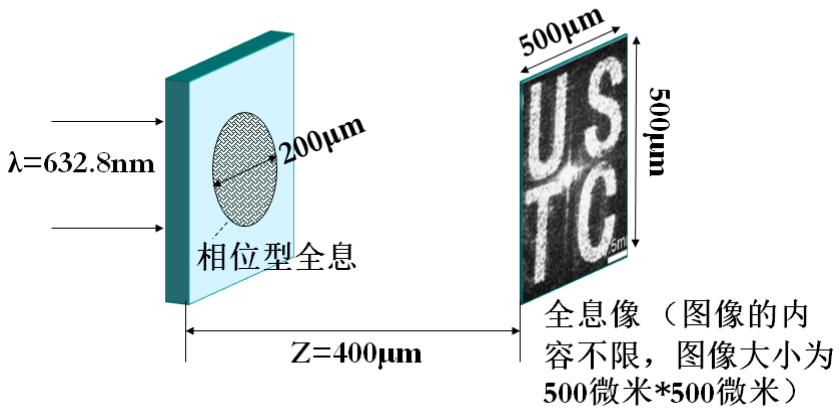
计算全息作业报告

PB18020539 黄韫飞

题目



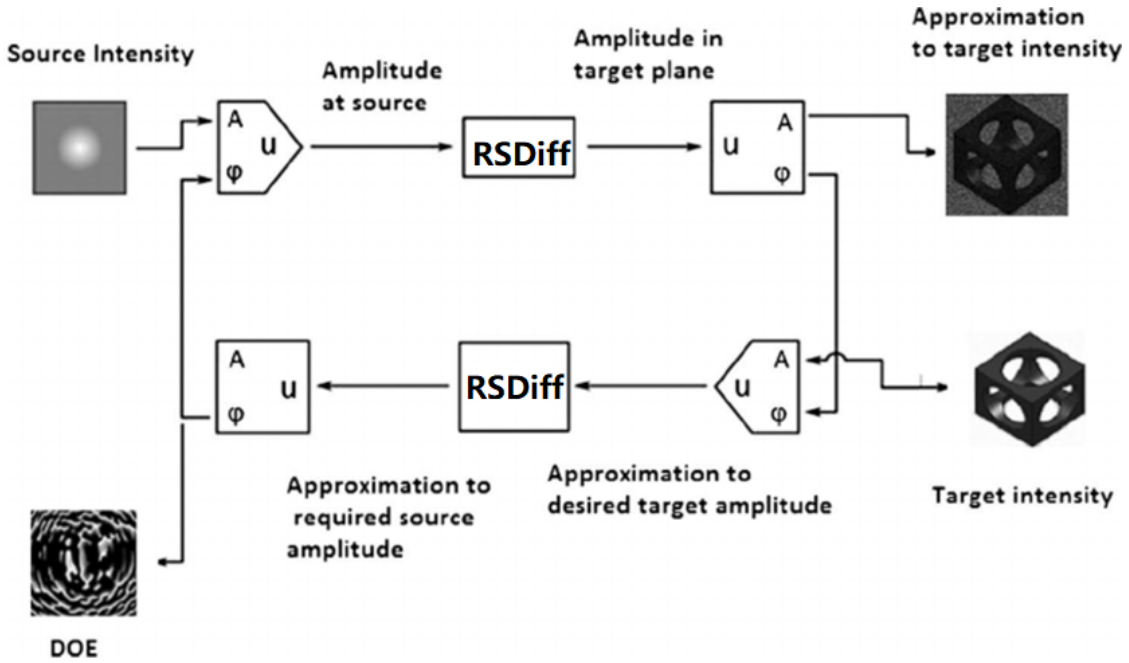
如上图所示，一束632.8nm激光，入射到直径为200微米的相位型全息片上，其透射光在z=400微米处，形成一个500μm *500μm的图像（图像内容自定）。结合角谱方法或Rayleigh-Sommerfeld衍射，自主搭建GS算法，优化以下两种情况下的全息片位相分布。

1.当全息片的像素单元尺寸为0.3μm*0.3μm时的位相分布

2.当全息片的像素单元尺寸为2μm*2μm时的位相分布

3.比较两种情况的区别，并分析原因

算法:



代码:

HW3_1.m

```
%Holography Simulation(GS Algorithm)
clear,clc,clf
```

```
rng(1)
Figure=im2double(imread('Wujing.jpg'));
Target(:,:)=Figure(:,:,1);
z=400;lambda=0.6328;tolerance=0.8;stepxy=0.3;k=2*pi/lambda;ntarget=500;
Target=imresize(Target,ntarget/length(Target));
x=(-ntarget/2):stepxy:(ntarget/2);y=(-ntarget/2):stepxy:(ntarget/2);
nscreen=length(x);
Target=imresize(Target,nscreen/ntarget);
[xx,yy]=meshgrid(x,y);
r=sqrt(xx.^2+yy.^2);
Amp=zeros(nscreen,nscreen);
Amp(r<100)=1;
phase0=2*pi*rand(nscreen,nscreen);screen=Amp.*exp(1i*phase0);
U=RSDiff(z,x,k,screen);
% U=fft2(screen);
correlate=ErrorCriterion(abs(U).^2,Target);
fprintf('n\t correlate\n')
count=1;
fprintf('%d\t %f\n',count,correlate);
while correlate<tolerance && count<=200
    A=sqrt(Target).*exp(1i*angle(U));
    B=RSDiff(-z,x,-k,A);
% B=ifft2(A);
    C=Amp.*exp(1i*angle(B));
    U=RSDiff(z,x,k,C);
% U=fft2(C);
    count=count+1;
    correlate=ErrorCriterion(abs(U).^2,Target);
    fprintf('%d\t %f\n',count,correlate);
end
imagesc(0:nscreen*stepxy,0:nscreen*stepxy,abs(U).^2)
saveas(gcf,'Wujing_Holo.jpg')
figure;
imagesc(0:nscreen*stepxy,0:nscreen*stepxy,angle(B));colorbar
saveas(gcf,'Wujing_Phase.jpg')
```

RSDiff.m

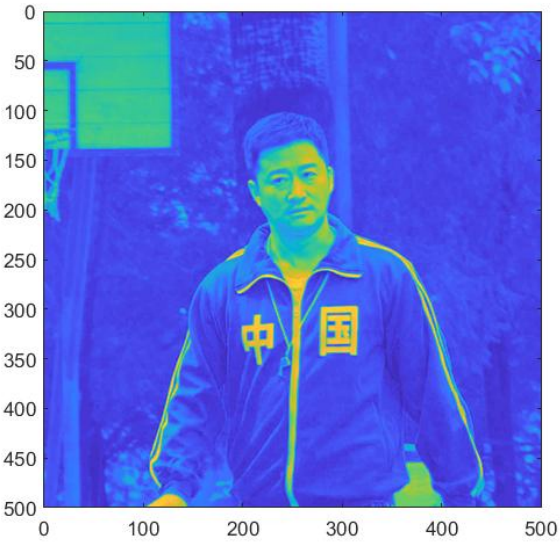
```
function Out = RSDiff(z,s,k,object)
%function RSDiff uses the formula in article2006 to calculate image
eta=s;x=s;y=s;
n=length(s);
U=[object,zeros(n,n-1);zeros(n-1,n),zeros(n-1,n-1)];
X=[x(1)-s(n+1-(1:(n-1))),x((n:2*n-1)-n+1)-s(1)];
Y=[y(1)-eta(n+1-(1:(n-1))),y((n:2*n-1)-n+1)-eta(1)];
[XX,YY]=meshgrid(X,Y);
r=sqrt(XX.^2+YY.^2+z^2);
G=1/(2*pi)*exp(1i*k*r)./r*z./r.*(1./r-1i*k);
S=ifft2(fft2(U).*fft2(G));
Out=S(n:end,n:end);
end
```

ErrorCriterion.m

```
function correlate = ErrorCriterion(Target,Figure)
%Returns the correlation coefficient of Target and Figure
Target1=Target/max(max(Target));
Figure1=Figure/max(max(Figure));
correlate=min(min(corrcoef(Target1,Figure1)));
end
```

