CEng 583 - Computational Vision

2012 Spring

Week – 2

5th of March, 2012

Tentative Schedule:

Week & Date		Topic
0	20th of Feb.	**No Lectures**
1	27th of Feb.	**Introduction to the Course & Vision.** What is vision? What are its goals and problems? What are the main processing stages?
2	5th of March	**Low-level Vision.** Cameras. Projective geometry. Calibration.
3	12th of March	**Early Vision.** Edges. Corners. Texture. Segmentation. Optic Flow.
4	19th of March	**3D Vision.** Monocular and binocular cues. 3D reconstruction.
5	26th of March	**Applications.** Video surveillance. Human behaviour understanding. Object recognition. Image/video retrieval. Image annotation.
6	2nd of April	**Paper presentations with theme**: Monocular depth estimation.
7	9th of April	**Paper presentations with theme**: Image annotation.
8	16th of April	**Paper presentations with theme**: Object/shape modelling. Object recognition.
9	23rd of April	**Paper presentations with theme**: Feature Descriptors.
10	30th of April	**Paper presentations with theme**: Context. Saliency. Attention.
11	7th of May	**Project Presentations**
12	14th of May	**Project presentations**
13	21st of May	**Project presentations**
14	28th of May	**Project presentations**

Today

- Low-level vision.
- Pre-vision & Low-level vision.
  - Cameras
  - Projective geometry
  - Calibration
  - **Early vision**

First of all, what is ‘low-level’ and ‘high-level’ vision?

- .. and, while we are at it:
  - What is Marr’s “vision system” and how does this course follow that?

Low-level → High-level

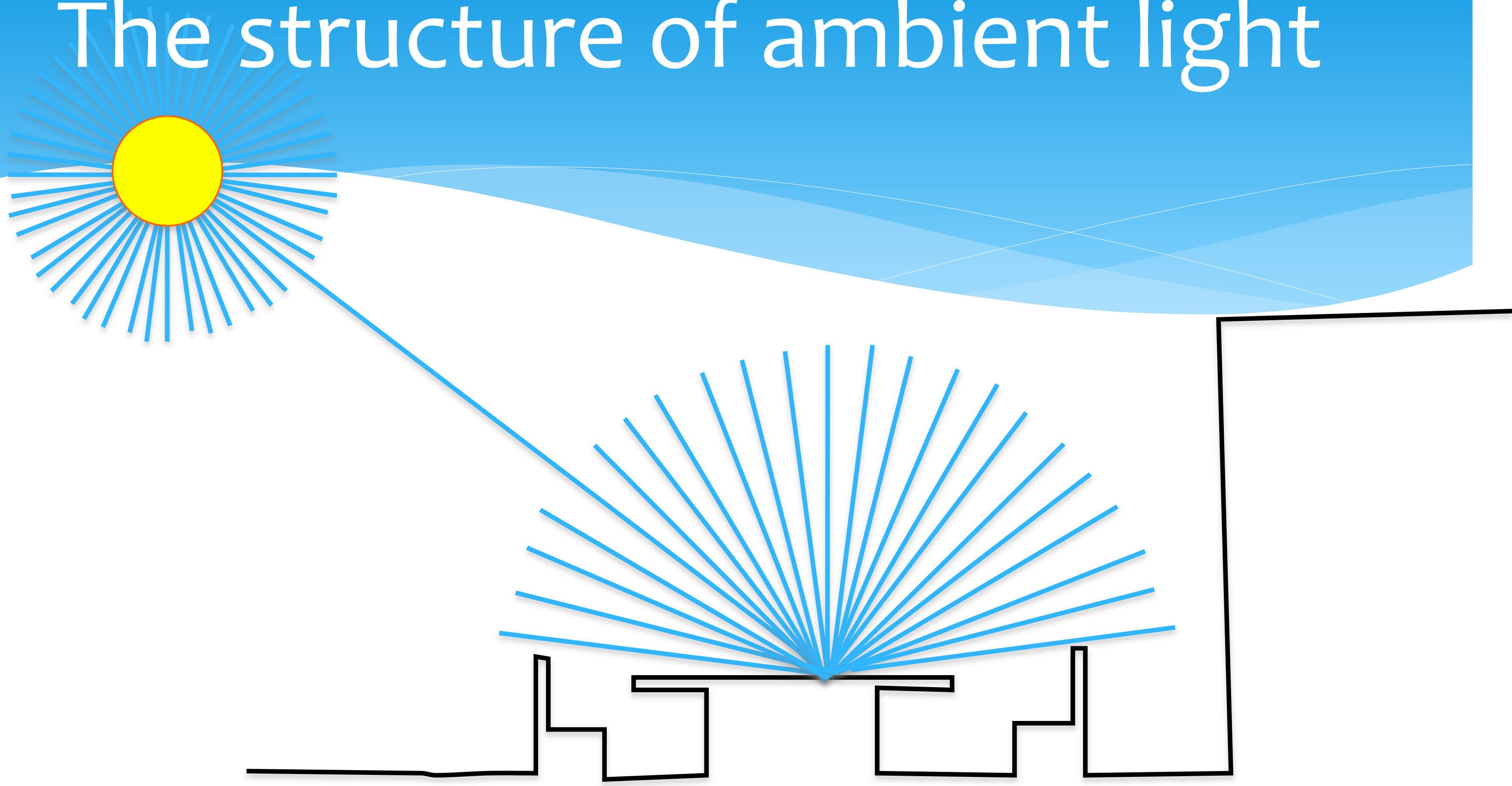
Low-level **High-level**

- Dense
- Raw
- Local
- Not interpretable

- Sparse
- Abstract
- Global
- Explicit, interpretable

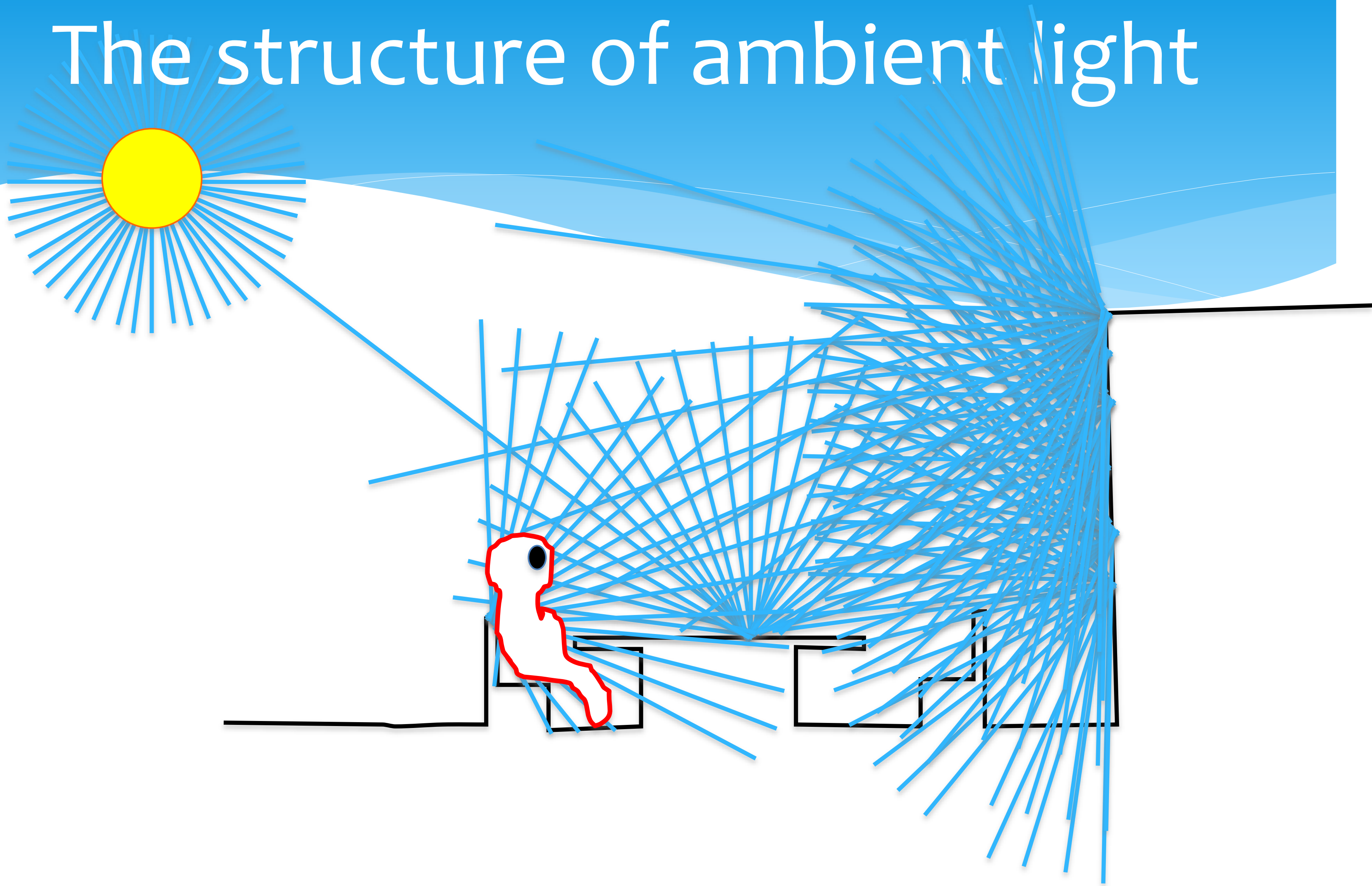
The environment and what we get from it as images.

The structure of ambient light



Slide: A. Torralba

The structure of ambient light



Slide: A. Torralba

The Plenoptic Function

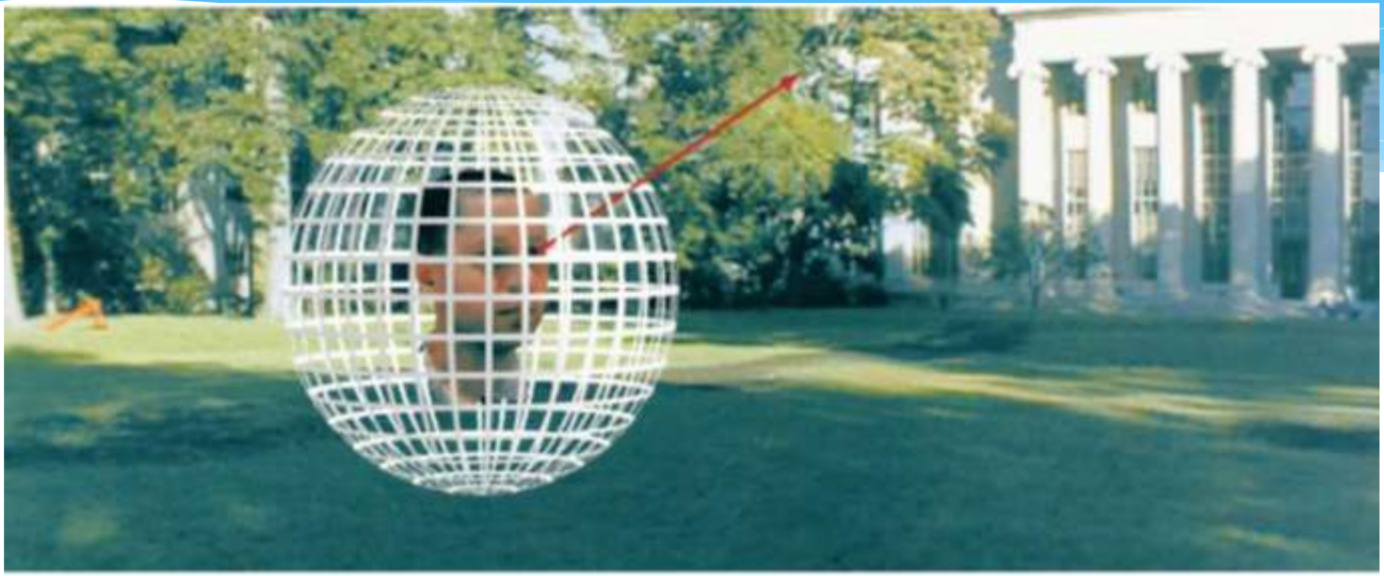
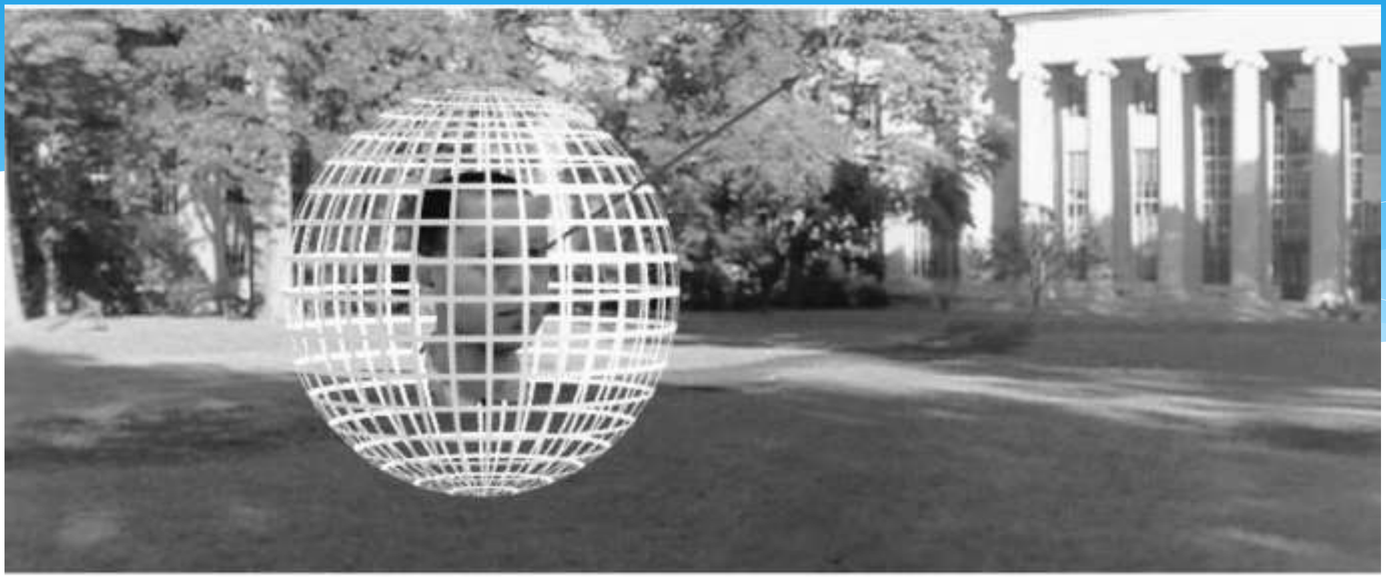


Figure by Leonard McMillan

- Q: What is the set of all things that we can ever see?
- A: The Plenoptic Function (Adelson & Bergen)

Slide: A. Efros

Grayscale snapshot

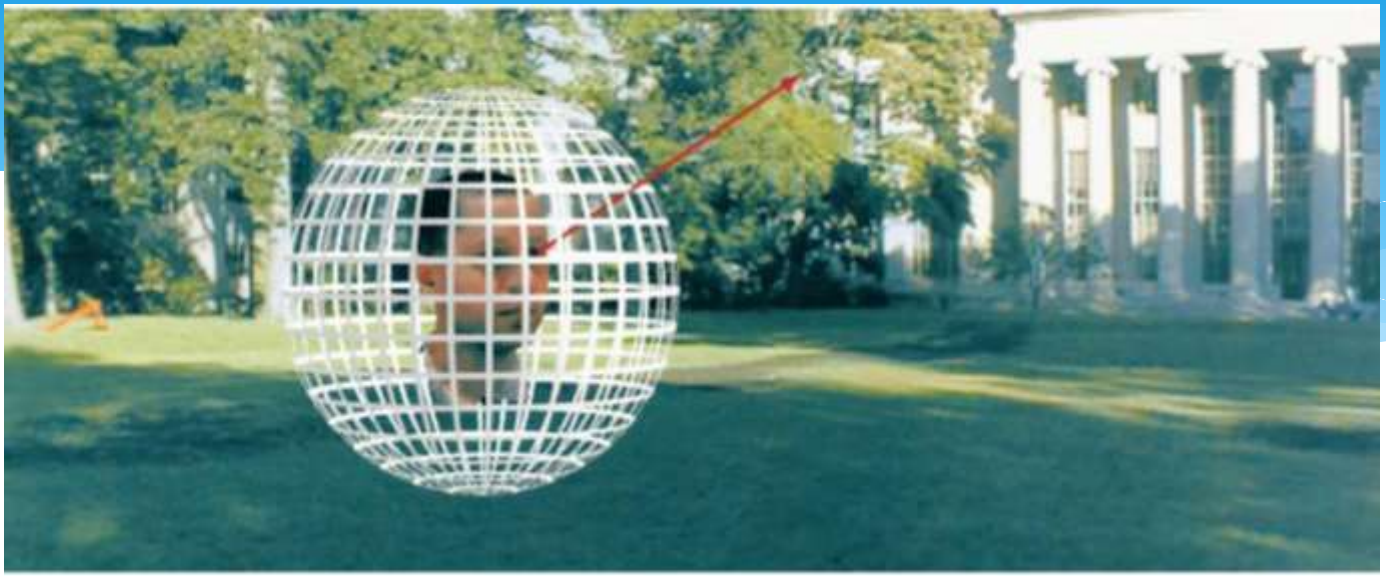


$$P(\theta,\phi)$$

- intensity of light
  - Seen from a single view point

Slide: A. Efros

Color snapshot

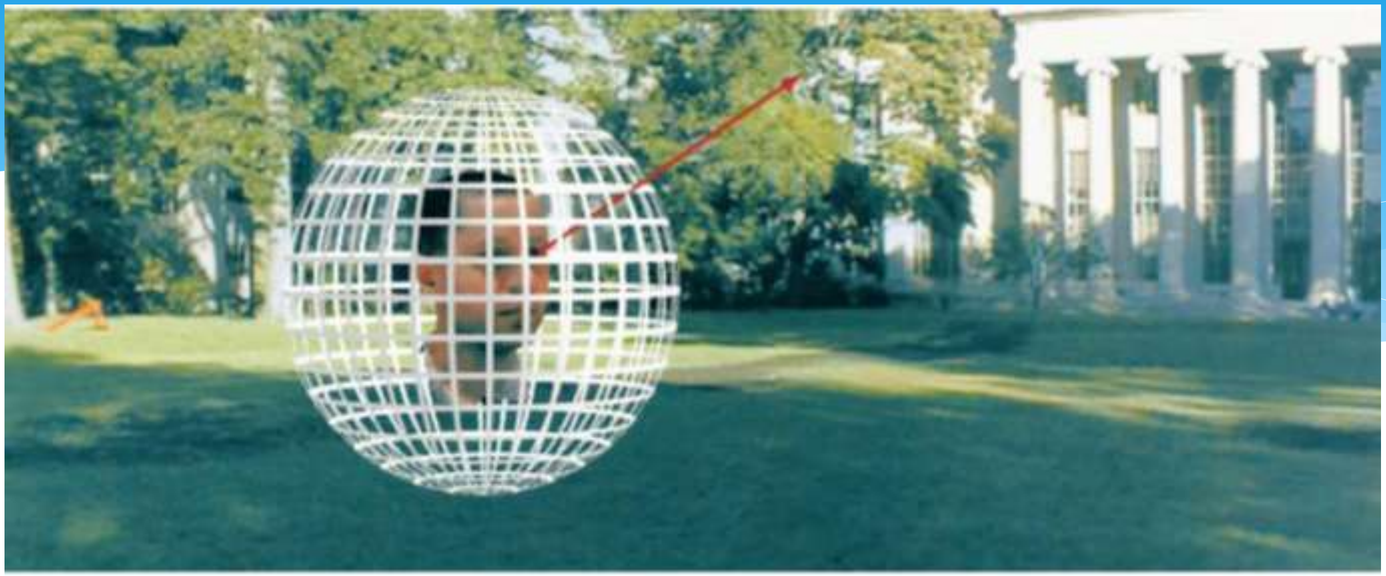


$$P(\theta,\phi,\lambda)$$

* is intensity of light
  * Seen from a single view point
  * At a single time
  * As a function of wavelength

Slide: A. Efros

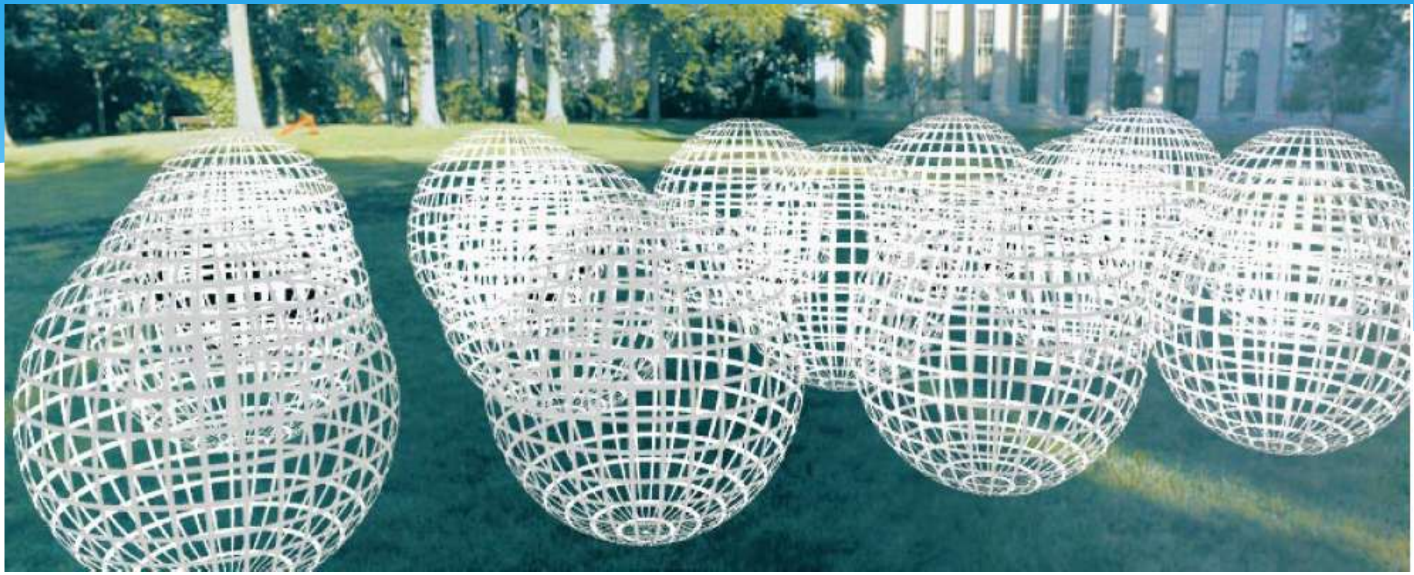
A movie



$$P(\theta,\phi,\lambda,t)$$

- is intensity of light
  - Seen from a single view point
  - Over time
  - As a function of wavelength

Slide: A. Efros



$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

- is intensity of light
  - Seen from ANY viewpoint
  - Over time
  - As a function of wavelength

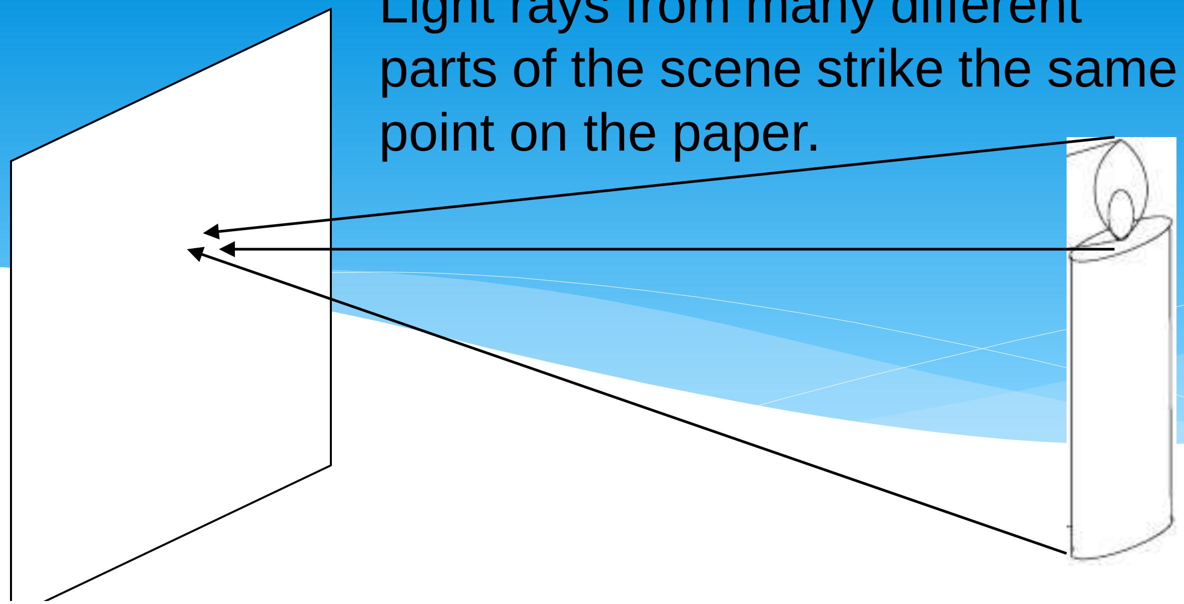
Slide: A. Efros

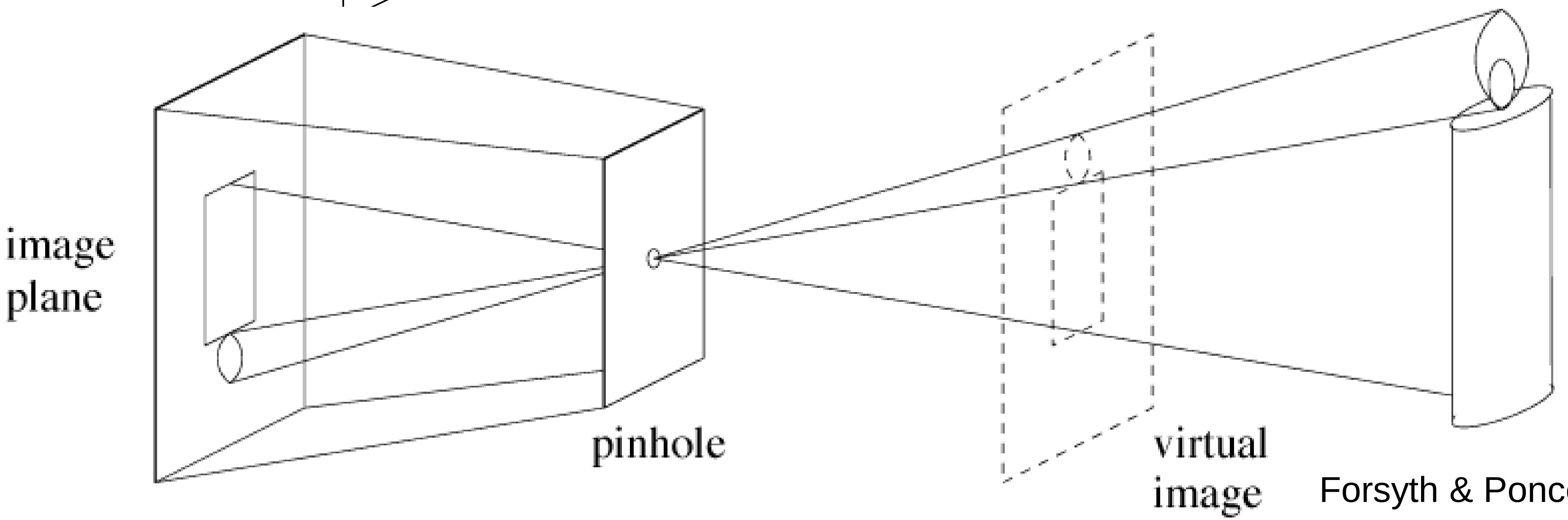
Vision is to estimate the Plenoptic Function.

$$I(x,y) \rightarrow P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

Cameras

Light rays from many different parts of the scene strike the same point on the paper.



