Lighting direction vs. Viewer direction

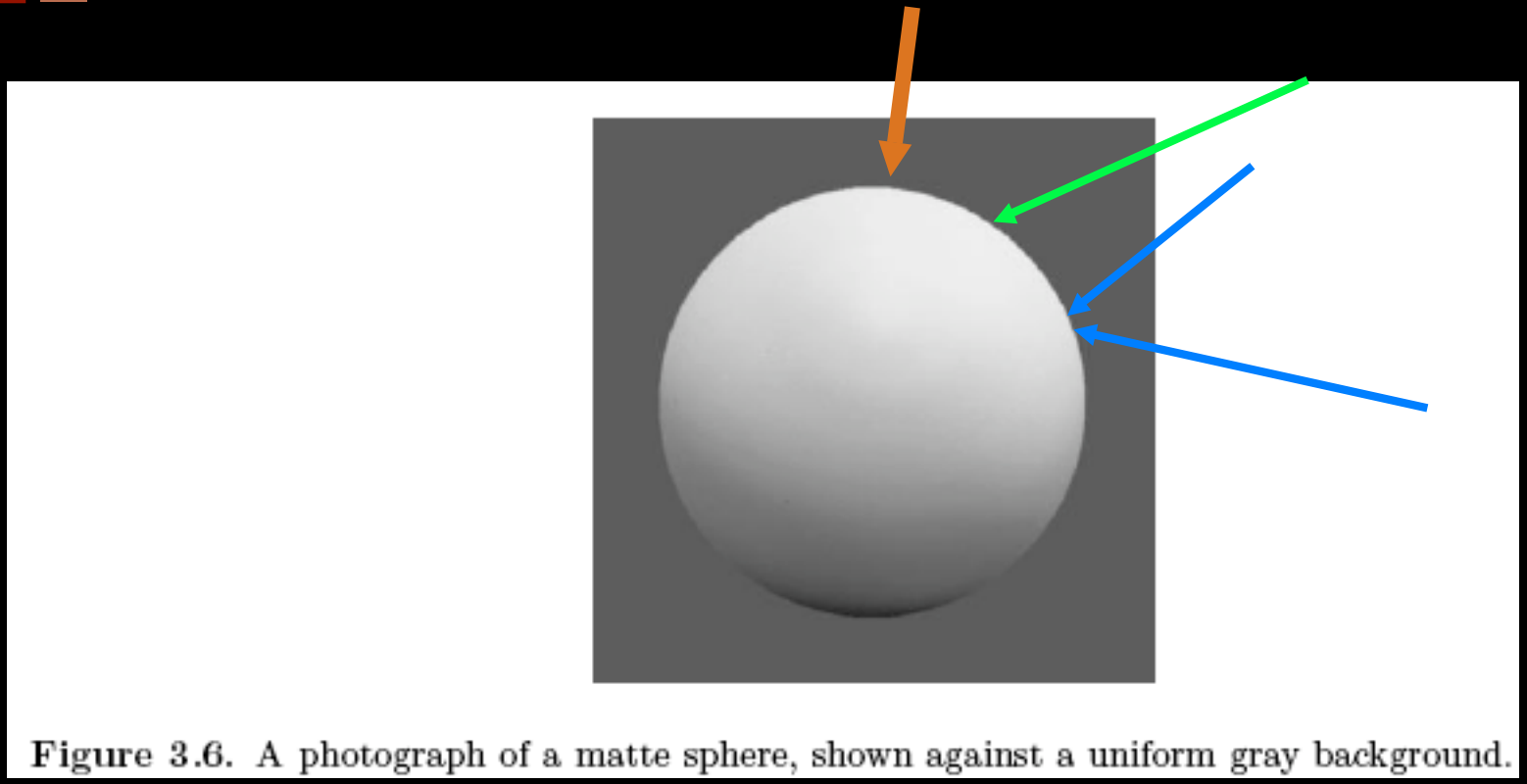


Figure 3.6. A photograph of a matte sphere, shown against a uniform gray background.

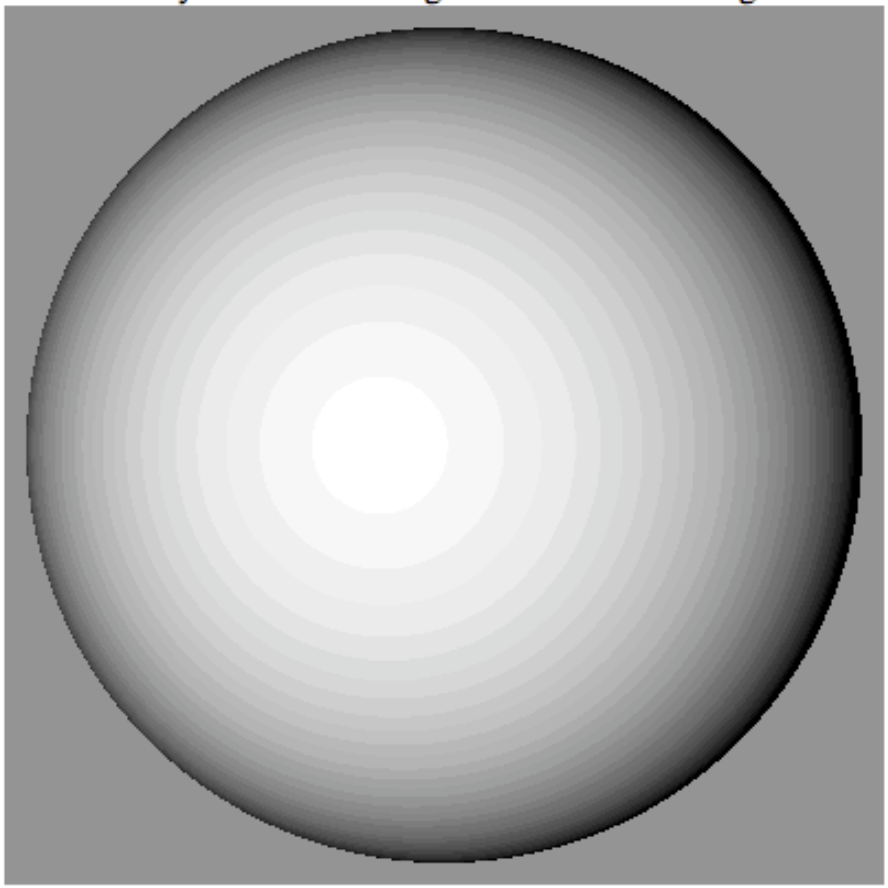
For Lambertian surface:

Viewer direction is irrelevant

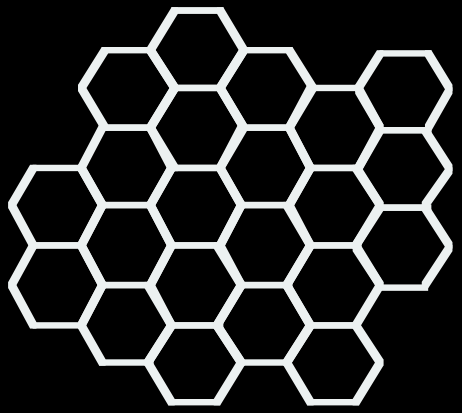
Lighting direction is *very relevant*

Lambertian Sphere

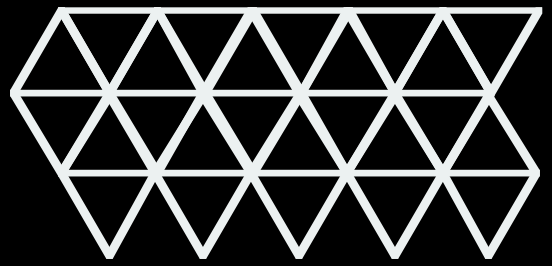
Orthogonal Projection,
Infinitely Distant Point Light from -90 to +90 degrees



