Peaks, Valleys, and Ridges

Dr. Daniel B. Rowe
Professor of Computational Statistics
Department of Mathematical and Statistical Sciences
Marquette University

Outline

Introduction

Function Critical Points

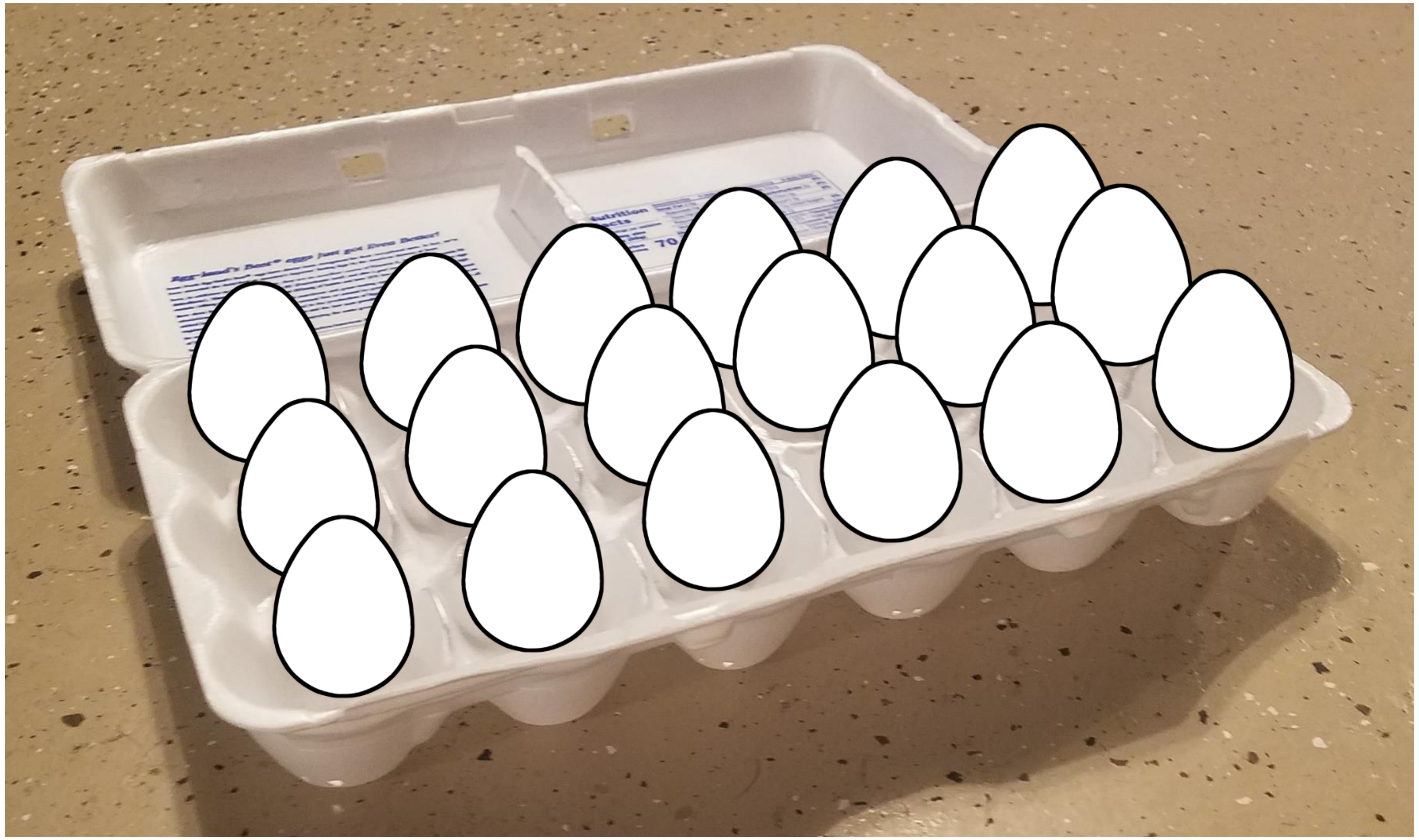
Image Critical Points

Discussion

Homework

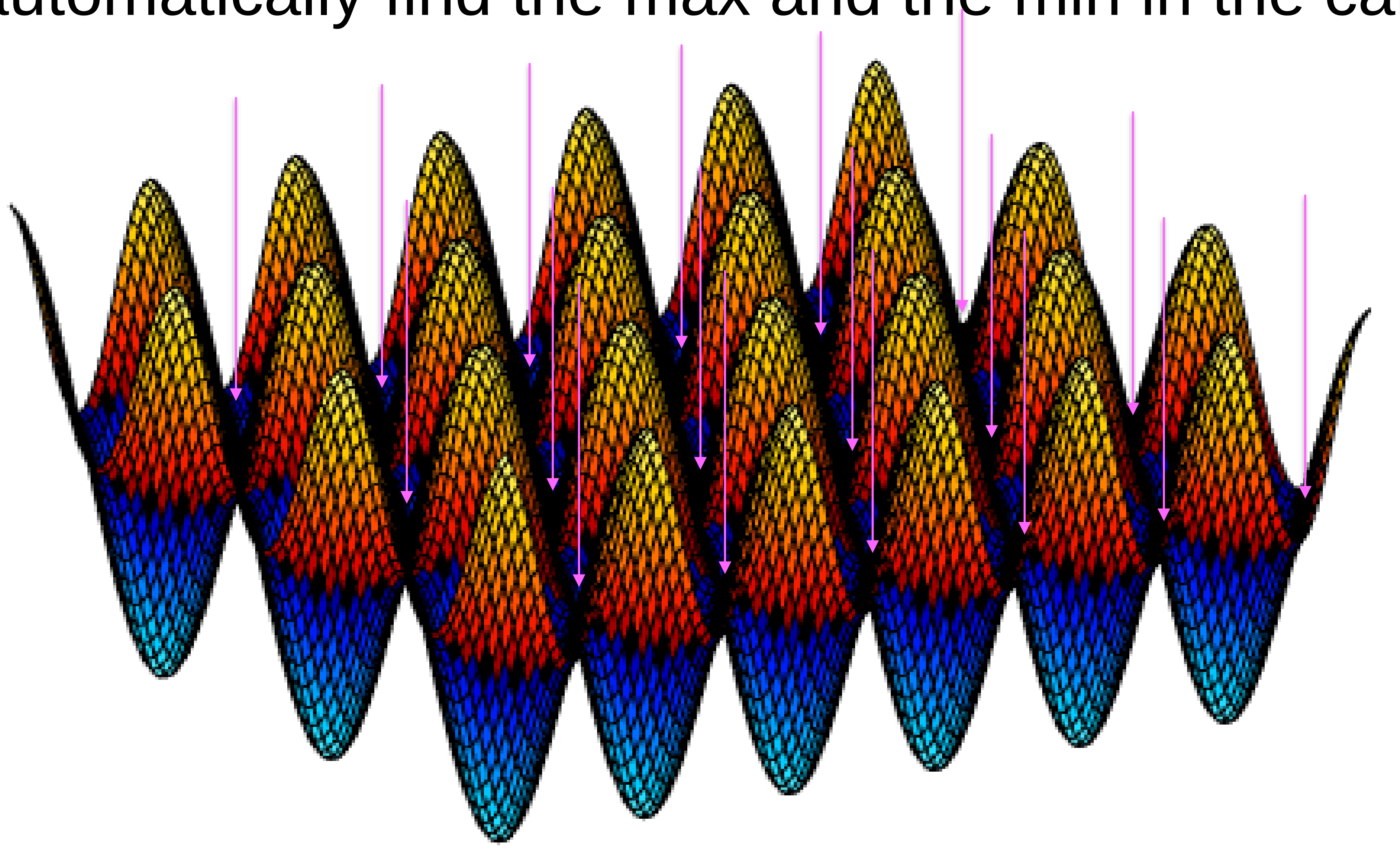
Introduction

A recent Marquette University graduate just landed their first job to develop an image recognition system to place eggs in an egg carton.



Introduction

The recent graduate was given a theoretical model for the egg carton and needs to automatically find the max and the min in the carton images.



Introduction

The new graduate decides to develop their computer vision system with the theoretical model for the egg carton.

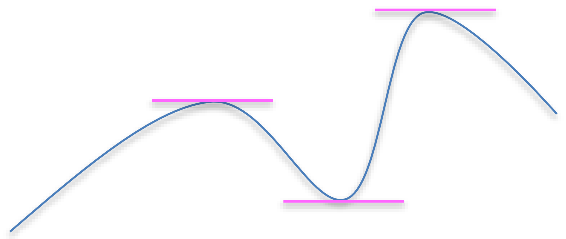
The new graduate wants to develop a general system for finding local maxima and minima in the images because there will be egg cartons of various shapes holding different numbers of eggs and it may be used for other purposes.

The graduate recalls being required to take Calculus 3 for their major.

In Calculus 3 was theoretical techniques for finding local max and min of continuous functions $f(x,y)$.

Function Critical Points

In one dimension, we learned that if we have a function $f(x)$, that we can find extrema such as maxima and minima by finding the locations where the slope is zero.



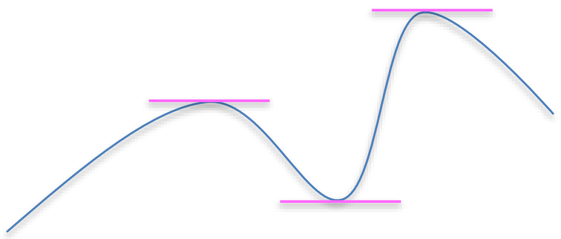
This is performed by taking the derivative and setting it to zero.

$$\frac{d}{dx} f(x) = 0$$

This tells us where the slope is zero (flat spots) called critical points.

Function Critical Points

We determine whether the critical point is a local maxima or minima.