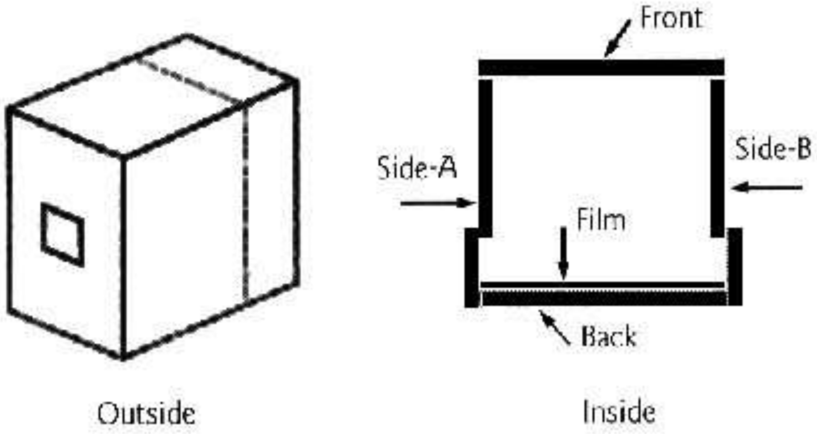


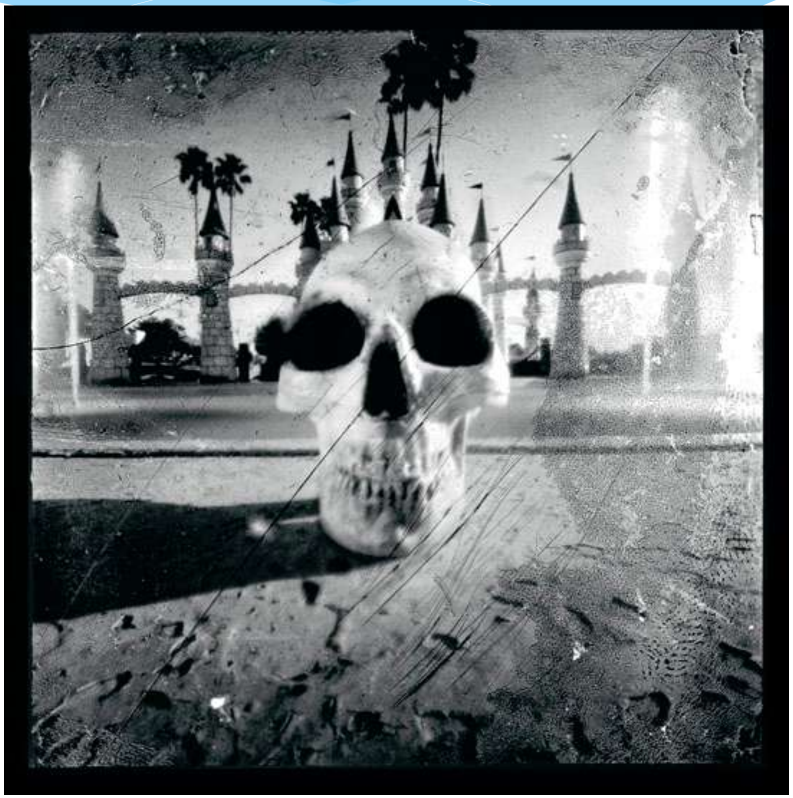
Forsyth & Ponce

The pinhole camera only allows rays from one point in the scene to strike each point of the paper.

Slide: A. Torralba

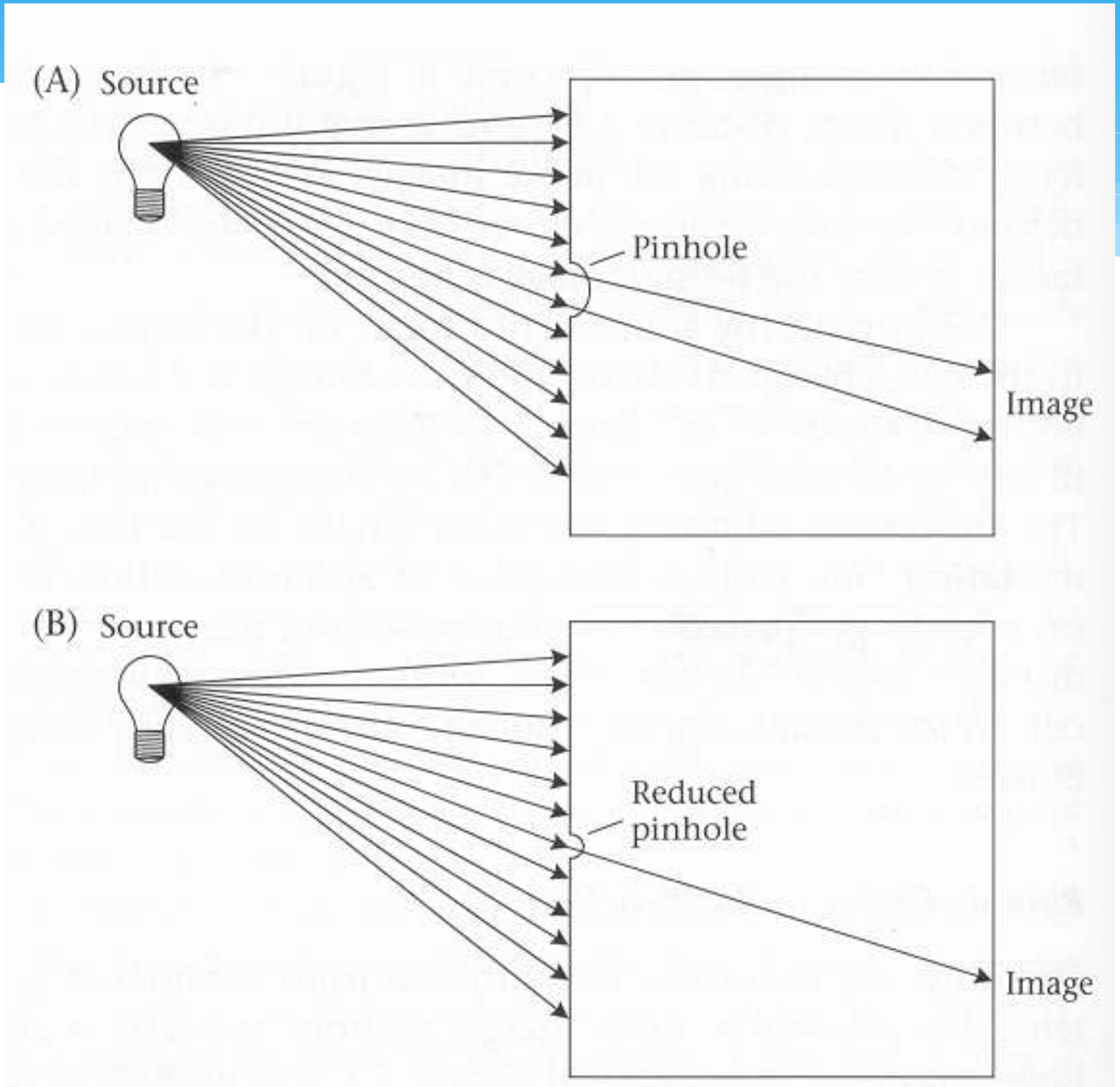
Construct your own camera





http://www.pinhole.org/make/index.cfm

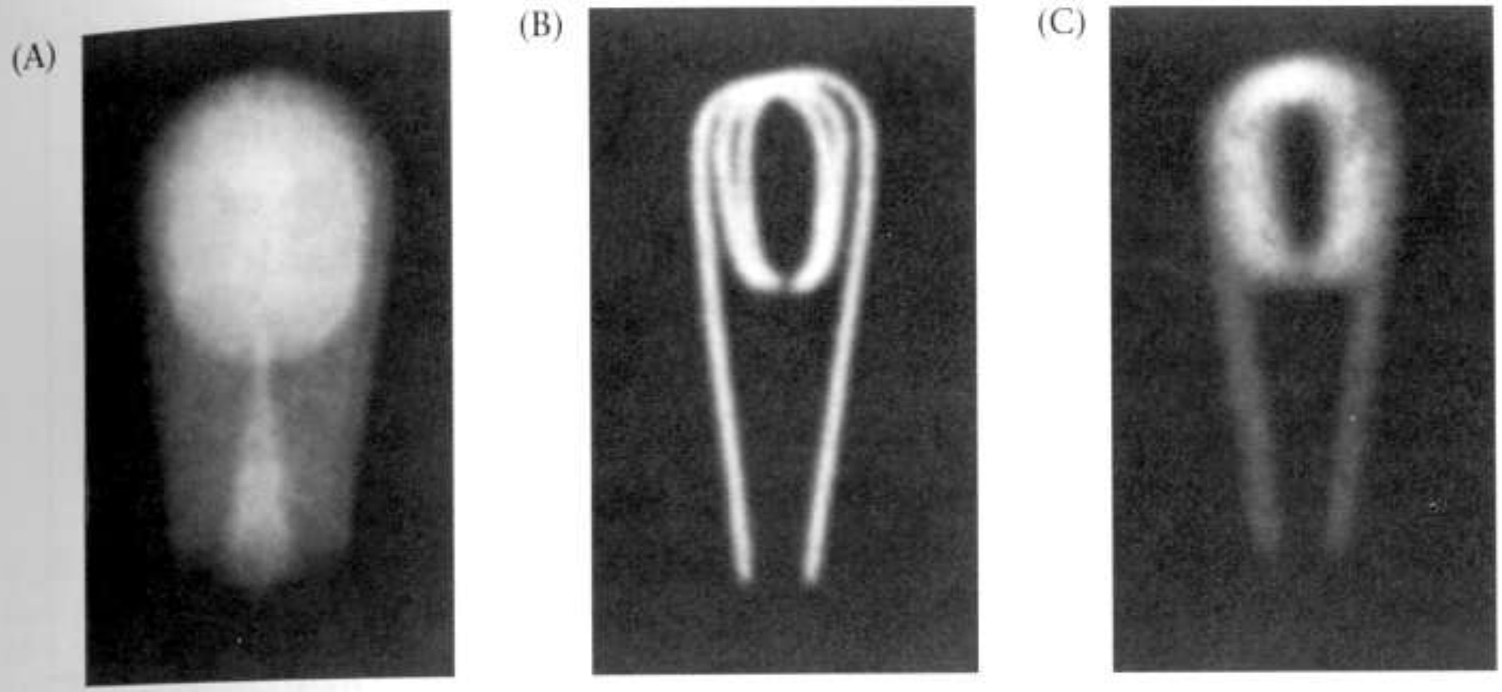
Effect of pinhole size



Wandell, Foundations of Vision, Sinauer, 1995

Slide: A. Torralba

Effect of pinhole size

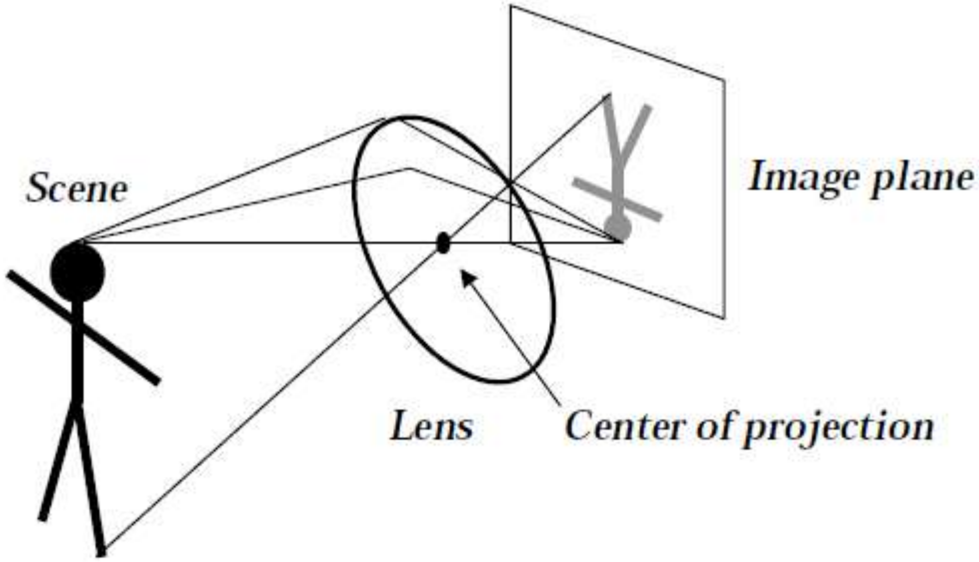


2.18 DIFFRACTION LIMITS THE QUALITY OF PINHOLE OPTICS. These three images of a bulb filament were made using pinholes with decreasing size. (A) When the pinhole is relatively large, the image rays are not properly converged, and the image is blurred. (B) Reducing the size of the pinhole improves the focus. (C) Reducing the size of the pinhole further worsens the focus, due to diffraction. From Ruechardt, 1958.

Wandell, Foundations of Vision, Sinauer, 1995

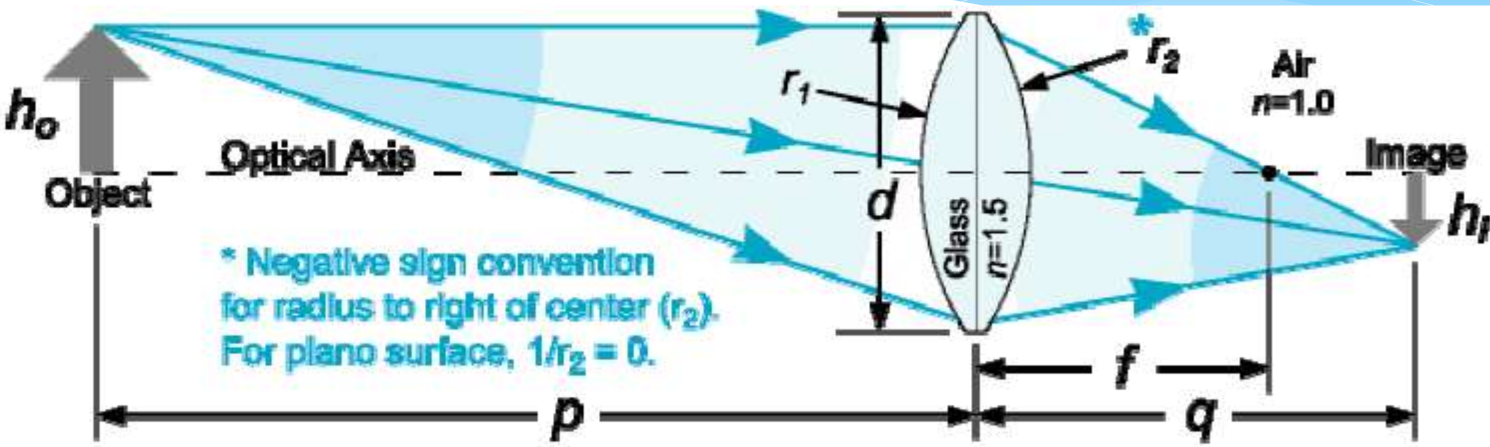
Slide: A. Torralba

Extending Cameras



Slide: L. Davis, R. Duraiswami, D. DeMenthon, and C. Fermüller

Extending Cameras



Lens Equation:

$$\frac{1}{f}=\frac{1}{p}+\frac{1}{q}$$

Lens Maker's Equation:

$$\frac{1}{f}=(n-1)\left(\frac{1}{r_1}-\frac{1}{r_2}\right)$$

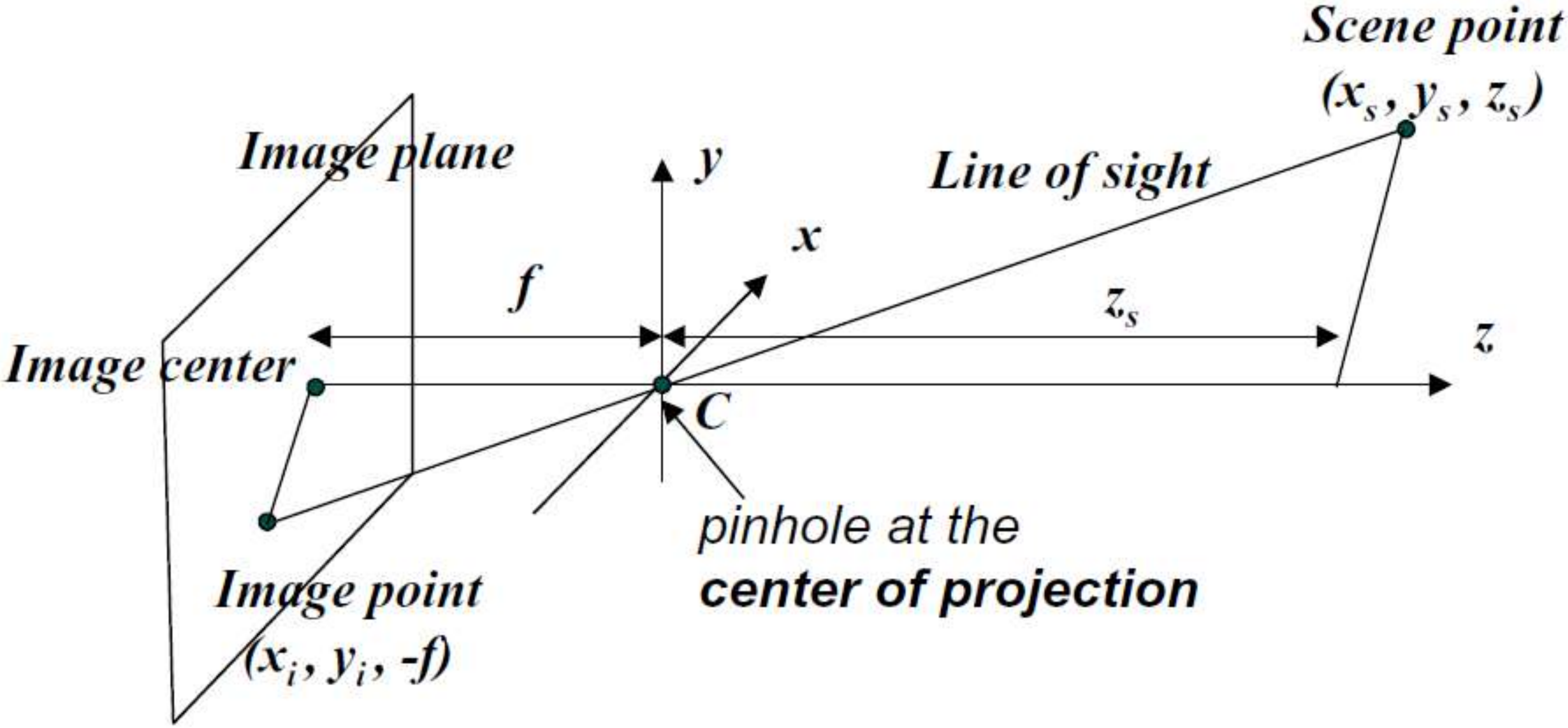
Magnification:

$$m=\frac{h_i}{h_o}=-\frac{q}{p}$$

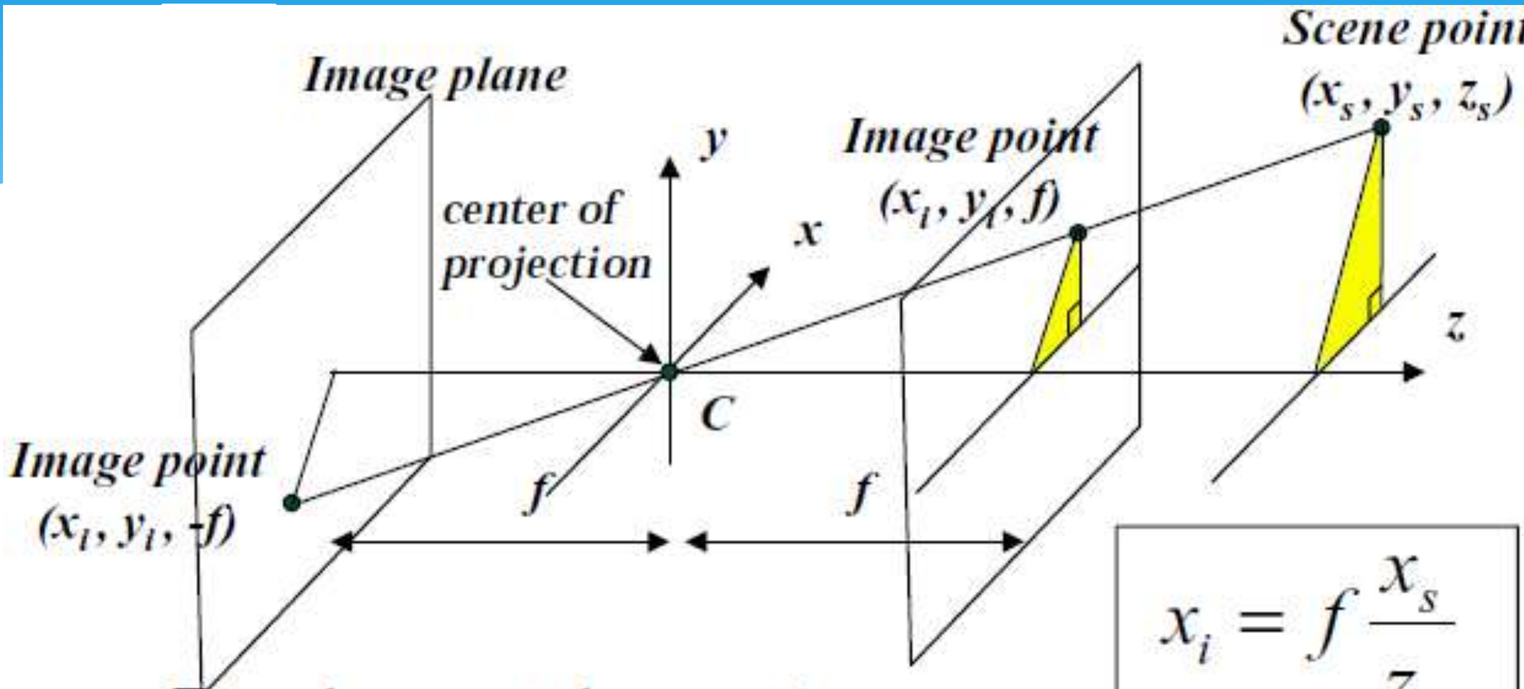
F-number:

$$f/\#=\frac{f}{d}$$

Slide: L. Davis, R. Duraiswami, D. DeMenthon, and C. Fermüller



Slide: L. Davis, R. Duraiswami, D. DeMenthon, and C. Fermüller

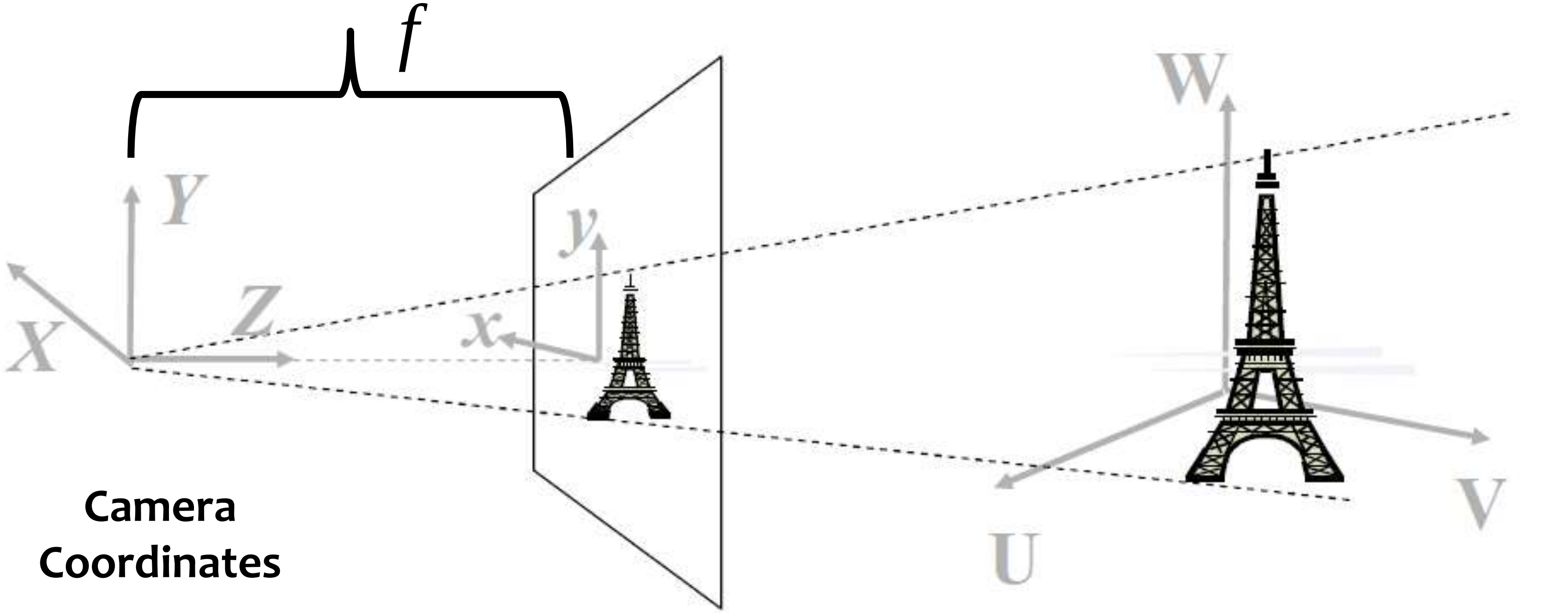


- Fundamental equations for perspective projection onto a plane

$$x_i = f\frac{x_s}{z_s}$$

$$y_i = f\frac{y_s}{z_s}$$

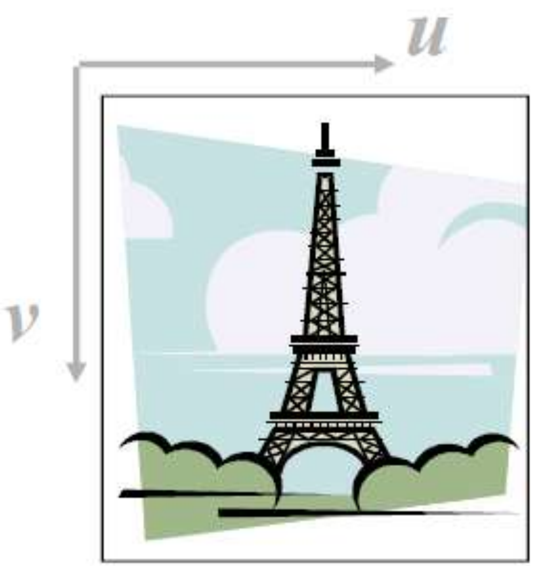
Slide: L. Davis, R. Duraiswami, D. DeMenthon, and C. Fermüller



Camera Coordinates

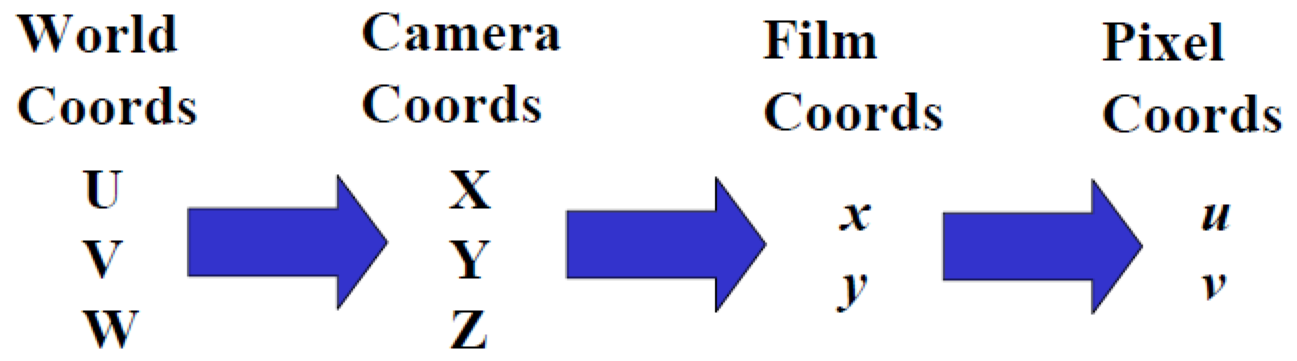
Image/Film Coordinates

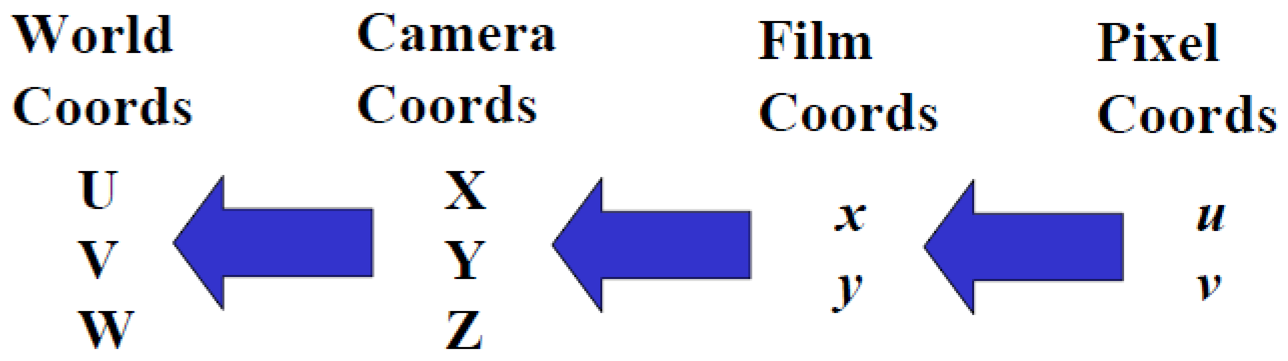
World Coordinates



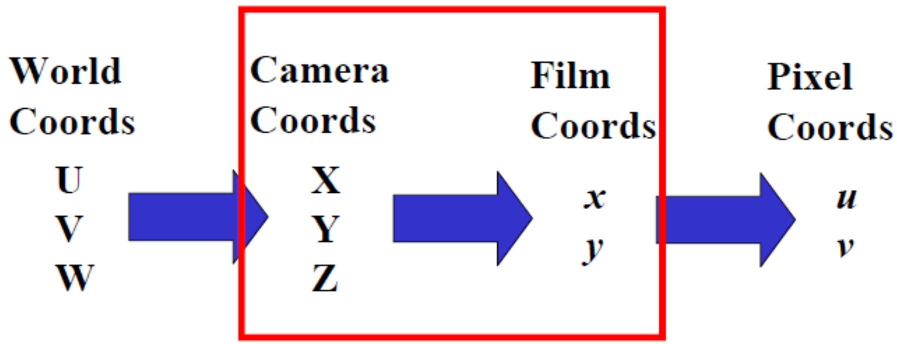
Pixel Coordinates

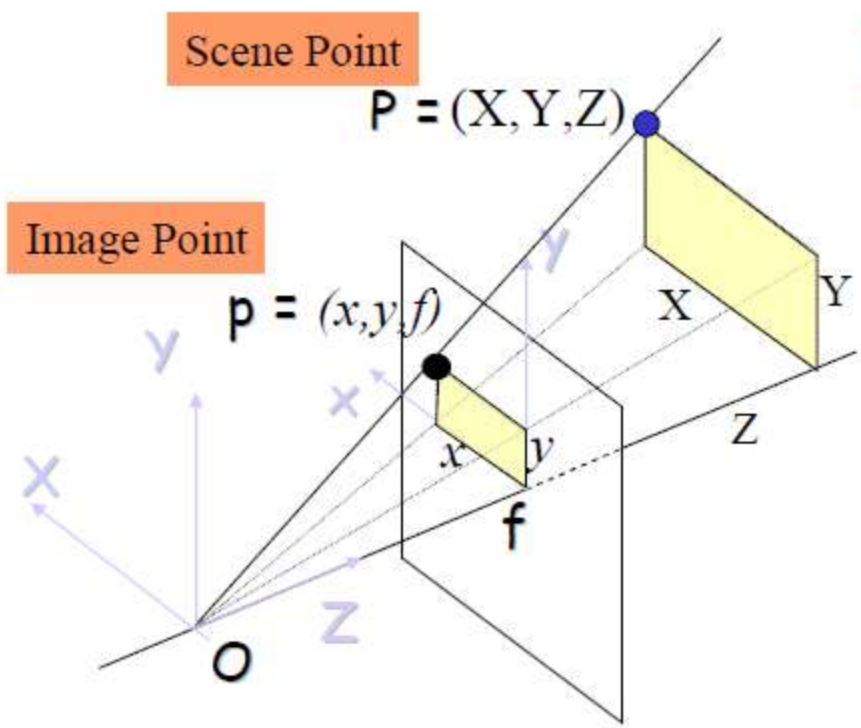
Slide: R. Collins





Slide: R. Collins





Perspective Projection Eqns

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

$$y = f\frac{Y}{Z}$$

Slide: R. Collins

How to represent this as a matrix equation?