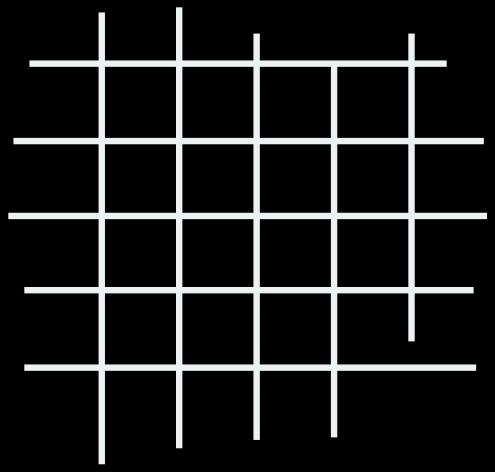
Tesselation Patterns



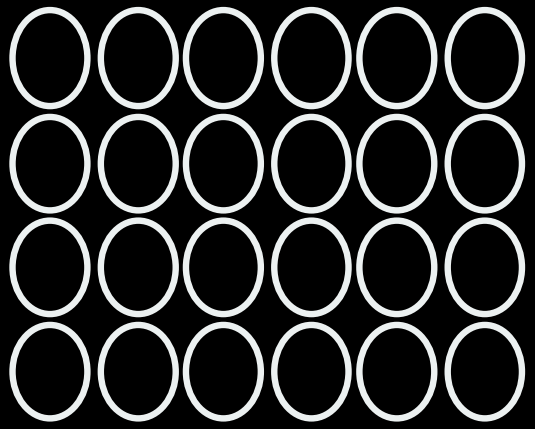
Hexagonal



Triangular



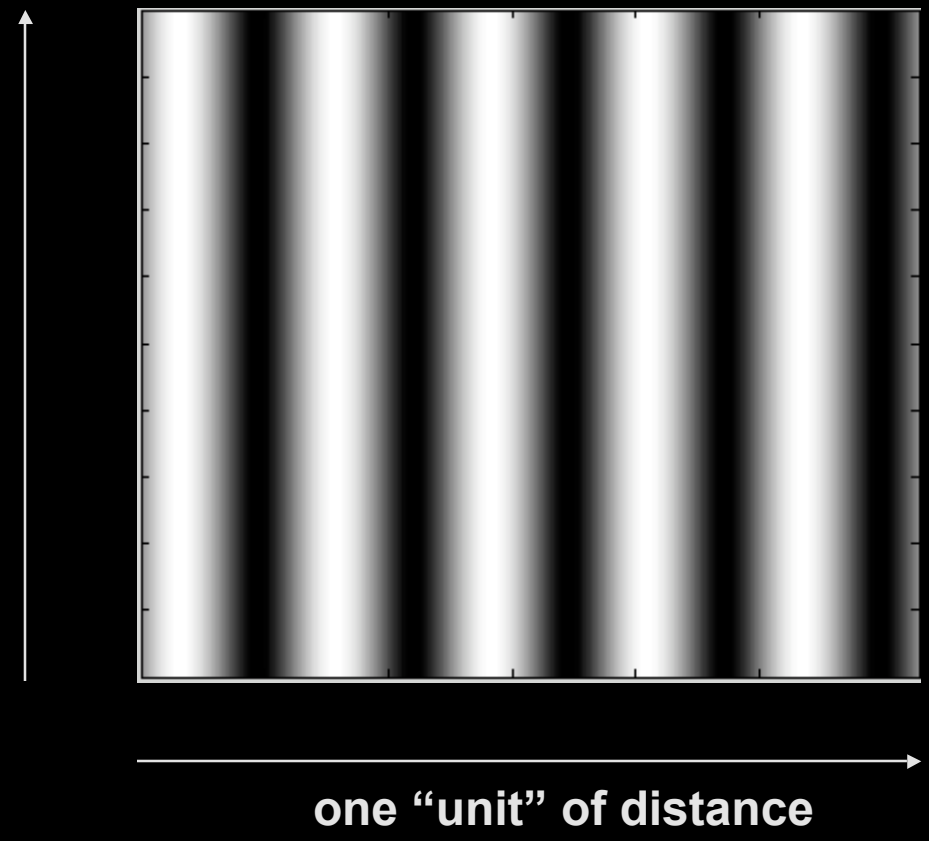
Rectangular



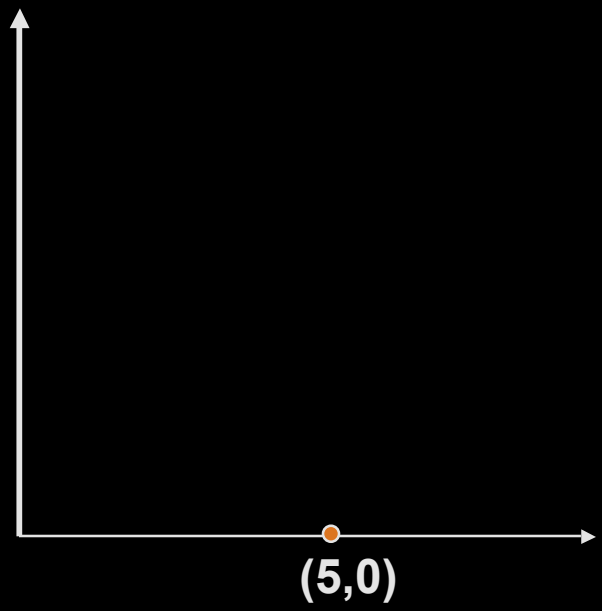
Typical

Spatial Frequencies

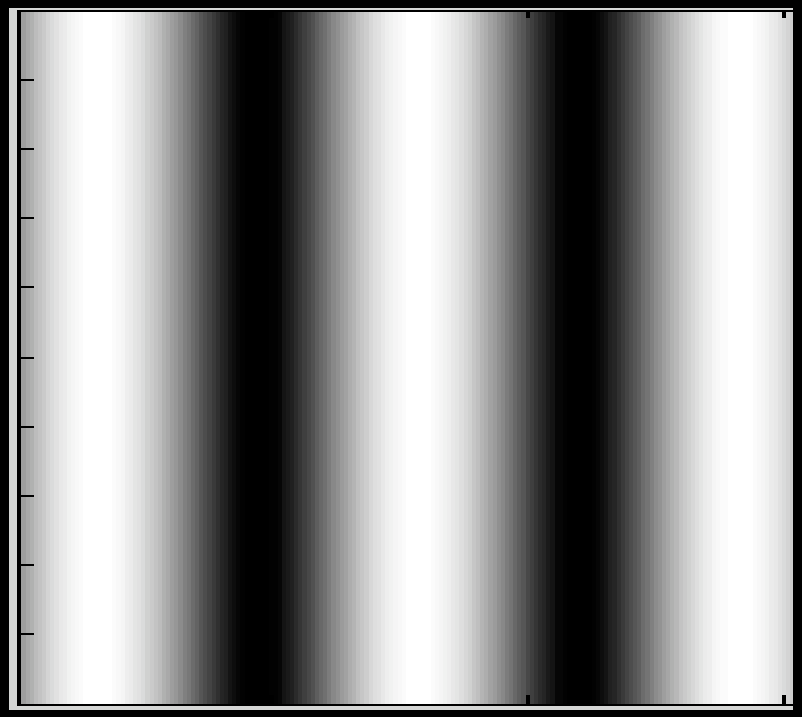
Image



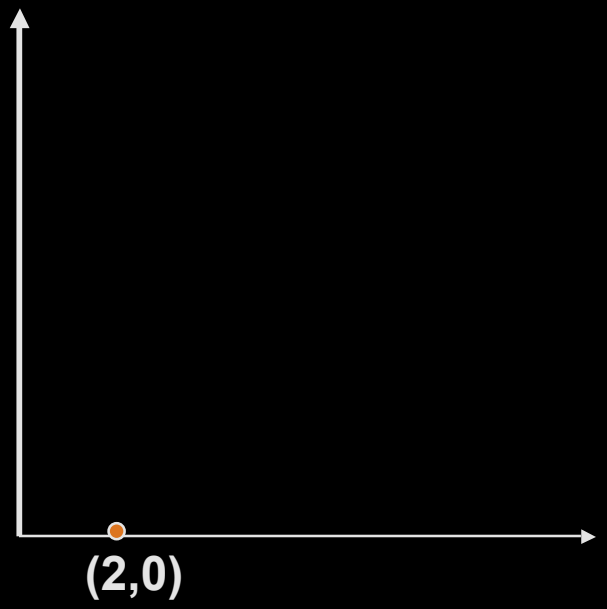
Fourier Power Spectrum



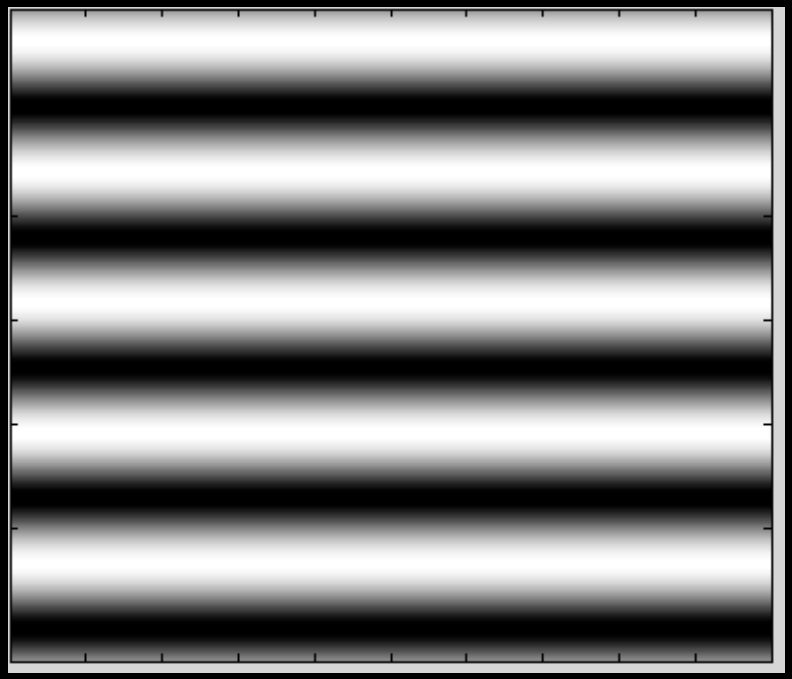
Spatial Frequencies



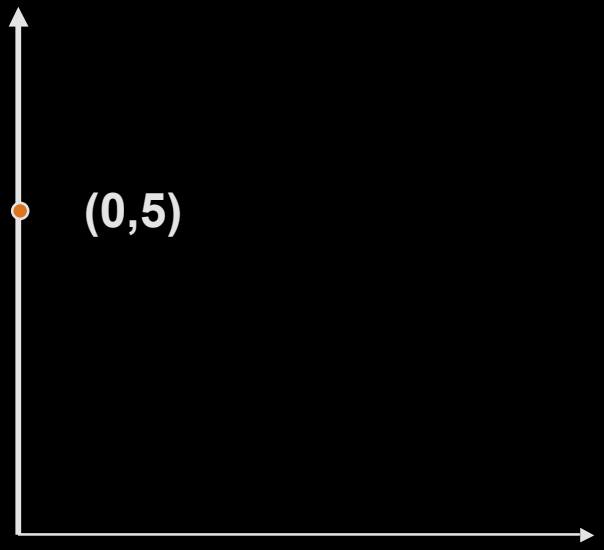
Fourier Power Spectrum



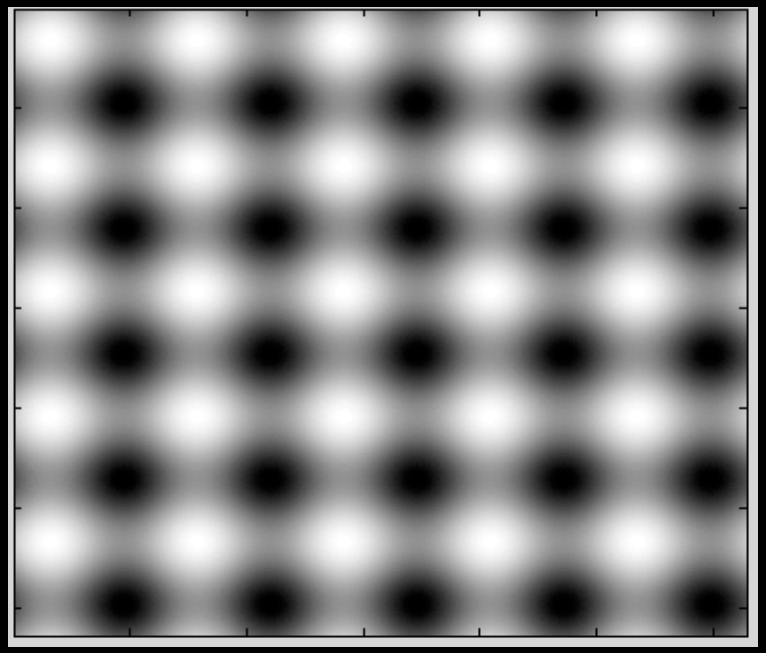
