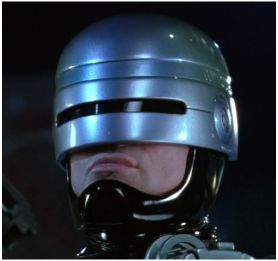
CS325 Artificial Intelligence
Ch. 24, Computer Vision I – Object Recognition

Cengiz Günay, Emory Univ.

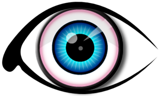


Spring 2013

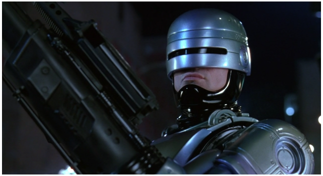
Computer Vision

- Done with games, except homework :)

- Done with games, except homework :)

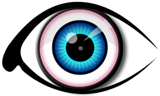


- **Vision** is one of our main perceptions
- **Computer vision** is what robots use to understand their surrounding



Computer Vision

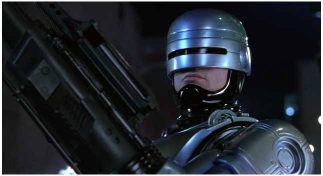
- Done with games, except homework :)



- **Vision** is one of our main perceptions
- **Computer vision** is what robots use to understand their surrounding

3 lectures:

1. Object recognition (today)
2. 3D reconstruction
3. Motion analysis



Entry/Exit Surveys

Exit survey: Advanced Planning

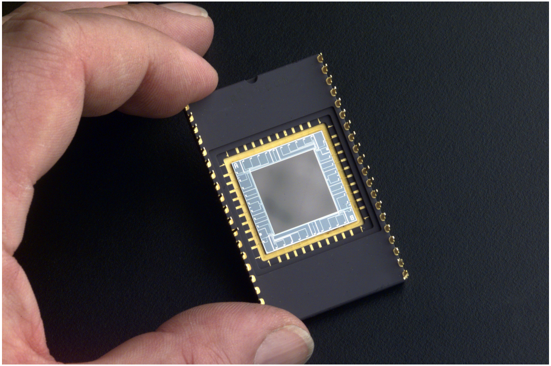
- Why isn't classical planning schema adequate for resource planning?
- What is the advantage gained in abstract plans by having *surely-reachable* versus *potentially-reachable* states?

Entry survey: Computer Vision I – Image Processing (0.25 points)

- List three specific tasks where computer vision would be desirable.
- What do you think are the major hurdles in computer vision?

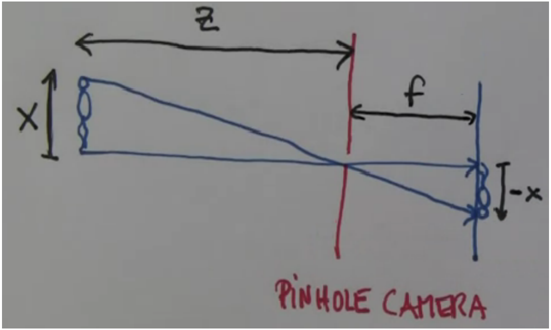
How Machines See: Cameras

A charge-coupled device (CCD) photo sensor array:



Focal Optics for Determining Distance and Size

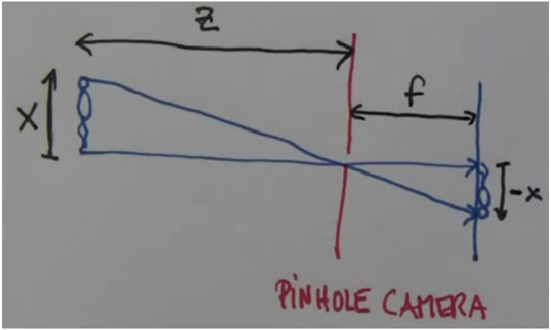
See the videos, I'll summarize:



$$\frac{X}{Z} = \frac{x}{f}$$

Focal Optics for Determining Distance and Size

See the videos, I'll summarize:

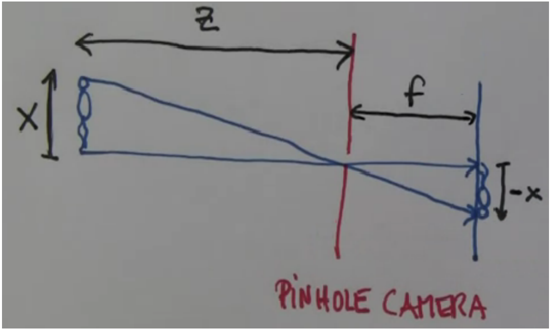


$$\frac{X}{Z} = \frac{x}{f}$$

What can we can figure out from this?

Focal Optics for Determining Distance and Size

See the videos, I'll summarize:



$$\frac{X}{Z} = \frac{x}{f}$$

What can we can figure out from this?

- Object's distance (Z) & height (X) based on projection height (x) and focal distance (f)

We All See a Perspective Projection

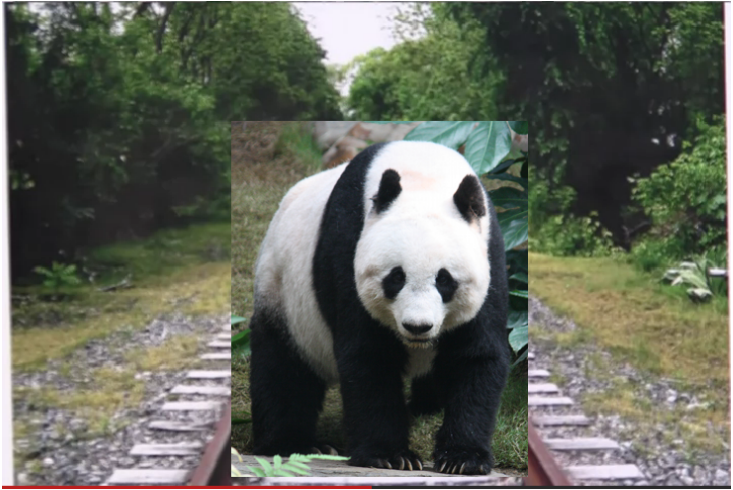
Vanishing points from parallel lines:

We All See a Perspective Projection

Vanishing points from parallel lines:

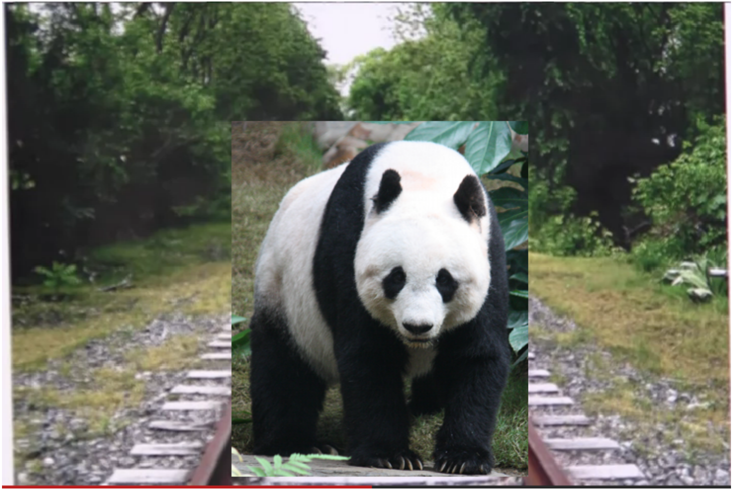
We All See a Perspective Projection

Vanishing points from parallel lines:



We All See a Perspective Projection

Vanishing points from parallel lines:



- Giant panda, or just close?