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

$$\Longleftrightarrow$$

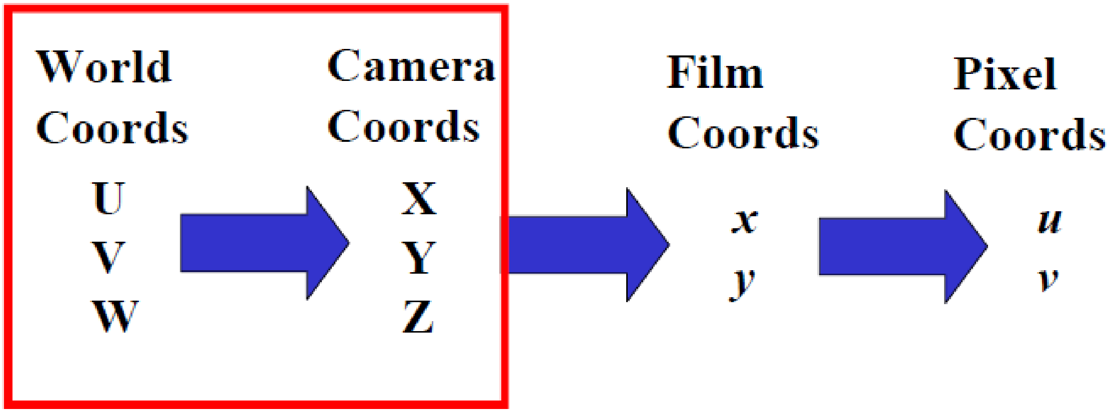
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

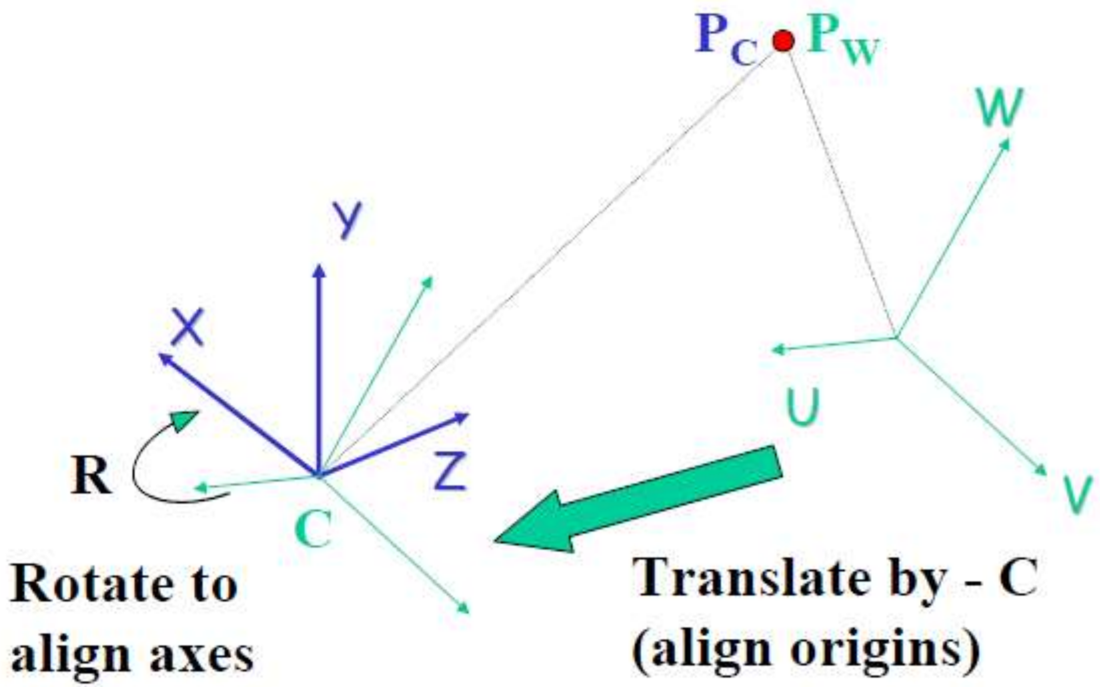
Homogeneous Coordinates

Slide: R. Collins

What is a ‘homogeneous coordinate’? Sounds fishy…

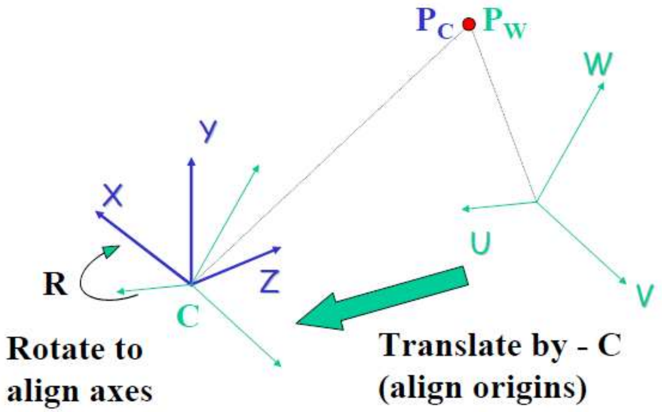
- Add another dimension to the existing coordinate.
  - 2D case: (x,y) → (x’, y’, z’)
  - 3D case: (x, y, z) → (x’, y’, z’, w’)
- such that:
  - 2D case: x = x’ / z’ and y = y’ / z’
  - 3D case: x = x’ / w’ and y = y’ / w’ and z = z’ / w’
- So, (x, y) → (x, y, 1) = (2x, 2y, 2) = (3x, 3y, 3) …





$$P_C = R\,(\,P_W - C\,)$$

Slide: R. Collins

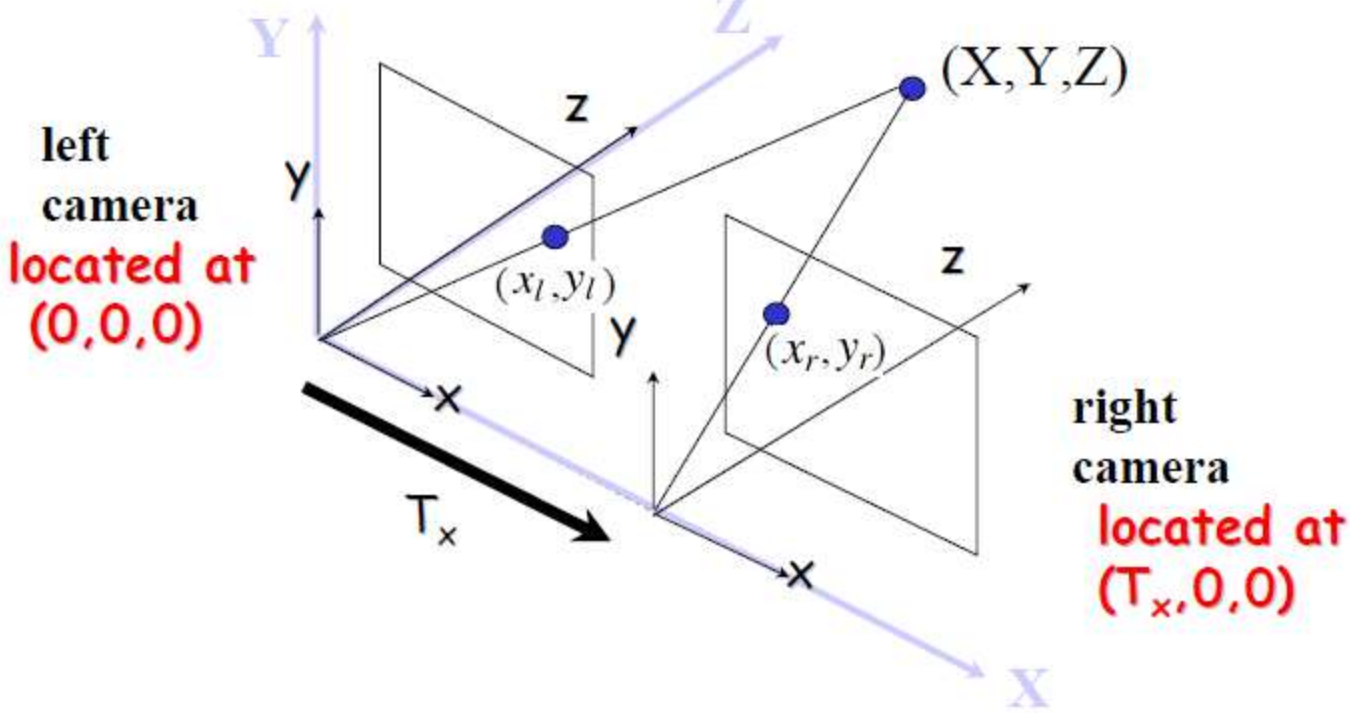


$$P_C = R\,(P_W - C)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

Slide: R. Collins

Example: A simple stereo system



Slide: R. Collins

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

camera axes aligned with world axes

located at world position (0,0,0)

Left camera

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_l = f\frac{X}{Z} \qquad y_l = f\frac{Y}{Z}$$

Slide: R. Collins

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

camera axes aligned with world axes

located at world position (T_x,0,0)

Right camera

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

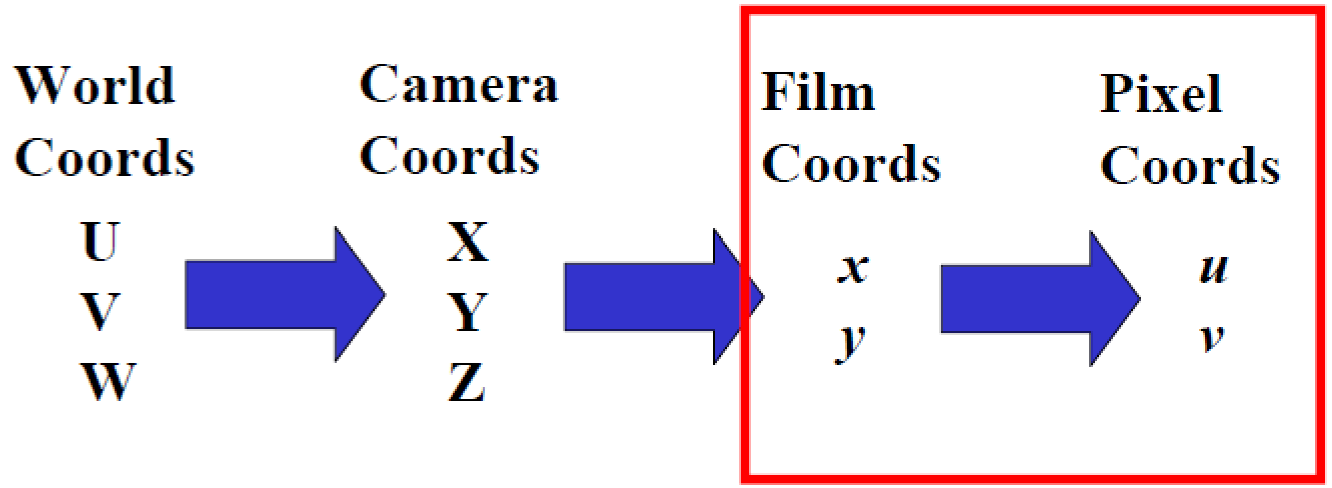
$$x_r = f\,\frac{X - T_x}{Z} \qquad y_r = f\,\frac{Y}{Z}$$

Slide: R. Collins

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

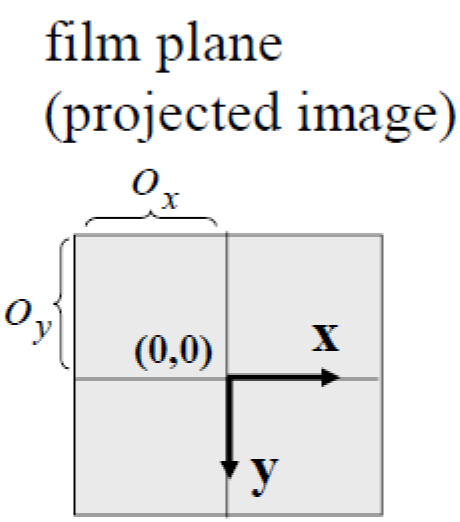
Slide: R. Collins

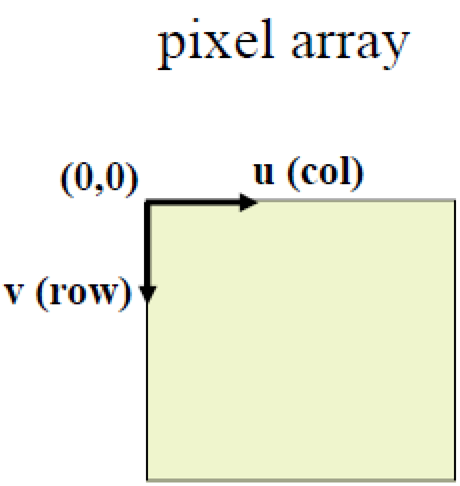
Intrinsic Camera Parameters: Affine Transformation



Slide: R. Collins

Intrinsic Camera Parameters



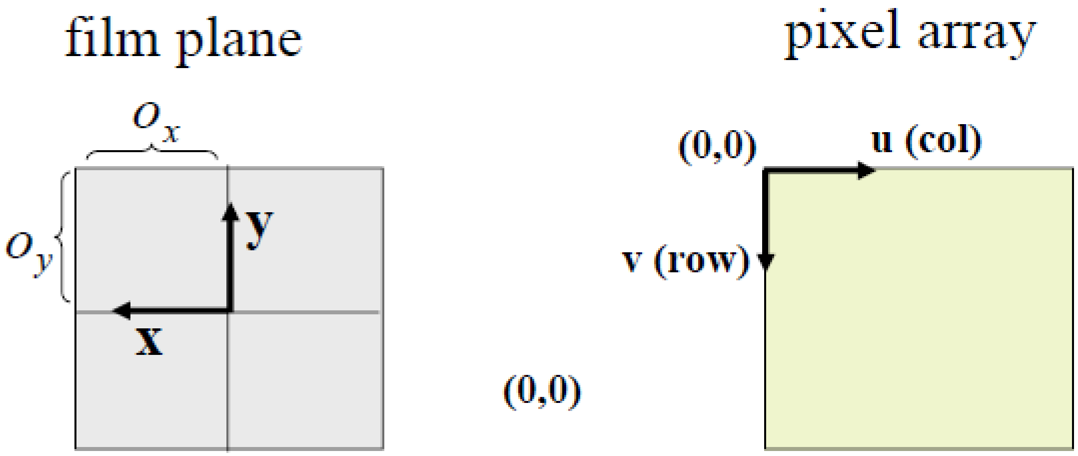


$$u = f\frac{X}{Z} + o_x \qquad v = f\frac{Y}{Z} + o_y$$

o_x and o_y called image center or principle point

Slide: R. Collins

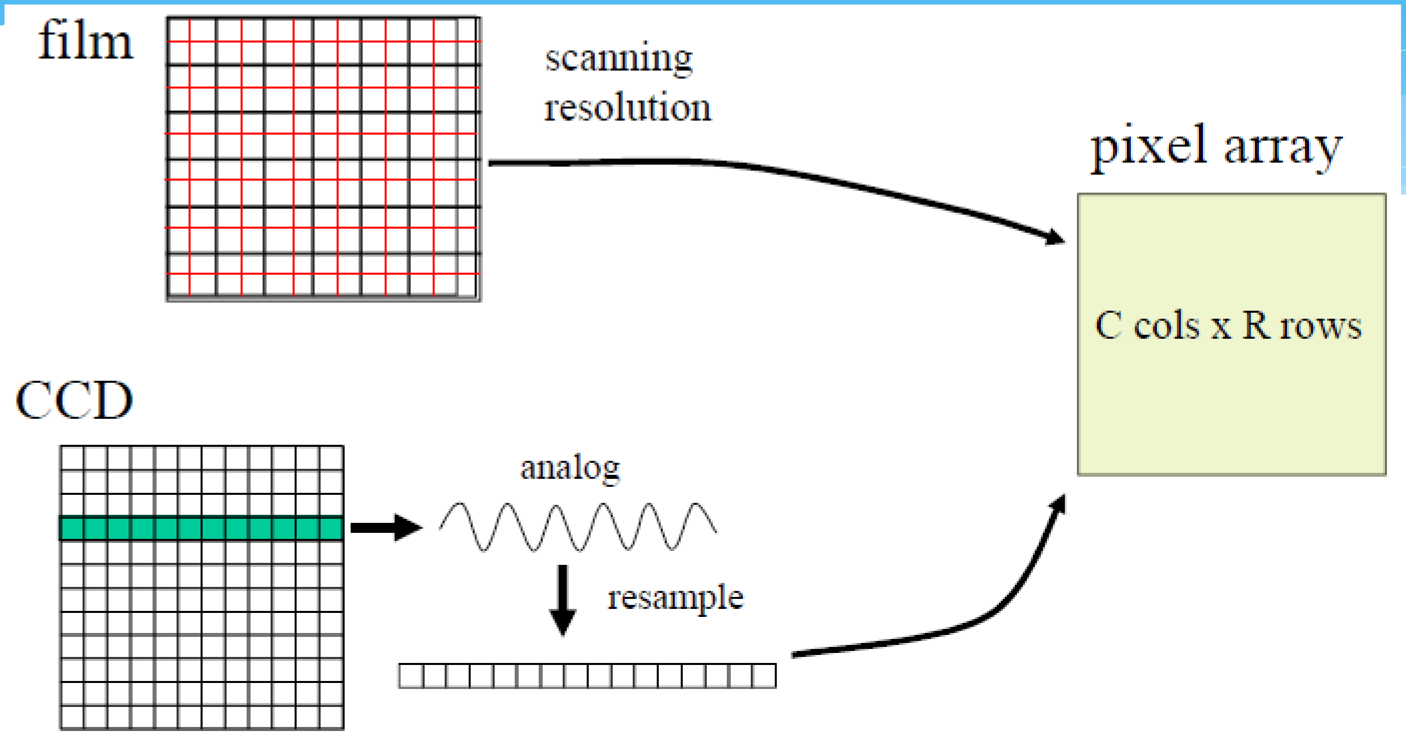
Intrinsic Camera Parameters



$$u = -f\frac{X}{Z} + o_x \qquad v = -f\frac{Y}{Z} + o_y$$

Slide: R. Collins

Intrinsic Camera Parameters



Slide: R. Collins

$$u = \frac{1}{s_x} f \frac{X}{Z} + o_x \qquad v = \frac{1}{s_y} f \frac{Y}{Z} + o_y$$

Adding the intrinsic parameters into the perspective projection matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f/s_x & 0 & o_x & 0 \\ 0 & f/s_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Slide: R. Collins

$$
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim
\underbrace{
\underbrace{
\underbrace{\overset{\text{Film plane to pixels}}{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}}}_{\mathbf{M}_{\mathbf{aff}}}
\underbrace{\overset{\text{Perspective projection}}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}}_{\mathbf{M}_{\mathbf{proj}}}
}_{\mathbf{M}_{\mathbf{int}}}
\underbrace{\overset{\text{World to camera}}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}}_{\mathbf{M}_{\mathbf{ext}}}
}_{\mathbf{M}}
\begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}
$$

Slide: R. Collins

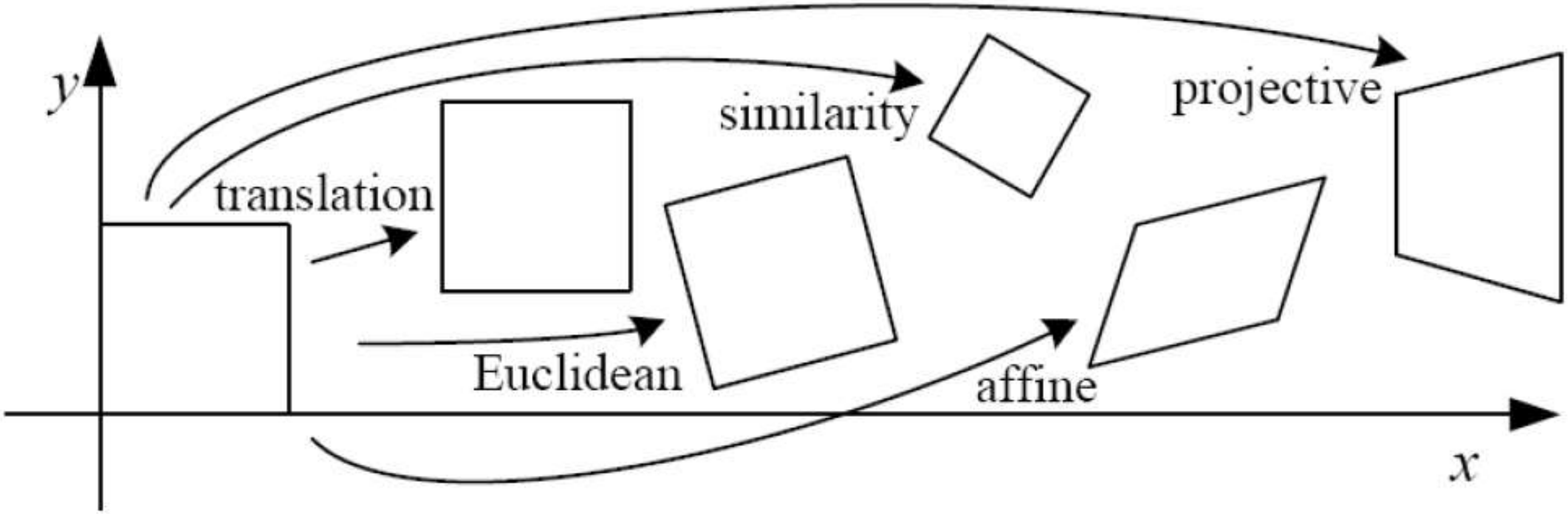
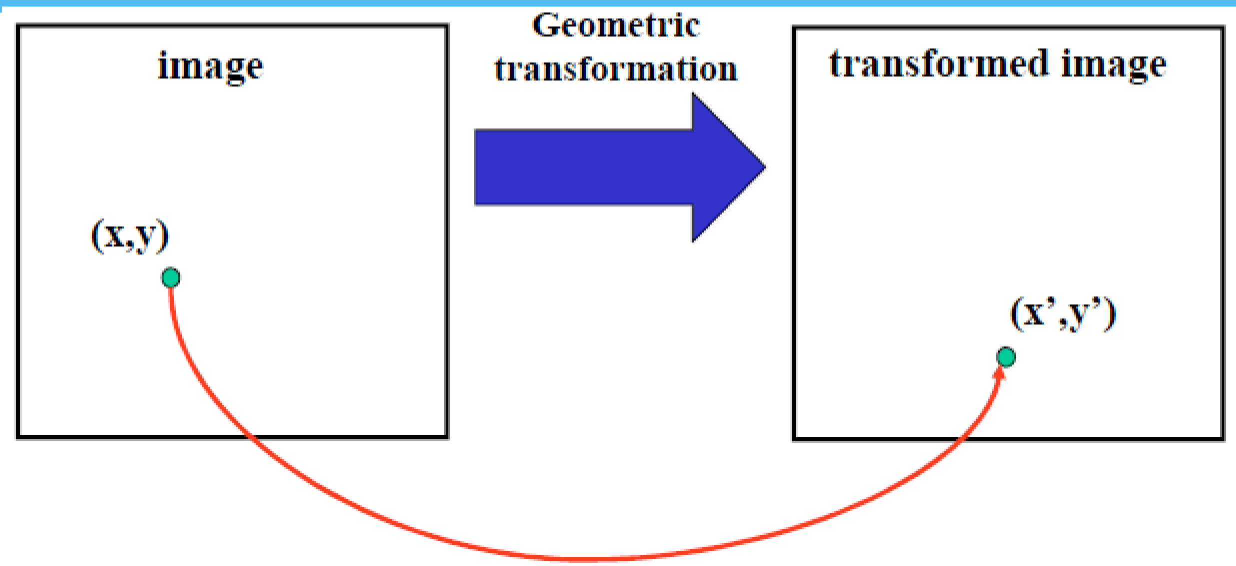


FIGURE 1. Basic set of 2D planar transformations

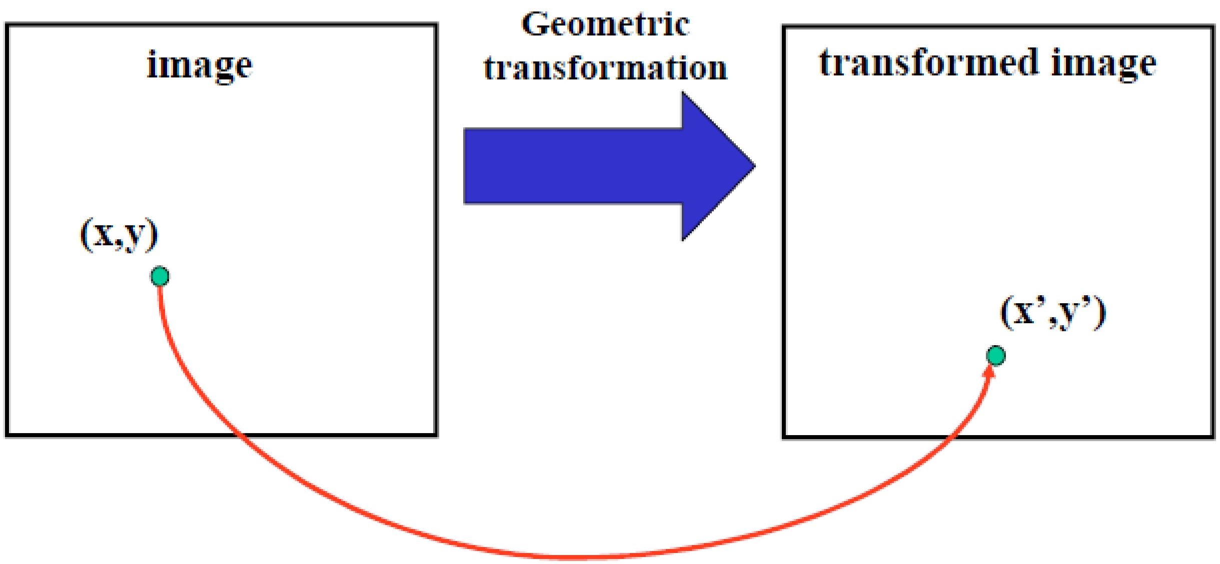
from R.Szeliski

Slide: R. Collins



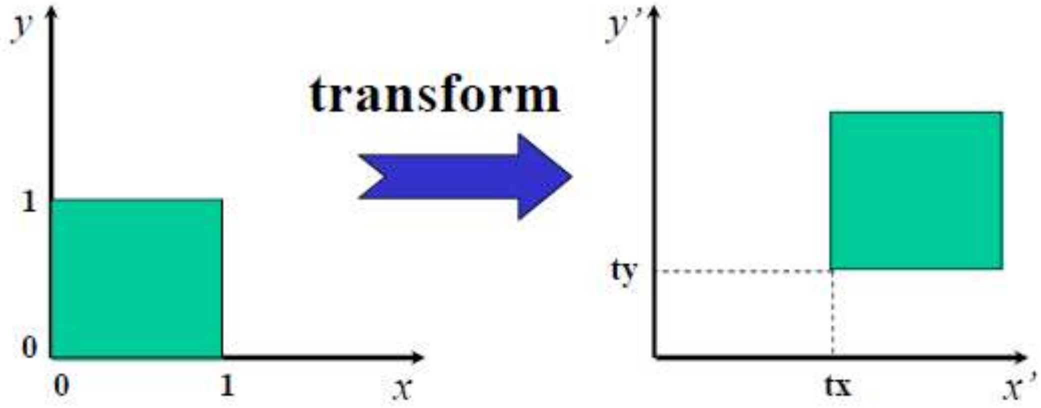
x' = f(x, y, {parameters})
y' = g(x, y, {parameters})