Spatial Frequencies



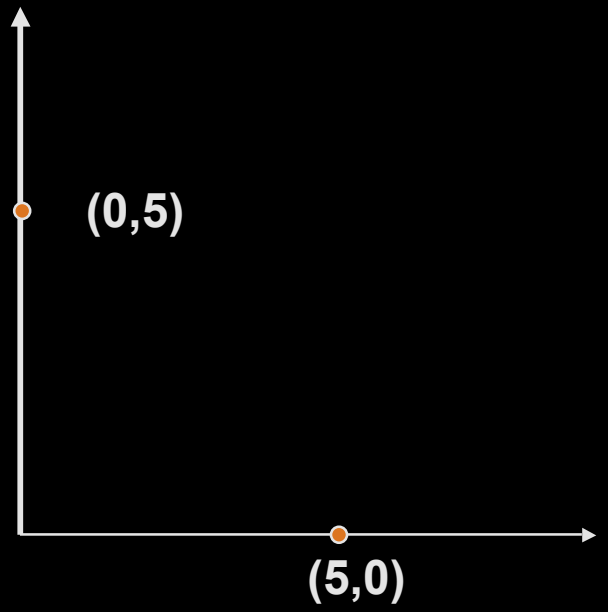
Fourier Power Spectrum



Spatial Frequencies

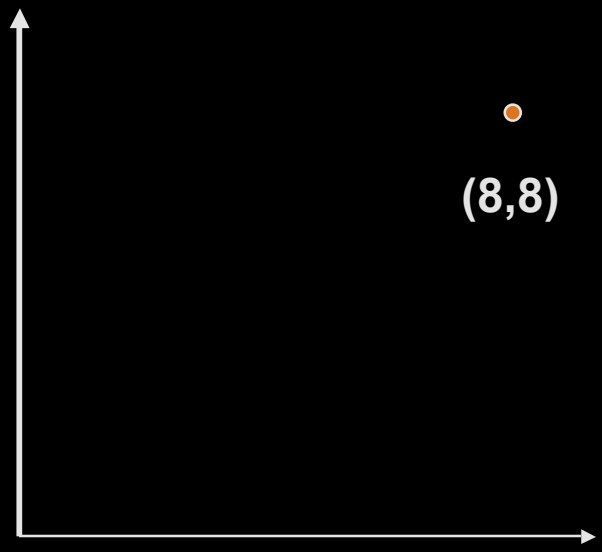


Fourier Power Spectrum



Spatial Frequencies

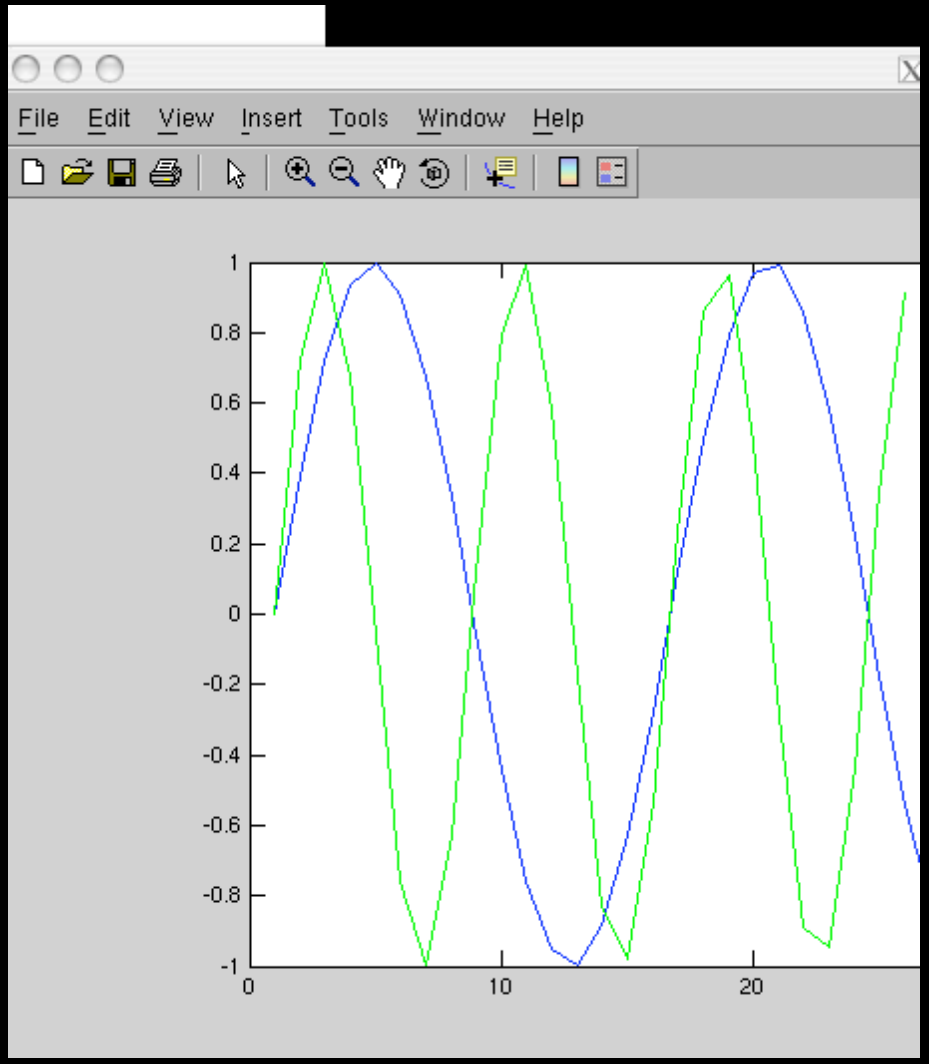
Fourier Power Spectrum



Sampling efficiency

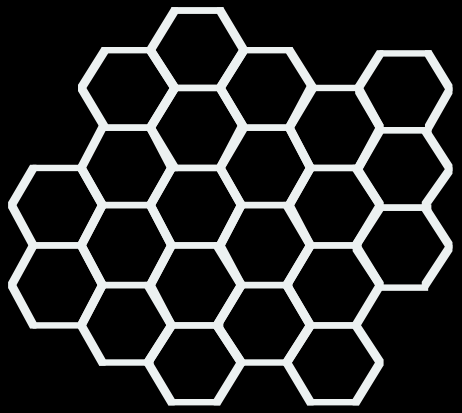
- Every sampling scheme captures some spatial frequencies but not others:
  - Low frequency sampling doesn't capture the picket fence
  - High frequency does.
- Which two-dimensional sampling scheme is most "efficient"?