Object Recognition: How Hard Can It Be?

Object Recognition: How Hard Can It Be?

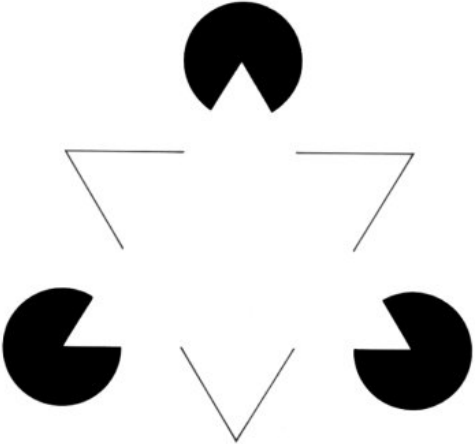
Problems?

Object Recognition: How Hard Can It Be?

Problems?

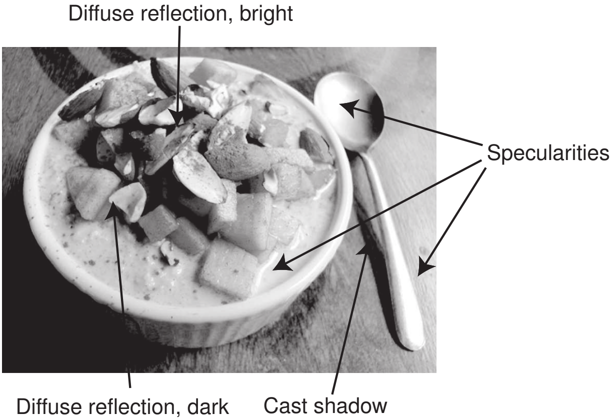
- Rotation, scale, illumination, occlusion, viewpoint, deformation

Not Hard for Us

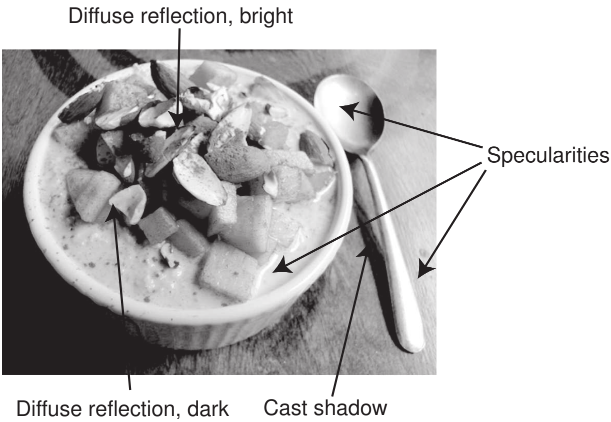


Not Hard for Us

Not Hard for Us

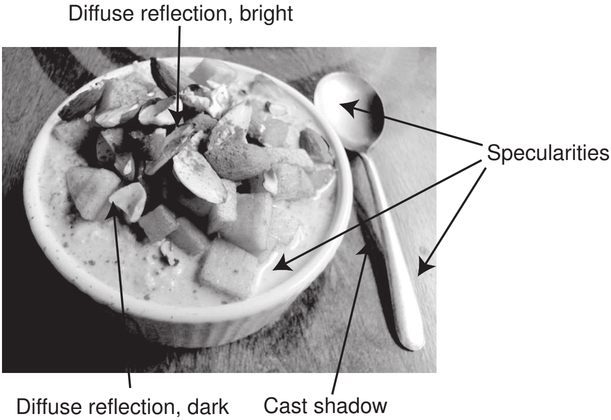


Not Hard for Us



How does our brain do it?

Not Hard for Us



How does our brain do it? Will have examples later.

Invariance is Crucial for Computer Vision

Must recognize objects **invariant** of their:

- Rotation, scale, illumination, occlusion, viewpoint, deformation

Invariance is Crucial for Computer Vision

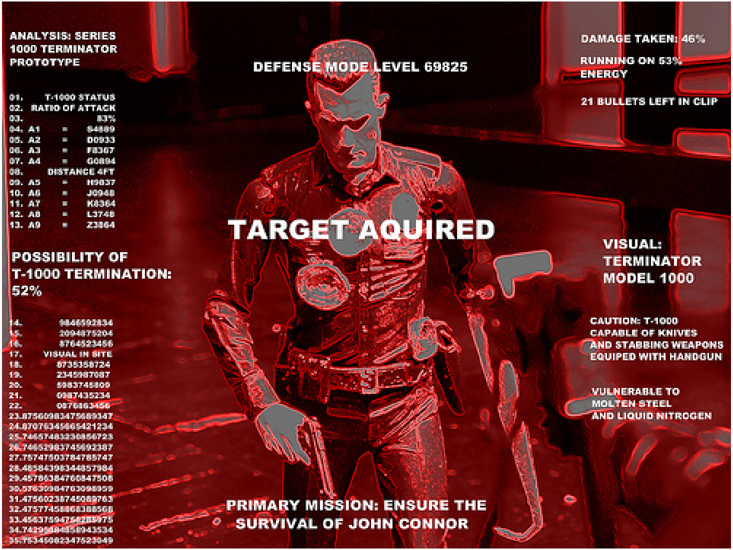
Must recognize objects **invariant** of their:

- Rotation, scale, illumination, occlusion, viewpoint, deformation

Let's start by simplifying:

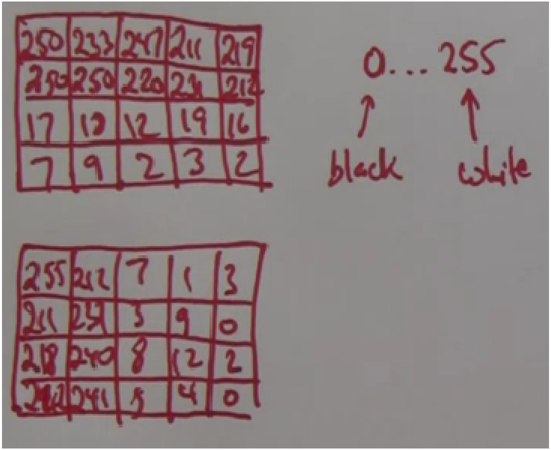
1. Greyscale (monochrome) images
2. Pixels can have values: $0 \ldots 255$