If the second derivative at the critical point is negative,

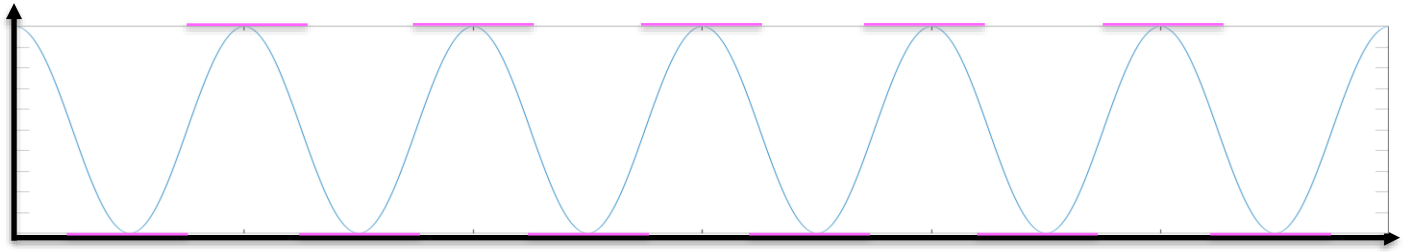
$$\frac{d^2}{dx^2} f(x) < 0$$ then it is a maxima

if it is positive

$$\frac{d^2}{dx^2} f(x) > 0$$ then it is a minima.

If it is zero then the second derivative doesn't tell us anything.

Function Critical Points



Example:

$$f(x) = \cos^2(2\pi x)$$

$$f_x = \frac{d}{dx} f(x)$$

$$f_x(x) = -4\pi\cos(2\pi x)\sin(2\pi x) = -2\pi\sin(4\pi x)$$

$$0 = -2\pi\sin(4\pi x)$$

$$f_{xx} = \frac{d}{dx^2} f(x)$$

$$x_{crit} = \pm\tfrac{n}{4}, \quad n \in \square$$

$$f_{xx}(x) = -8\pi^2\cos(4\pi x)$$

$$f_{xx}(\tfrac{n}{4}) = -8\pi^2\cos(4\pi\tfrac{n}{4})$$

$-8\pi^2\cos(4\pi\tfrac{n}{4}) < 0$ for n even (maxima) and $-8\pi^2\cos(4\pi\tfrac{n}{4}) > 0$ for n odd (minima).

Function Critical Points

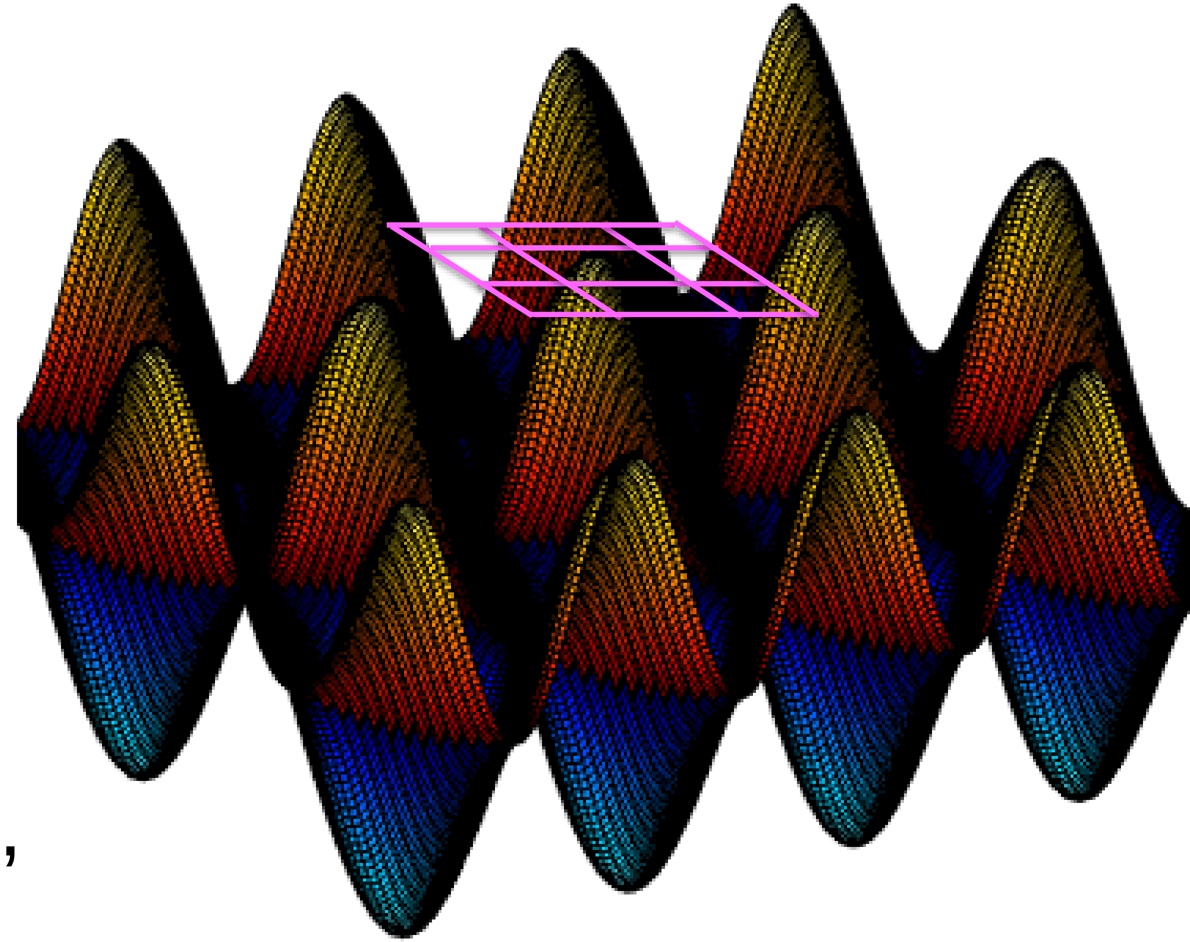
In two dimensions, a similar process is followed.

We can look to see when the slope in the x and y directions is zero.

Take the derivatives and setting to zero.

$$\frac{d}{dx} f(x, y) = 0$$

$$\frac{d}{dy} f(x, y) = 0$$

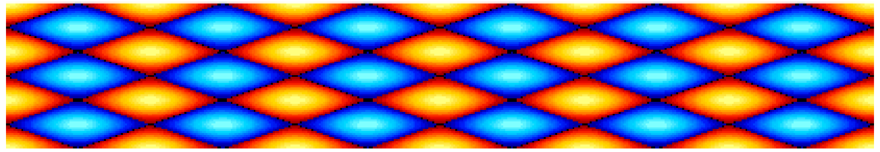


This tells us where the slope is zero (flat spots), called critical points.

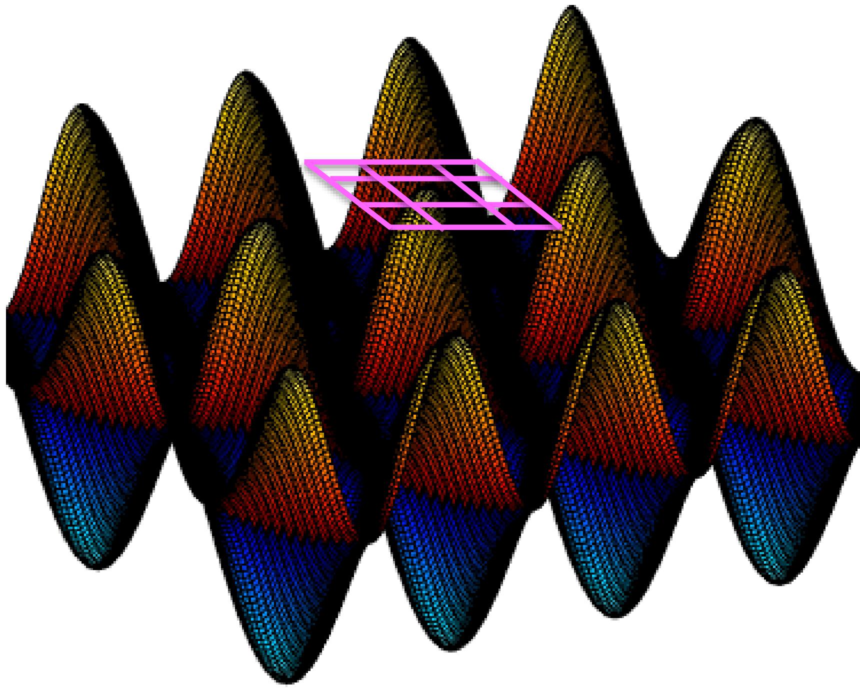
Function Critical Points

We determine if the critical point is a local maxima, minima, or saddle point.

To determine what type of point we have, we need second derivatives.



$$f_{xx} = \frac{d^2}{dx^2} f(x,y) \qquad f_{yy} = \frac{d^2}{dy^2} f(x,y) \qquad f_{xy} = \frac{d^2}{dxdy} f(x,y)$$



And the determinant of the Hessian matrix.

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix} \qquad D = |H| = f_{xx} f_{yy} - f_{xy}^2$$

Function Critical Points

Second Derivative Test for Functions of Two Variables

Suppose (x_0,y_0) is a point where $f_x(x_0,y_0)=0$ and $f_y(x_0,y_0)=0$.

Let

$$D = f_{xx}(x_0,y_0)f_{yy}(x_0,y_0) - (f_{xy}(x_0,y_0))^2$$

- If $D>0$ and $f_{xx}(x_0,y_0)>0$, then f has a local minimum at (x_0,y_0).
- If $D>0$ and $f_{xx}(x_0,y_0)<0$, then f has a local maximum at (x_0,y_0).
- If $D<0$, then f has a saddle point at (x_0,y_0).
- If $D=0$, anything can happen: f can have a local maximum, or a local minimum, or a saddle point, or none of these at (x_0,y_0).

Function Critical Points

- $D>0$ & $f_{xx}(x_0,y_0)>0$, then local min at (x_0,y_0).
- $D>0$ & $f_{xx}(x_0,y_0)<0$, then local max at (x_0,y_0).
- $D<0$, then f has a saddle point at (x_0,y_0).
- If $D=0$, anything can happen at (x_0,y_0).