Aliasing

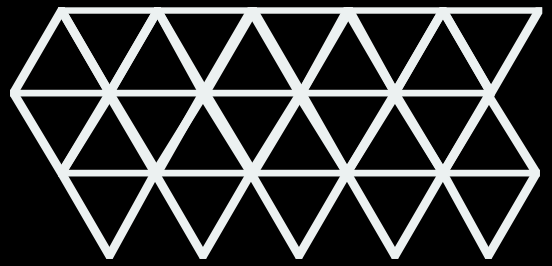

File Edit View Insert Tools Window Help
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20

Wagon Wheel illusion

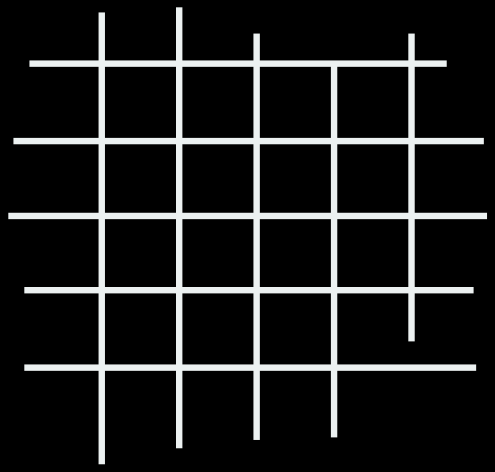
Tesselation Patterns



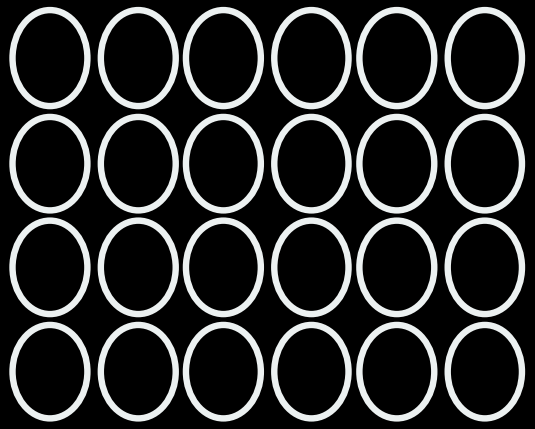
Hexagonal



Triangular

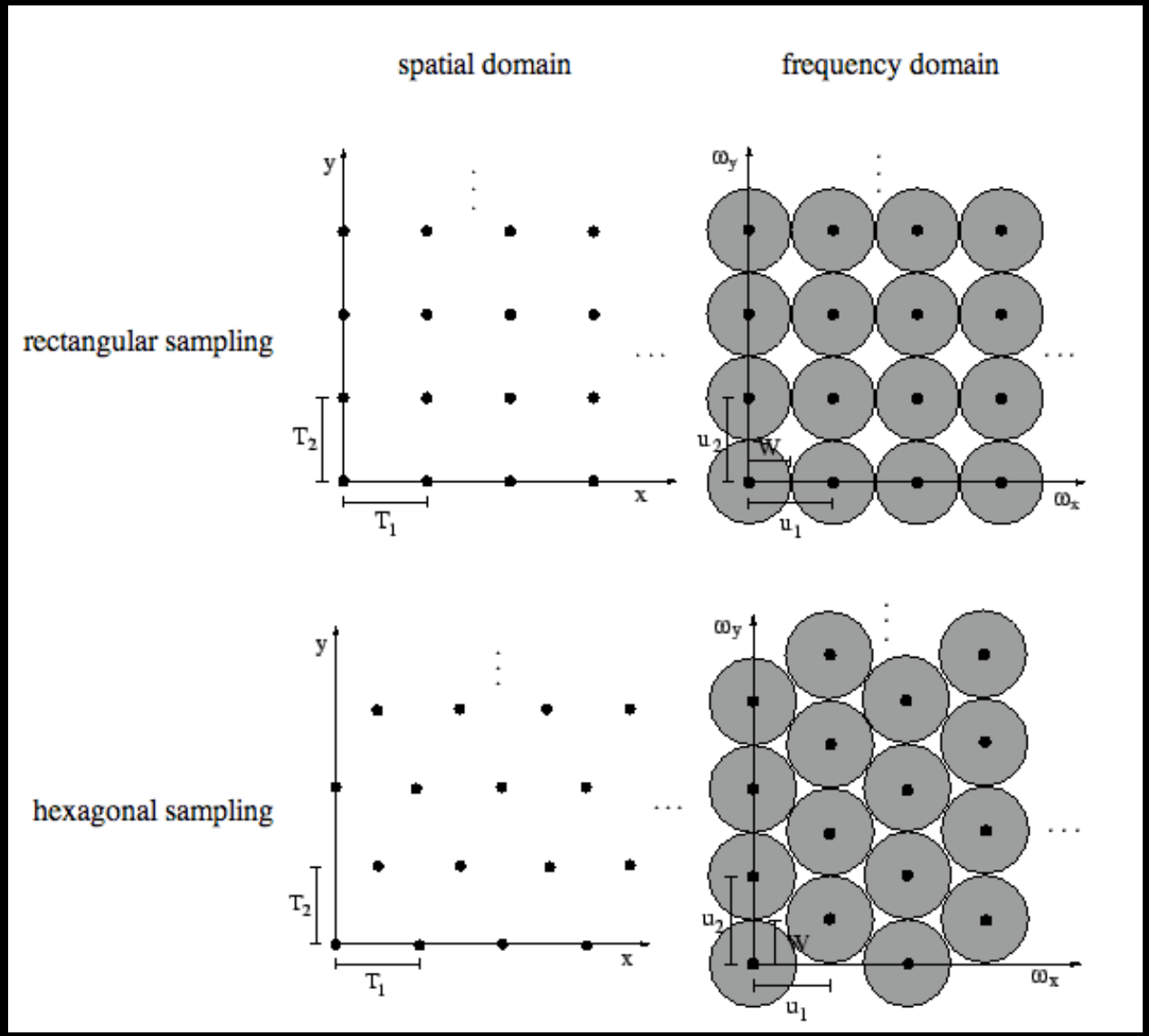


Rectangular



Typical

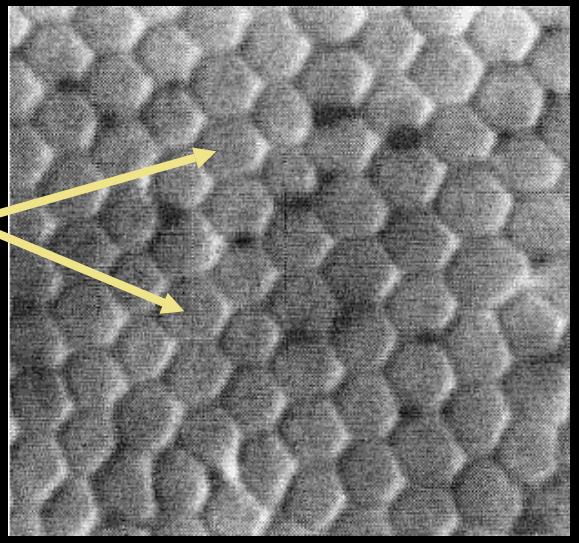
Sampling Grids



Retina

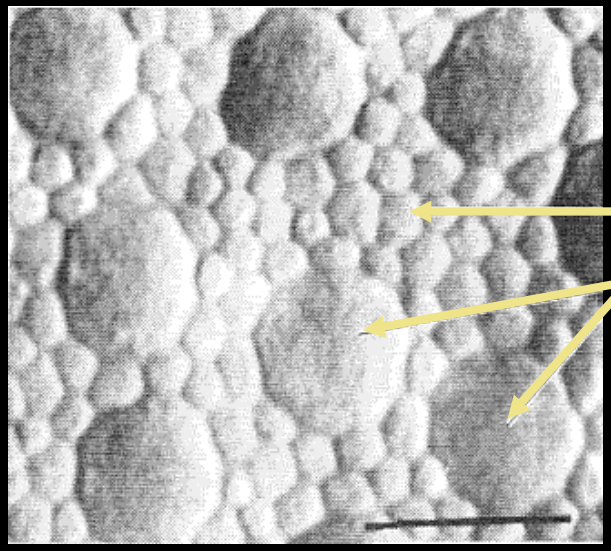
Cones in the fovea

Cones

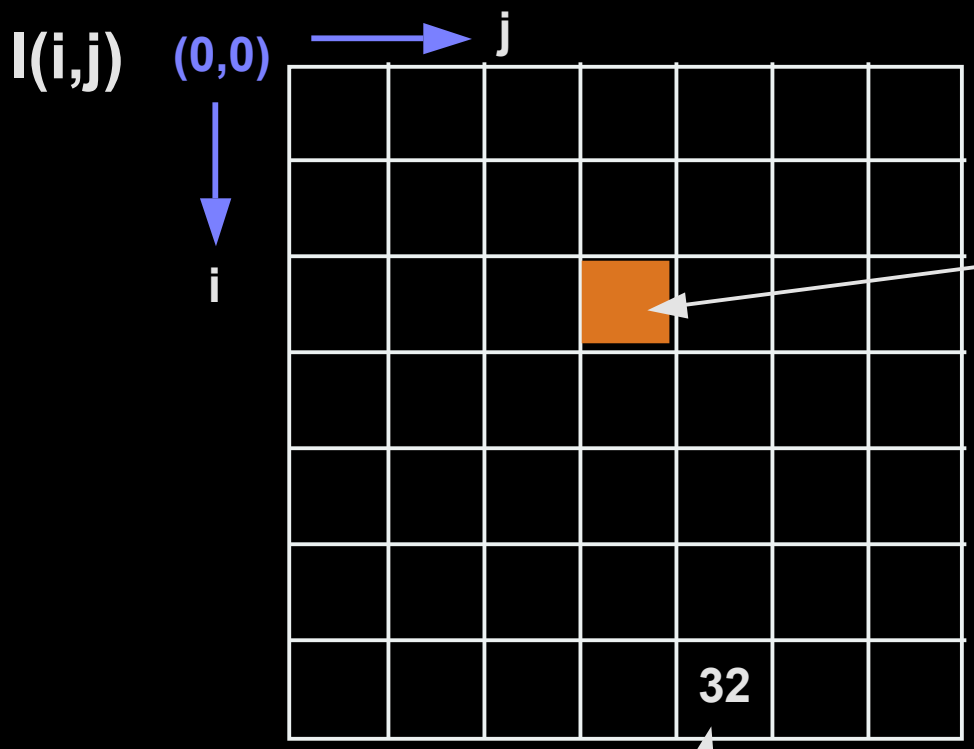


All of them are cones!

Moving outward from fovea