Slide: R. Collins



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} \\ M(\text{params}) \\ \\ \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Slide: R. Collins

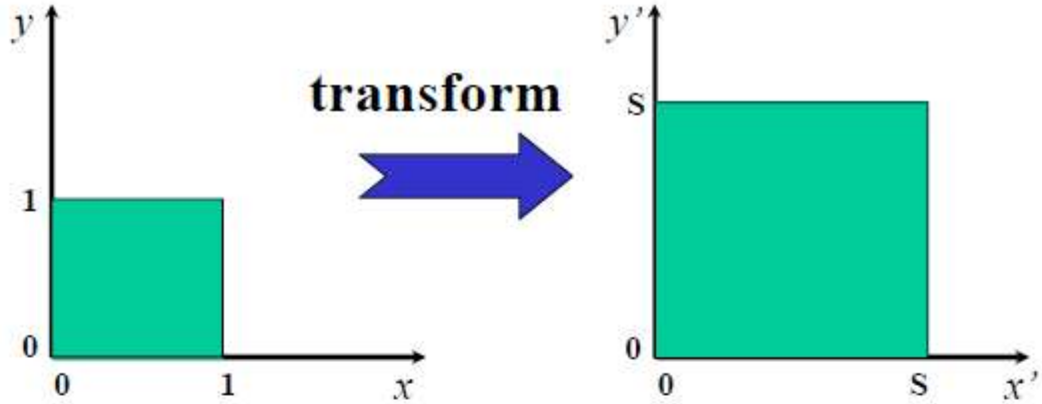


$$x' = x + t_x$$
$$y' = y + t_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

equations **matrix form**

Slide: R. Collins



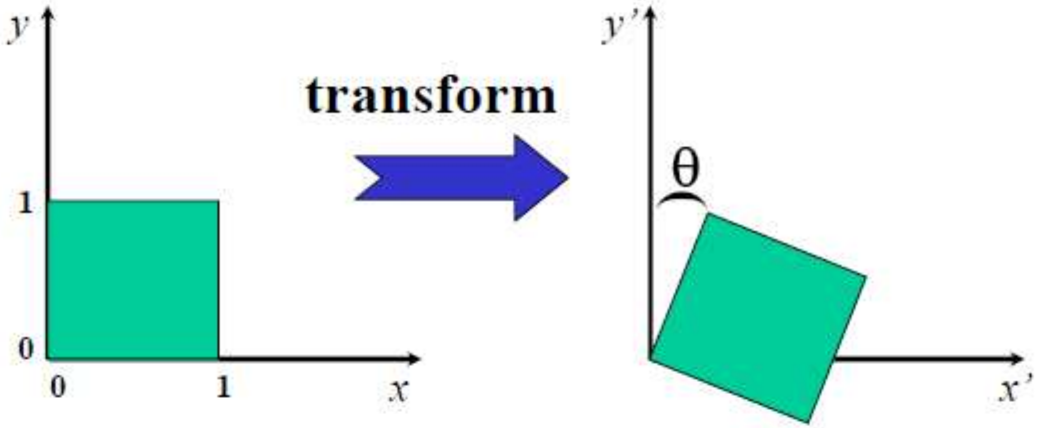
$$x' = s\,x_i$$
$$y' = s\,y_i$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Slide: R. Collins



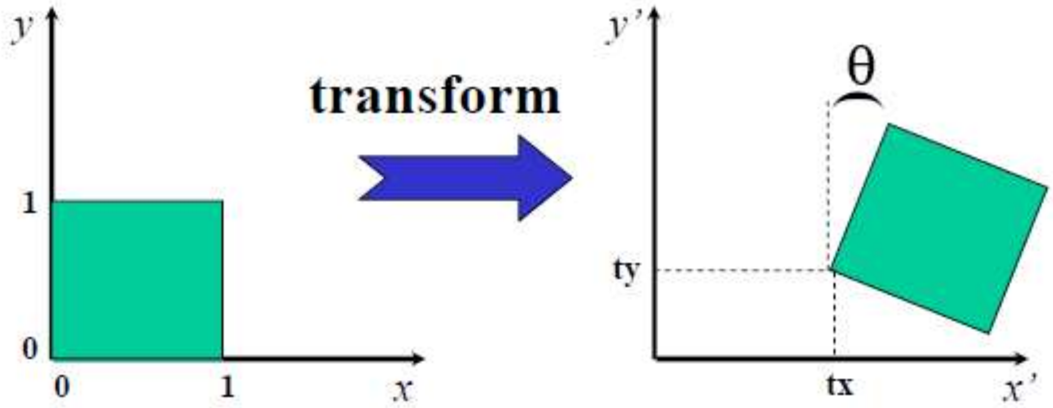
$$x' = x_i \cos\theta - y_i \sin\theta$$
$$y' = x_i \sin\theta + y_i \cos\theta$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Slide: R. Collins



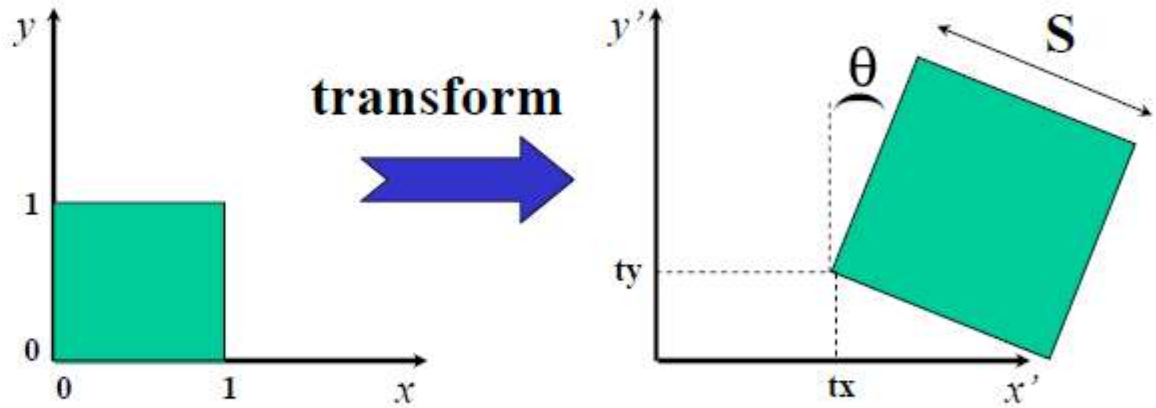
$$x' = x_i \cos\theta - y_i \sin\theta + t_x$$
$$y' = x_i \sin\theta + y_i \cos\theta + t_y$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Slide: R. Collins



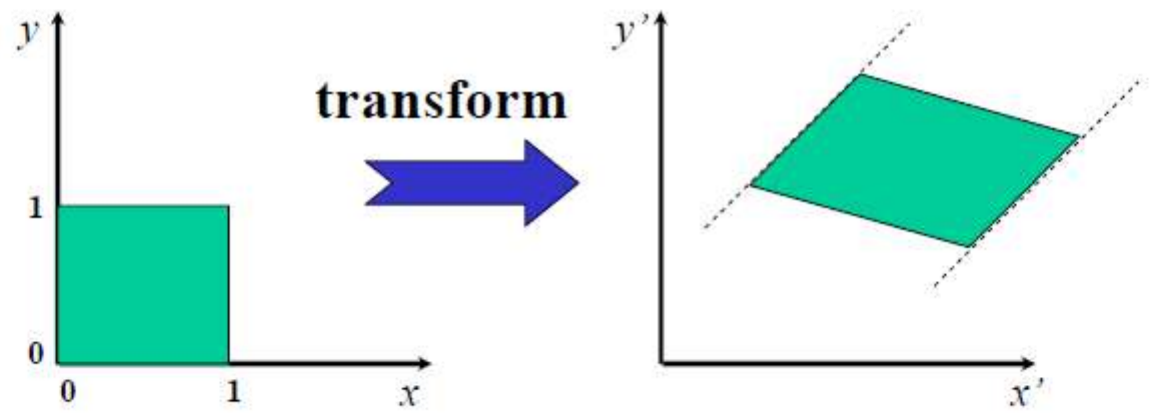
$$p' = sRp + t$$

equations

$$\begin{bmatrix} p' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

matrix form

Slide: R. Collins



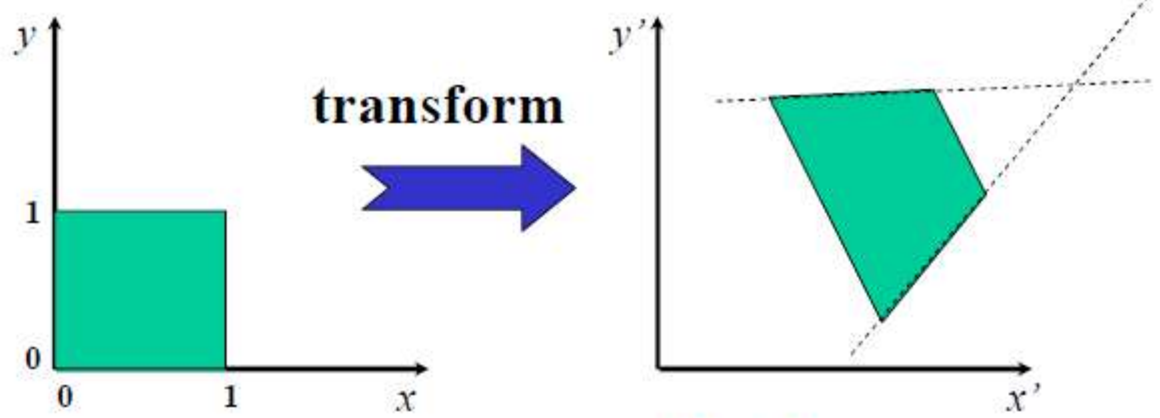
$$p' = Ap + b$$

equations

$$\begin{bmatrix} p' \\ 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

matrix form

Slide: R. Collins



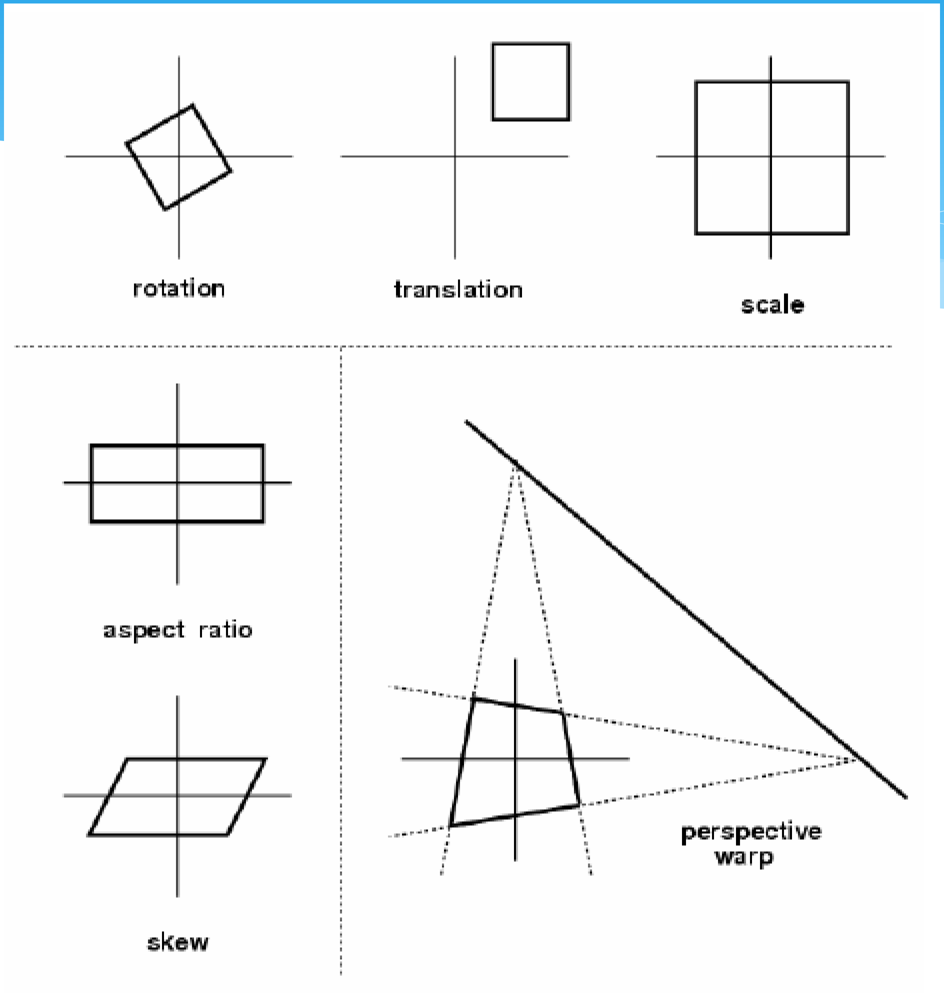
Note!

$$p' = \frac{Ap + b}{c^T p + 1}$$

$$\begin{bmatrix} p' \\ 1 \end{bmatrix} \sim \begin{bmatrix} A & b \\ c^T & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

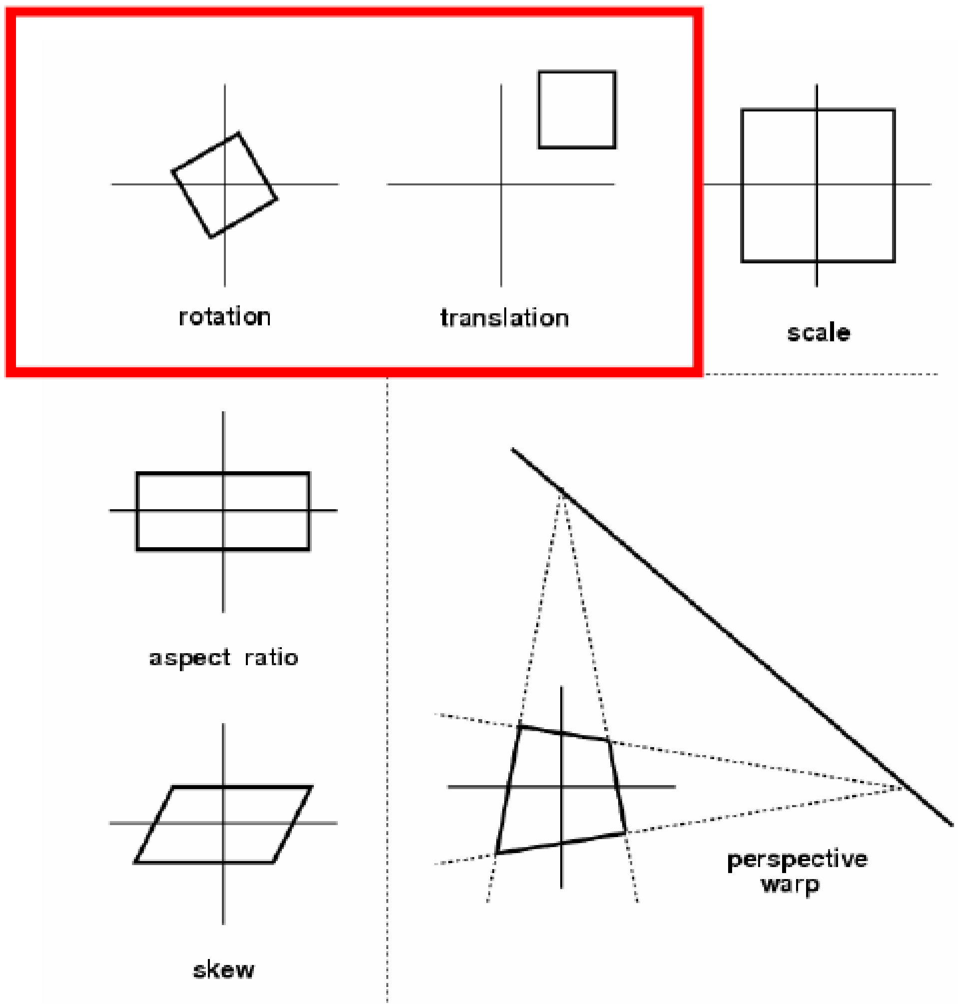
equations **matrix form**

Slide: R. Collins



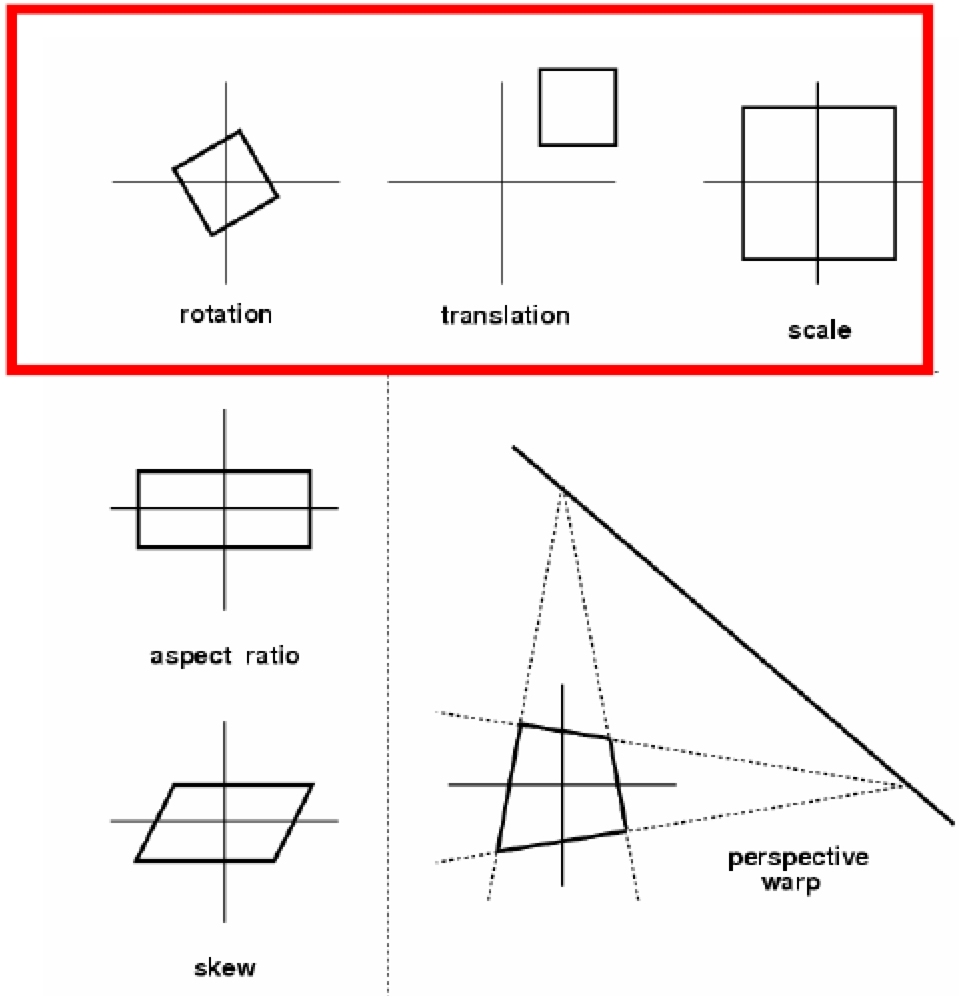
Slide: R. Collins

Euclidean



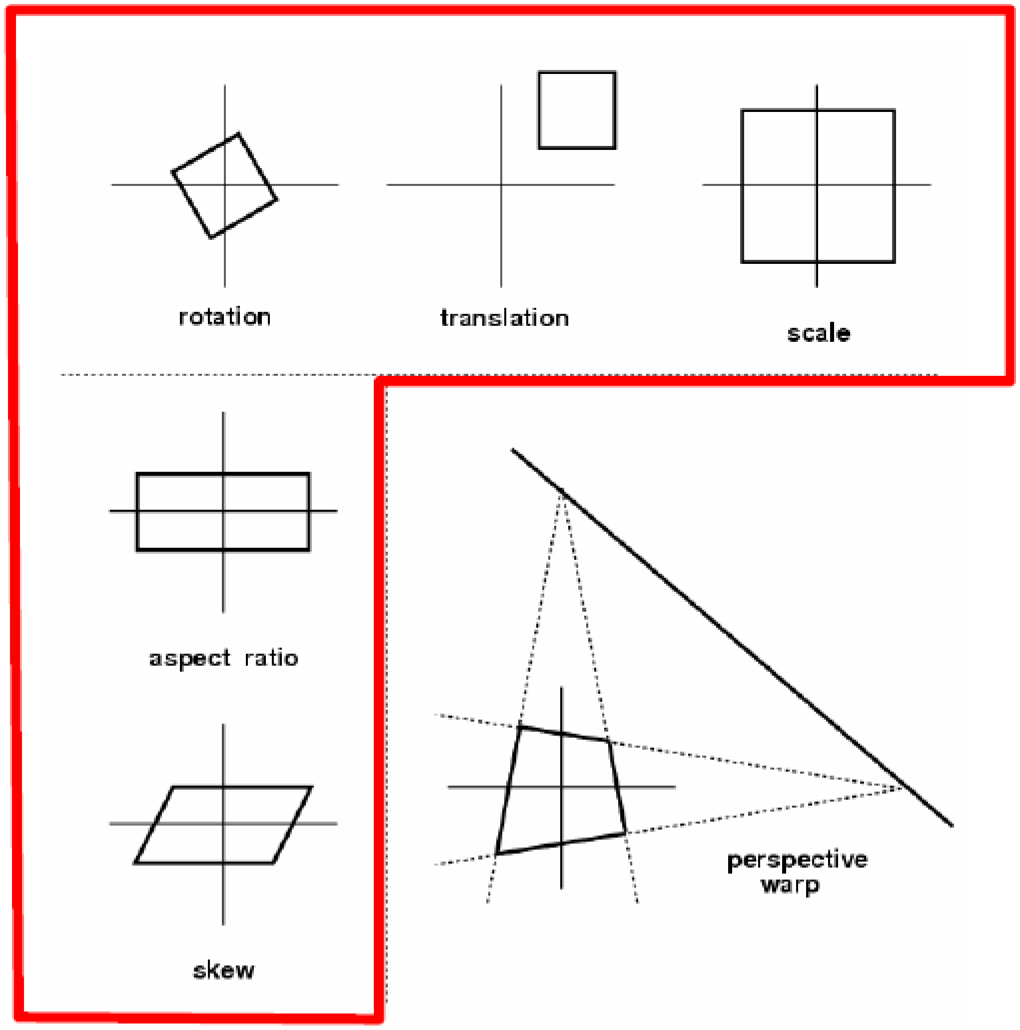
Slide: R. Collins

Similarity



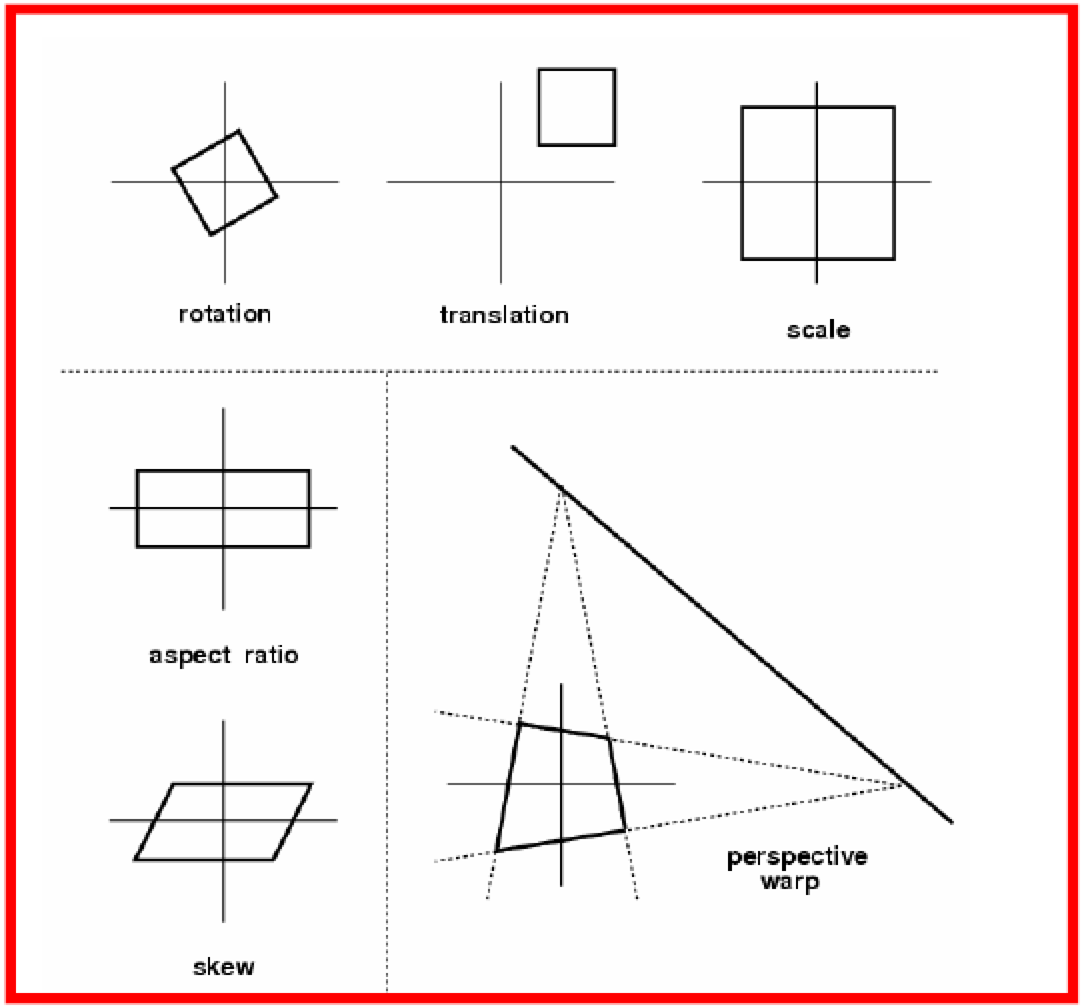
Slide: R. Collins

Affine



Slide: R. Collins

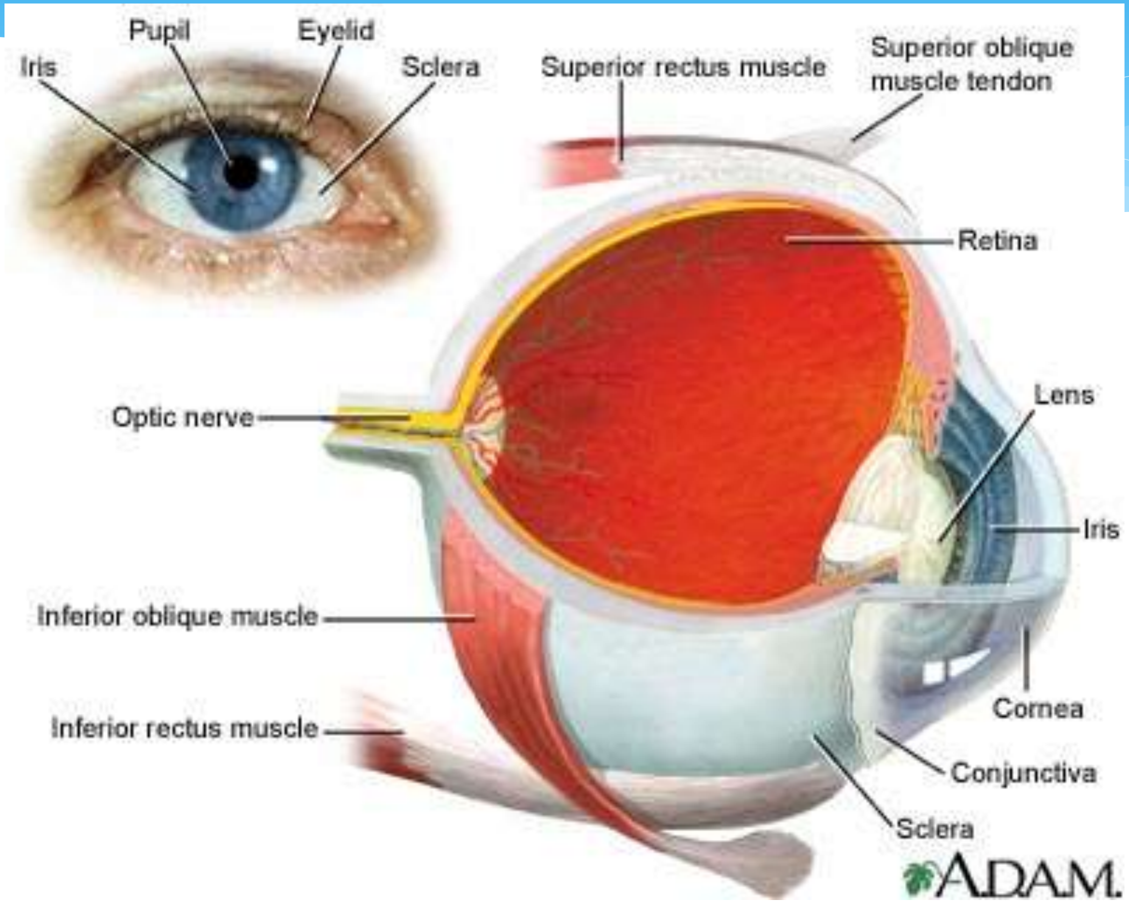
Projective



Slide: R. Collins

Euclidean Geometry vs Projective Geometry

* Euclidean geometry keeps objects/shapes as they are
* Projective geometry describes things as they appear



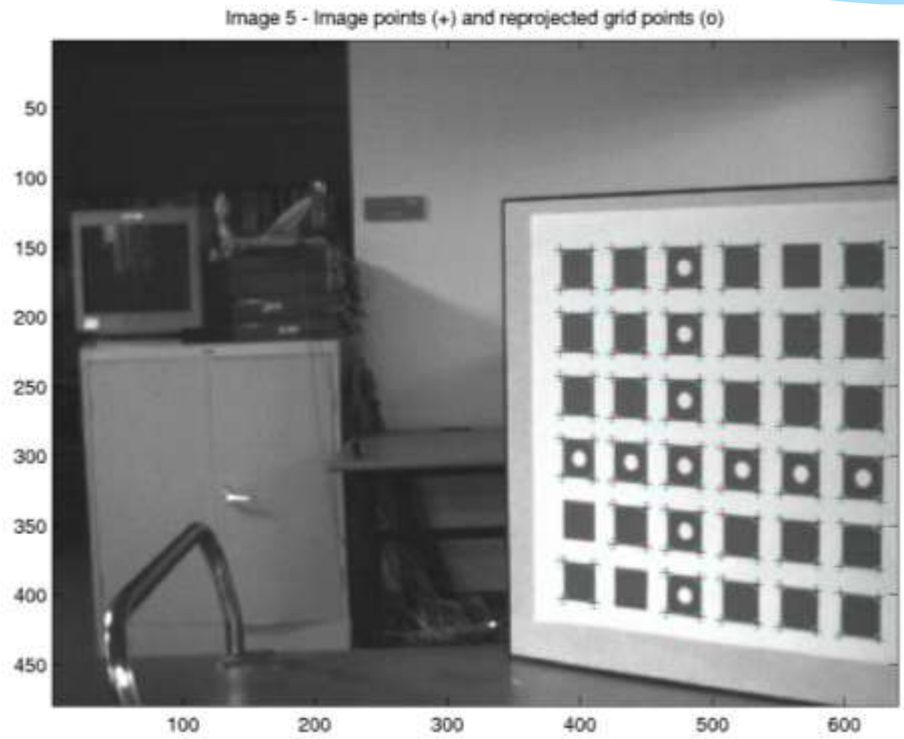
Picture: http://www.nlm.nih.gov/medlineplus/ency/imagepages/1094.htm

Camera Calibration

Camera Calibration

- Determine:
  - Focal length
  - Position of the image center
  - Scaling for the row and the column pixels
  - Skew
  - Lens Distortion
- Why important?
  - For 3D reconstruction.
  - Hand-eye coordination → robotic manipulation.