Example:

$$f(x,y)=\cos^2(2\pi x)+\cos^2(2\pi y)$$

$$f_x(x,y)=-4\pi\cos(2\pi x)\sin(2\pi x)=-2\pi\sin(4\pi x)$$

$$f_y(x,y)=-4\pi\cos(2\pi y)\sin(2\pi y)=-2\pi\sin(4\pi y)$$

$$0=-2\pi\sin(4\pi x) \qquad x_{crit}=\pm\tfrac{n}{4},\ n\in\mathbb{Z}$$

$$0=-2\pi\sin(4\pi y) \qquad y_{crit}=\pm\tfrac{m}{4},\ m\in\mathbb{Z}$$

$$f_{xx}(x,y)=-8\pi^2\cos(4\pi x)$$

$$f_{yy}(x,y)=-8\pi^2\cos(4\pi y)$$

$$f_{xy}(x,y)=0$$

$$D=(-8\pi^2\cos(4\pi x))(-8\pi^2\cos(4\pi y))-0$$

$$D=64\pi^4\cos(4\pi x)\cos(4\pi y)$$

And we can classify the points.

x	y	f_{xx}	D	f
0	0	−	+	2
$\frac{1}{4}$	$\frac{1}{4}$	+	+	0
$\frac{1}{4}$	$\frac{2}{4}$	+	−	1
$\frac{2}{4}$	$\frac{1}{4}$	−	−	1

Image Critical Points

In the same way that we can find critical points for a function, we can find critical points in an image.

We can calculate all of the necessary discrete derivatives using kernels.

1	0	-1
1	0	-1
1	0	-1

/6

$$f_x = \frac{df(x,y)}{dx}$$

1	1	1
0	0	0
-1	-1	-1

/6

$$f_y = \frac{df(x,y)}{dy}$$

1	-2	1
2	-4	2
1	-2	1

/16

$$f_{xx} = \frac{d^2 f(x,y)}{dx^2}$$

1	2	1
-2	-4	-2
1	2	1

/16

$$f_{yy} = \frac{d^2 f(x,y)}{dy^2}$$

1	0	-1
0	0	0
-1	0	1

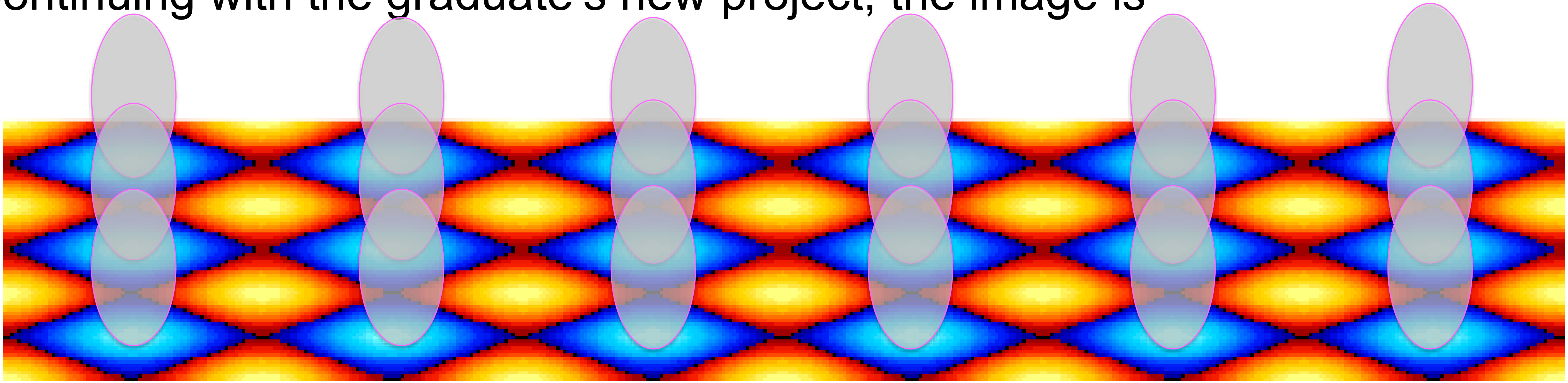
/16

$$f_{xy} = \frac{d^2 f(x,y)}{dxdy}$$

$$D = |H| = f_{xx} f_{yy} - f_{xy}^2$$

Image Critical Points

Continuing with the graduate's new project, the image is

$I=$ 

and we need to find the minima to place the eggs.

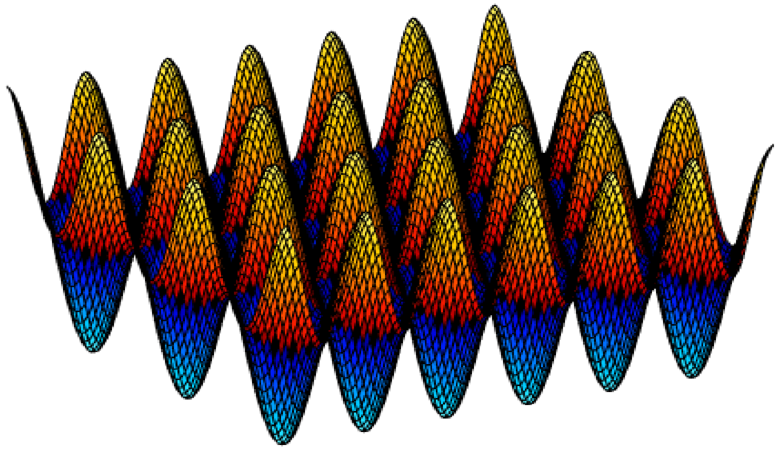
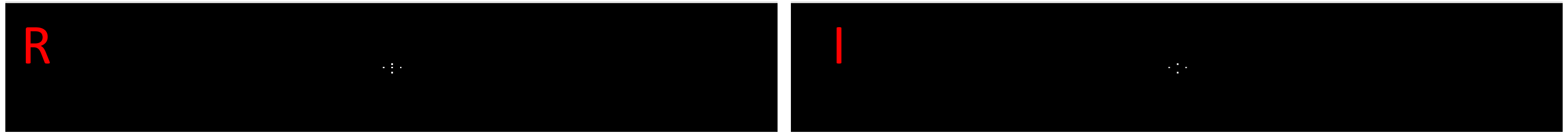


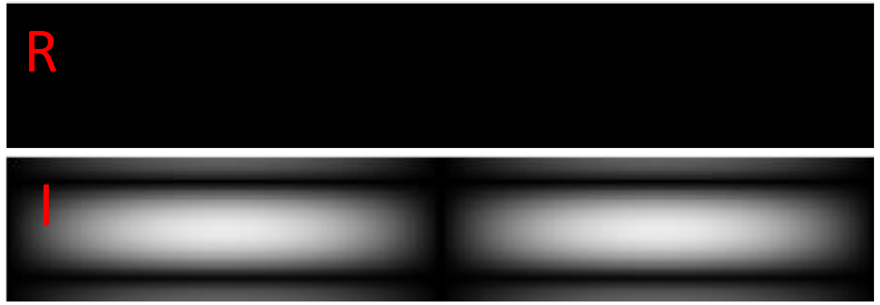
Image Critical Points

$\mathcal{F}\{I\}=$

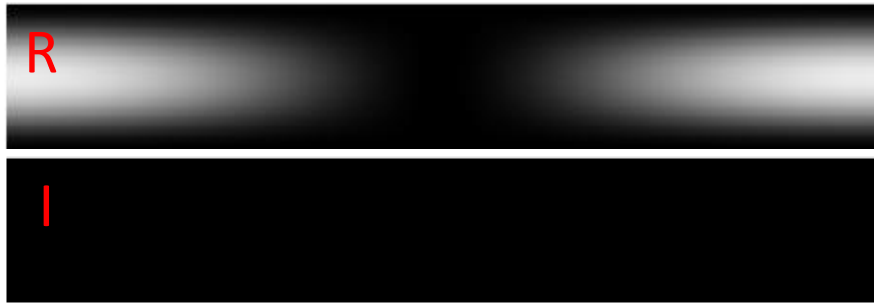


Continuing with the graduate's new project, the DFTs of the centered kernels

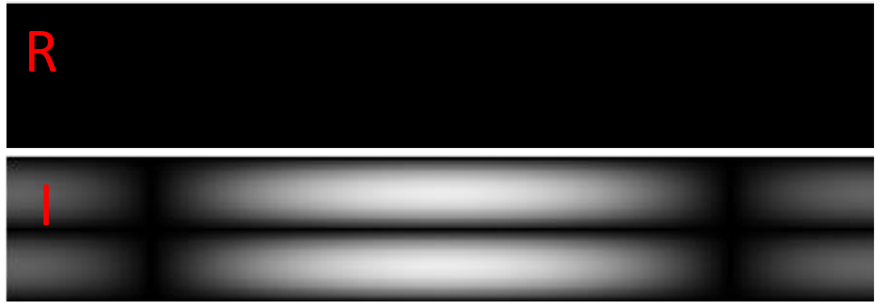
$$k_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} /6$$



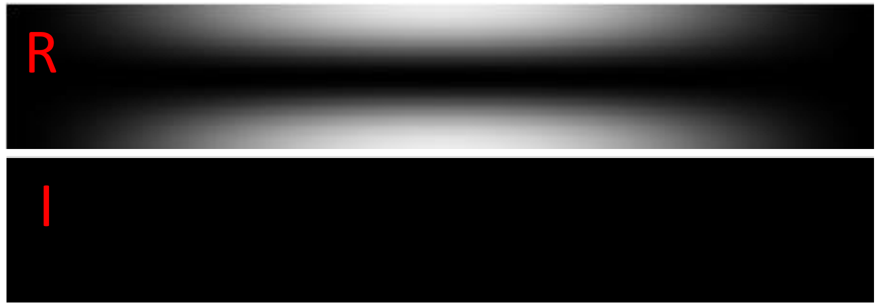
$$k_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} /16$$



$$k_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} /6$$



$$k_{yy} = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{bmatrix} /16$$



$$k_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} /16$$

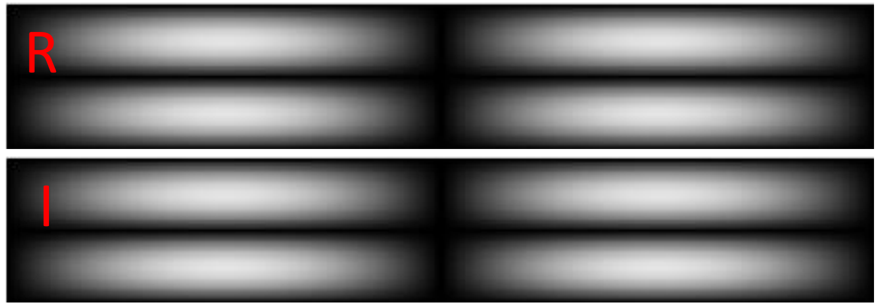
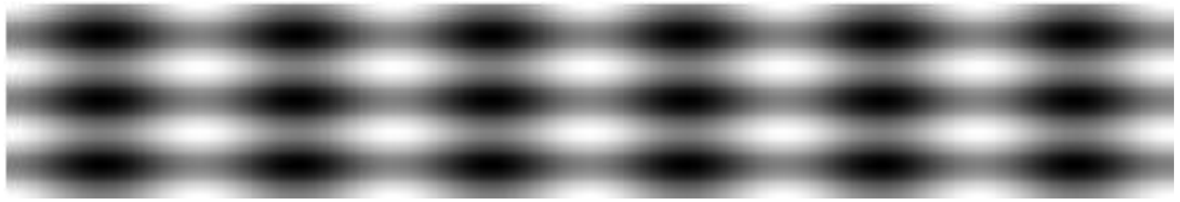


Image Critical Points

$I=$ 

Continuing with the graduate's new project, the DFTs of kernel times image

Derivative images using kernels.

