Simulación del proceso de pticografía de Fourier

Contents

Emmanuel Isaac Juárez Caballero- Facultad de Física Universidad Veracruzana

Archivos Originales

Para la amplitud se ocuparon las imagenes siguientes, primero siendo tratadas para que estas tuvieran un aspecto monocromático y estuvieran de acuerdo a las dimensiones del código.

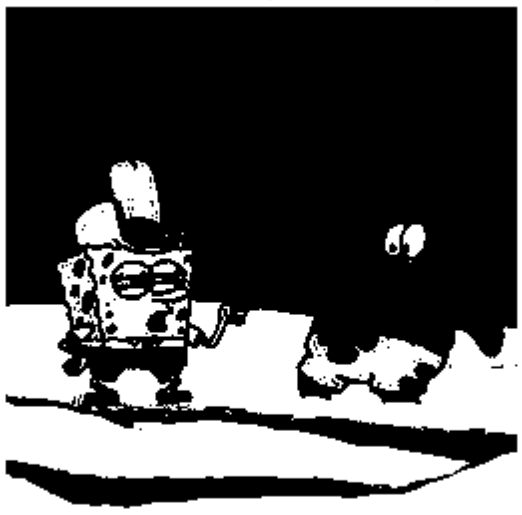
```
%%Fotos originales
sucio=imread('sucio.png');
flor=imread('flor-azul.png');
imshow(sucio);
```

```
imshow(flor);
```

Simulación del objeto complejo

```
%objectAmplitud=double(imread('cameraman.tif')); %CARGA UNA IMAGEN
objectAmplitud=double(imread('sucio.tiff')); %CARGA UNA IMAGEN
%objectAmplitud=pi*imresize(phase,[256 256])./max(max(objectAmplitud));
phase=double(imread('flor-azul.tiff')); %CARGA OTRA IMAGEN
%phase=double(imread('city.jpg'));
phase=pi*imresize(phase,[256 256])./max(max(phase)); %HACE DE LAS MISMAS DIMENSIONES ALA SEGUNDA IMAGEN
object=objectAmplitud.*exp(1i.* phase); %CREA EL OBJETO QUE ES LA AMPLITUD POR UNA EXPONENCIAL POR UNA FACE
imshow(abs(object),[]);title('Entrada del objeto complejo') %MUESTRA EN PANTALLA AL OBJETO COMPLEJO
```

Entrada del objeto complejo



Montaje de parámtros para el sistema de imagenes coherentes

```
waveLength = 0.63e-6;
k0 = 2*pi/waveLength;
spsize = 2.75e-6; % Medida del espaciamiento de pixeles en la
psize = spsize / 4; % Tamaño final de los pixeles en la reconstrucción
NA = 0.08;
```

Creando los vectores de onda para la reconstrucción

```
arraysize = 15;
xlocation = zeros(1,arraysize^2);
ylocation = zeros(1,arraysize^2);
LEDgap = 4; % Distancia en milímetros de los leds
LEDheight = 90; % Distancia en mm del arreglo de leds a la muestra

for i=1:arraysize % from top left to bottom right
    xlocation(1,1+arraysize*(i-1):15+arraysize*(i-1)) = (-(arraysize-1)/2:1:(arraysize-1)/2)*LEDgap;
    ylocation(1,1+arraysize*(i-1):15+arraysize*(i-1)) = ((arraysize-1)/2-(i-1))*LEDgap;
end;
kx_relative = -sin(atan(xlocation/LEDheight));
ky_relative = -sin(atan(ylocation/LEDheight));
```