Camera Calibration



Camera calibration

From before, we had these equations relating image positions, u,v, to points at 3-D positions P (in homogeneous coordinates):

$$u = \frac{m_1 \cdot \vec{P}}{m_3 \cdot \vec{P}}$$

$$v = \frac{m_2 \cdot \vec{P}}{m_3 \cdot \vec{P}}$$

So for each feature point, i, we have:

$$(m_1 - u_i m_3) \cdot \vec{P}_i = 0$$

$$(m_2 - v_i m_3) \cdot \vec{P}_i = 0$$

W. Freeman

Camera calibration

Stack all these measurements of i=1… n points

$$(m_1 - u_i m_3) \cdot \vec{P}_i = 0$$
$$(m_2 - v_i m_3) \cdot \vec{P}_i = 0$$

into a big matrix:

$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

W. Freeman

Camera calibration

In vector form:

$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Showing all the elements:

$$\begin{pmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & 0 & 0 & 0 & 0 & -u_1 P_{1x} & -u_1 P_{1y} & -u_1 P_{1z} & -u_1 \\ 0 & 0 & 0 & 0 & P_{1x} & P_{1y} & P_{1z} & 1 & -v_1 P_{1x} & -v_1 P_{1y} & -v_1 P_{1z} & -v_1 \\ & & & & & & \cdots & \cdots & \cdots & & & \\ P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_n P_{nx} & -u_n P_{ny} & -u_n P_{nz} & -u_n \\ 0 & 0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_n P_{nx} & -v_n P_{ny} & -v_n P_{nz} & -v_n \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

W. Freeman

Once you have the M matrix, can recover the intrinsic and extrinsic parameters as in Forsyth&Ponce, sect. 3.2.2.

$$\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}$$

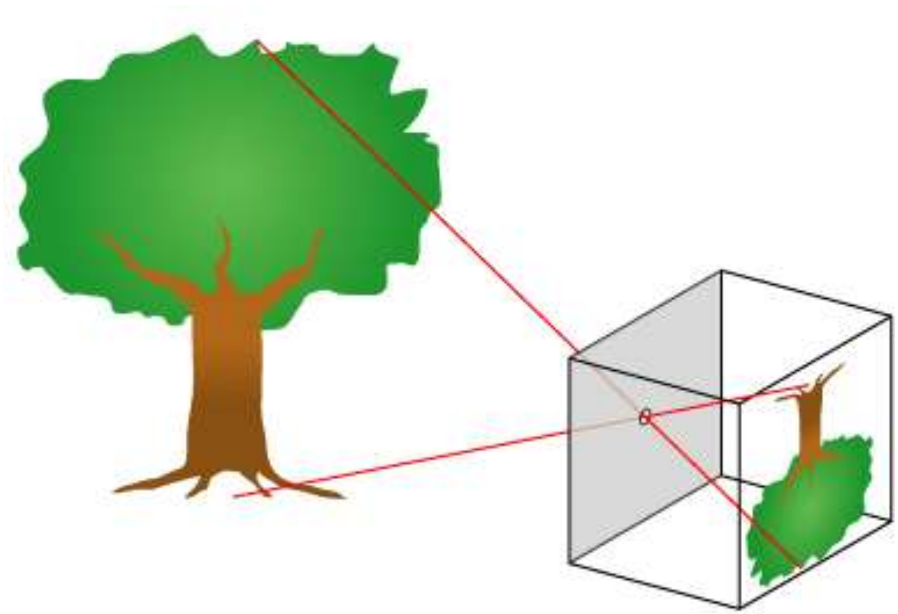
W. Freeman

End of Cameras & Projective Geometry & Calibration

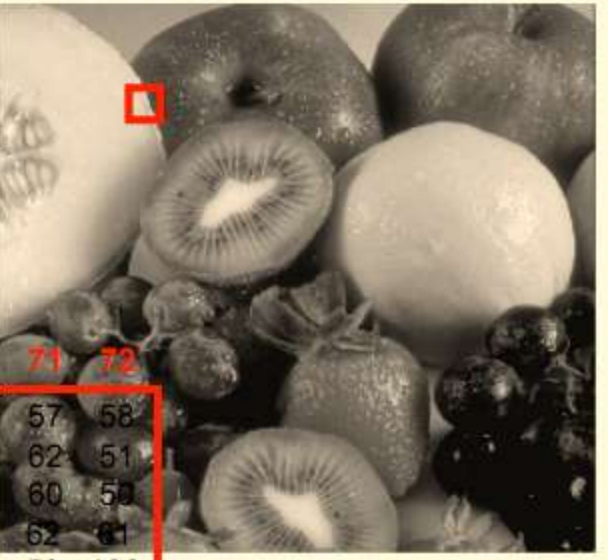
* What did I skip?
  * Image formation:
    * Light, Energy, Color

We have the image… so, what?

* That is just the beginning…



Grayscale Image



y \ x	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

Slide: M. J. Black

A Crash Tutorial on
Filtering, Convolution, The Universe and everything

Filtering

Slide: A. Torralba

Linear filtering

For a linear system, each output is a linear combination of all the input values:

$$f[m,n] = \sum_{k,l} h[m,n,k,l]\, g[k,l]$$

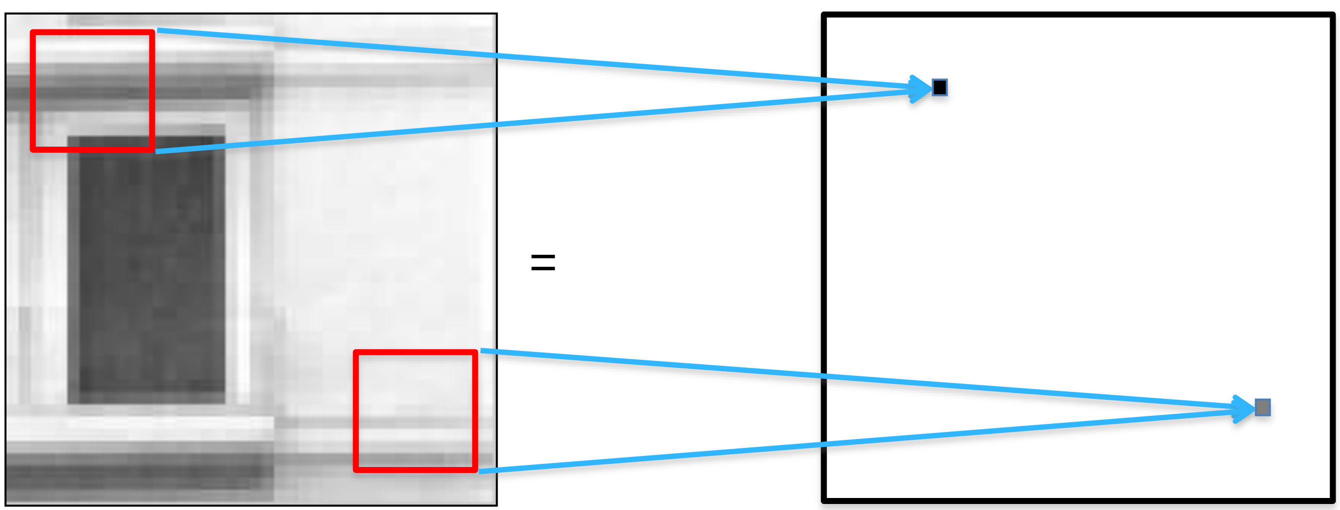
In matrix form:

F = H G

Slide: A. Torralba

Linear filtering

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l] g[k,l]$$



Slide: A. Torralba

Linear filtering

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$

m=0 1 2 …

111	115	113	111	112	111	112	111
135	138	137	139	145	146	149	147
163	168	188	196	206	202	206	207
180	184	206	219	202	200	195	193
189	193	214	216	104	79	83	77
191	201	217	220	103	59	60	68
195	205	216	222	113	68	69	83
199	203	223	228	108	68	71	77

g[m,n]

$\otimes$

-1	2	-1
-1	2	-1
-1	2	-1

h[m,n]

=

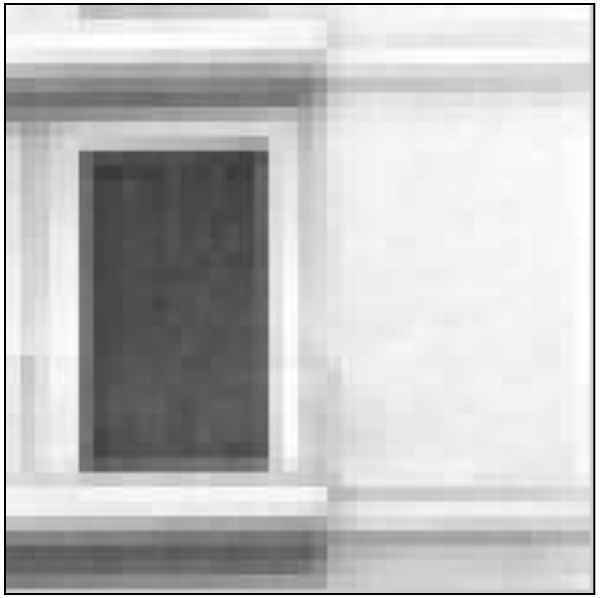
?	?	?	?	?	?	?	?
?	-5	9	-9	21	-12	10	?
?	-29	18	24	4	-7	5	?
?	-50	40	142	-88	-34	10	?
?	-41	41	264	-175	-71	0	?
?	-24	37	349	-224	-120	-10	?
?	-23	33	360	-217	-134	-23	?
?	?	?	?	?	?	?	?

f[m,n]

Slide: A. Torralba

Impulse

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$



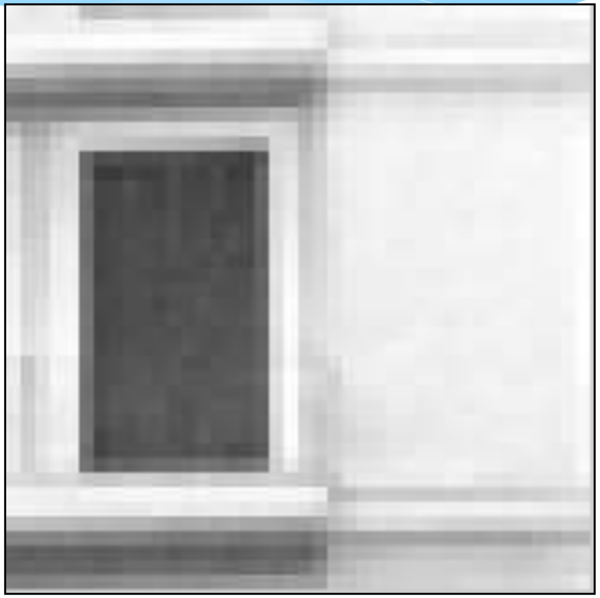
g[m,n]

$\otimes$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

h[m,n]

=

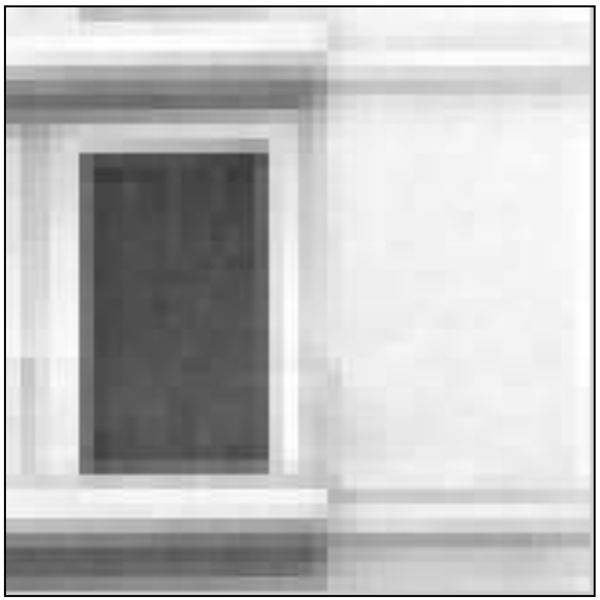


f[m,n]

Slide: A. Torralba

Shifts

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$



g[m,n]

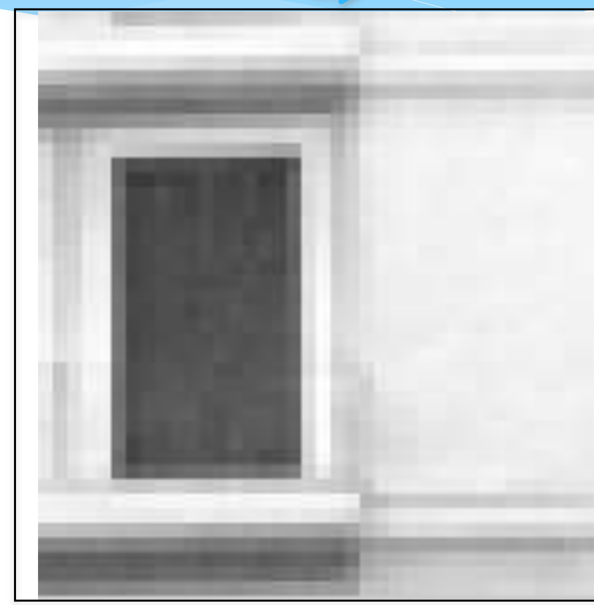
$\otimes$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0

h[m,n]

=

2pixels



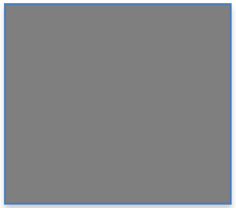
f[m,n]

Slide: A. Torralba

Rectangular filter

g[m,n]

⊗



h[m,n]

=

f[m,n]

Slide: A. Torralba

Rectangular filter

⊗ h[m,n] =

g[m,n]

f[m,n]

Slide: A. Torralba

Rectangular filter

⊗

h[m,n]

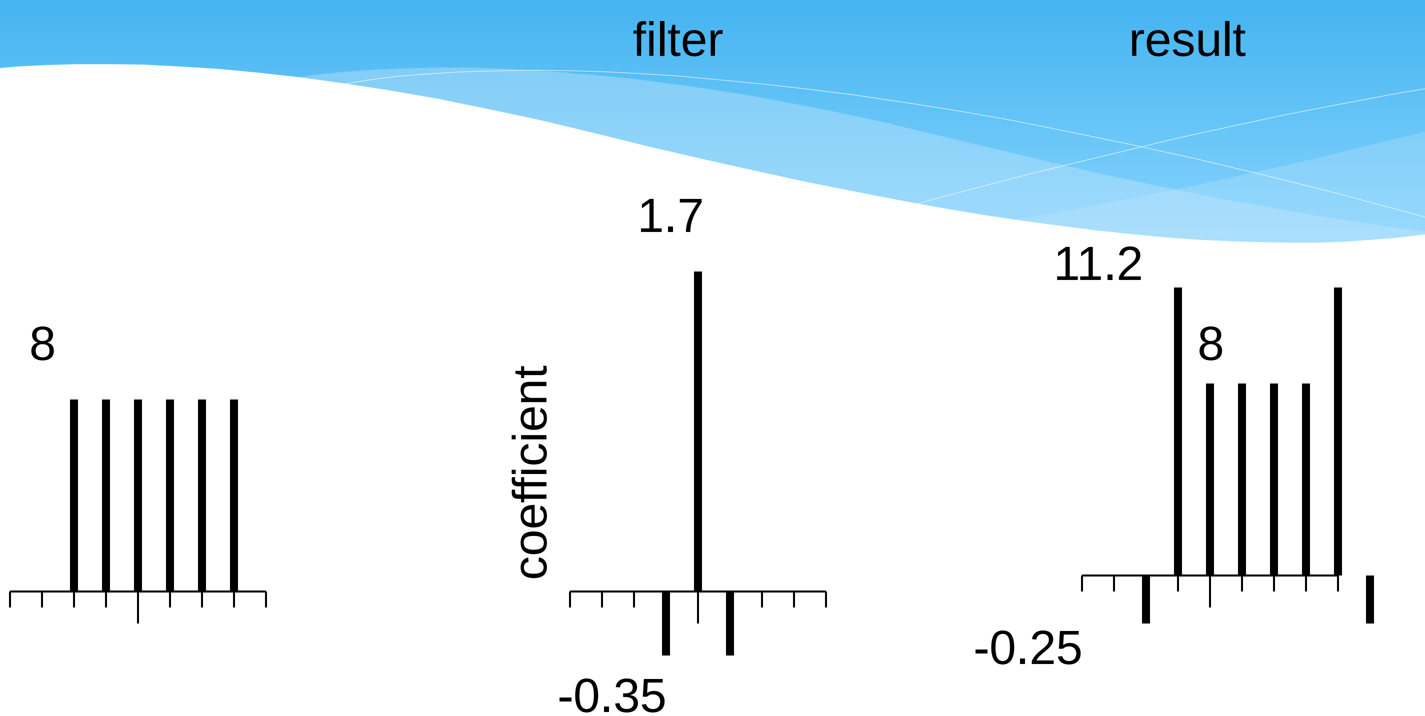
=

g[m,n]

f[m,n]

Slide: A. Torralba

Sharpening example

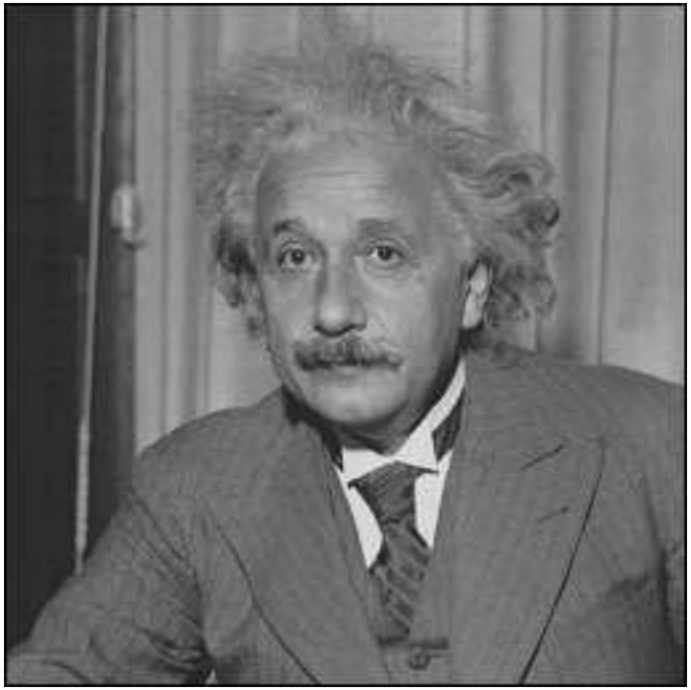


original

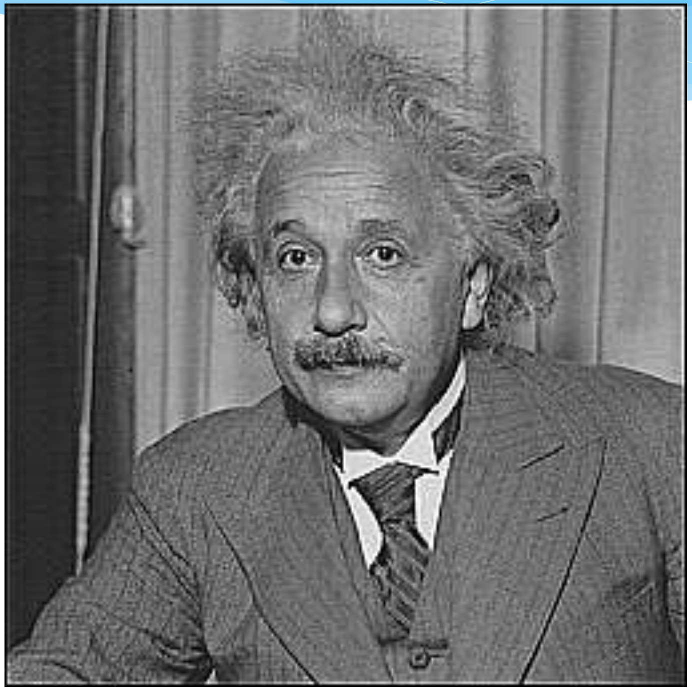
Sharpened (differences are accentuated; constant areas are left untouched).

Slide: A. Torralba

Sharpening



before

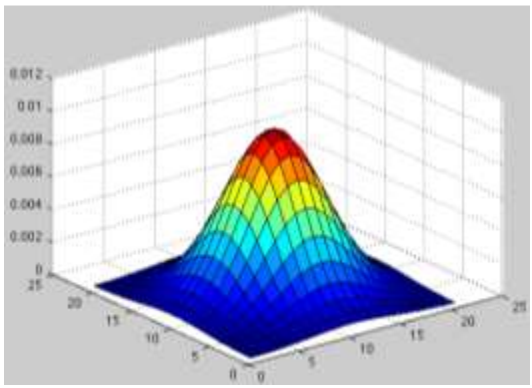


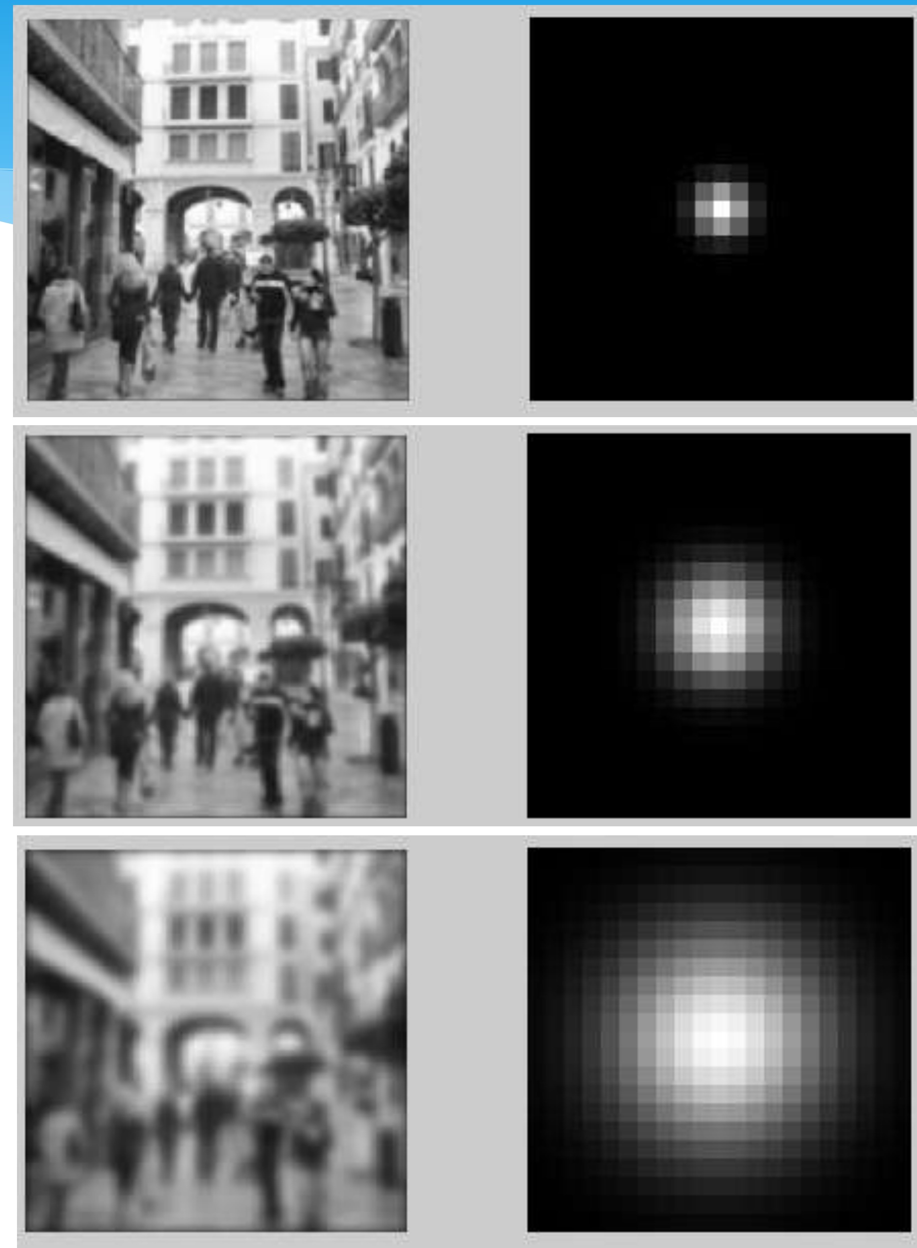
after

Slide: A. Torralba

Gaussian filter

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$





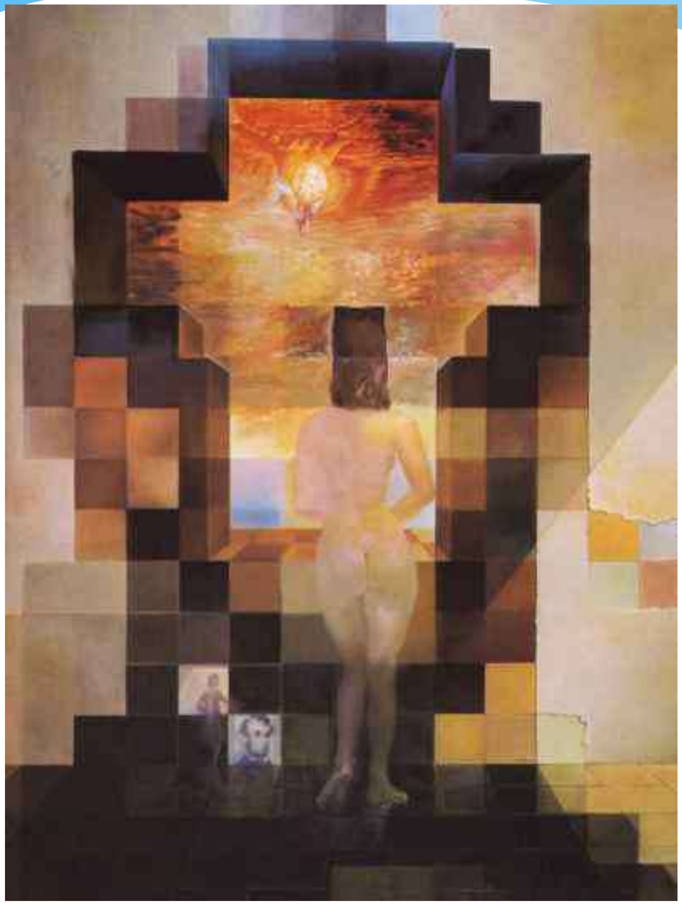
σ=1

σ=2

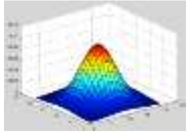
σ=4

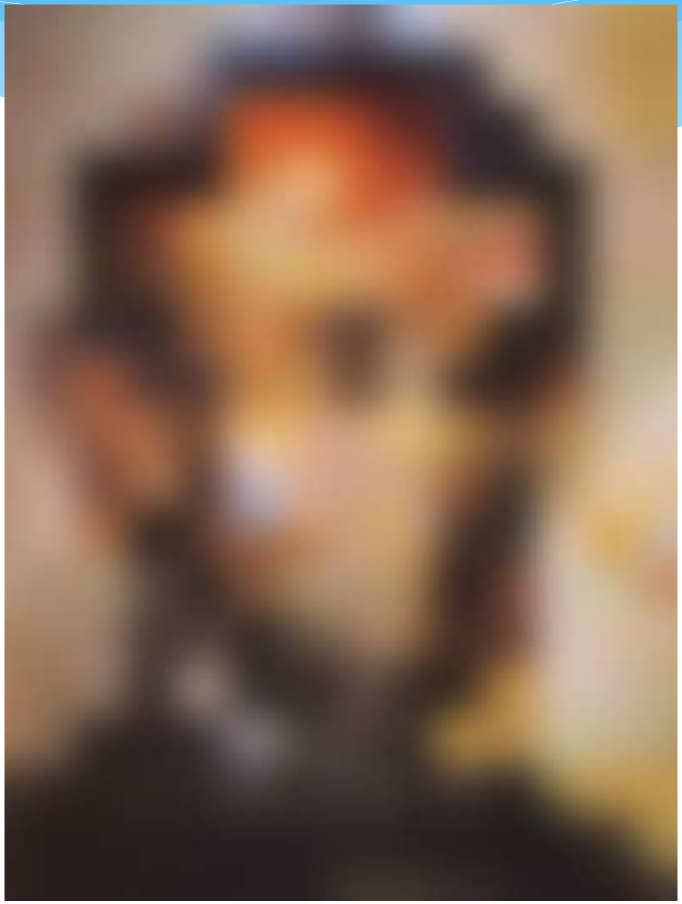
Slide: A. Torralba

Global to Local Analysis



Dali





Slide: A. Torralba

[-1 1]