$\mathcal{F}\{K\}\mathcal{F}\{I\}$

$I_x(x,y)$ $\mathcal{F}^{-1}$

$I_y(x,y)$ $\mathcal{F}^{-1}$

$I_{xx}(x,y)$ $\mathcal{F}^{-1}$

$I_{yy}(x,y)$ $\mathcal{F}^{-1}$

$I_{xy}(x,y)$ $\mathcal{F}^{-1}$

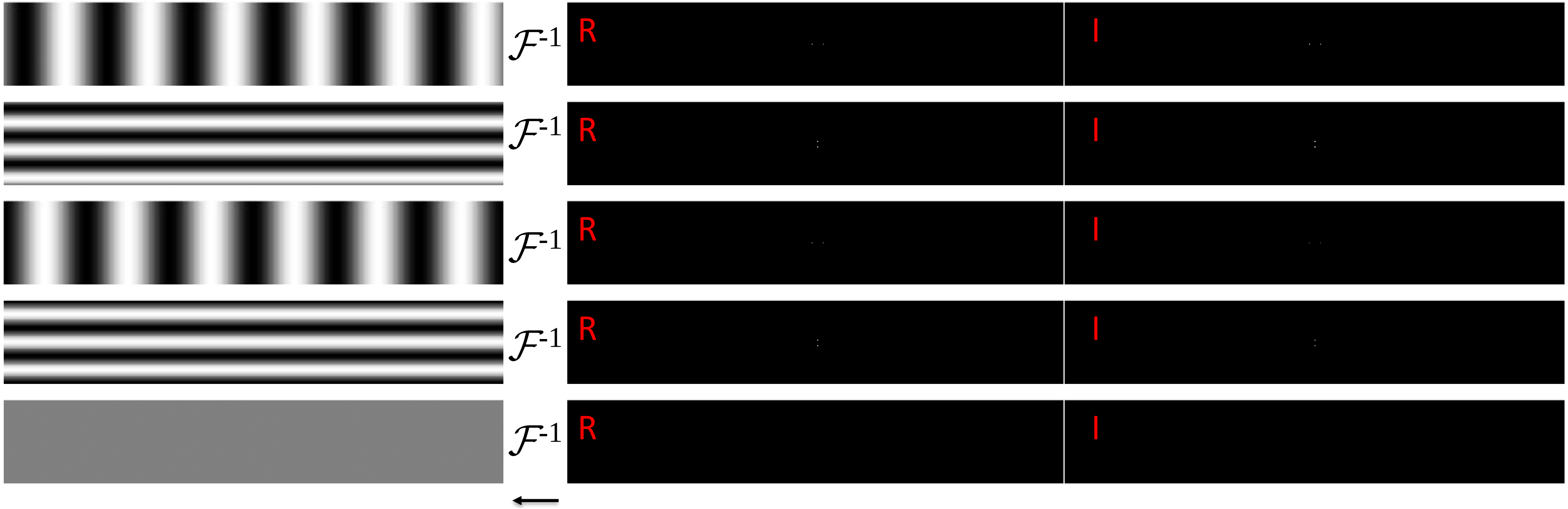
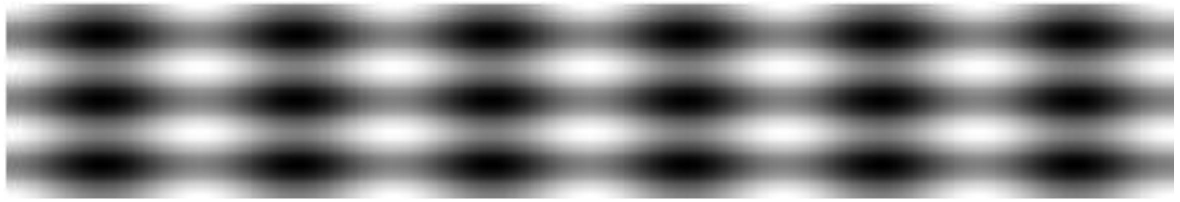


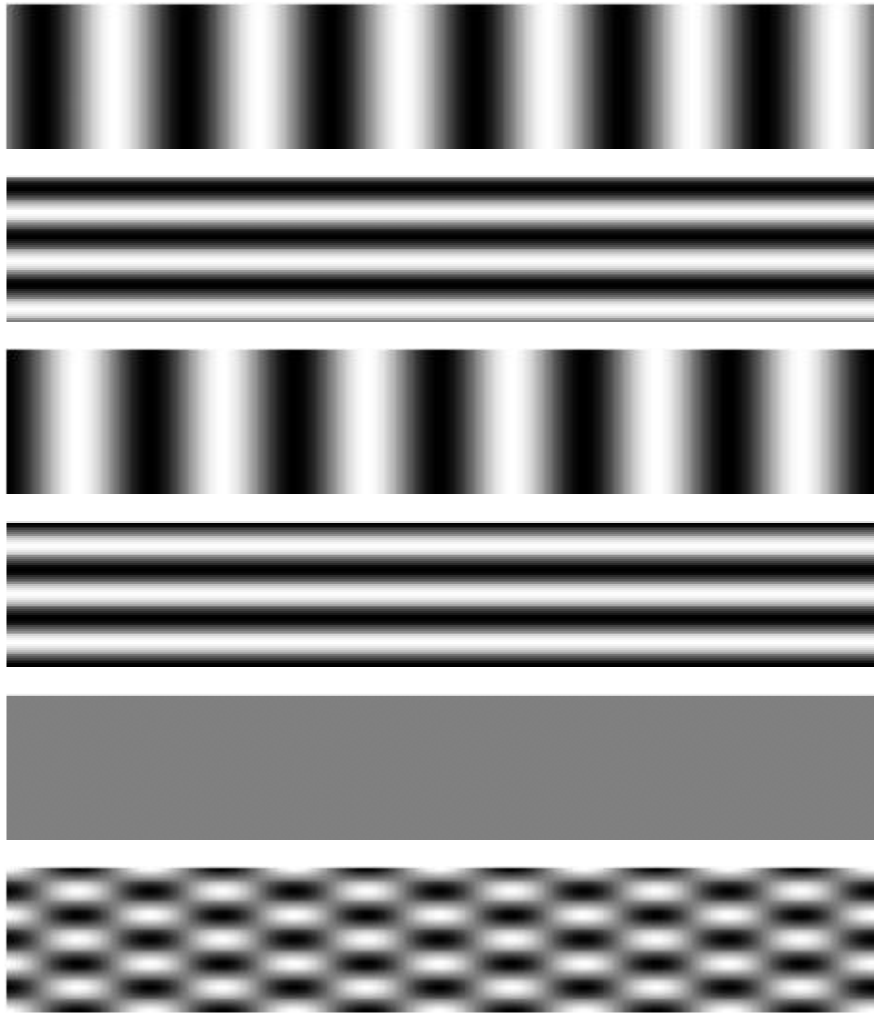
Image Critical Points

$I=$ 

Continuing with the graduate's new project, $f(x,y)=\cos^2(2\pi x)+\cos^2(2\pi y)$

Derivative images using kernels.

$I_x(x,y)$, $I_y(x,y)$, $I_{xx}(x,y)$, $I_{yy}(x,y)$, $I_{xy}(x,y)$, $I_D(x,y)$



compare

Derivative images using Calculus.