Digital Geometry



Picture Element or Pixel

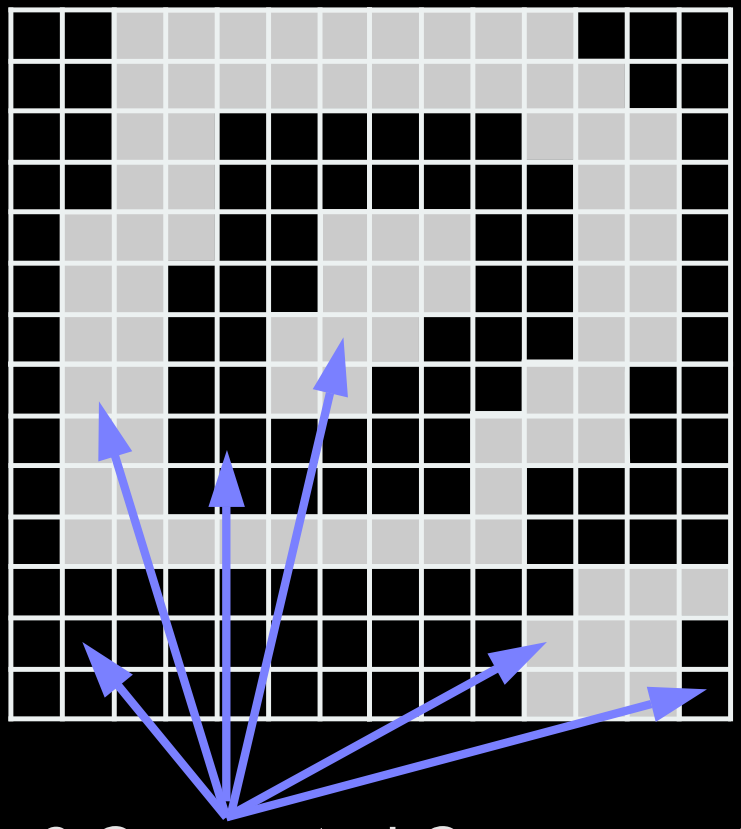
Pixel value I(l,j) =

- 0,1 Binary Image
- 0 - K-1 Gray Scale Image
- Vector: Multispectral Image

- Neighborhood
- Connectedness
- Distance Metrics

Connected Components

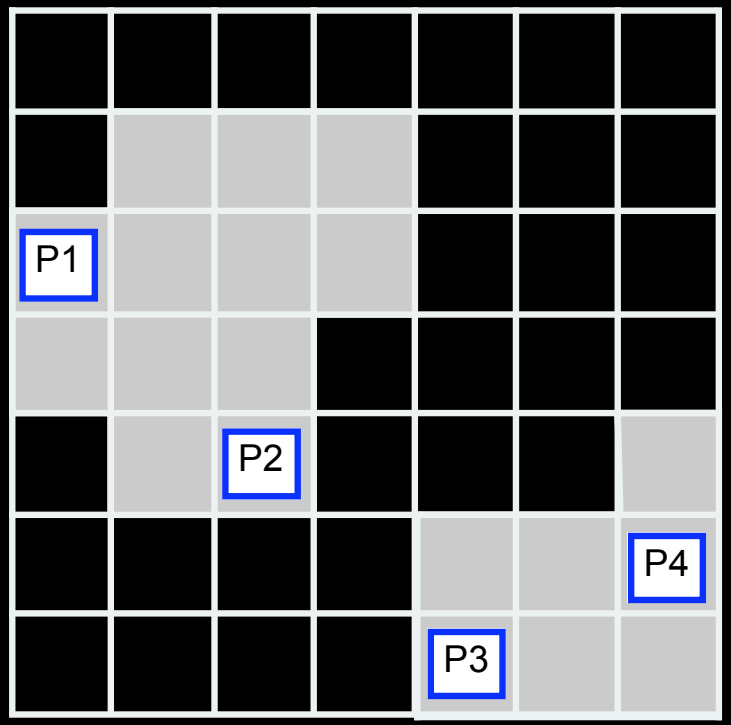
- Binary image with multiple 'objects'
- Separate 'objects' must be labeled individually



6 Connected Components

Finding Connected Components

- Two points in an image are 'connected' if a path can be found for which the value of the image function is the same all along the path.