$\otimes$

[-1, 1]

h[m,n]

=

g[m,n]

f[m,n]

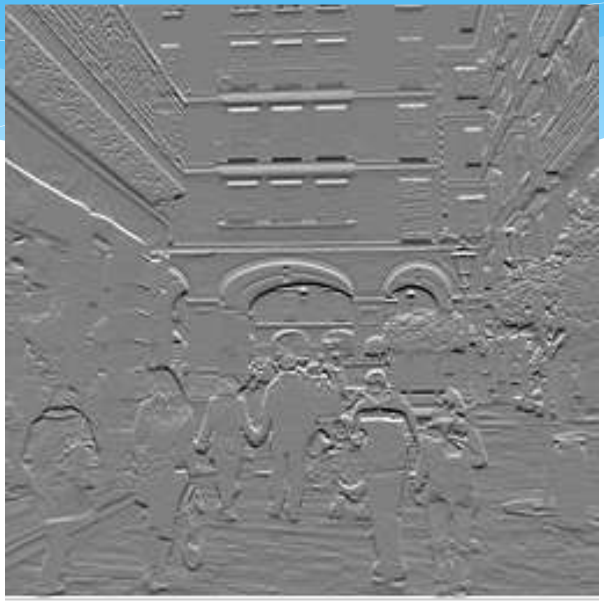
Slide: A. Torralba

$[-1\ 1]^T$

$\otimes$

$[-1, 1]^T$ =

h[m,n]



g[m,n]

f[m,n]

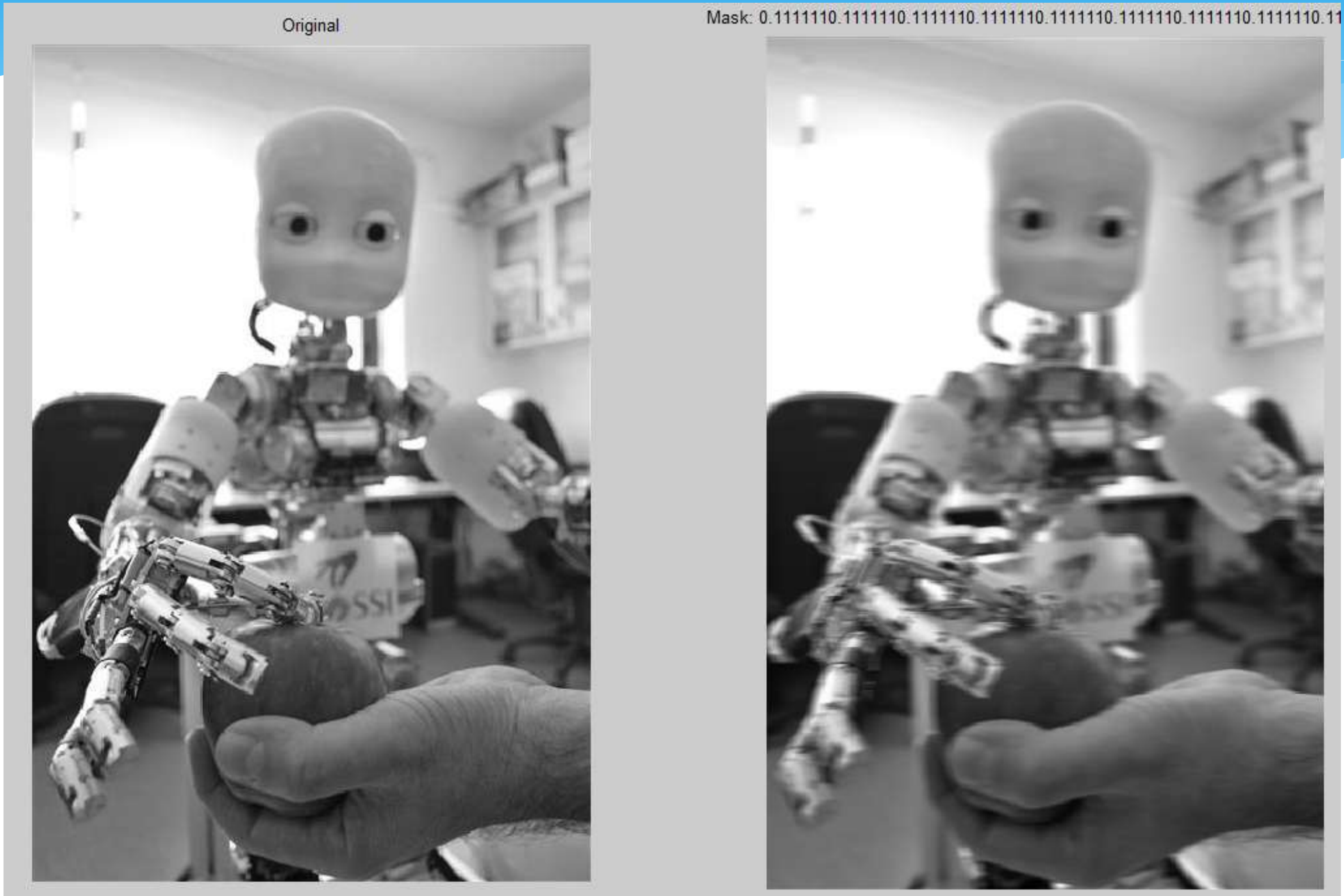
Slide: A. Torralba

Test it yourselves!!

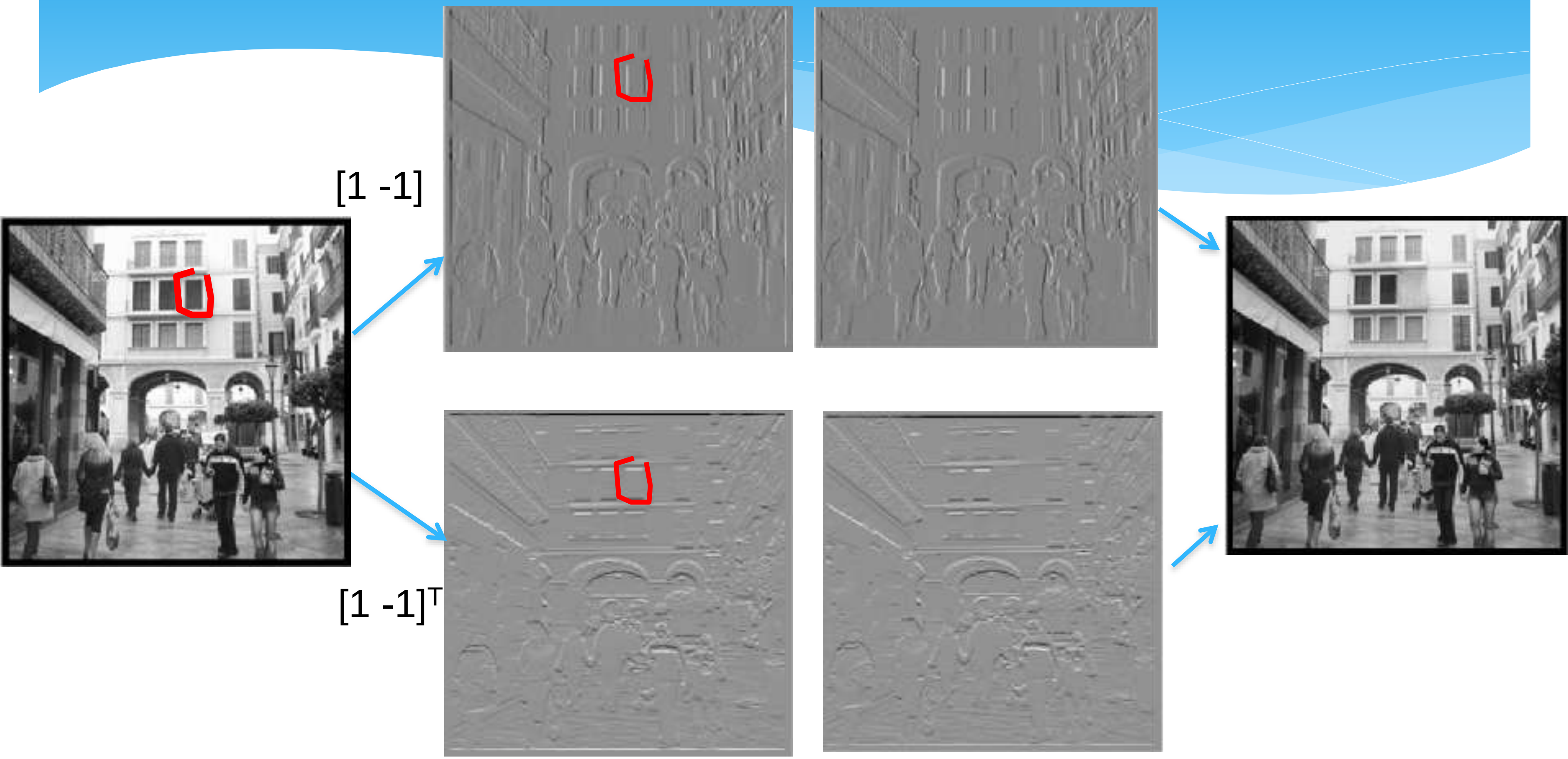
- I will upload this piece of code to the webpage.

```
Image = rgb2gray(imread('icub.jpg'));
mask = [ [1/9 1/9 1/9], [1/9 1/9 1/9], [1/9 1/9 1/9] ];
Imout = conv2(double(Image), double(mask), 'valid');
figure;
    subplot(1,2,1);
            imshow(Image, []);
            title('Original');
    subplot(1,2,2);
            imshow(Imout, []);
            title(sprintf('Mask: %s', sprintf('%f', mask)));
```

Test it yourselves!!



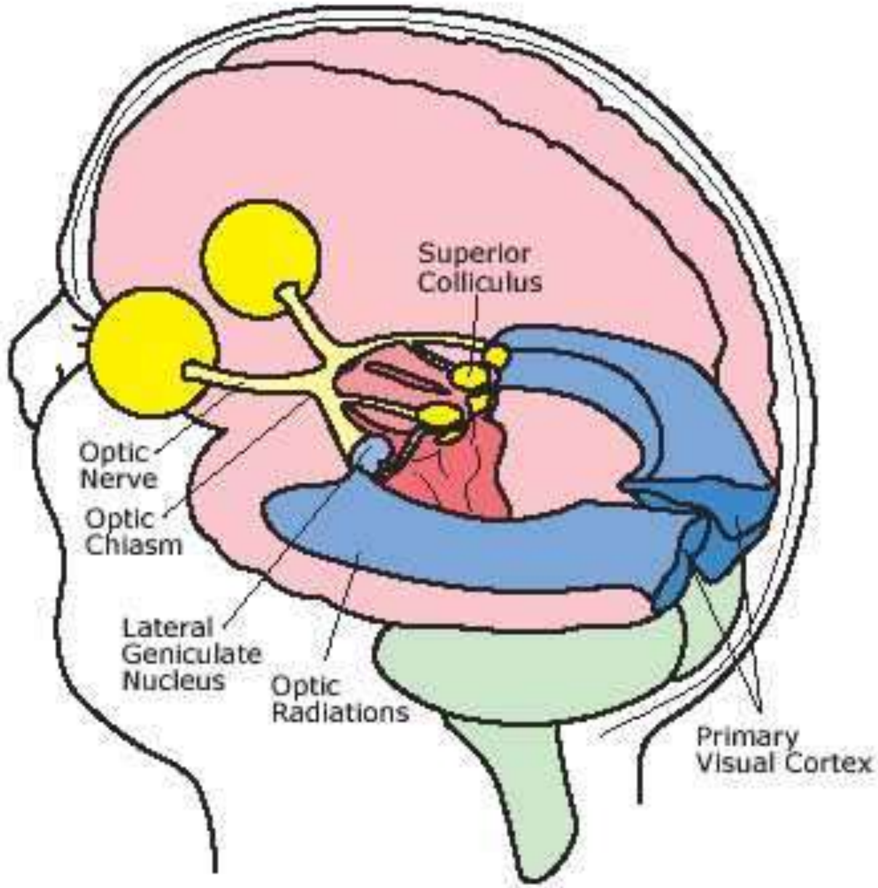
Editing the edge image



Slide: A. Torralba

Early Vision

* Unfortunately, we don’t have a concrete definition.
* Here is a try:
  * The first representation/impression extracted from ‘images’.
  * Involves edges, corners, textures, optic flow, disparity.
  * Local processing!!
  * Incomplete, ambiguous information.

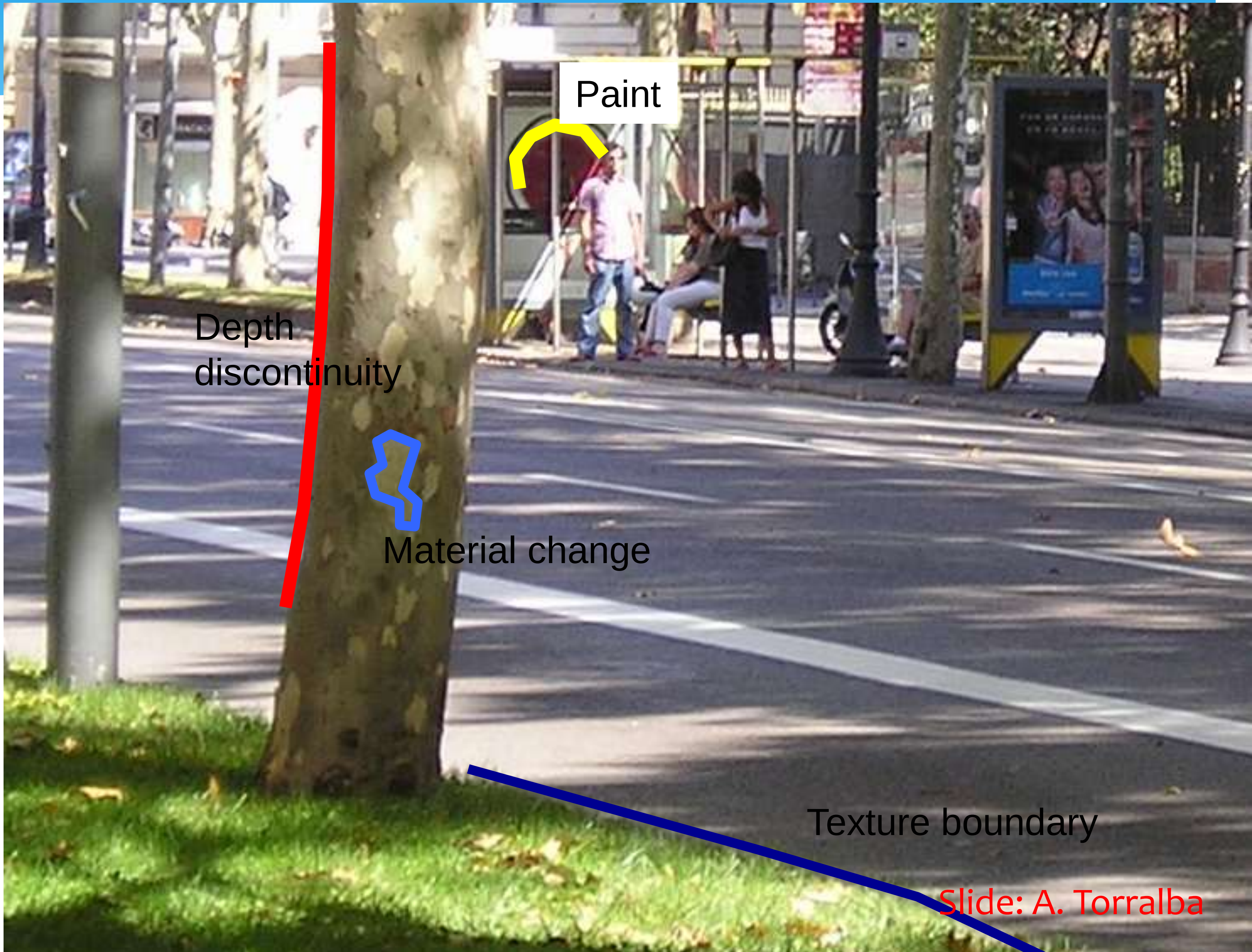


Picture: http://articles-and-essays.blogspot.com/2010/09/new-finding-on-blindsight.html

What is an edge?

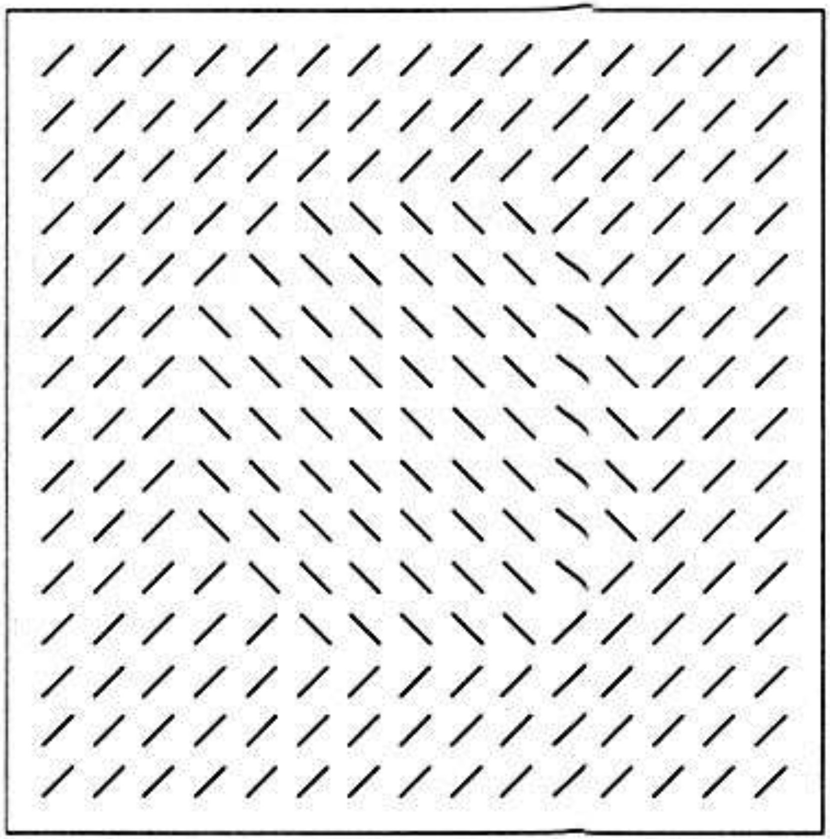
Slide: A. Torralba

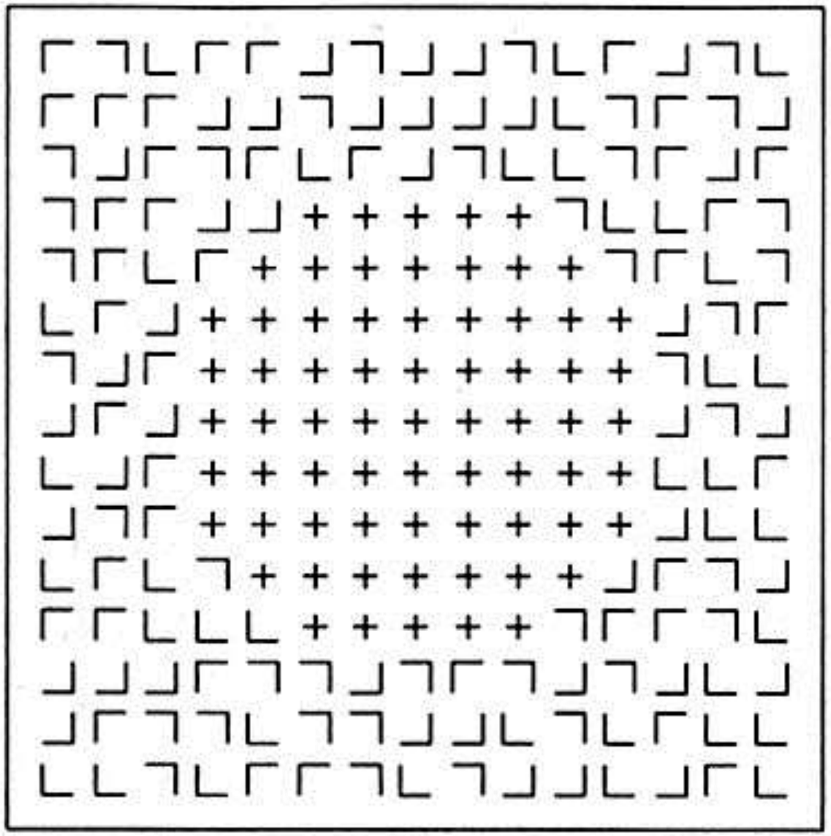
What is an edge?



Slide: A. Torralba

Edges





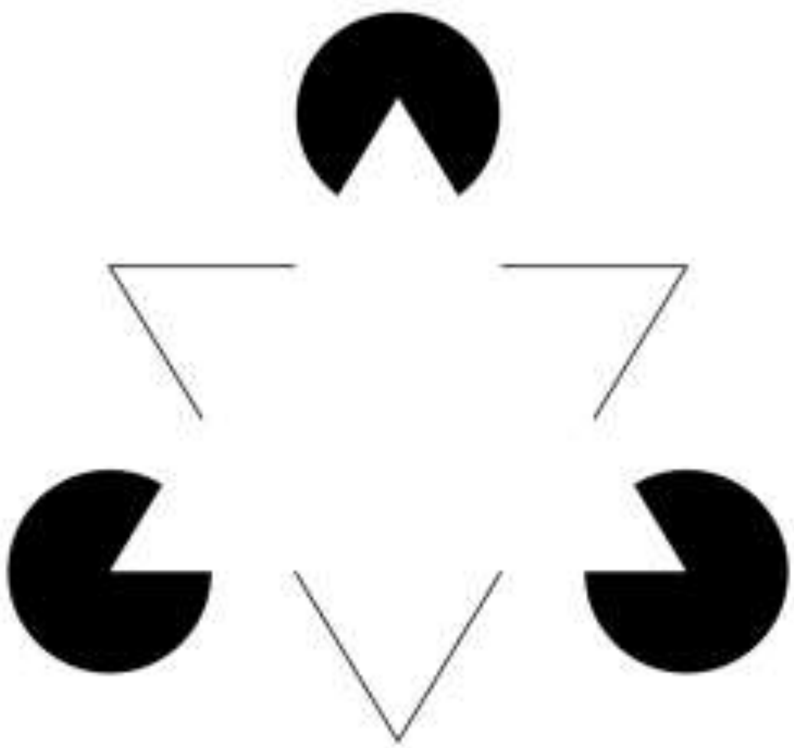
Slide: A. Torralba

Gaussian gradient boundaries

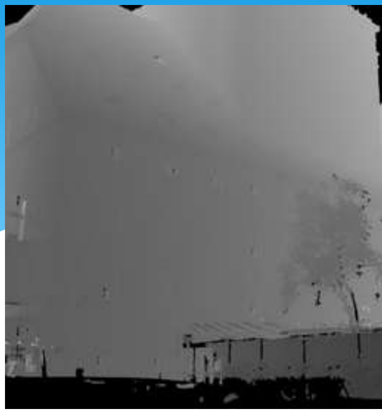
Human boundaries

Slide: A. Torralba

What is an edge?