$f_x(x,y)$, $f_y(x,y)$, $f_{xx}(x,y)$, $f_{yy}(x,y)$, $f_{xy}(x,y)$, $f_D(x,y)$

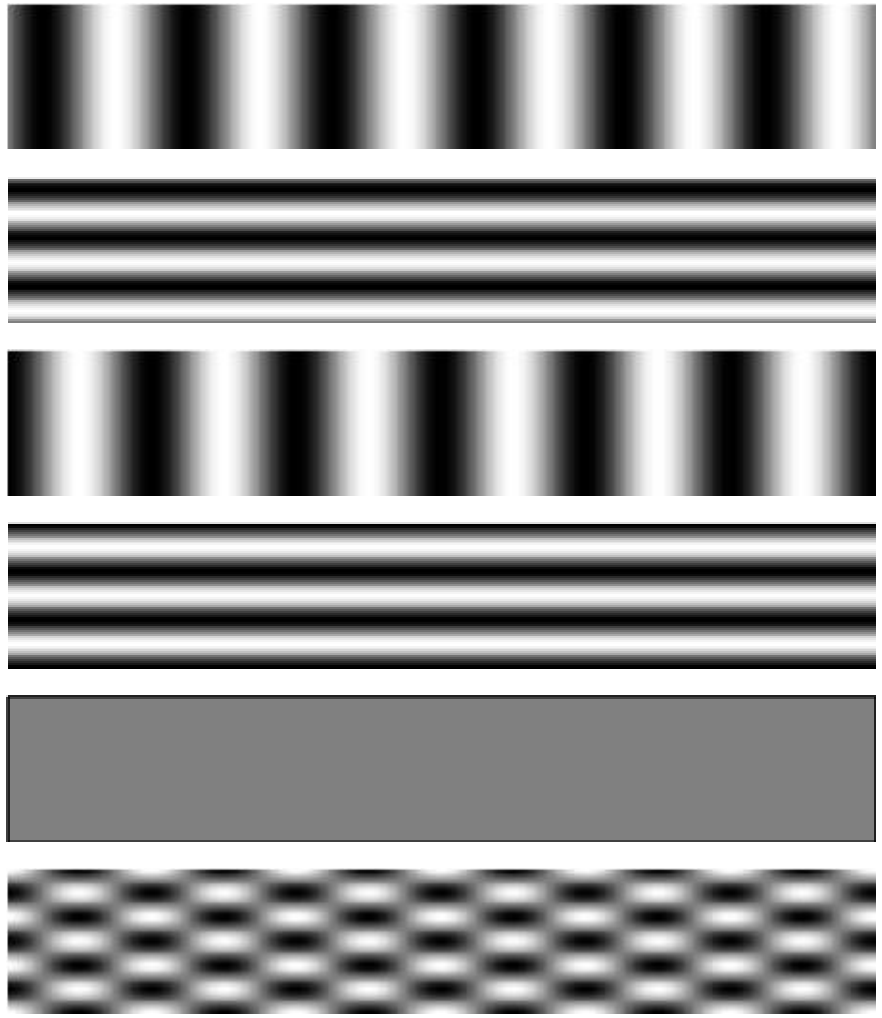


Image Critical Points

Continuing with the graduate's new project,

- $D>0$ & $f_{xx}(x_0,y_0)>0$, then local min at (x_0,y_0).
- $D>0$ & $f_{xx}(x_0,y_0)<0$, then local max at (x_0,y_0).
- $D<0$, then f has a saddle point at (x_0,y_0).
- If $D=0$, anything can happen at (x_0,y_0).

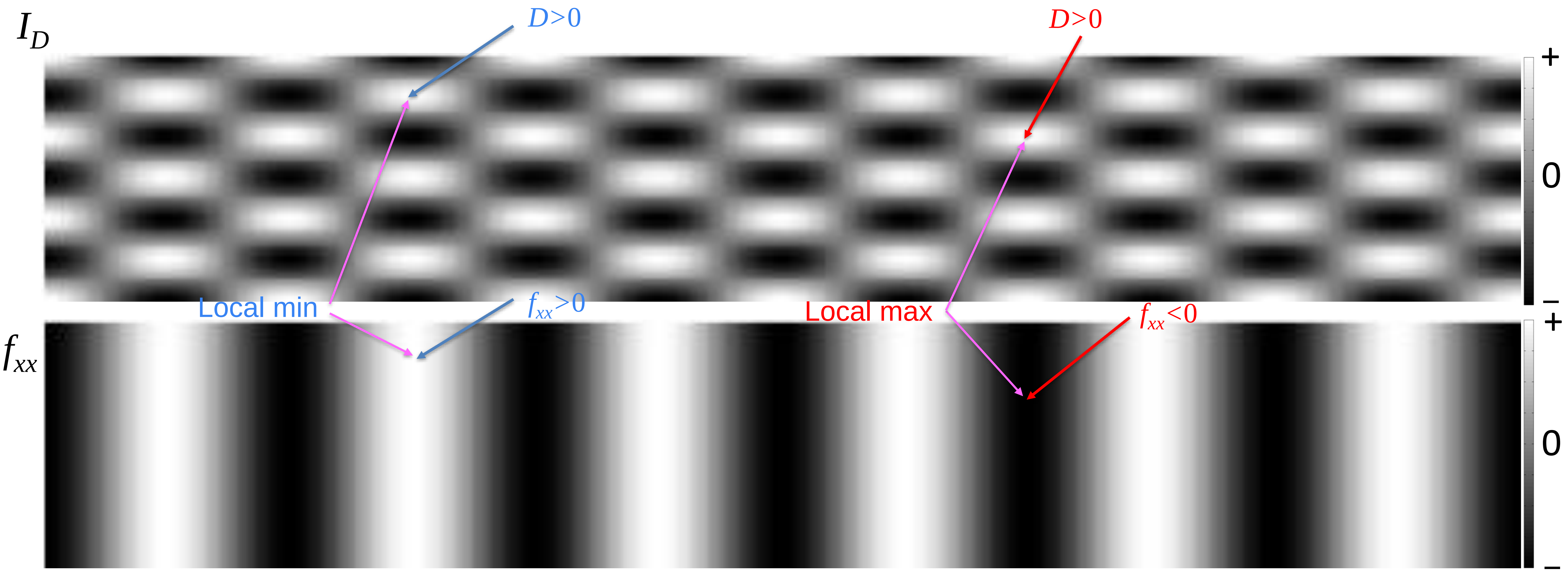


Image Critical Points

- $D>0$ & $f_{xx}(x_0,y_0)>0$, then local min at (x_0,y_0).
- $D>0$ & $f_{xx}(x_0,y_0)<0$, then local max at (x_0,y_0).
- $D<0$, then f has a saddle point at (x_0,y_0).
- If $D=0$, anything can happen at (x_0,y_0).

And this is where we place the eggs!

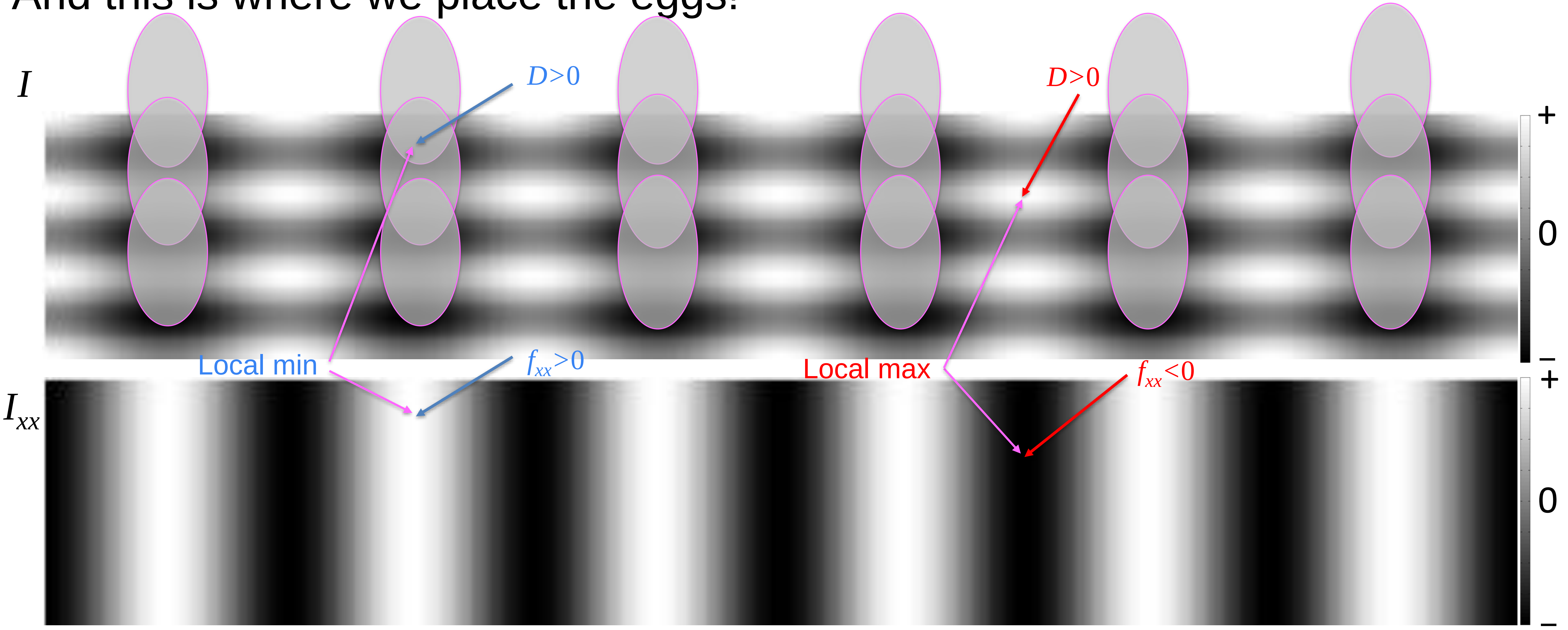


Image Critical Points

```
load myposnegmapblk.txt
load myposmapblk.txt
% 1 D
dx=1;
x=(dx:dx:3*180);
nx=length(x);
y=cos(2*pi*x/180).^2;
figure;
plot(x/180,y)
%2 D
dy=1;
y=(dy:dy:90);
ny=length(y);
scale=128;
% function image
[X,Y]=meshgrid(x,y);
f=scale*(cos(2*pi*X/180).^2+cos(2*pi*Y/60).^2);
figure;
imagesc(f)
colormap(gray), axis off, axis image
figure;
imagesc(f)
colormap(myposnegmapblk), axis off, axis image
```

```
figure;
surf(X,Y,f)
az=-30; el=30; view(az,el)
colormap(myposnegmapblk), axis off
```

Image Critical Points

```
% x derivative image
fx=-scale*2*pi/180*sin(4*pi*X/180);
figure;
imagesc(fx,[-scale*2*pi/180,scale*2*pi/180])
colormap(gray), axis off, axis image
% y derivative image
fy=-scale*2*pi/60*sin(4*pi*Y/60);
figure;
imagesc(fy,[-scale*2*pi/60,scale*2*pi/60])
colormap(gray), axis off, axis image
% xx derivative image
fxx=-2*pi^2/8100/4*scale*cos(4*pi*X/180);
figure;
imagesc(fxx,[-2*pi^2/8100/4*scale,2*pi^2/8100/4*scale])
colormap(gray), axis off, axis image
% yy derivative image
fyy=-2*pi^2/900/4*scale*cos(4*pi*Y/60);
figure;
imagesc(fyy,[-2*pi^2/900/4*scale,2*pi^2/900/4*scale])
colormap(gray), axis off, axis image
```

```
% xy derivative image
fxy=zeros(ny,nx);
figure;
imagesc(fxy,[-1/10,1/10])
colormap(gray), axis off, axis image
% Determinant
fD=fxx.*fyy-(fxy).^2;
figure;
imagesc(fD,[-2*pi^2/8100/4*scale*2*pi^2/900/4*scale,...
    2*pi^2/8100/4*scale*2*pi^2/900/4*scale])
colormap(gray), axis off, axis image
```