P_1 connected to P_2

P_3 connected to P_4

P_1 not connected to P_3 or P_4

P_2 not connected to P_3 or P_4

P_3 not connected to P_1 or P_2

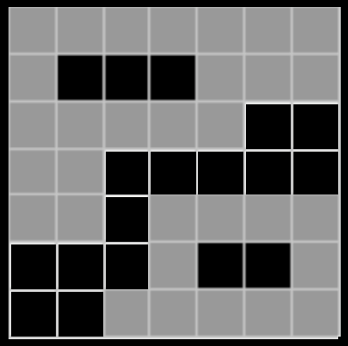
P_4 not connected to P_1 or P_2

Algorithm

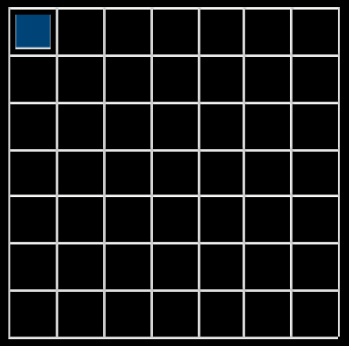
- Pick any pixel in the image and assign it a label
- Assign same label to any neighbor pixel with the same value of the image function
- Continue labeling neighbors until no neighbors can be assigned this label
- Choose another label and another pixel not already labeled and continue
- If no more unlabeled image points, stop.

Who's my neighbor?

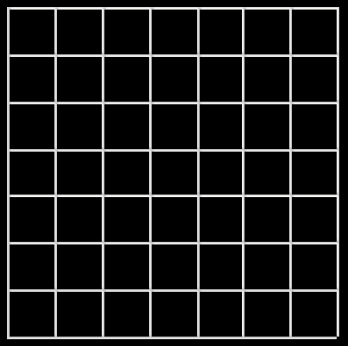
Example

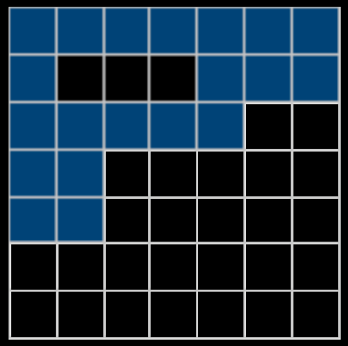


Image

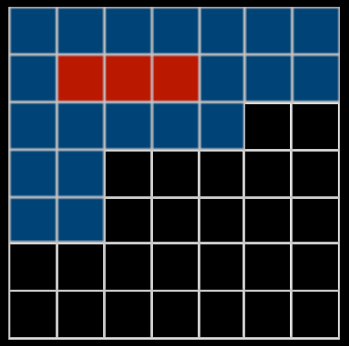


'Label' Image

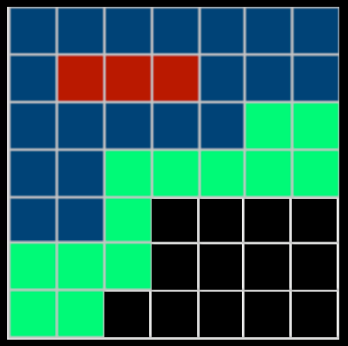




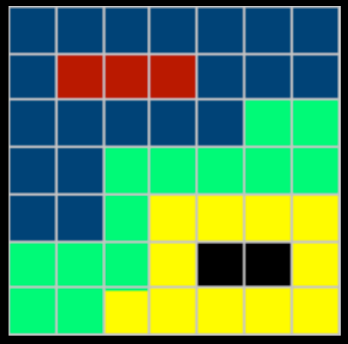
Lab. Im. - 1st Component



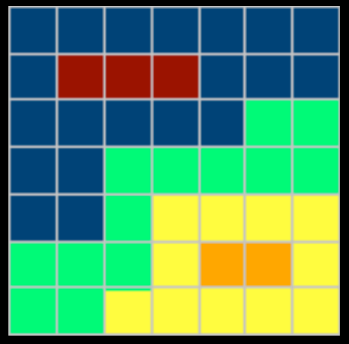
Lab. Im. - 2nd Component



Lab. Im. - 3rd Component



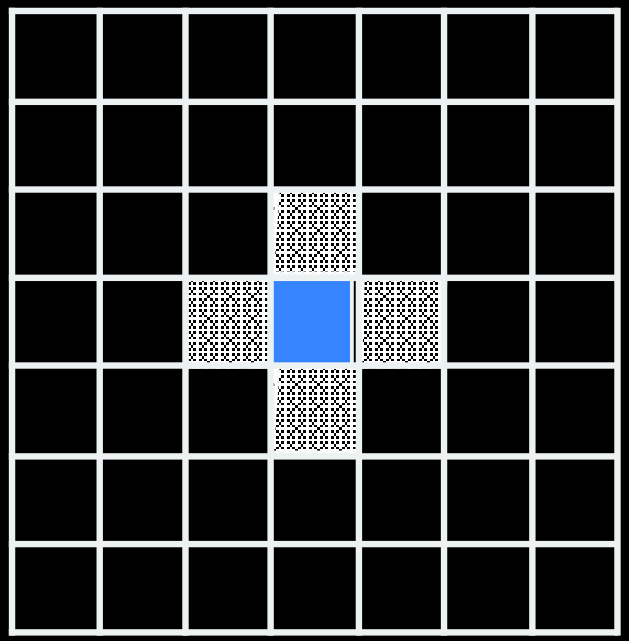
Lab. Im. - 4th Component



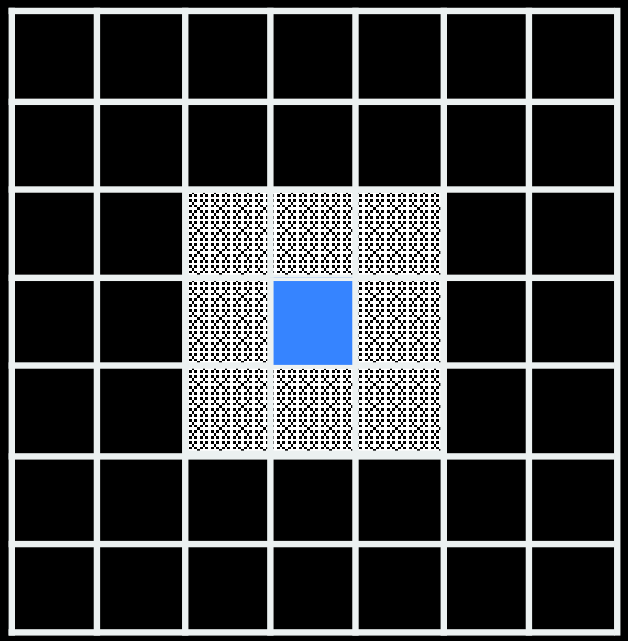
Final Labeling

Neighbor

- Consider the definition of the term 'neighbor'
- Two common definitions:



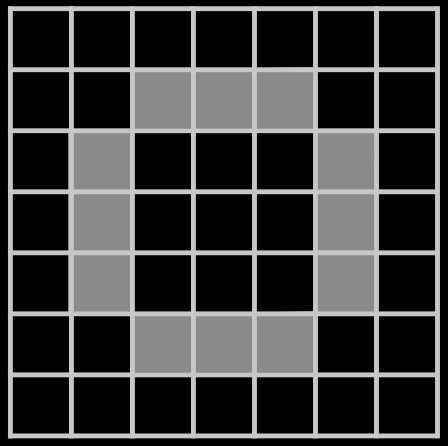
Four Neighbor



Eight Neighbor

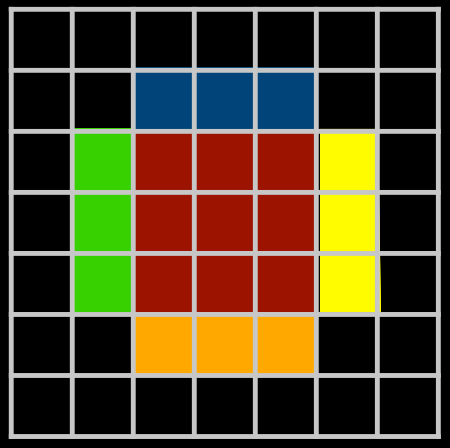
- Consider what happens with a closed curve.
- One would expect a closed curve to partition the plane into two connected regions.

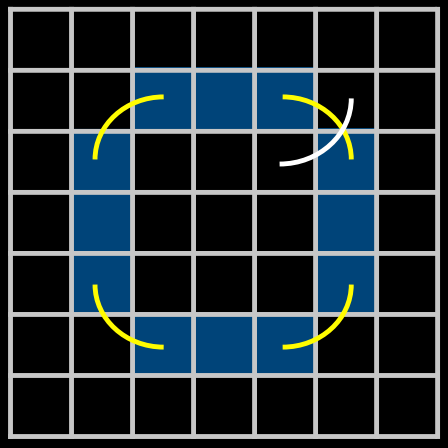
Alternate Neighborhood Definitions



4-neighbor connectedness

8-neighbor connectedness

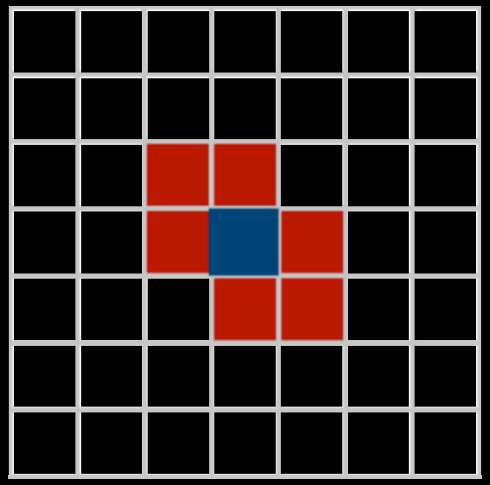




Neither neighborhood definition satisfactory!

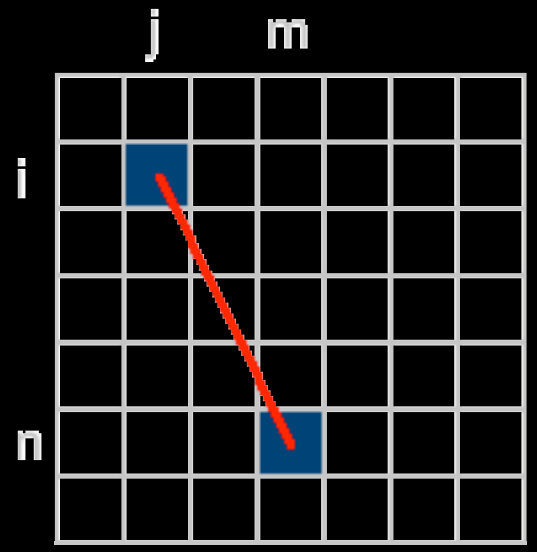
Possible Solutions

- Use 4-neighborhood for object and 8-neighborhood for background
  - requires a-priori knowledge about which pixels are object and which are background
- Use a six-connected neighborhood:



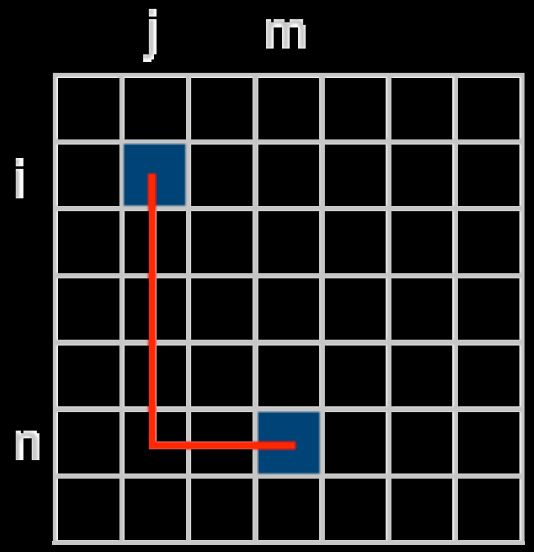
Digital Distances

- Alternate distance metrics for digital images



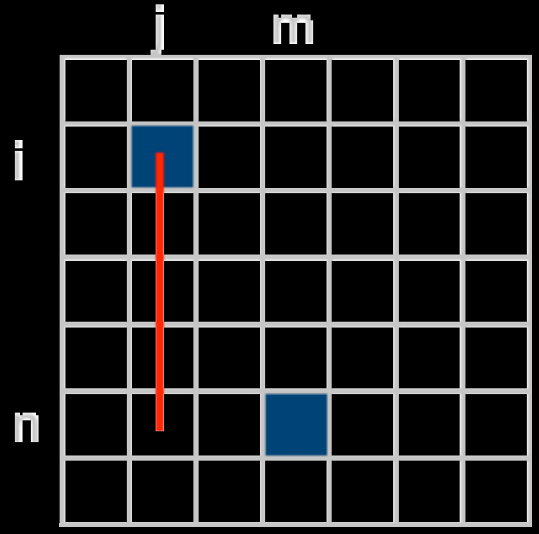
Euclidean Distance

$$= \sqrt{(i-n)^2 + (j-m)^2}$$



City Block Distance

$$= |i-n| + |j-m|$$



Chessboard Distance

$$= \max[\, |i-n|, |j-m| \,]$$