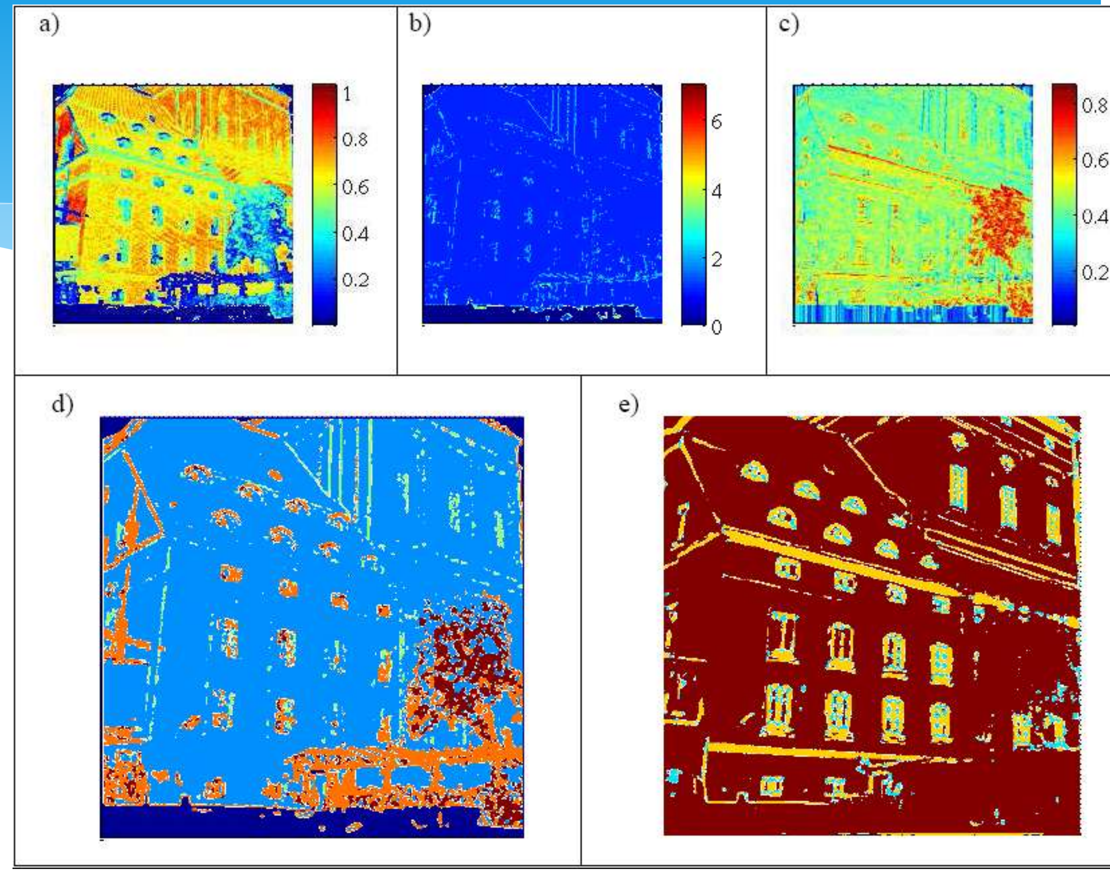


My previous investigations…



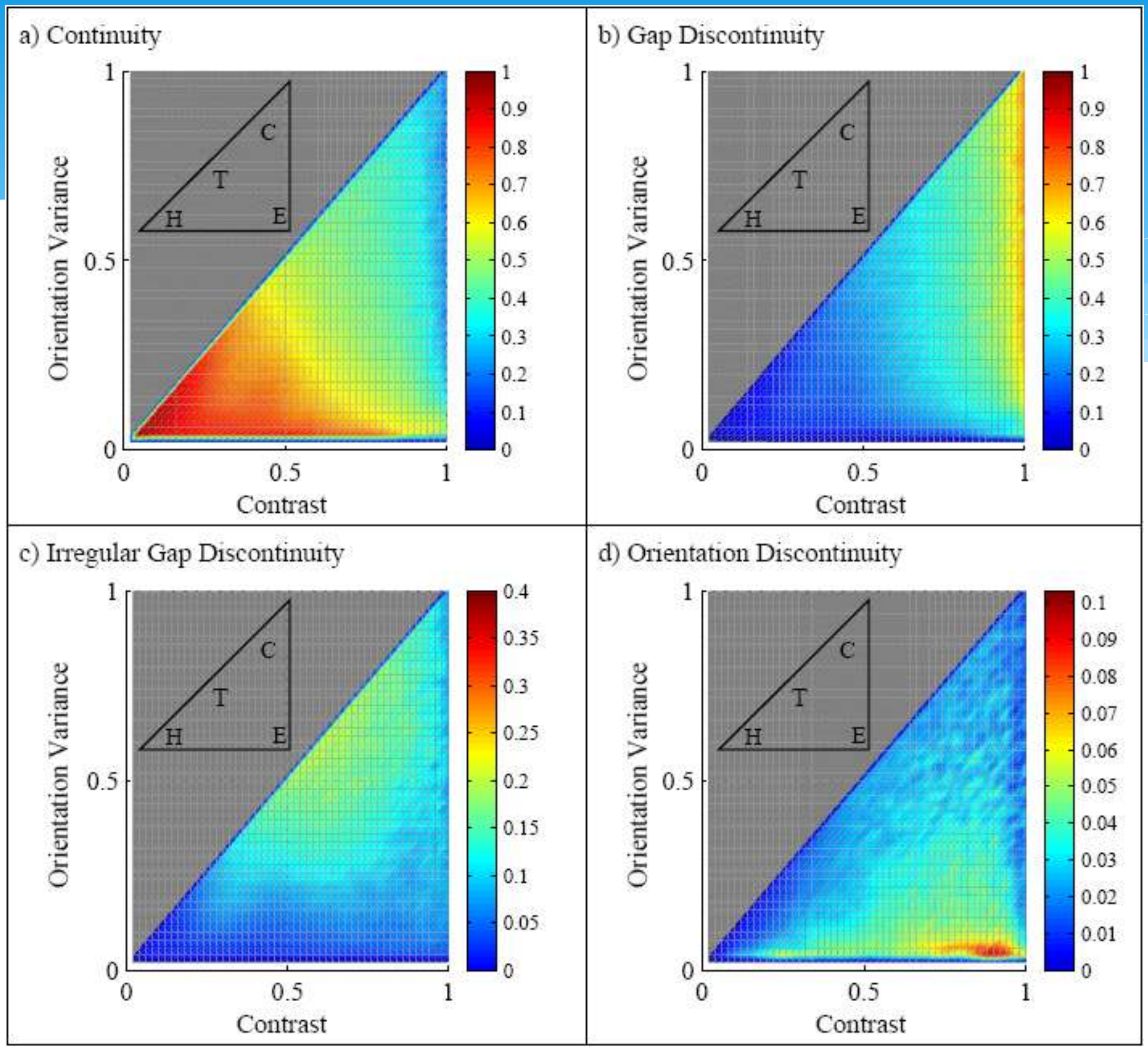
Range Image

Color Image



(a) Gap discontinuity map.
(b) Orientation discontinuity map.
(c) Irregular gap discont. map.
(d) Combination of (a), (b) and (c).
(e) Local image structures.

Kalkan et al., IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.



Kalkan et al., IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

A Computational Approach to Edge Detection

JOHN CANNY, MEMBER, IEEE

John Canny (S'81–M'82) was born in Adelaide, Australia, in 1958. He received the B.Sc. degree in computer science and the B.E. degree from Adelaide University in 1980 and 1981, respectively, and the S.M. degree from the Massachusetts Institute of Technology, Cambridge, in 1983.

He is with the Artificial Intelligence Laboratory, M.I.T. His research interests include low-level vision, model-based vision, motion planning for robots, and computer algebra.

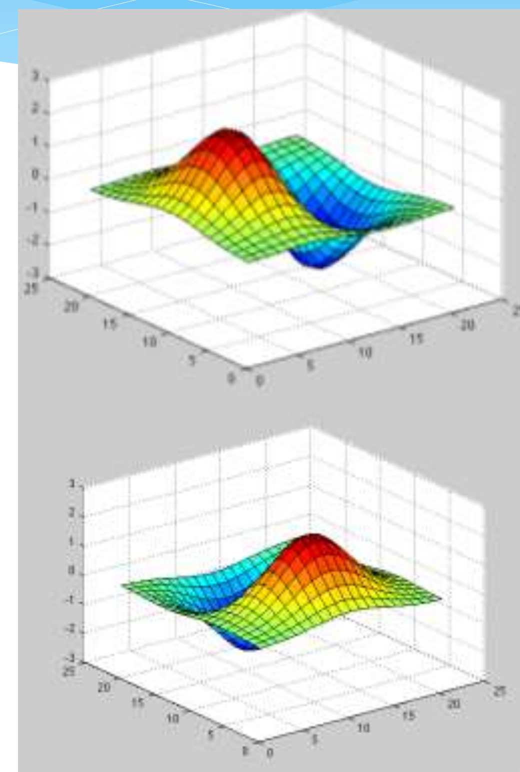
Mr. Canny is a student member of the Association for Computing Machinery.

Slide: A. Torralba

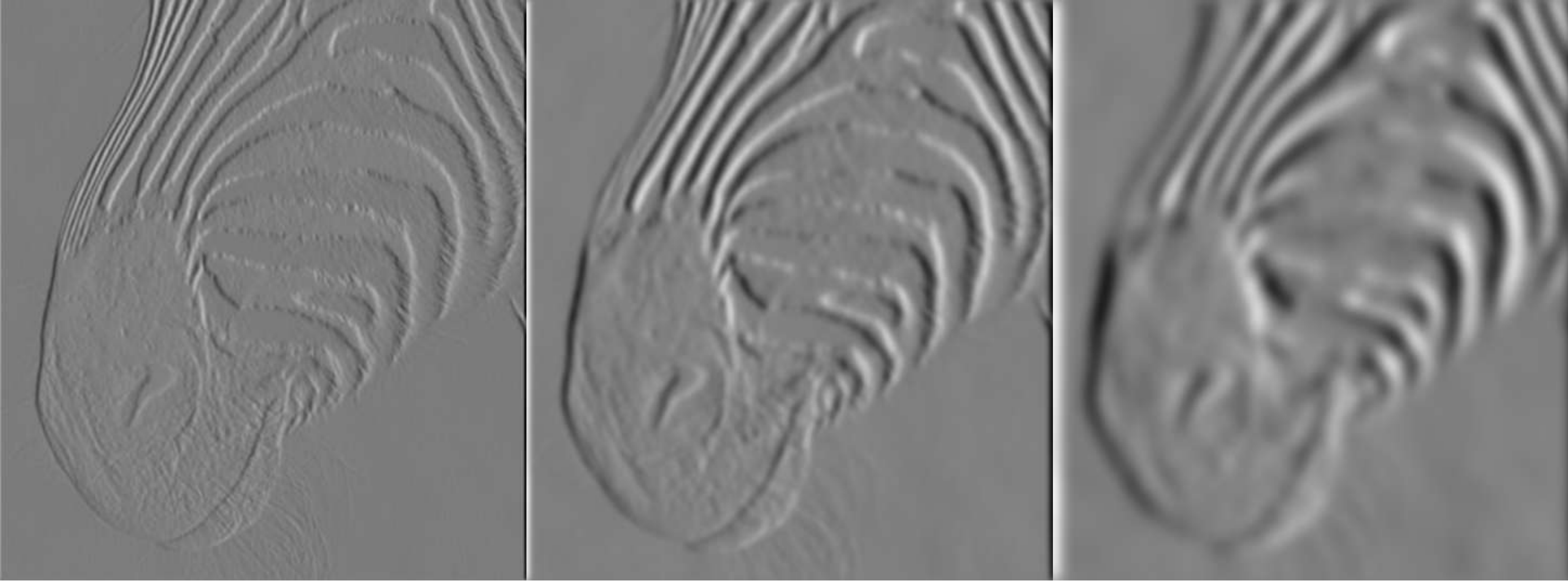
$$h_x(x,y) = \frac{\partial h(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$h_y(x,y) = \frac{\partial h(x,y)}{\partial x} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Scale



Slide: A. Torralba



1 pixel   3 pixels   7 pixels

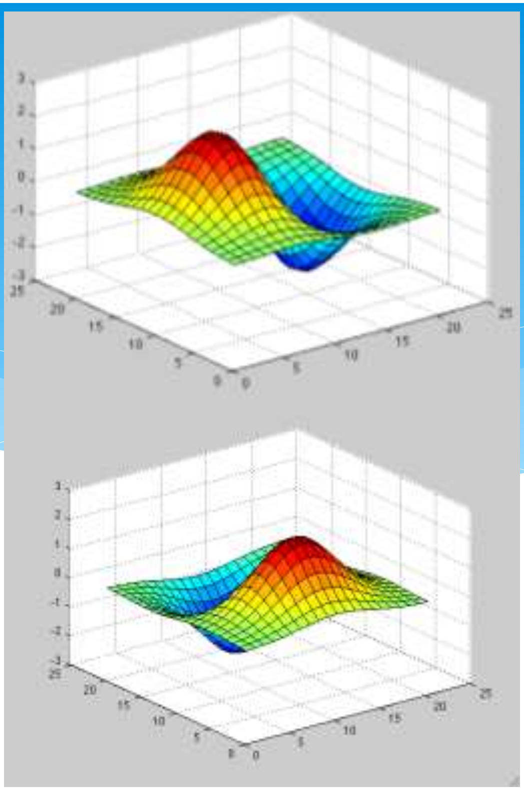
The scale of the smoothing filter affects derivative estimates, and also the semantics of the edges recovered.

Forsyth, 2002

Slide: A. Torralba

$$h_x(x,y) = \frac{\partial h(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

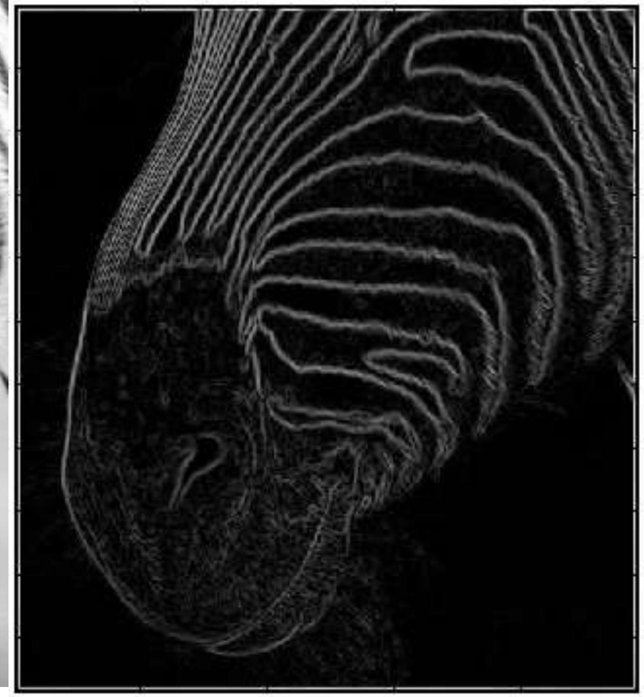
$$h_y(x,y) = \frac{\partial h(x,y)}{\partial x} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



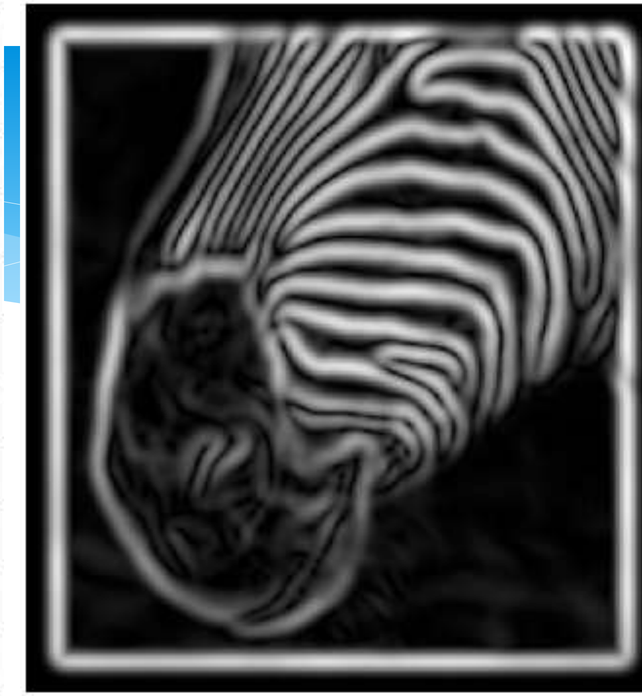
Magnitude: $h_x(x,y)^2 + h_y(x,y)^2$ *Edge strength*

Angle: $\arctan\left(\frac{h_y(x,y)}{h_x(x,y)}\right)$ *Edge normal*

Slide: A. Torralba



Gradient magnitudes at scale 1



Gradient magnitudes at scale 2

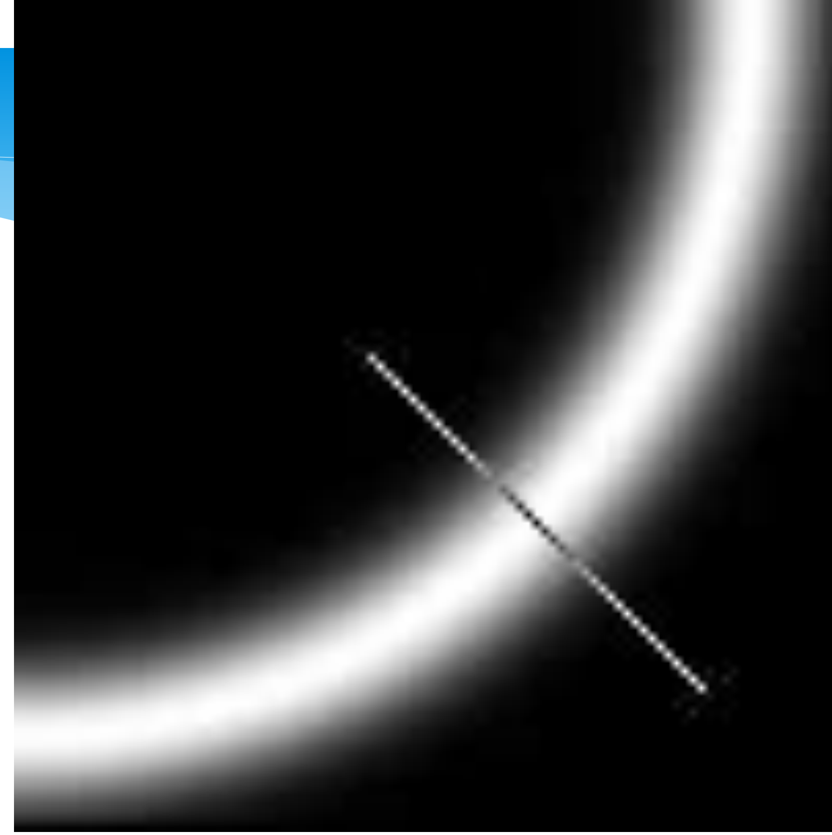
Issues:

1) The gradient magnitude at different scales is different; which should we choose?
2) The gradient magnitude is large along thick trail; how do we identify the significant points?
3) How do we link the relevant points up into curves?
4) Noise.

The scale of the smoothing filter affects derivative estimates, and also the semantics of the edges recovered.

Forsyth, 2002

Slide: A. Torralba

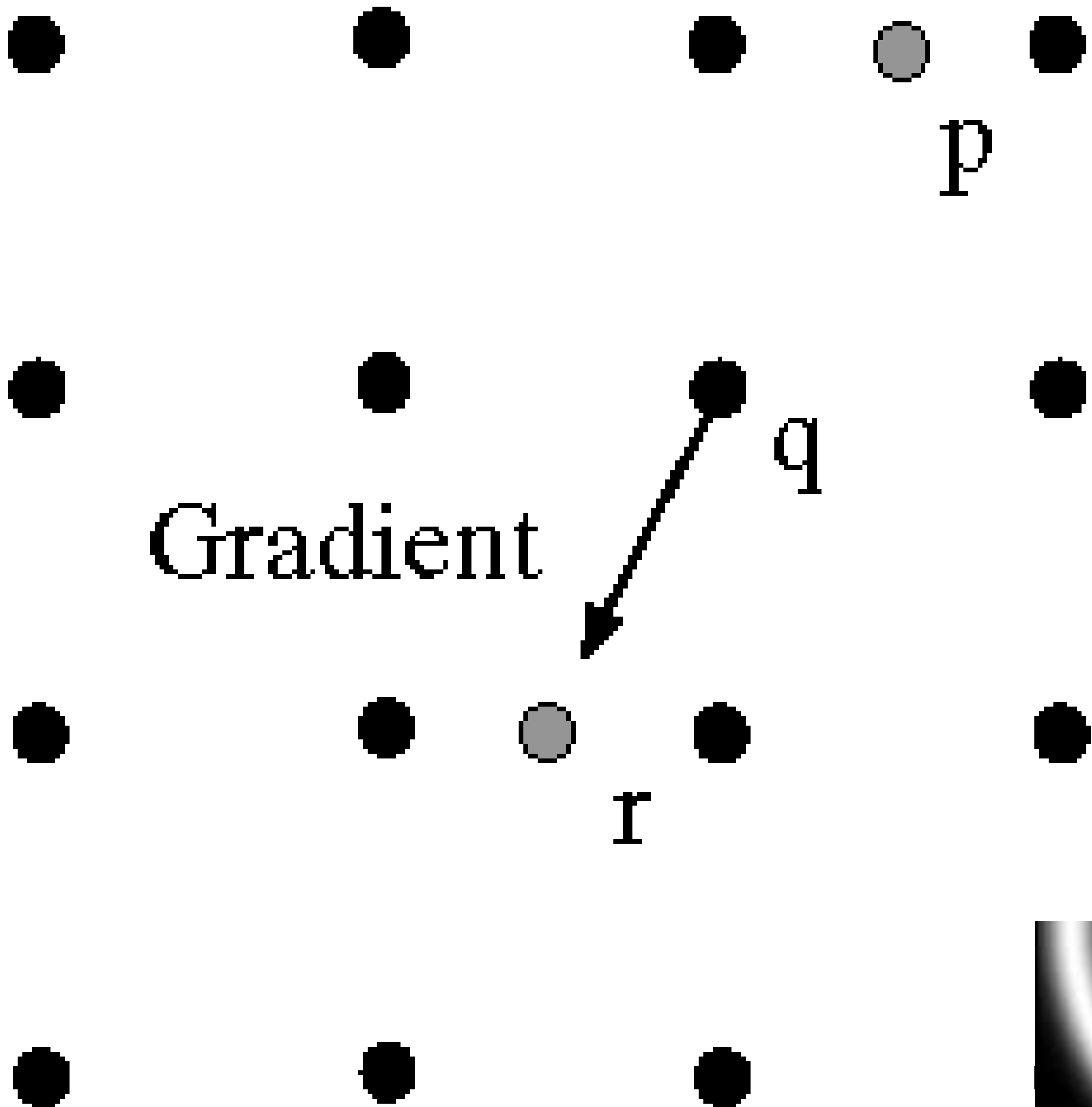


We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Forsyth, 2002

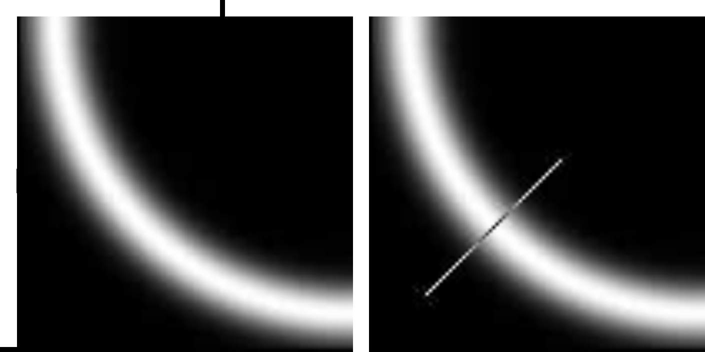
Slide: A. Torralba

Non-maximum suppression

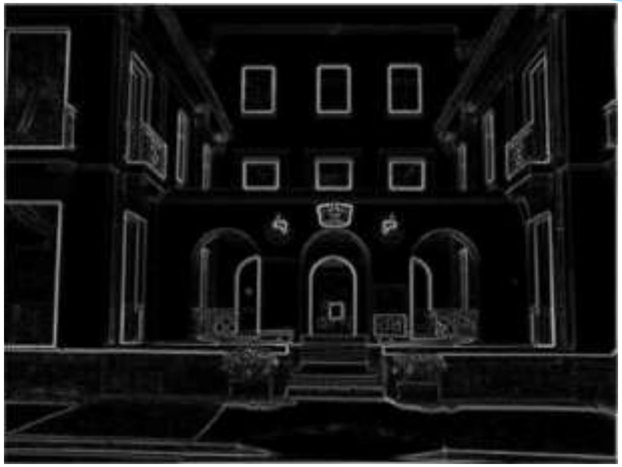
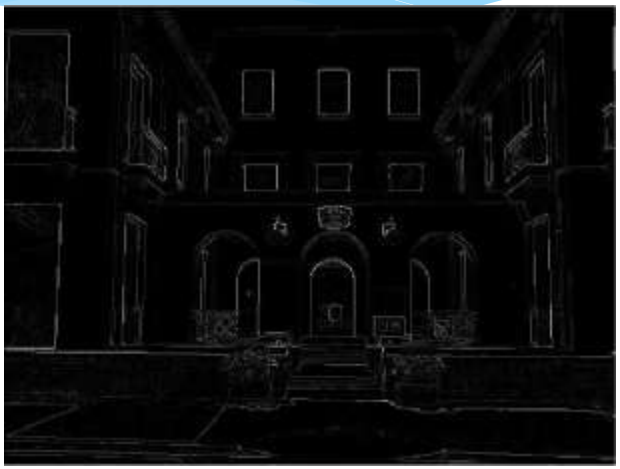


Forsyth, 2002

At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.



Examples: Non-Maximum Suppression

courtesy of G. Loy

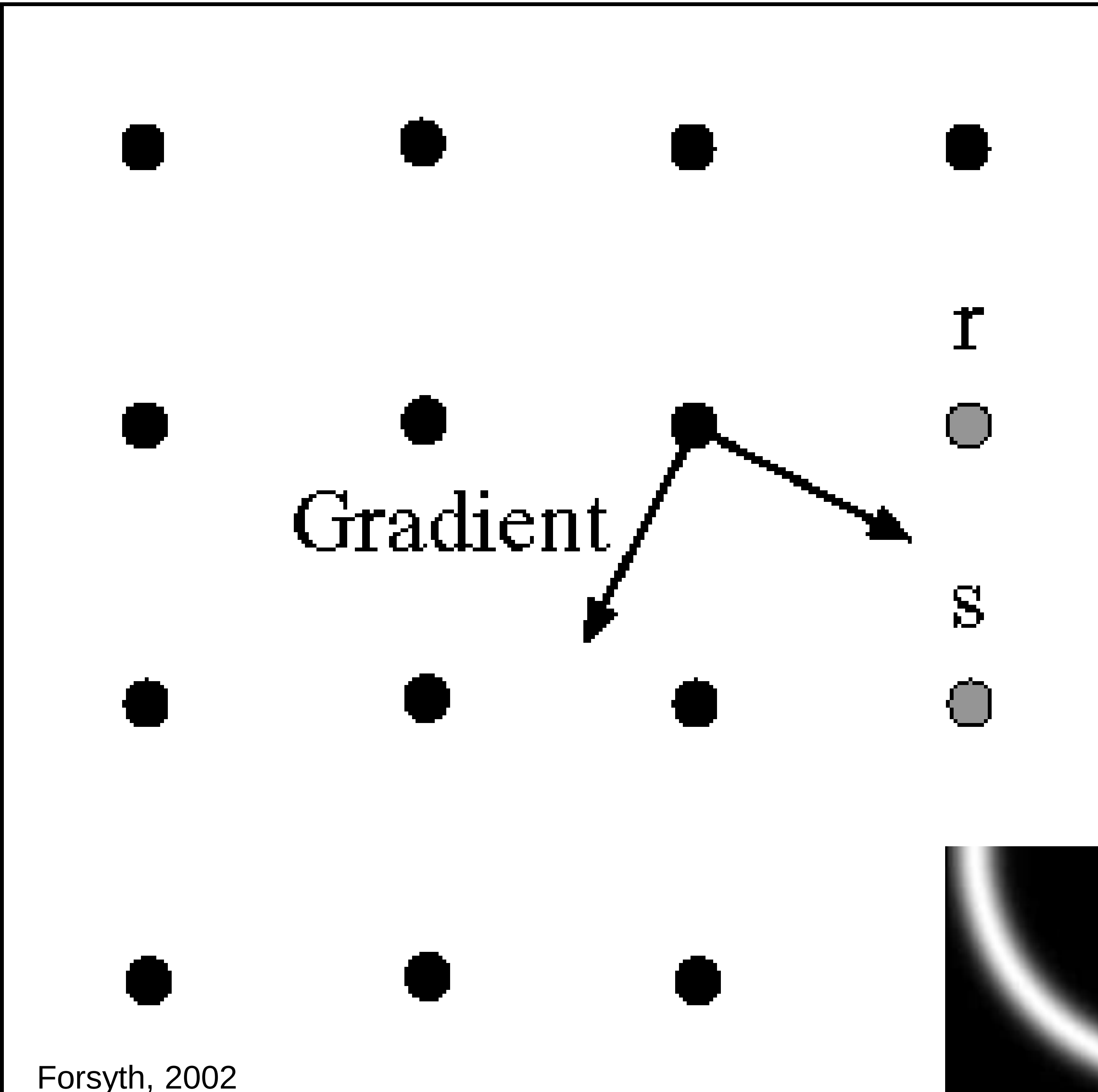
Original image

Gradient magnitude

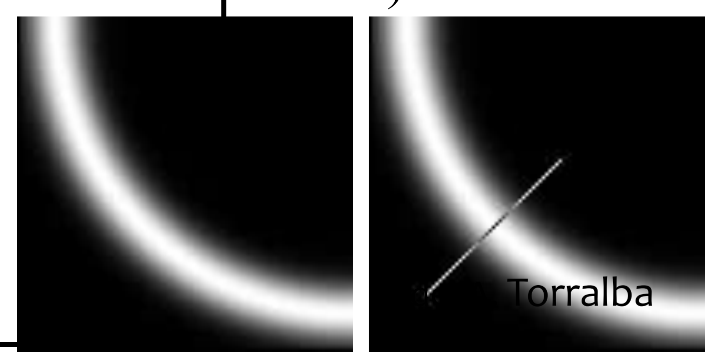
Non-maxima suppressed

Slide credit: Christopher Rasmussen

Predicting the next edge point

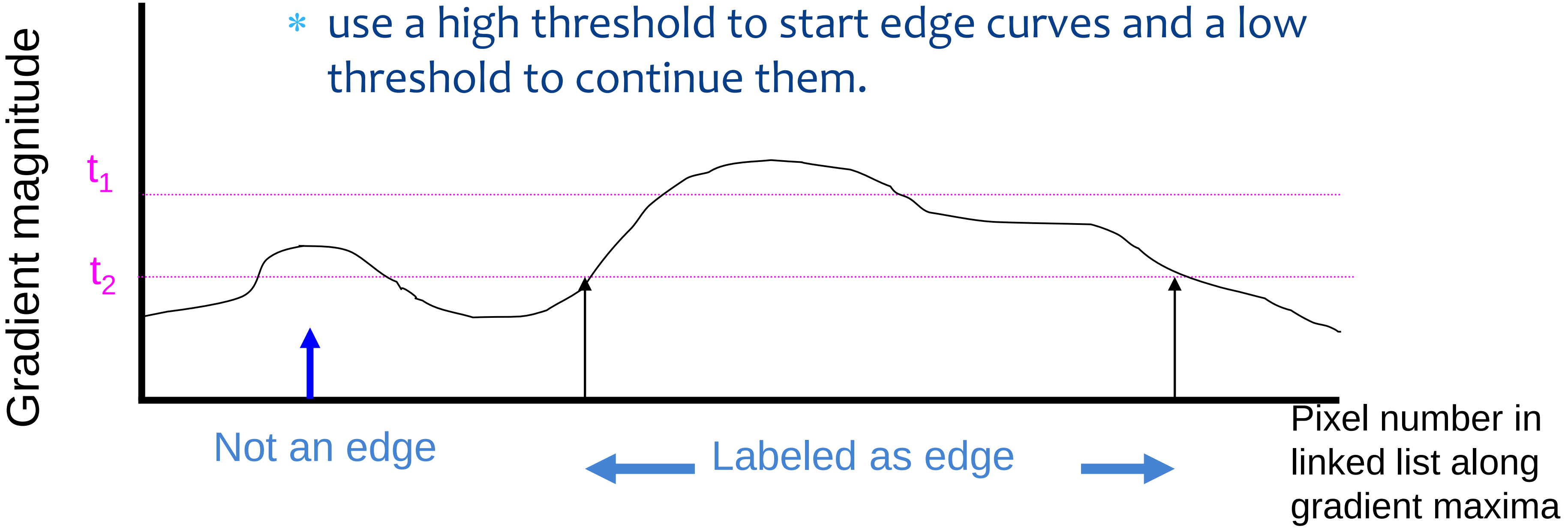


Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).