Image Critical Points

```
% derivatives using kernels
I=f;
kernelx=[ones(3,1),zeros(3,1),-ones(3,1)]/6;
kernely=[ones(1,3);zeros(1,3);-ones(1,3)]/6;
kernelxx=[[1;2;1],[-2;-4;-2],[1;2;1]]/16;
kernelyy=[[1,2,1];[-2,-4,-2];[1,2,1]]/16;
kernelxy=[1,0,-1;0,0,0;-1,0,1]/16;

kernelfillx=zeros(ny,nx);  kernelfilly=zeros(ny,nx);
kernelfillxx=zeros(ny,nx); kernelfillyy=zeros(ny,nx);
kernelfillxy=zeros(ny,nx); [ky,kx]=size(kernelx);
if (mod(ky,2)==1)
    kernelfillx(ny/2-(ky-1)/2+1:ny/2+(ky-1)/2+1,nx/2-(kx-1)/2+1:nx/2+(kx-1)/2+1)=kernelx;
    kernelfilly(ny/2-(ky-1)/2+1:ny/2+(ky-1)/2+1,nx/2-(kx-1)/2+1:nx/2+(kx-1)/2+1)=kernely;
    kernelfillxx(ny/2-(ky-1)/2+1:ny/2+(ky-1)/2+1,nx/2-(kx-1)/2+1:nx/2+(kx-1)/2+1)=kernelxx;
    kernelfillyy(ny/2-(ky-1)/2+1:ny/2+(ky-1)/2+1,nx/2-(kx-1)/2+1:nx/2+(kx-1)/2+1)=kernelyy;
    kernelfillxy(ny/2-(ky-1)/2+1:ny/2+(ky-1)/2+1,nx/2-(kx-1)/2+1:nx/2+(kx-1)/2+1)=kernelxy;
elseif (mod(ky,2)==0)
    kernelfillx(ny/2-ky/2+1:ny/2+ky/2,nx/2-kx/2+1:nx/2+kx/2)=kernelx;
    kernelfilly(ny/2-ky/2+1:ny/2+ky/2,nx/2-kx/2+1:nx/2+kx/2)=kernely;
    kernelfillxx(ny/2-ky/2+1:ny/2+ky/2,nx/2-kx/2+1:nx/2+kx/2)=kernelxx;
    kernelfillyy(ny/2-ky/2+1:ny/2+ky/2,nx/2-kx/2+1:nx/2+kx/2)=kernelyy;
    kernelfillxy(ny/2-ky/2+1:ny/2+ky/2,nx/2-kx/2+1:nx/2+kx/2)=kernelxy;
end
```

Image Critical Points

```
ftI      =fftshift(fft2(fftshift(I          )));
figure;
imagesc(log(abs(real(ftI))+1),[0,15])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftI))+1),[0,15])
colormap(gray), axis off, axis image

ftkernx =fftshift(fft2(fftshift(kernelfillx )));
figure;
imagesc(log(abs(real(ftkernx))+1),[0,1/10])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernx))+1),[0,3/4])
colormap(gray), axis off, axis image

ftkerny =fftshift(fft2(fftshift(kernelfilly )));
figure;
imagesc(log(abs(real(ftkerny))+1),[0,1/10])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkerny))+1),[0,3/4])
colormap(gray), axis off, axis image
```

```
ftkernxx=fftshift(fft2(fftshift(kernelfillxx)));
figure;
imagesc(log(abs(real(ftkernxx))+1),[0,3/4])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernxx))+1),[0,1/10])
colormap(gray), axis off, axis image

ftkernyy=fftshift(fft2(fftshift(kernelfillyy)));
figure;
imagesc(log(abs(real(ftkernyy))+1),[0,3/4])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernyy))+1),[0,1/10])
colormap(gray), axis off, axis image

ftkernxy=fftshift(fft2(fftshift(kernelfillxy)));
figure;
imagesc(log(abs(real(ftkernxy))+1),[0,1/4])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(real(ftkernxy))+1),[0,1/4])
colormap(gray), axis off, axis image
```

Image Critical Points

```
ftkernxftI =ftkernx .*ftI;
figure;
imagesc(log(abs(real(ftkernxftI))+1),[0,15])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernxftI))+1),[0,15])
colormap(gray), axis off, axis image

ftkernyftI =ftkerny .*ftI;
figure;
imagesc(log(abs(real(ftkernyftI))+1),[0,15])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernyftI))+1),[0,15])
colormap(gray), axis off, axis image
ftkernxxftI=ftkernxx.*ftI;
figure;
imagesc(log(abs(real(ftkernxxftI))+1),[0,15])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernxxftI))+1),[0,15])
colormap(gray), axis off, axis image
ftkernyyftI=ftkernyy.*ftI;
```

```
figure;
imagesc(log(abs(real(ftkernyyftI))+1),[0,15])
colormap(gray), axis off, axis image
figure;
imagesc(log(abs(imag(ftkernyyftI))+1),[0,15])
colormap(gray), axis off, axis image

ftkernxyftI=ftkernxy.*ftI;
figure;
imagesc(log(abs(real(ftkernxyftI))+1),[0,15])
colormap(gray), axis off, axis image
print(gcf,'-dtiffn','-r100','ftkernxyftIR')
figure;
imagesc(log(abs(imag(ftkernxyftI))+1),[0,15])
colormap(gray), axis off, axis image
print(gcf,'-dtiffn','-r100','ftkernxyftII')

Ix      =real(fftshift(ifft2(fftshift(ftkernx .*ftI))));
Iy      =real(fftshift(ifft2(fftshift(ftkerny .*ftI))));
Ixx     =real(fftshift(ifft2(fftshift(ftkernxx.*ftI))));
Iyy     =real(fftshift(ifft2(fftshift(ftkernyy.*ftI))));
Ixy     =real(fftshift(ifft2(fftshift(ftkernxy.*ftI))));
ID      =Ixx.*Iyy-(Ixy).^2; % Determinant
```

Image Critical Points

```
figure;
imagesc(Ix,[-scale*2*pi/180,scale*2*pi/180])
colormap(gray), axis off, axis image
figure;
imagesc(Iy,[-scale*2*pi/60,scale*2*pi/60])
colormap(gray), axis off, axis image
figure;
imagesc(Ixx,[-2*pi^2/8100/4*scale,2*pi^2/8100/4*scale])
colormap(gray), axis off, axis image
figure;
imagesc(Iyy,[-2*pi^2/900/4*scale,2*pi^2/900/4*scale])
colormap(gray), axis off, axis image
figure;
imagesc(Ixy,[-1/10,1/10])
colormap(gray), axis off, axis image
figure;
imagesc(ID,[-
2*pi^2/8100/4*scale*2*pi^2/900/4*scale,2*pi^2/8100/4*scale*2*pi^2/900/4*scale])
colormap(gray), axis off, axis image
```

Image Critical Points

This process can be applied to other images

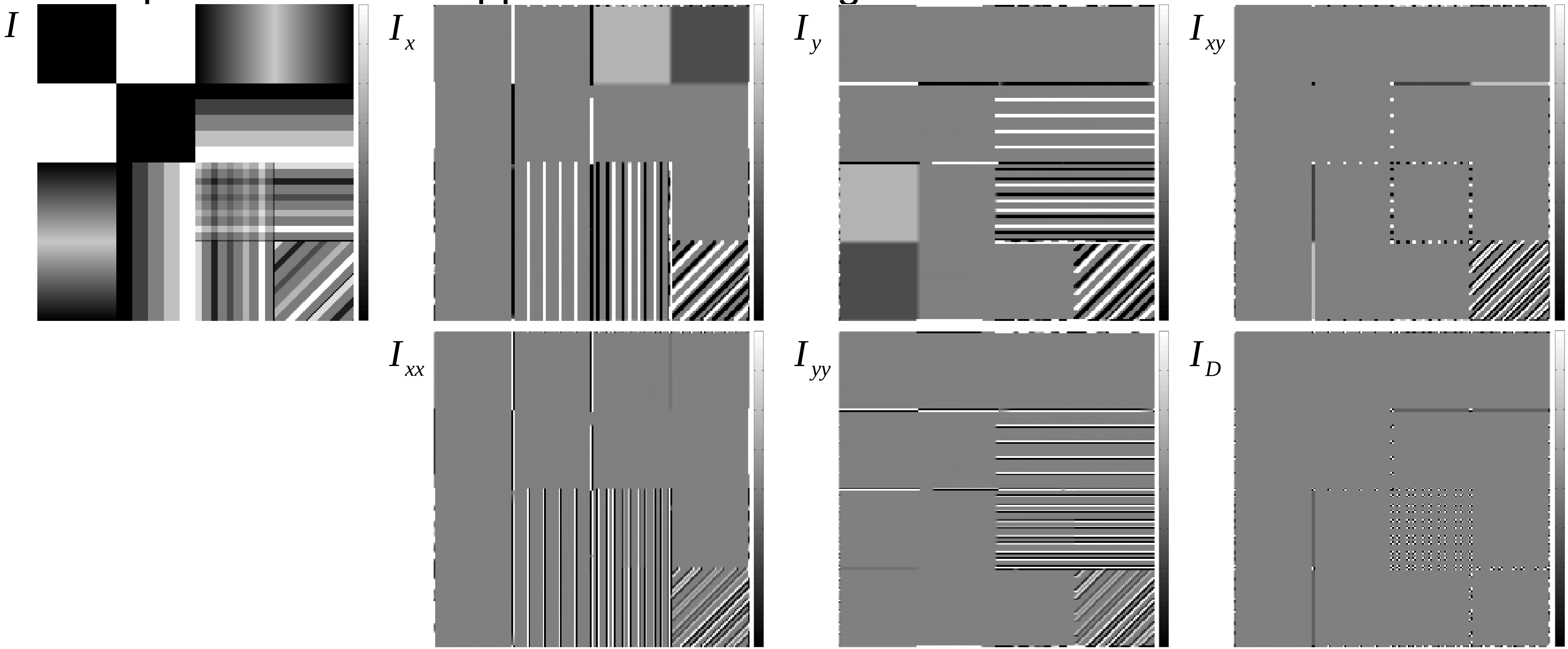
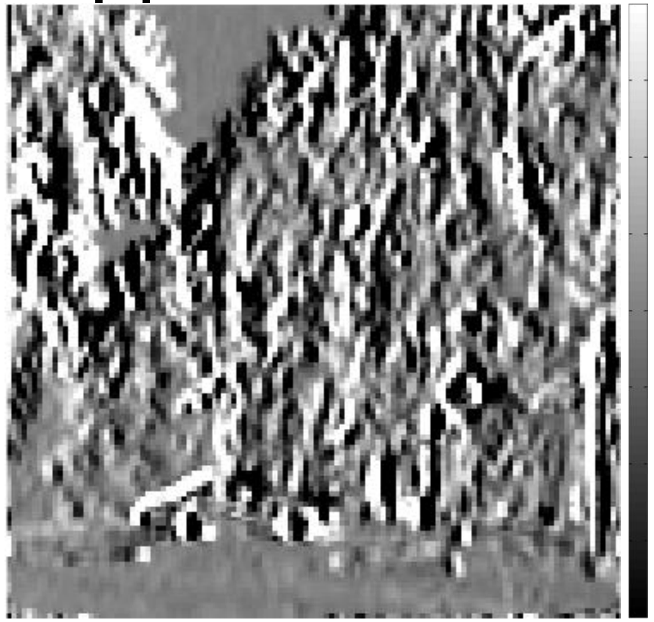