对于像素$2\mu m \times 2\mu m$，只要将程序中的stepxy=0.3改为stepxy=2即可

目标图像，文件 'Wujing.jpg' 的Red分量



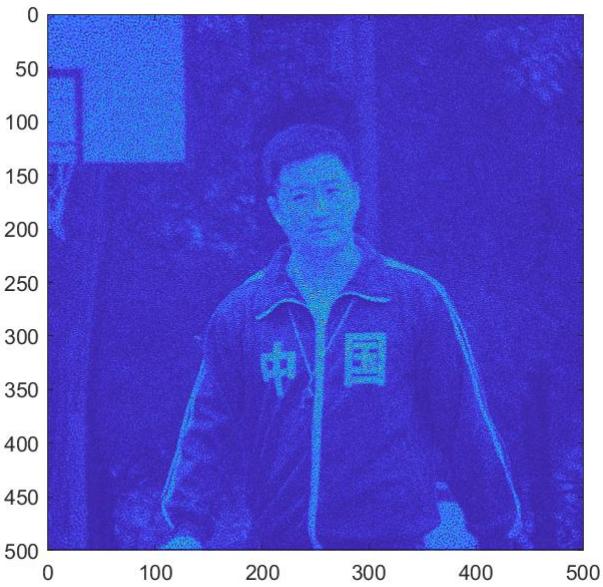
运行结果：

当stepxy=0.3时

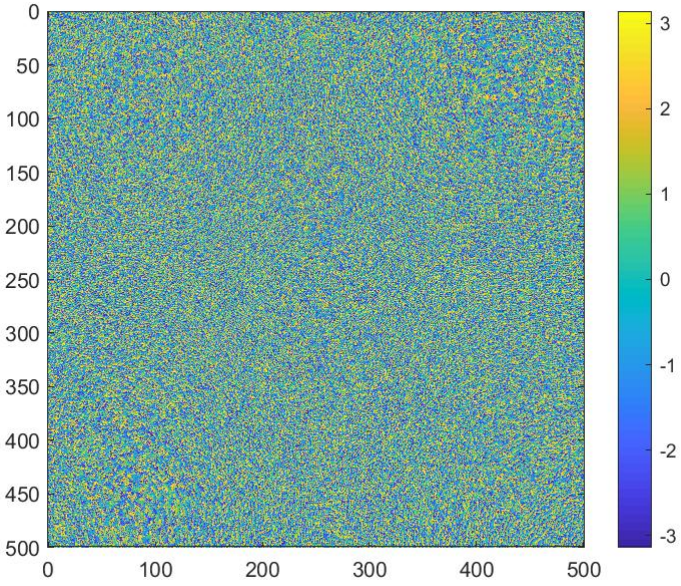
输出：

```
n    correlate
1    0.010619
2    0.682883
3    0.730488
4    0.753342
5    0.767273
6    0.776642
7    0.783485
8    0.788775
9    0.793023
10   0.796542
11   0.799516
12   0.802057
```

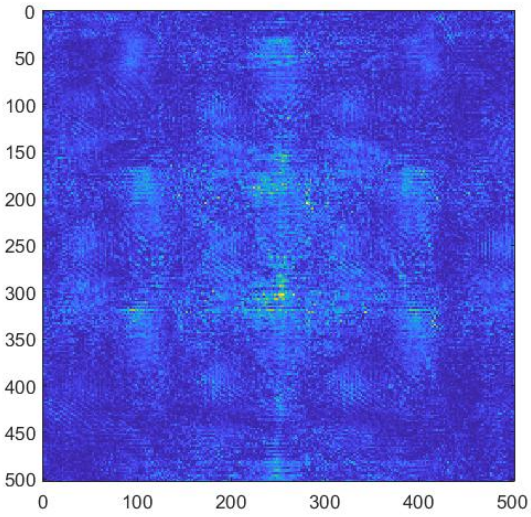
得到的光强分布



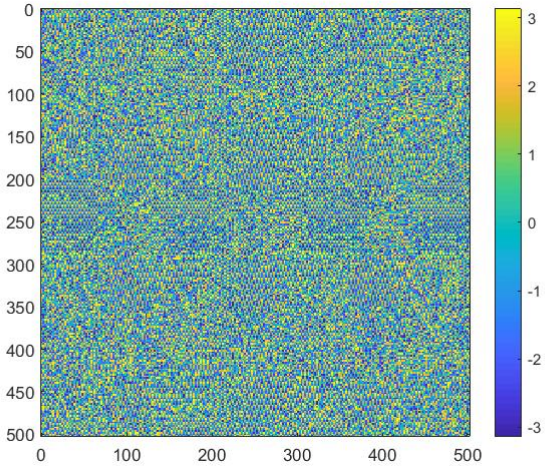
位相分布：



当stepxy=2时，无法得到预期的图像，correlate始终在0.2-0.25之间，无法趋近于1，甚至在几次循环后开始下降，循环200次后退出，得到的图像：

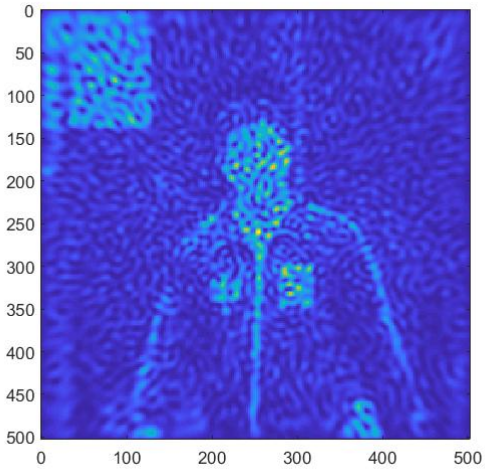


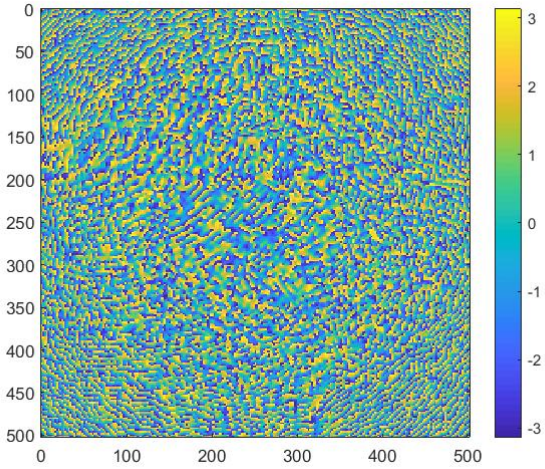
相位：



分析原因：

可能是因为取的step过大导致stepxy大于大于波长。在计算全息的问题中，调制光场的频率不能大于载波的频率，就例如在第二次作业中，我被建议使用小于半个波长的网格。假如在stepxy=2时取波长lambda=4，使得格点小于波长，则可以得到收敛的correlate和较为正常的全息图，如下图所示。与stepxy=0.3相比，分辨率下降，但是仍然可以得到图像的大致形状。




0
50
100
150
200
250
300
350
400
450
500
0
100
200
300
400
500
3
2
1
0
-1
-2
-3