Even Terminator Has Monochrome Vision

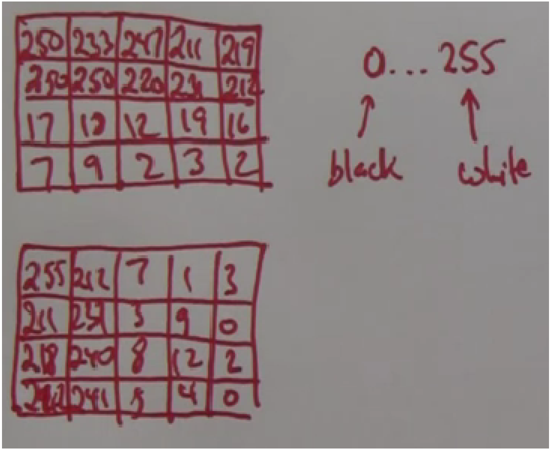


Extracting Features: Edge Detection

Extracting Features: Edge Detection

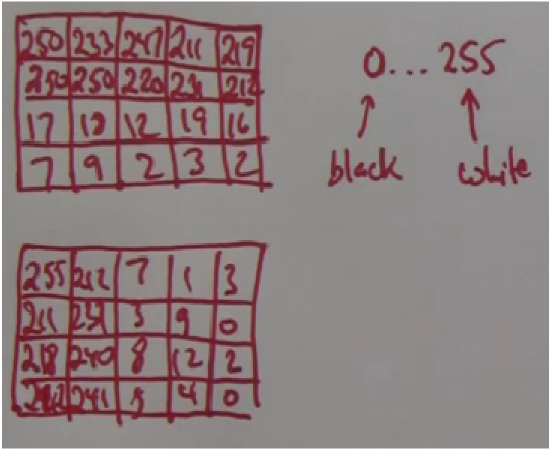


Extracting Features: Edge Detection



How to detect the vertical edge?

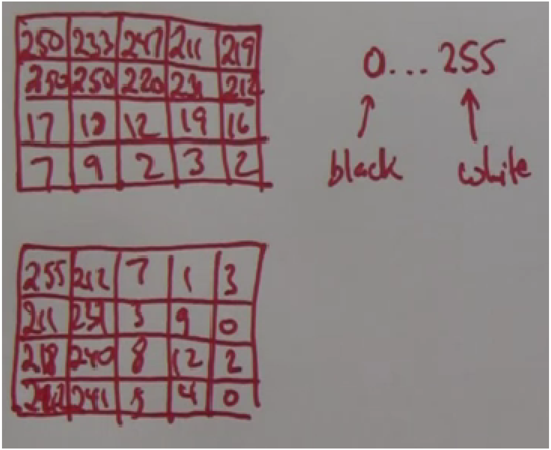
Extracting Features: Edge Detection



How to detect the vertical edge?

1. Spatial derivative?

Extracting Features: Edge Detection

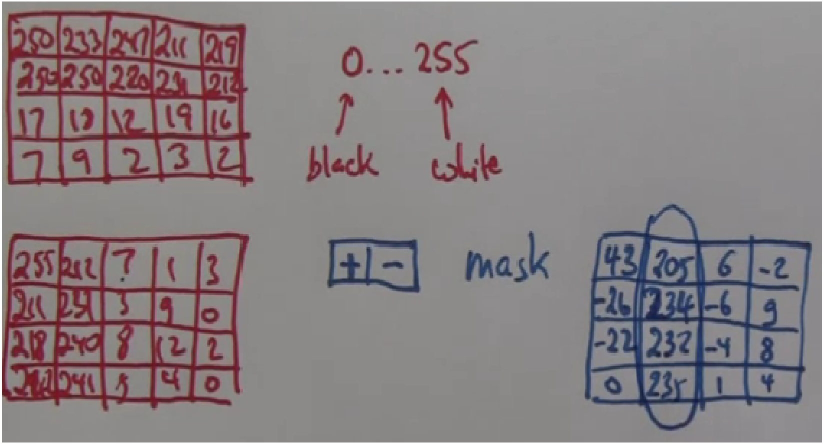


How to detect the vertical edge?

1. Spatial derivative?
2. Filter with mask:

$+1$	-1

Extracting Features: Edge Detection



How to detect the vertical edge?

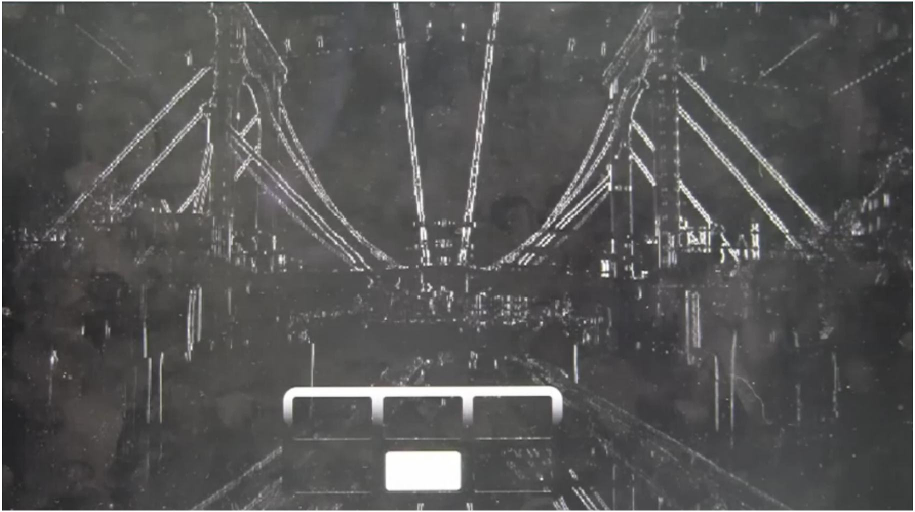
1. Spatial derivative?
2. Filter with mask: $\boxed{+1 \mid -1}$

Extracting Features: Edge Detection

How to detect the vertical edge?

1. Spatial derivative?

Extracting Features: Edge Detection



How to detect the vertical edge?

1. Spatial derivative?

Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \otimes g = I'$$

$$\otimes \quad = $$

Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \quad \otimes g = \quad I'$$

$$\otimes \ \square =$$

For each pixel, we multiply by **mask** and sum:

$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \otimes g = I'$$

$$\otimes \quad =$$

For each pixel, we multiply by **mask** and sum:

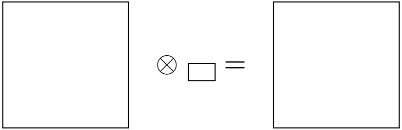
$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Does that equation look familiar?