Image Critical Points

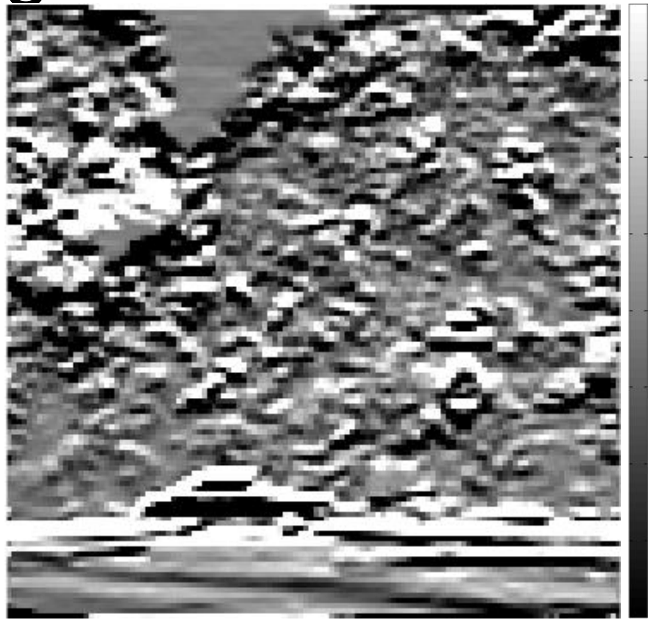
This process can be applied to other images

I

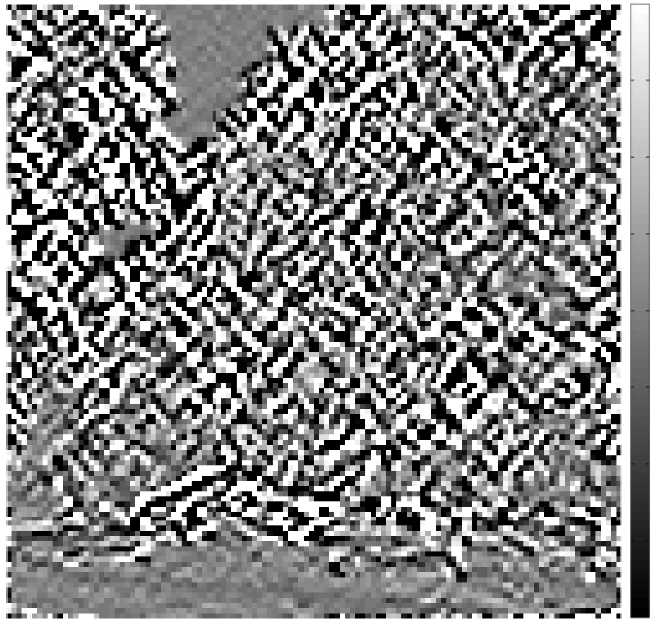
I_x



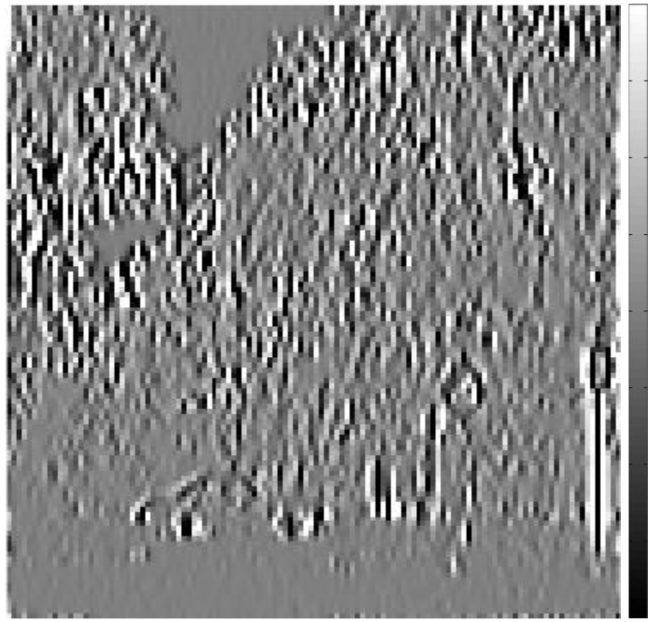
I_y



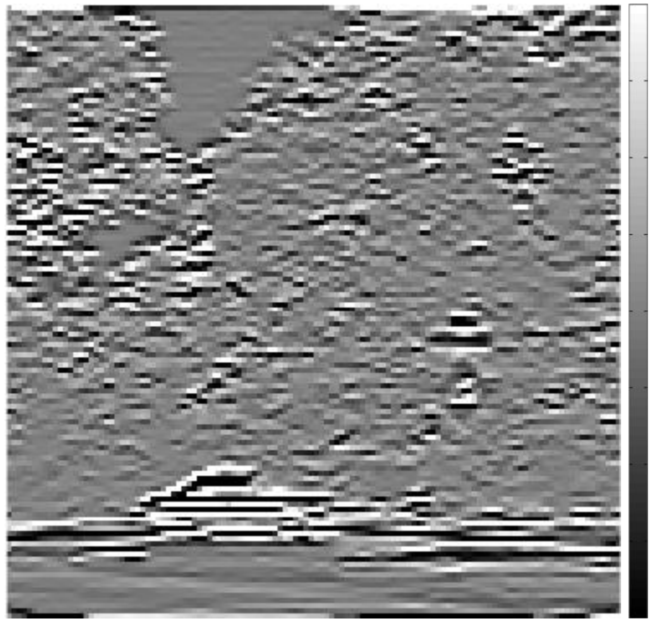
I_{xy}



I_{xx}



I_{yy}



I_D

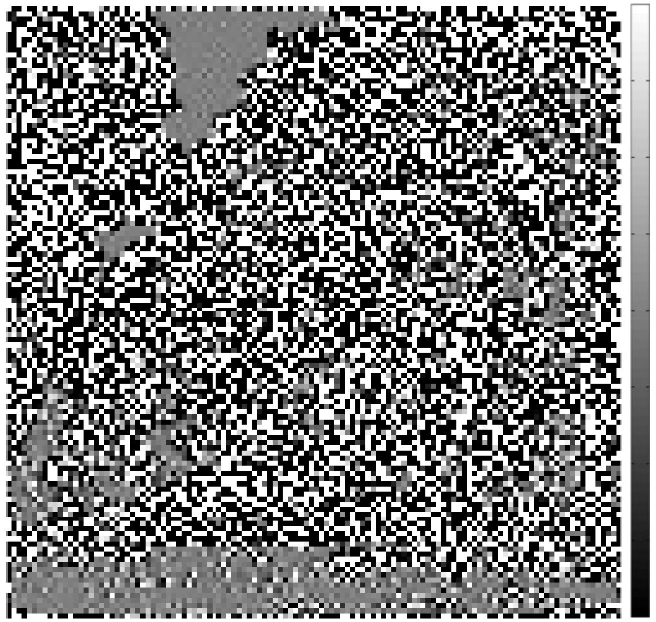


Image Critical Points

The largest eigenvector of H is perpendicular to the path of a curve.

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}$$

$$det(H\text{-}\lambda I)=(f_{xx}\text{-}\ \lambda)(f_{yy}\text{-}\ \lambda)\text{-}f_{xy}^{\,2}=0$$

solve for eigenvalue roots λ_1 and λ_2.

$$\lambda^2\text{-}(f_{xx}+f_{yy})\lambda+ f_{xx}f_{yy} - f_{xy}^{\,2}=0$$

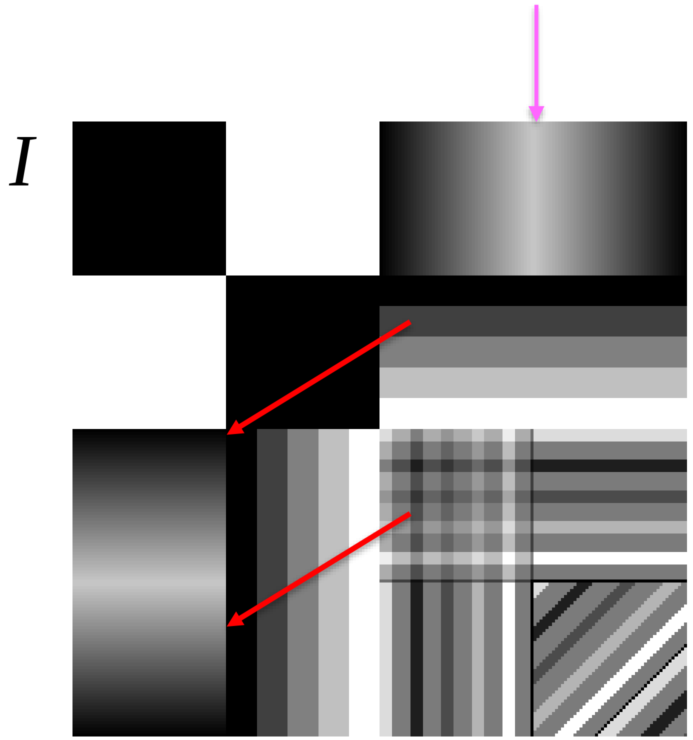
quadratic equation

Solve for eigenvectors $Av_1=\lambda_1 v_1$, $Av_2=\lambda_2 v_2$.