Generando el filtrado de imagen

```
[m,n] = size(object); % Tamaño de la resolución del objeto
m1 = m/(spsize/psize);n1 = n/(spsize/psize); % Tamaño final de la imagen
imSeqLowRes = zeros(m1, n1, arraysize^2); % La imagen final de baja resolución
kx = k0 * kx_relative;
ky = k0 * ky_relative;
dkx = 2*pi/(psize*n);
dky = 2*pi/(psize*m);
cutoffFrequency = NA * k0;
kmax = pi/spsize;
[kxm kym] = meshgrid(-kmax:kmax/((n1-1)/2):kmax,-kmax:kmax/((n1-1)/2):kmax);
CTF = ((kxm.^2+kym.^2)<cutoffFrequency^2); % pupil function circ(kmax); no aberration
% z = 10e-6; kzm = sqrt(k0^2-kxm.^2-kym.^2);
% pupil = exp(1i.*z.*real(kzm)).*exp(-abs(z).*abs(imag(kzm)));
% aberratedCTF = pupil.*CTF;
```

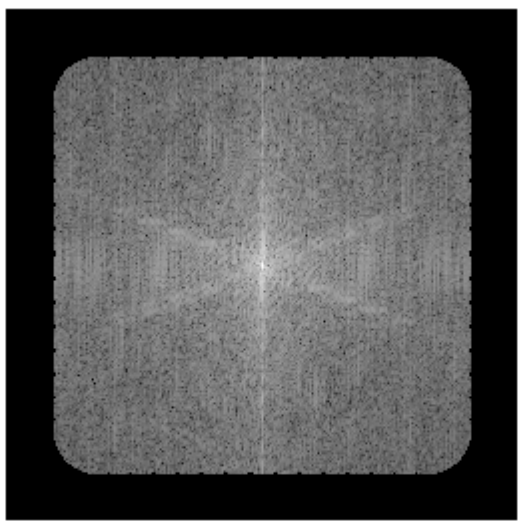
```
close all;clc;
objectFT = fftshift(fft2(object));
for tt =1:arraysize^2
    kxc = round((n+1)/2+kx(1,tt)/dkx);
    kyc = round((m+1)/2+ky(1,tt)/dky);
    kyl=round(kyc-(m1-1)/2);kyh=round(kyc+(m1-1)/2);
    kxl=round(kxc-(n1-1)/2);kxh=round(kxc+(n1-1)/2);
    imSeqLowFT = (m1/m)^2 * objectFT(kyl:kyh,kxl:kxh).*CTF;
    imSeqLowRes(:,:,tt) = abs(ifft2(ifftshift(imSeqLowFT)));
end;
figure;imshow(imSeqLowRes(:,:,1),[]);title('Prueba')
```

Prueba

Reconstruyendo la imagen de alta resolución

```
seq = gseq(arraysize);
objectRecover = ones(m,n);
objectRecoverFT = fftshift(fft2(objectRecover));
loop = 5;
for tt=1:loop
    for i3=1:arraysize^2
        i2=seq(i3);
        kxc = round((n+1)/2+kx(1,i2)/dkx);
        kyc = round((m+1)/2+ky(1,i2)/dky);
        kyl=round(kyc-(m1-1)/2);kyh=round(kyc+(m1-1)/2);
        kxl=round(kxc-(n1-1)/2);kxh=round(kxc+(n1-1)/2);
        lowResFT = (m1/m)^2 * objectRecoverFT(kyl:kyh,kxl:kxh).*CTF;
        im_lowRes = ifft2(ifftshift(lowResFT));
        im_lowRes = (m/m1)^2 * imSeqLowRes(:,:,i2).*exp(1i.*angle(im_lowRes));
        lowResFT=fftshift(fft2(im_lowRes)).*CTF;
        objectRecoverFT(kyl:kyh,kxl:kxh)=(1-CTF).*objectRecoverFT(kyl:kyh,kxl:kxh) + lowResFT;
     end;
end;
objectRecover=ifft2(ifftshift(objectRecoverFT));
imshow(abs(objectRecover),[]);
figure;imshow(angle(objectRecover),[]);
figure;imshow(log(objectRecoverFT),[]);
```

```
Warning: Displaying real part of complex input.
```



Published with MATLAB® R2015a