Edge Detection: Linear Filter

What we did is called **convolution**:

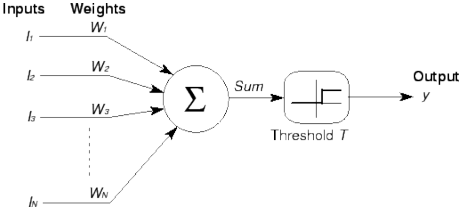
$$I \quad \otimes g = \quad I'$$



For each pixel, we multiply by **mask** and sum:

$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

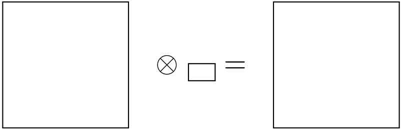
Does that equation look familiar?
Perceptron?



Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \otimes g = I'$$

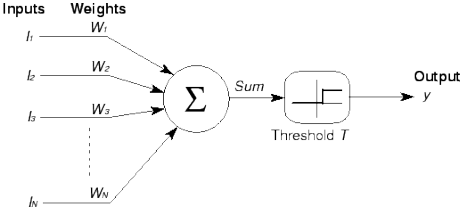


For each pixel, we multiply by **mask** and sum:

$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Does that equation look familiar?
Perceptron?

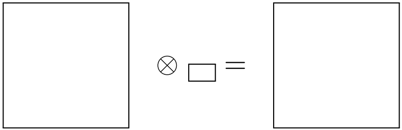
- What are the weights?



Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \quad \otimes g = \quad I'$$

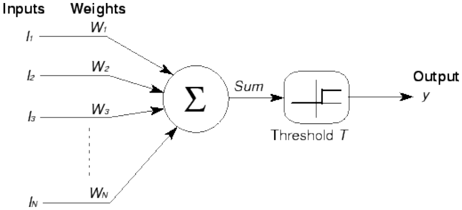


For each pixel, we multiply by **mask** and sum:

$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Does that equation look familiar? Perceptron?

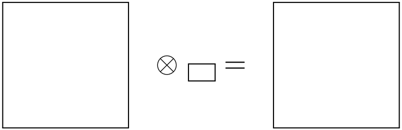
- What are the weights?
  The mask, g.



Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \quad \otimes g = \quad I'$$

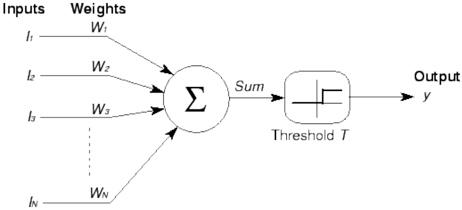


For each pixel, we multiply by **mask** and sum:

$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Does that equation look familiar?
Perceptron?

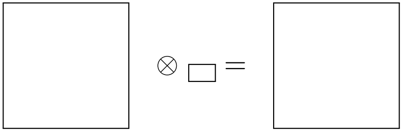
- What are the weights?
  The mask, g.
- What's the advantage?



Edge Detection: Linear Filter

What we did is called **convolution**:

$$I \otimes g = I'$$

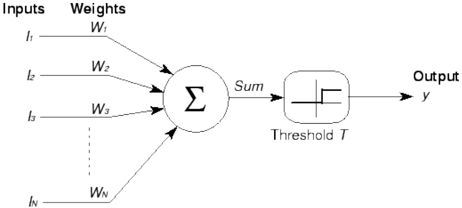


For each pixel, we multiply by **mask** and sum:

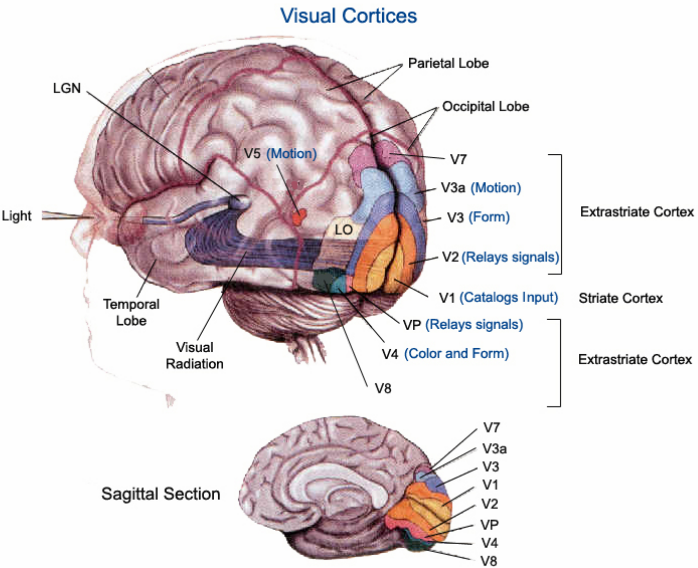
$$I'(x,y) = \sum_{u,v} I(x-u, y-v)\, g(u,v)$$

Does that equation look familiar?
Perceptron?

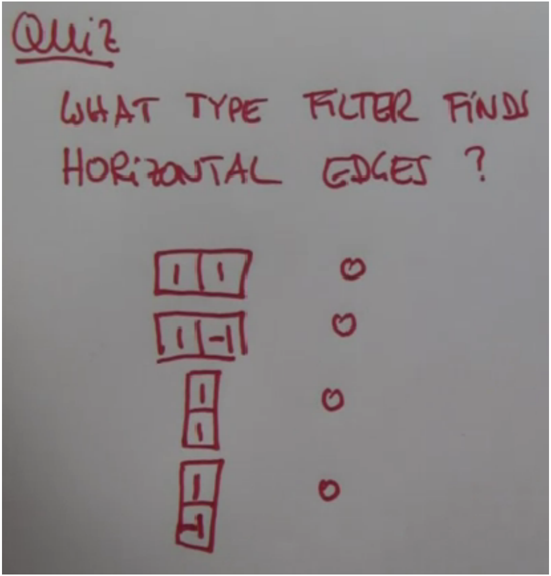
- What are the weights?
  The mask, g.
- What's the advantage?
  Works in parallel!



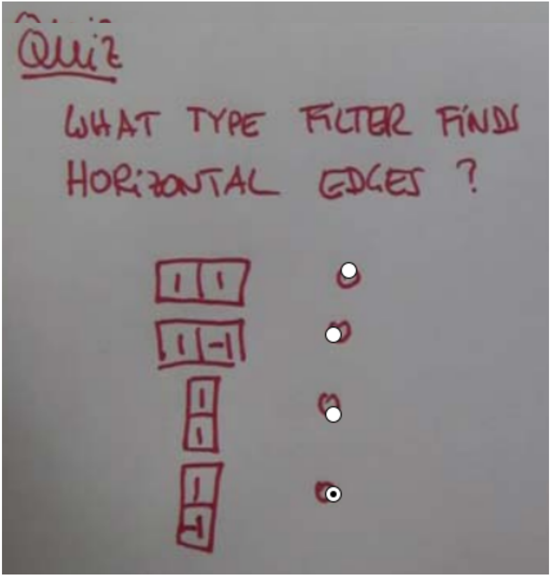
Neurons Can Do It Faster?