Obtain orthogonal v_1 and v_2.

We can calculate the angle that the eigenvectors are pointing θ_1 and θ_2.

Image Critical Points



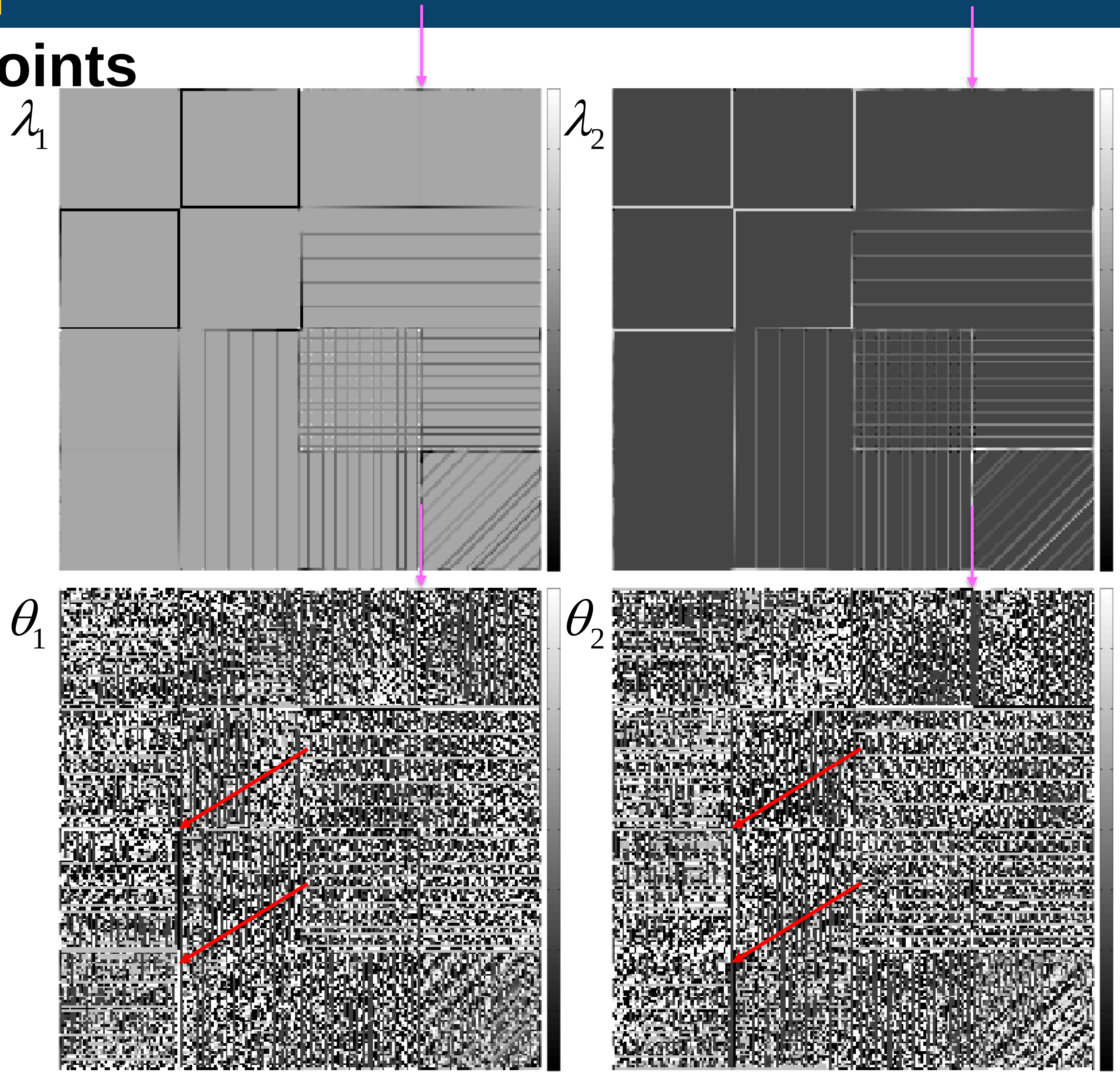


Image Critical Points

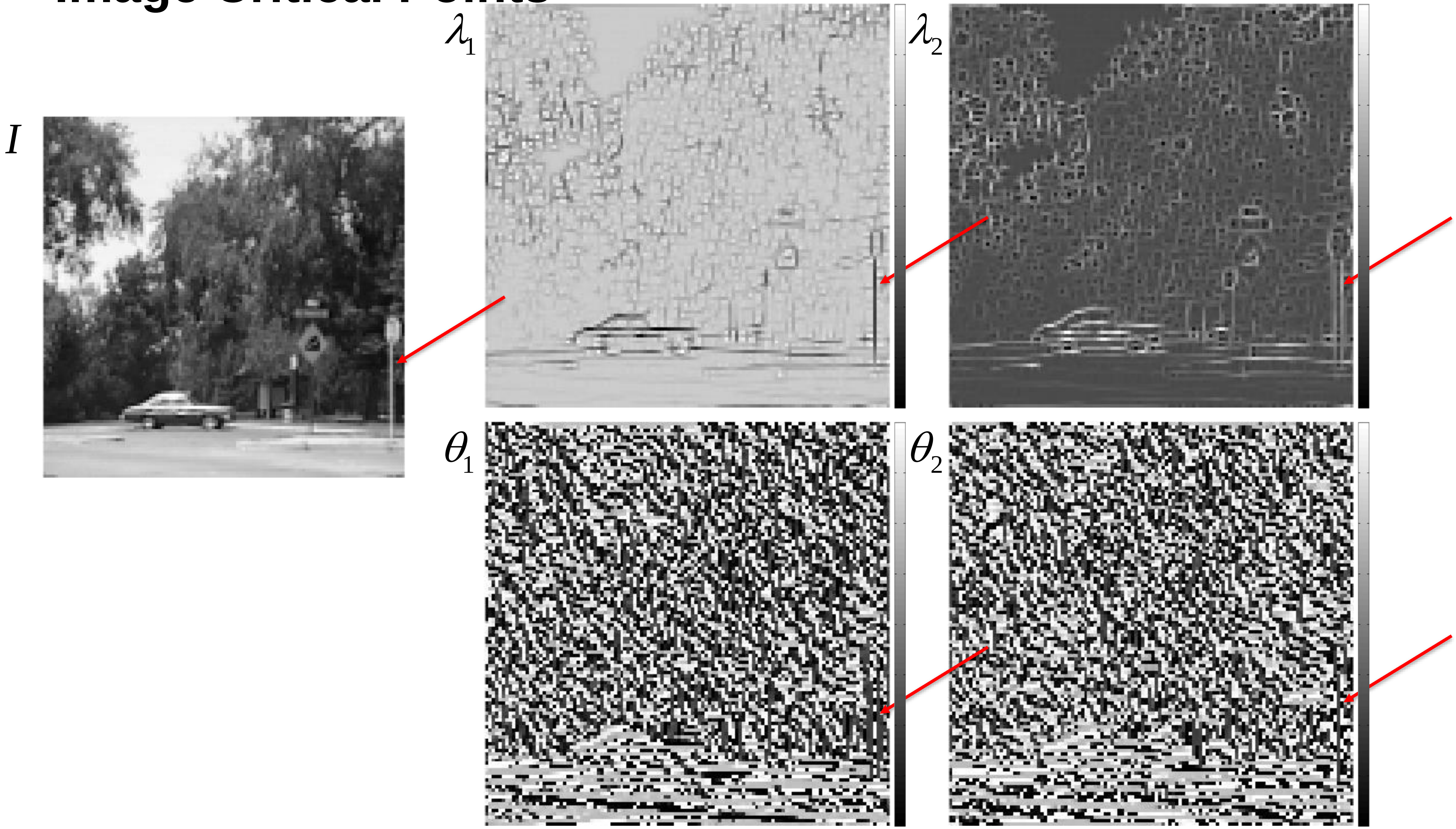


Image Critical Points

```
EV=zeros(ny,nx,2,2); ED=zeros(ny,nx,2,2);
theta1=zeros(ny,nx); theta2=zeros(ny,nx);
for j=1:ny
    for i=1:nx
        H=[Ixx(j,i),Ixy(j,i);Ixy(j,i),Iyy(j,i)]; % Hessian
        [V,D]=eig(H); % compute eigen vectors and eigen values
        EV(j,i,:,:)=V; ED(j,i,:,:)=D;
        theta1(j,i)=atan2(EV(j,i,2,1),EV(j,i,1,1)); % angle v1
        theta2(j,i)=atan2(EV(j,i,2,2),EV(j,i,1,2)); % angle v2
    end
end
figure;
imagesc(squeeze(ED(:,:,1,1)))
colormap(gray), axis off, axis image
figure;
imagesc(squeeze(ED(:,:,2,2)))
colormap(gray), axis off, axis image
figure;
imagesc(theta1)
colormap(gray), axis off, axis image
figure;
imagesc(theta2)
colormap(gray), axis off, axis image
```

Discussion

There is much more that can be done with derivatives.

Paths of relatively constant pixel intensity can be traced using the eigenvectors of the Hessian matrix.

The largest eigenvector of H is perpendicular to the path of a curve.

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}$$

Eigenvectors from adjacent pixels can be connected to trace out ridges.

Discussion

Questions?

Homework 11

1. Apply the first and second derivative kernels to an image of your own. Present images of each of the derivatives and D image. Comment in relation to your image features.

2. Determine pixels in an image that are local maxima or local minima.

3*. Use the derivative images and create your own homework problem. What can you learn from them?

*For students in MSSC 5770.