75));
figure;imagesc(abs(U0).^0.75,[Gmin,Gmax/1]);colormap(gray);
axis equal;axis tight;
xlabel(['Magnification : ',num2str(Gy)]);
title('Image reconstructed by D-FFT');
p=input('Display parameter (>1) : ');
while isempty(p) == 0
  imagesc(abs(U0).^0.75,[Gmin,Gmax/p]),colormap(gray);axis equal;axis
tight;ylabel('pixels');
  xlabel(['Width of the reconstruction plane =',num2str(L),' mm']);
  title(' Image reconstructed by D-FFT with adjustable magnification ');
  p=input('Display parameter (>1) (0=end) : ');
  if p==0,
    break
  end
end
```

A1.5.2. *Example of adjustable-magnification reconstruction with LIM5.m*

Using the hologram previously calculated by LIM3.m, Figure A1.6 shows the images reconstructed with magnifications of G_y = 0.05 and G_y = 0.1. During the execution of the program, we may diminish the perturbation by the zero order by

deleting the hologram smoothed by a 3×3 moving average filter (see section 5.3.4). Furthermore, the display of the calculated image plane by S-FFT is considered as the reference determining the spectral position of the object wave and its bandwidth (see section 5.4.1.2). To clearly display the reconstructed image, as with LIM4.m, we may modify the value of the display parameter p.



a) Magnification $G_y = 0.05$ b) Magnification $G_y = 0.1$

($L = 4.76$ mm, $N \times N = 1024 \times 1024$, $h = 0.000532$ nm, $z0 = 1500$ mm)

Figure A1.6. *Reconstruction of the image using D-FFT*

This program may be used for any experimental hologram.

Note that LIM5.m supposes that the reference wave is a plane wave.