Detect Only Vertical Edges?



Detect Only Vertical Edges?

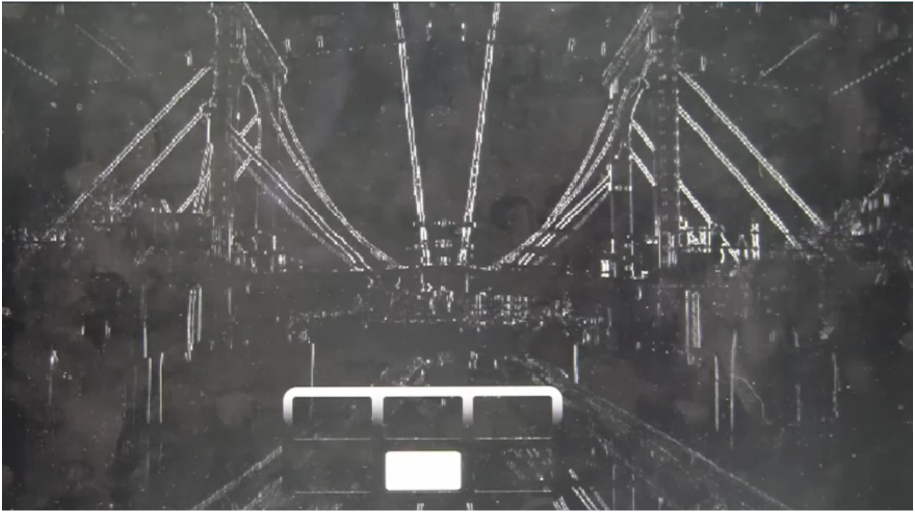


Horizontal and Vertical Gradients

Original:

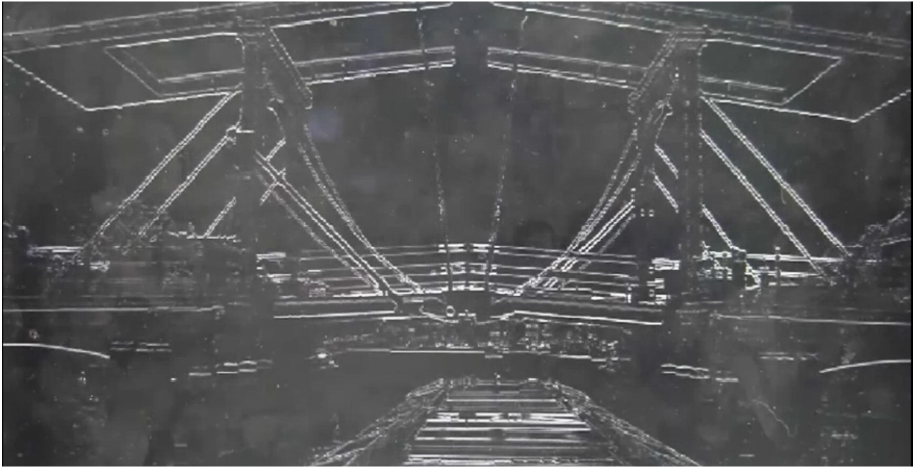
Horizontal and Vertical Gradients

Vertical gradient:



Horizontal and Vertical Gradients

Horizontal gradient:



Combining Gradients

Horizontal mask gives vertical gradient (I_x) and vice versa:

$$I_x = I \otimes \boxed{\begin{array}{cc} -1 & +1 \end{array}}$$

$$I_y = I \otimes \boxed{\begin{array}{c} -1 \\ +1 \end{array}}$$

Combining Gradients

Horizontal mask gives vertical gradient (I_x) and vice versa:

$$I_x = I \otimes \boxed{\begin{array}{cc} -1 & +1 \end{array}}$$

$$I_y = I \otimes \boxed{\begin{array}{c} -1 \\ +1 \end{array}}$$

How to combine them?

Combining Gradients

Horizontal mask gives vertical gradient (I_x) and vice versa:

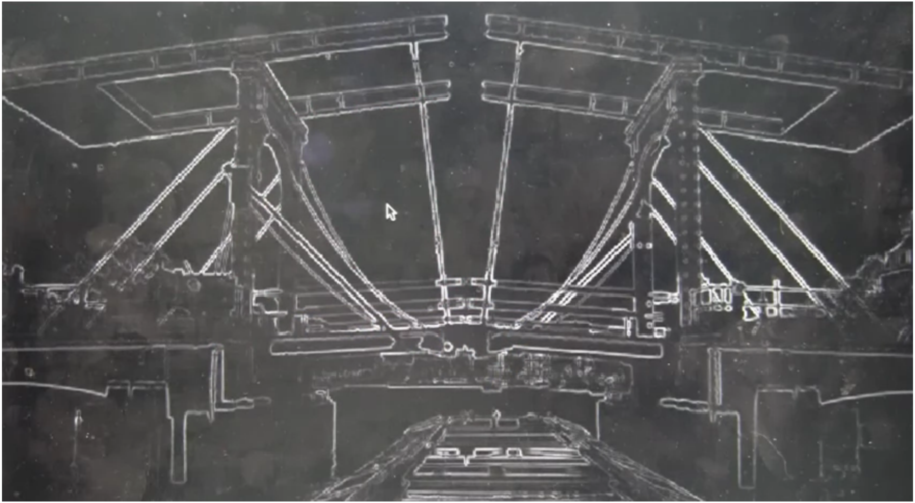
$$
\begin{aligned}
I_x &= I \otimes \boxed{\begin{matrix} -1 & +1 \end{matrix}} \\
I_y &= I \otimes \boxed{\begin{matrix} -1 \\ +1 \end{matrix}}
\end{aligned}
$$

How to combine them?

$$E = \sqrt{I_x^2 + I_y^2}$$

Horizontal and Vertical Gradients

Combined gradients:

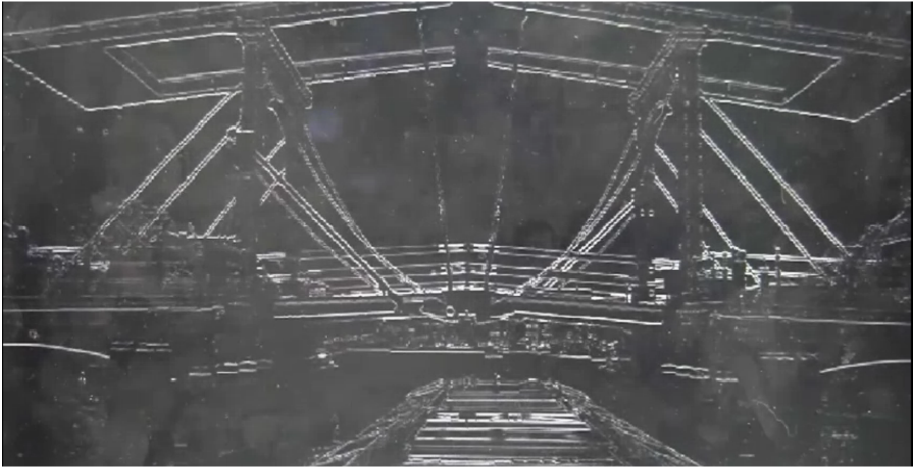


Horizontal and Vertical Gradients

Original:

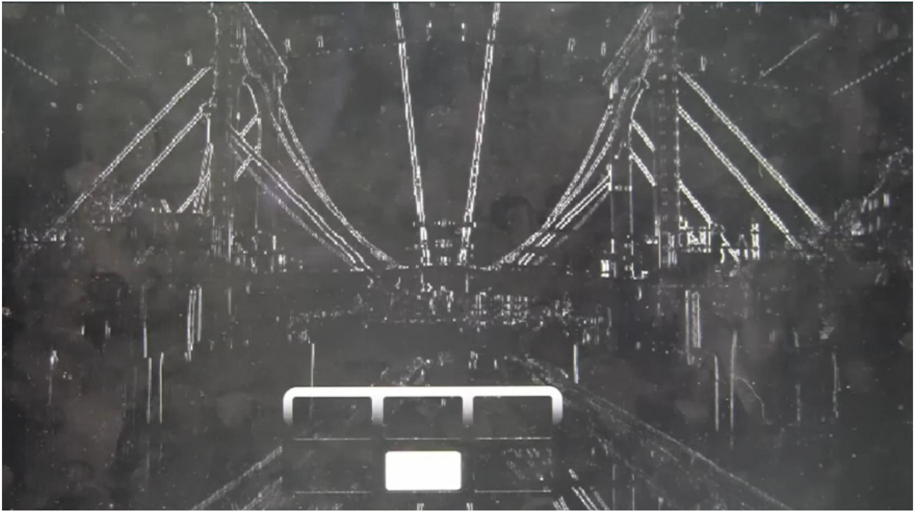
Horizontal and Vertical Gradients

Horizontal gradient:



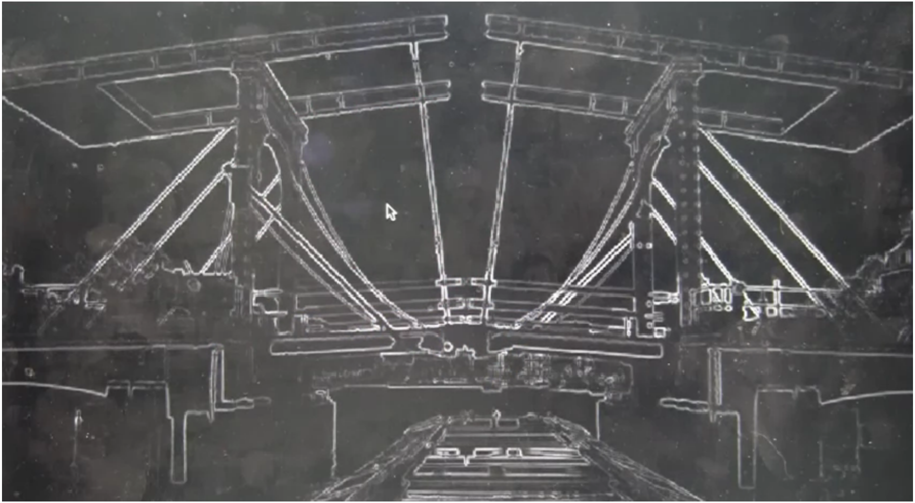
Horizontal and Vertical Gradients

Vertical gradient:



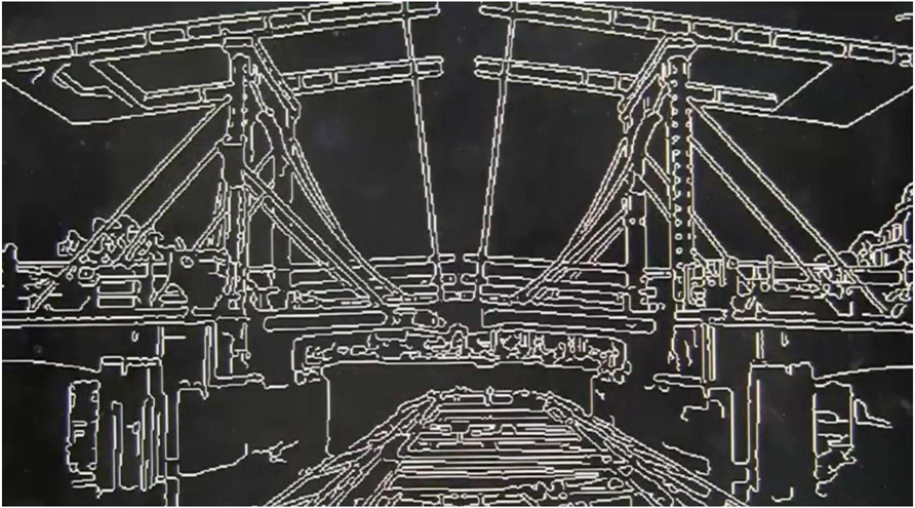
Canny Edge Detector is Uncanny!

Combined gradients:

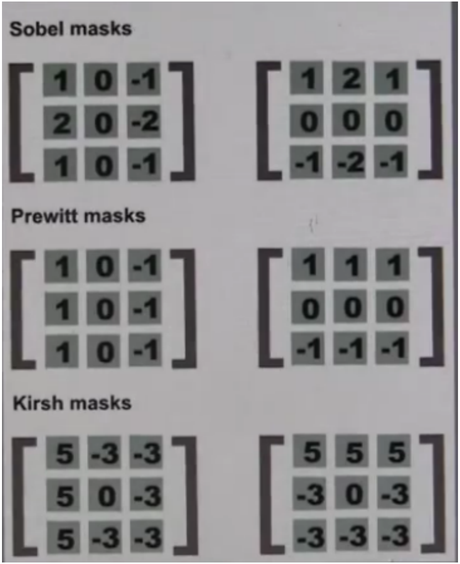


Canny Edge Detector is Uncanny!

Canny edge detector (by John Canny):



Other Edge Detection Masks



Sobel masks

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

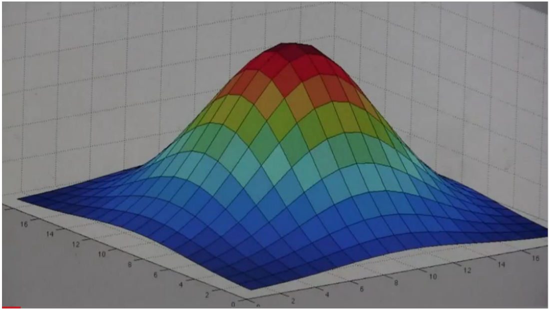
Prewitt masks

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

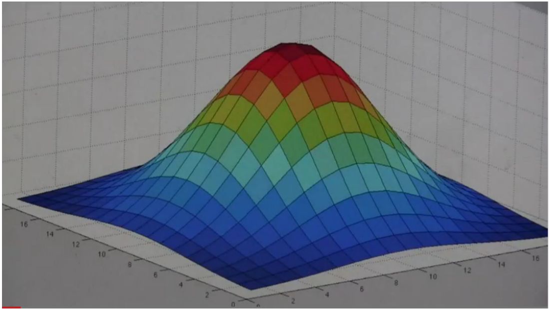
Kirsh masks

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \qquad \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

A Gaussian Mask?



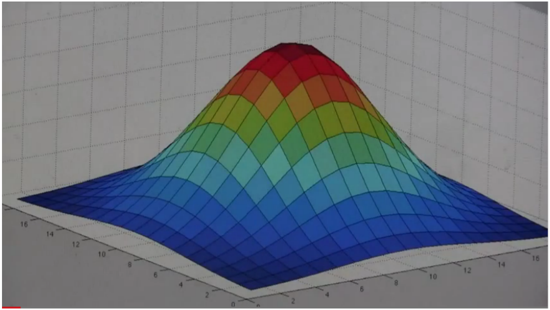
A Gaussian Mask?



What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. Blur
5. Sharpen

A Gaussian Mask?

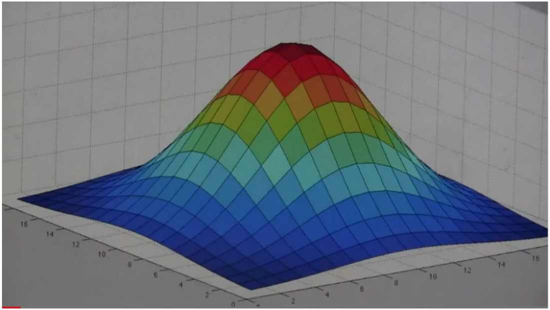


What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. **Blur**
5. Sharpen

What's the Point of Blurring Images?

A Gaussian Mask?



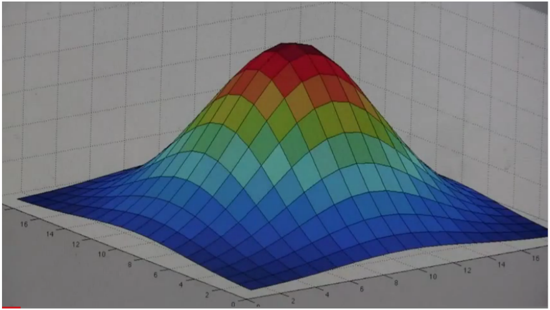
What will it do?

1. Edge filter
2. Dot filter
3. Corner
4. **Blur**
5. Sharpen

What's the Point of Blurring Images?

1. Downsampling

A Gaussian Mask?



What will it do?

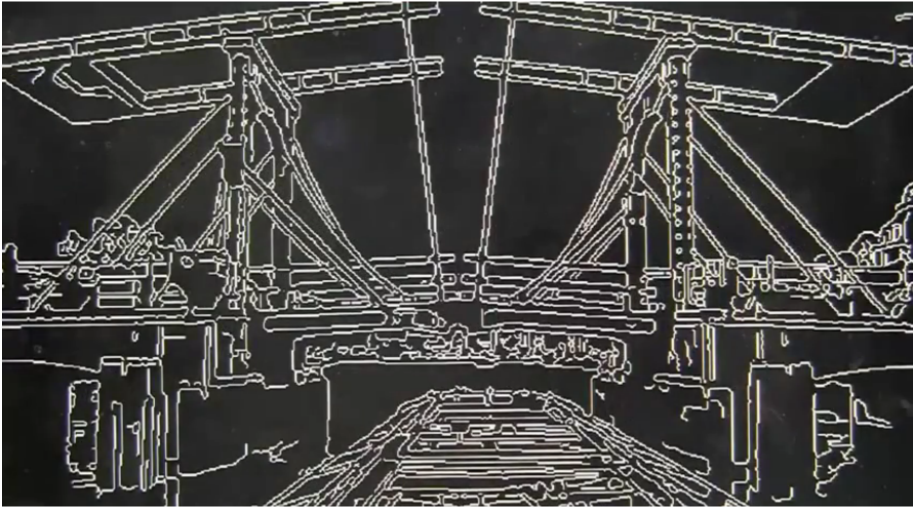
1. Edge filter
2. Dot filter
3. Corner
4. Blur
5. Sharpen

What's the Point of Blurring Images?

1. Downsampling
2. Noise reduction

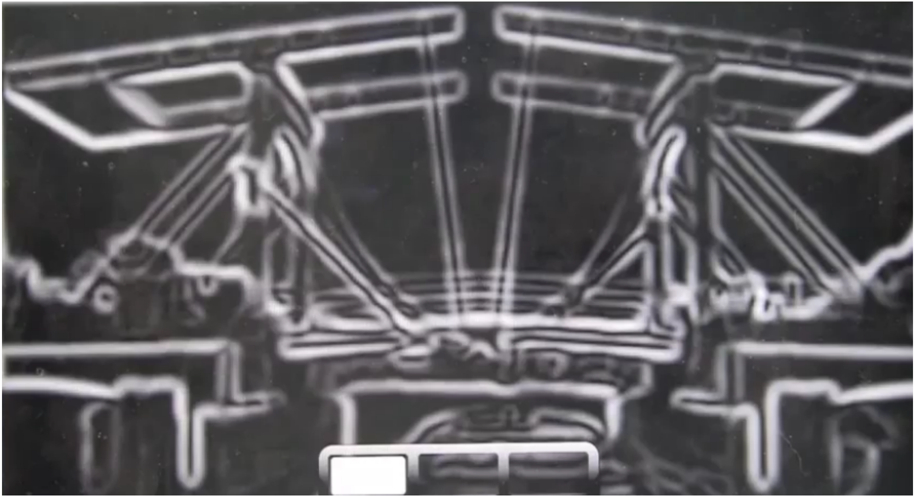
Gaussian Mask in Action

Canny filter:



Gaussian Mask in Action

Canny with Gaussian:



Tricks with Linear Filters

$$I' = I \otimes f \otimes g$$

where

- f is Gaussian mask and
- g is gradient mask.

Tricks with Linear Filters

$$I' = I \otimes f \otimes g$$

where

f is Gaussian mask and

g is gradient mask.

Does the order matter?

Tricks with Linear Filters

$$I' = I \otimes f \otimes g$$

where

f is Gaussian mask and

g is gradient mask.

Does the order matter? **No.** Linear operations are transitive.

$$= I \otimes g \otimes f$$

Tricks with Linear Filters

$$I' = I \otimes f \otimes g$$

where

f is Gaussian mask and

g is gradient mask.

Does the order matter? **No.** Linear operations are transitive.

$$= I \otimes g \otimes f$$

Can we combine them?

Tricks with Linear Filters

$$I' = I \otimes f \otimes g$$

where

f is Gaussian mask and

g is gradient mask.

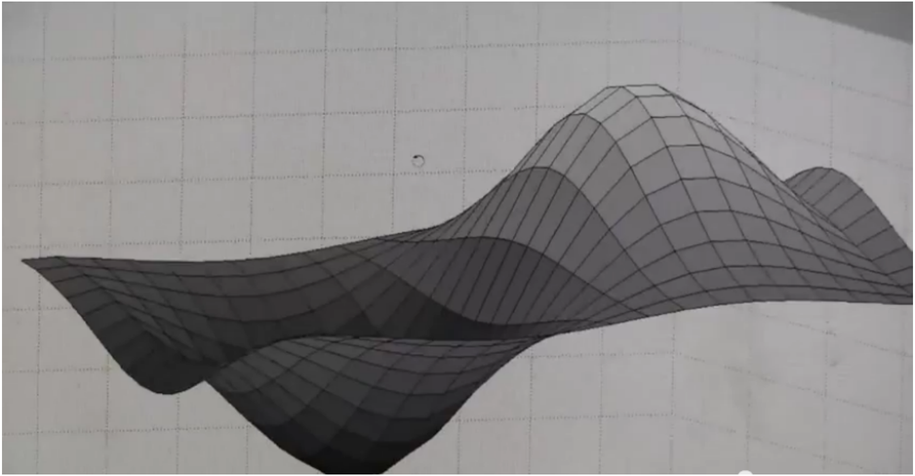
Does the order matter? **No.** Linear operations are transitive.

$$= I \otimes g \otimes f$$

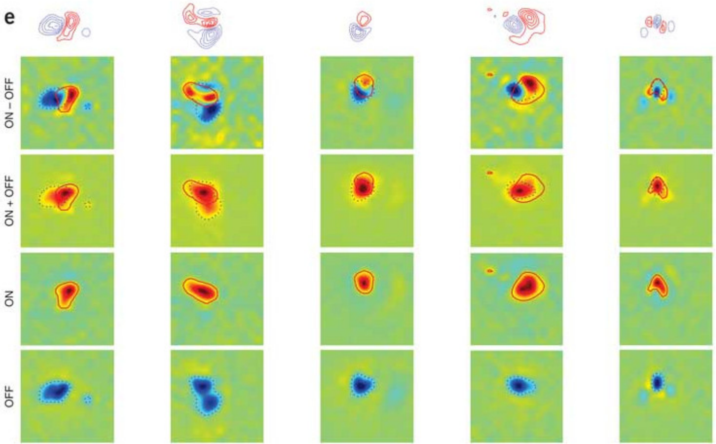
Can we combine them? **Yes.** We'll get a new linear mask/kernel.

$$= I \otimes (f \otimes g)$$

Gaussian Mask Combined with Gradient



Neurons Are Doing Exactly That!

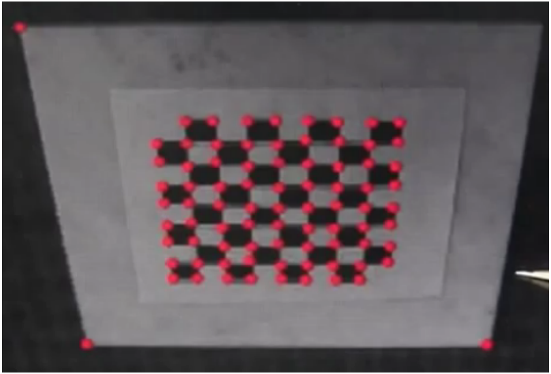


J Jin, Y Wang, HA Swadlow & JM Alonso (2011)

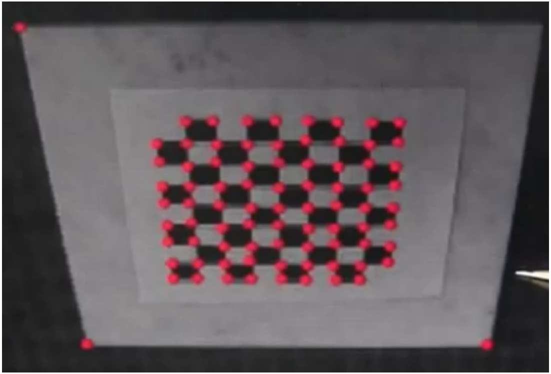
"Population receptive fields of ON and OFF thalamic inputs to an orientation column in visual cortex"

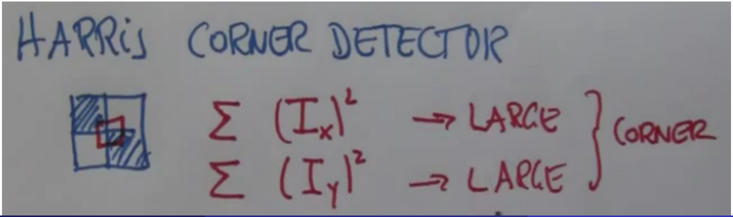
Nature Neuroscience 14(2): 232–238. doi:10.1038/nn.2729

Corner Detection

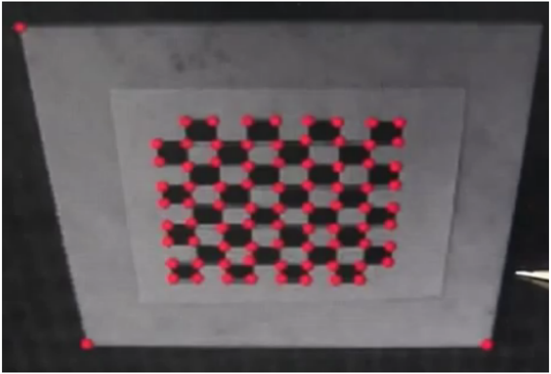


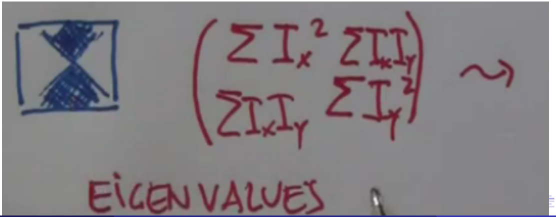
Corner Detection





Corner Detection





Modern Feature Detectors

They are:

1. Localizable
2. Unique signatures

Modern Feature Detectors

They are:

1. Localizable
2. Unique signatures

Two major algorithms:

1. HOG: Histogram of Oriented Gradients
2. SiFT: Scale-invariant Feature Transform

SiFT in Action