Closing edge gaps

* Check that maximum value of gradient value is sufficiently large
  * drop-outs? use **hysteresis**
    * use a high threshold to start edge curves and a low threshold to continue them.



Slide: A. Torralba

Example: Canny Edge Detection

Original image

gap is gone

Strong + connected weak edges

Strong edges only

Weak edges

courtesy of G. Loy

edges

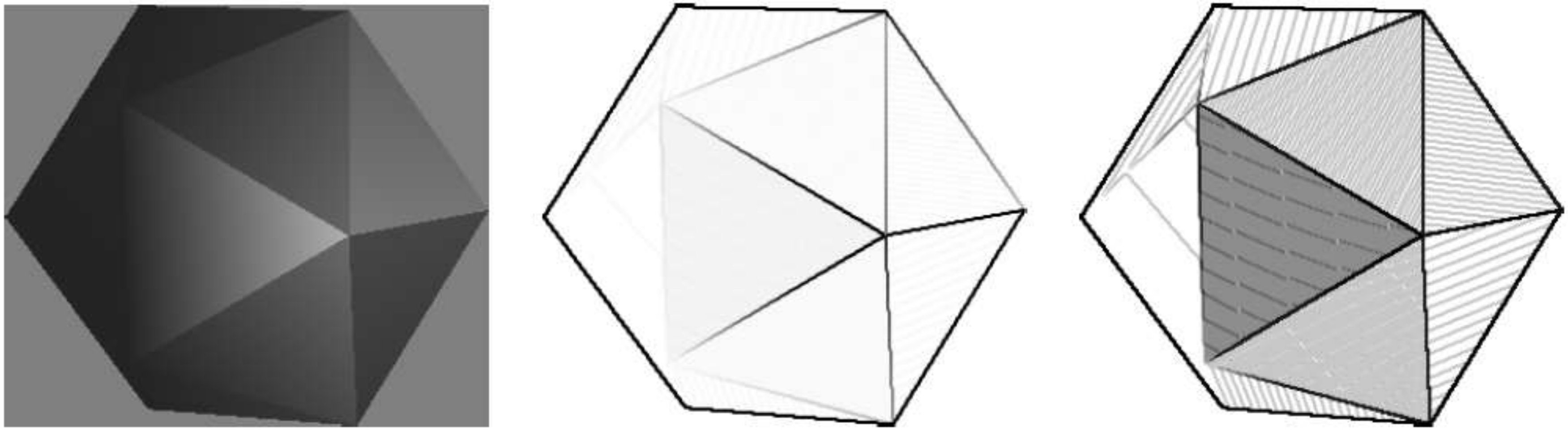
* Issues:
  * isn’t it way too early to be thresholding, based on local, low-level pixel information alone?

Slide: A. Torralba

Slide: A. Torralba

Slide: A. Torralba

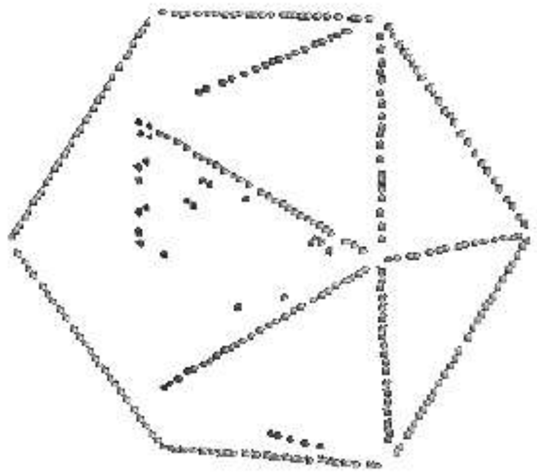
Effect of thresholding

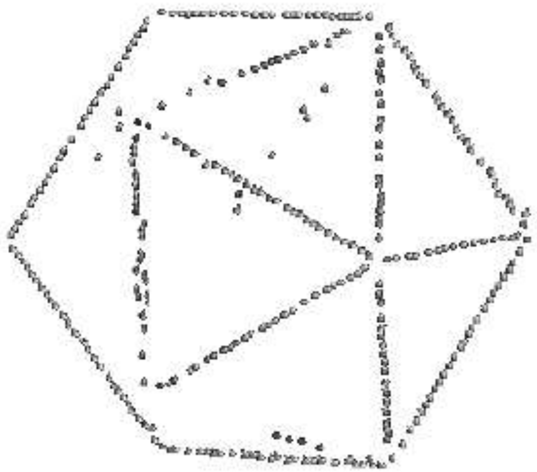


(a) (b) (c)

Kalkan et al., Int Conf. on Computer Vision Theory and Applications, 2008.

Effect of thresholding





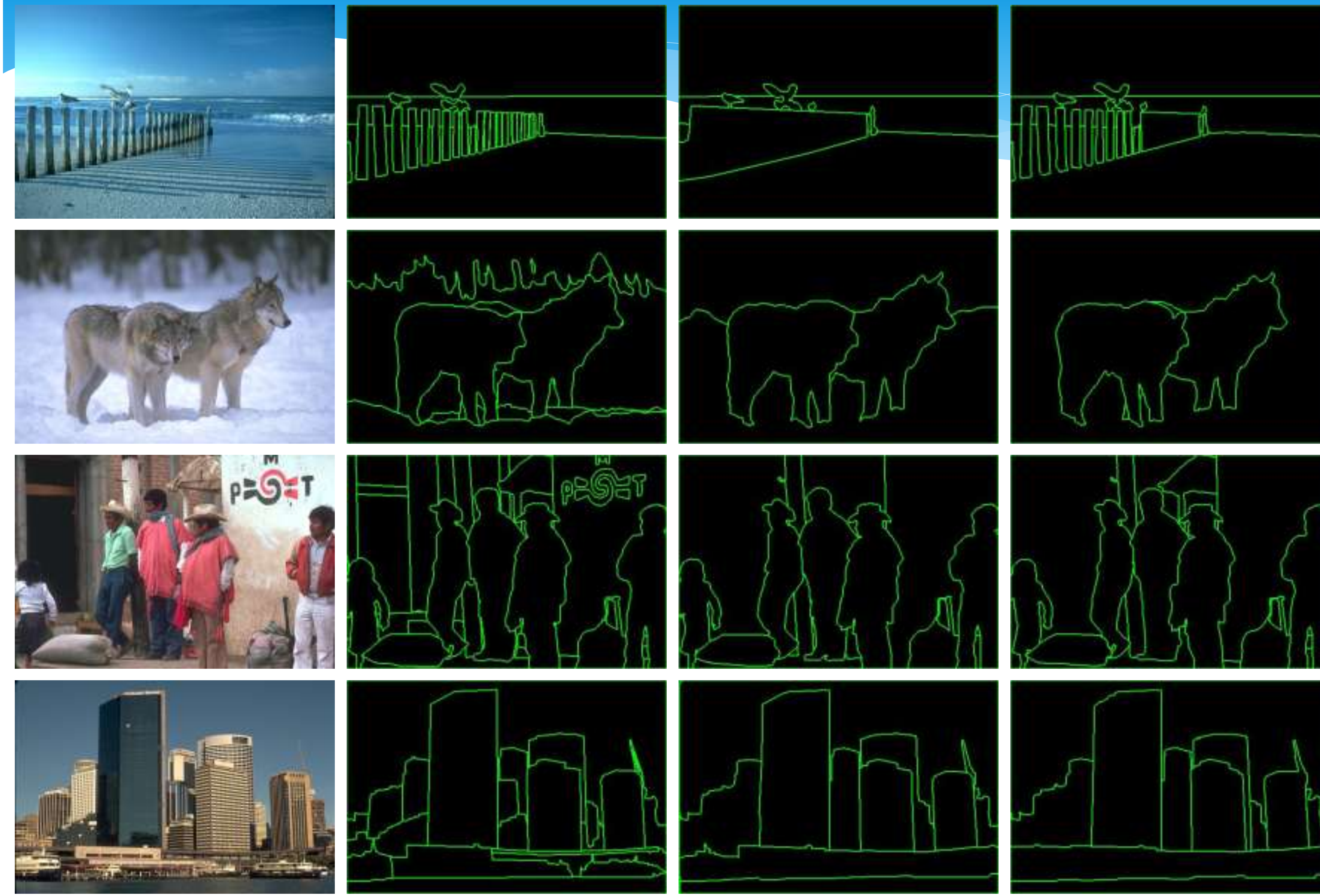
Kalkan et al., Int Conf. on Computer Vision Theory and Applications, 2008.

Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues

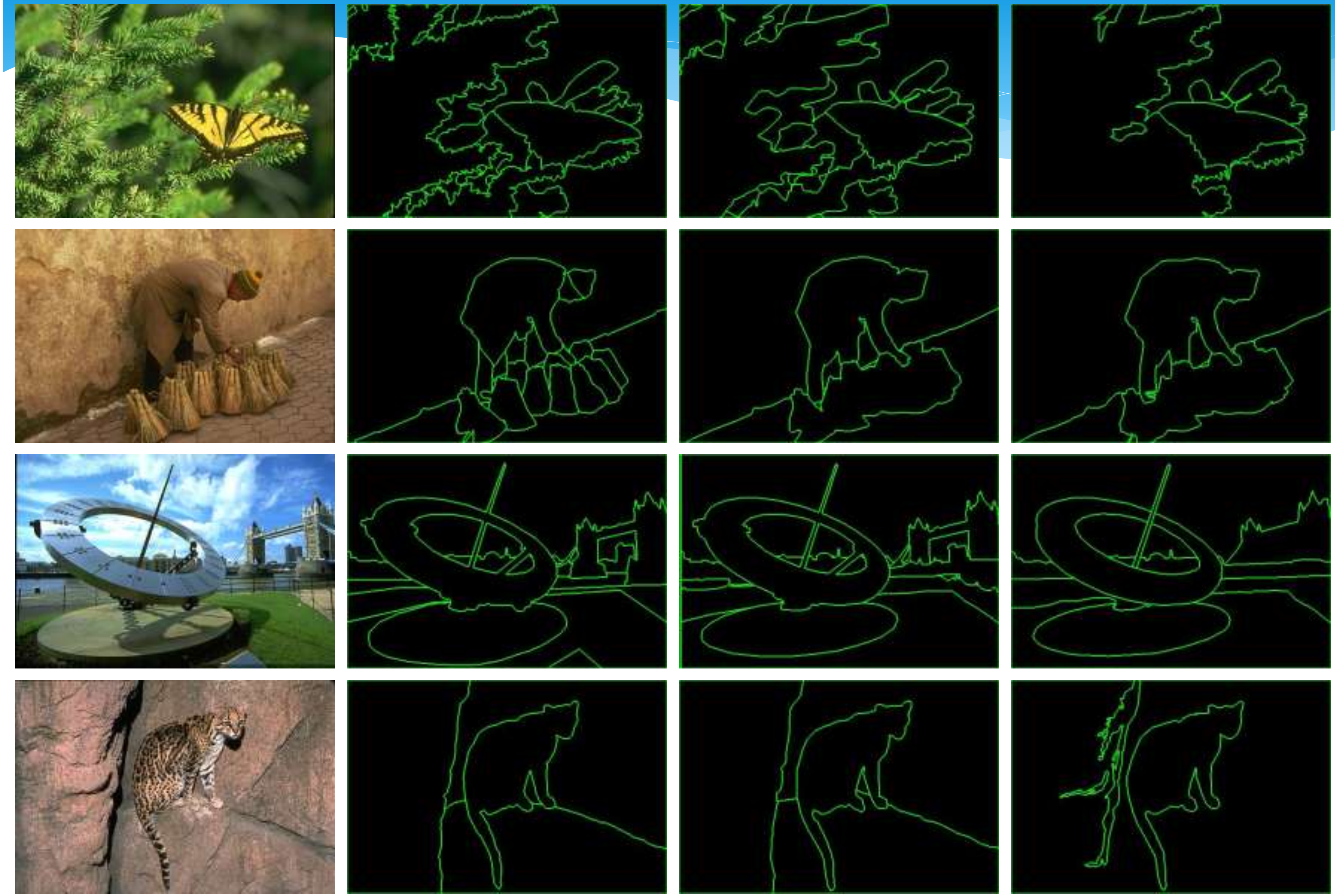
David R. Martin, *Member*, *IEEE*, Charless C. Fowlkes, and Jitendra Malik, *Member*, *IEEE*

Abstract—The goal of this work is to accurately detect and localize boundaries in natural scenes using local image measurements. We formulate features that respond to characteristic changes in brightness, color, and texture associated with natural boundaries. In order to combine the information from these features in an optimal way, we train a classifier using human labeled images as ground truth. The output of this classifier provides the posterior probability of a boundary at each image location and orientation. We present precision-recall curves showing that the resulting detector significantly outperforms existing approaches. Our two main results are 1) that cue combination can be performed adequately with a simple linear model and 2) that a proper, explicit treatment of texture is required to detect boundaries in natural images.

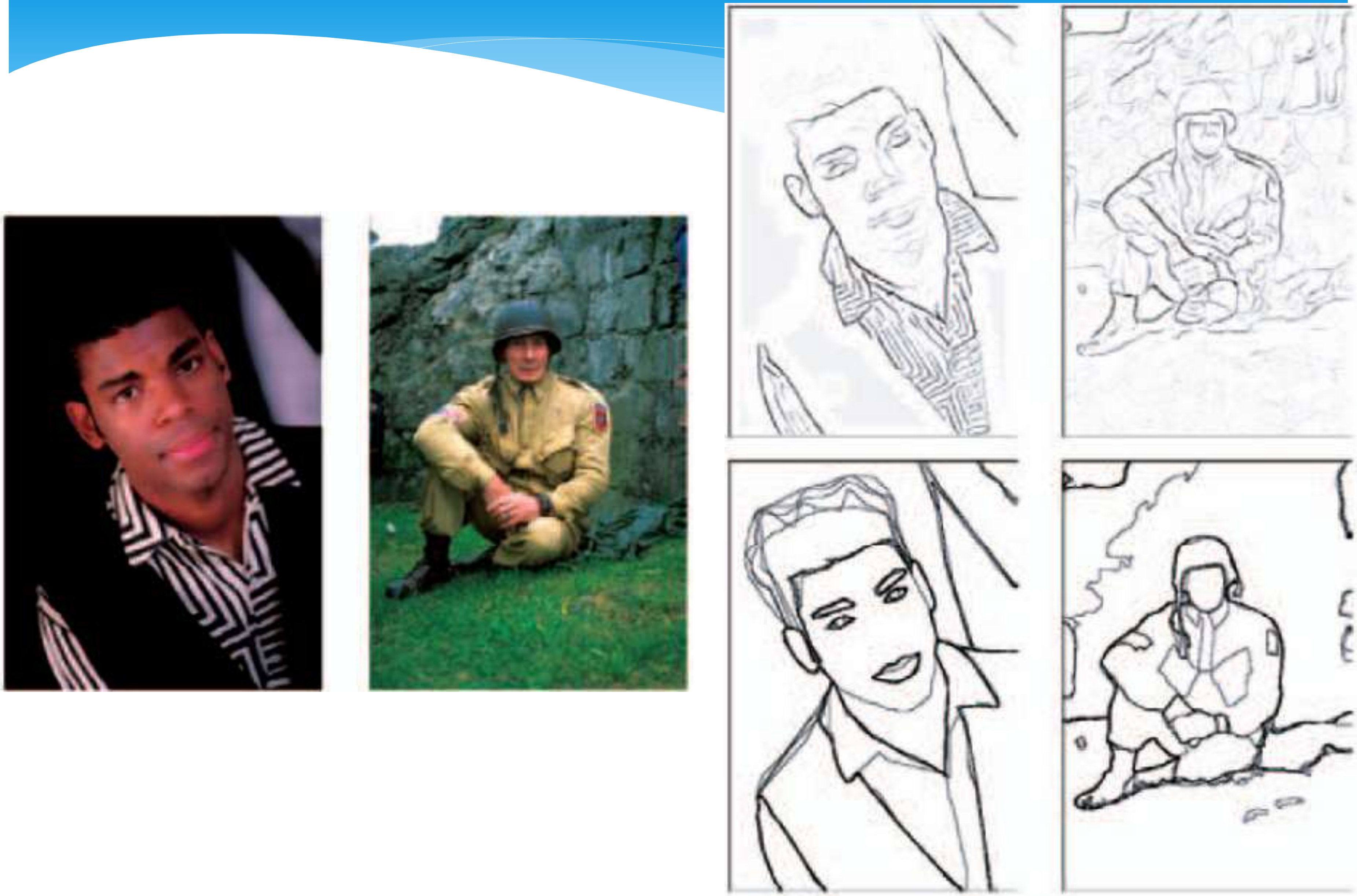
Slides credit: Jitendra Malik



Slides credit: Jitendra Malik



Slides credit: Jitendra Malik



Martin et al., PAMI, 2004.

Gabor Filters, Efficient Encoding and Edges

Sensory Coding

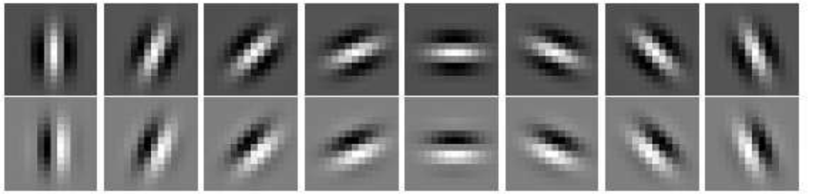
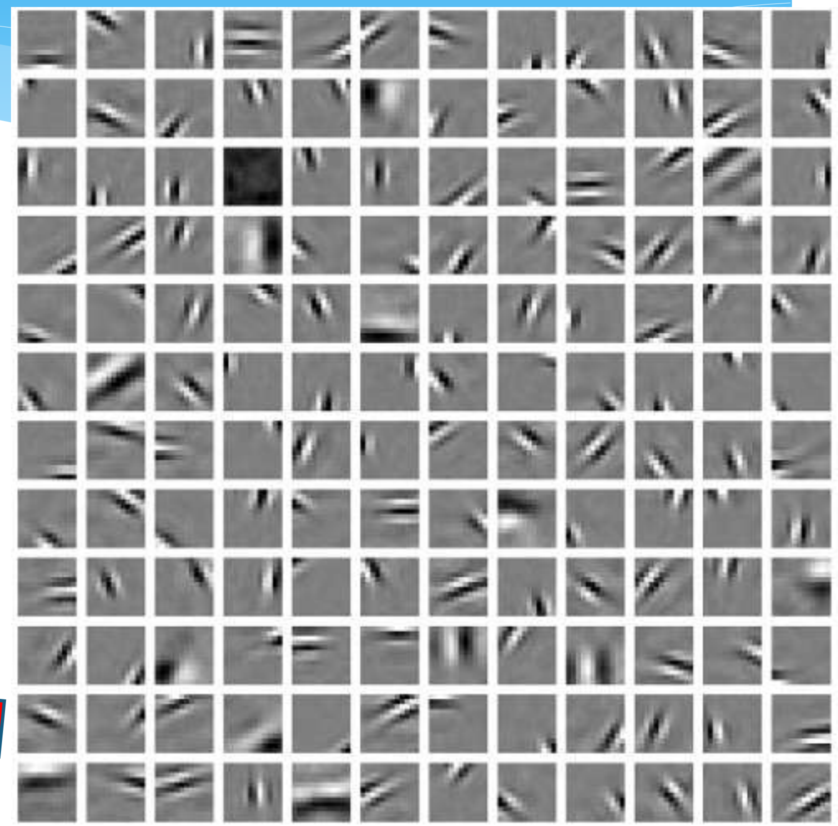


Figure 3: Real (first row) and imaginary (second row) parts of eight orientation Gabor wavelets.

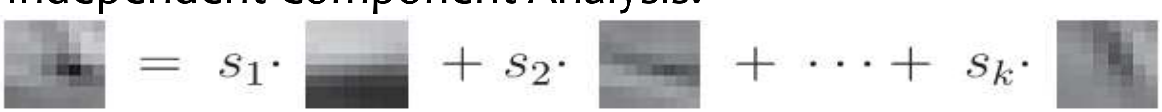
(Kalkan et al., 2008)

- Efficient Coding Hypothesis
  - "The goal of early vision (or, early visual processing) is to provide an efficient representation of the incoming visual signal"
  - (Field, 1987; Hateren, 1998; Bell & Sejknowski, 1996, 1997; Olshausen & Field, 1996; Hyvarinen, 2010)
- For a review & critics:
  - (Simoncelli & Olshausen, 2001; Simoncelli, 2003)



(Olshausen & Field, 1996)

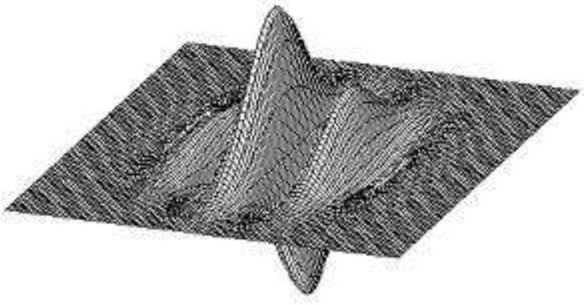
Independent Component Analysis:

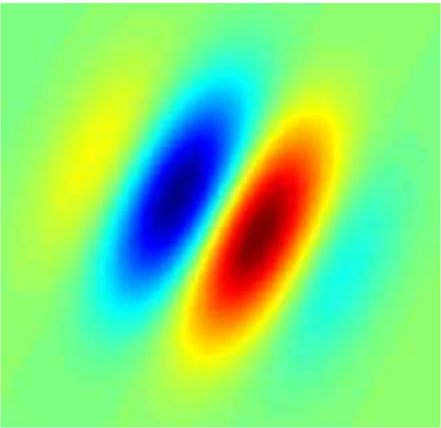


(Hyvarinen, 2010)

Gabor Filter

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right)$$



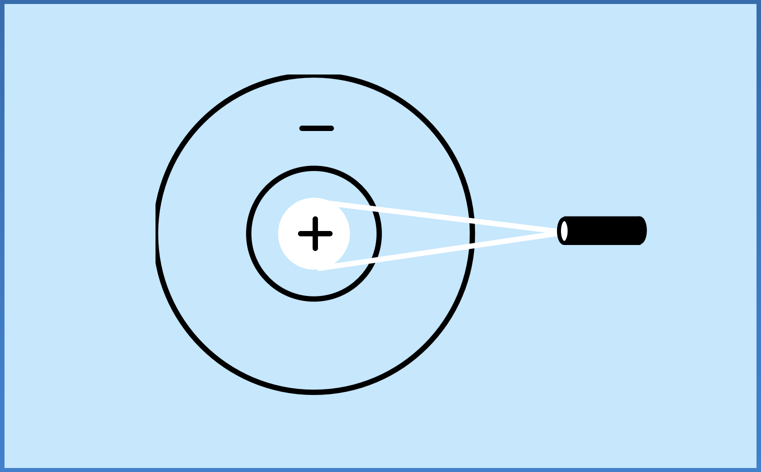


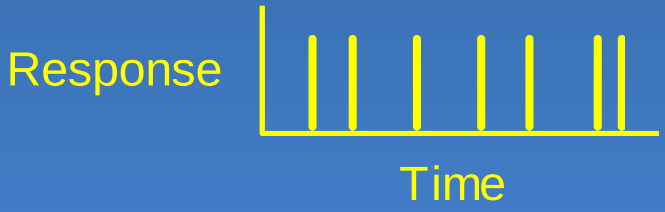
Gabor Filters vs.
Cortical Receptive Fields

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





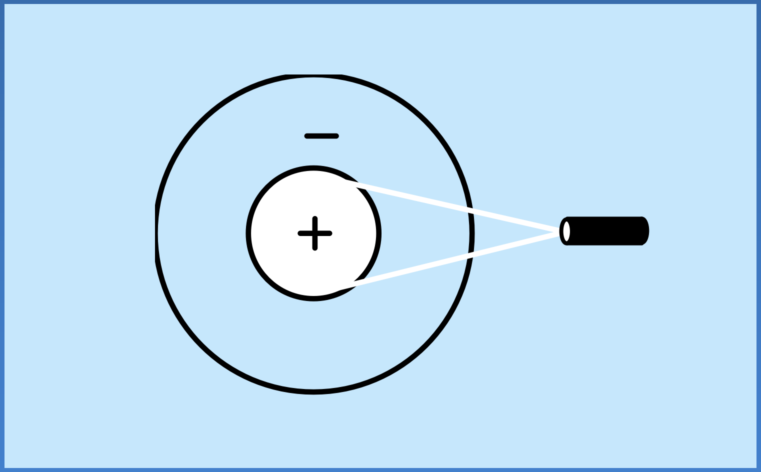
Stimulus condition

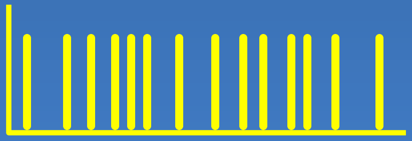
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





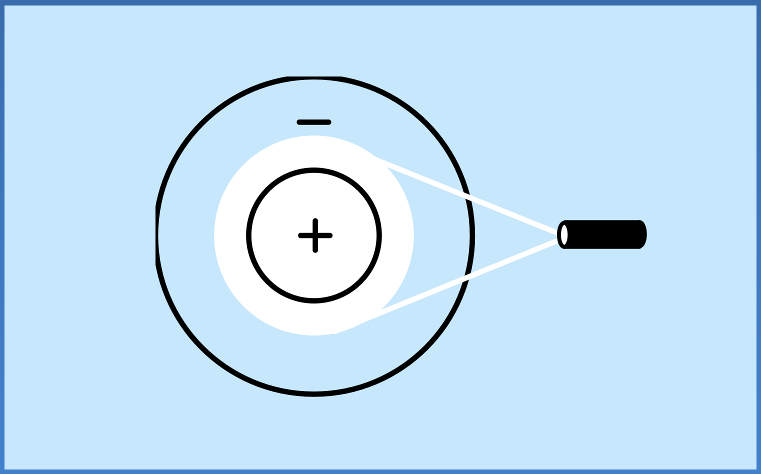
Stimulus condition

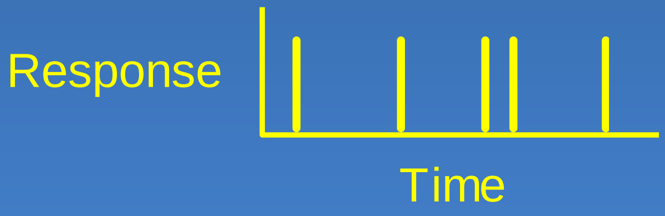
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





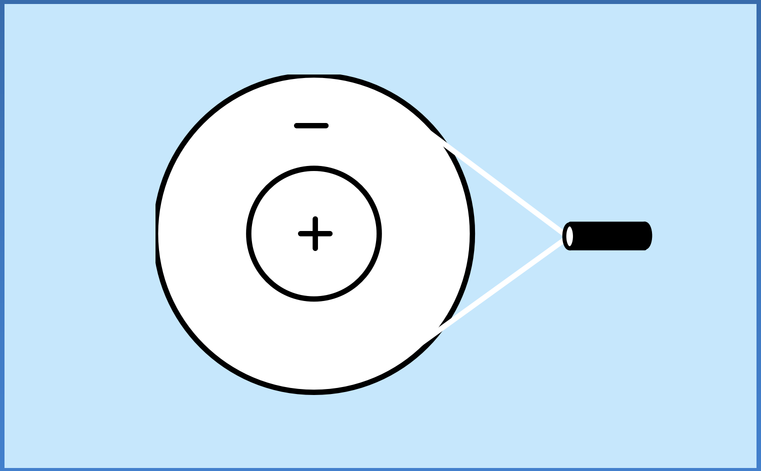
Stimulus condition

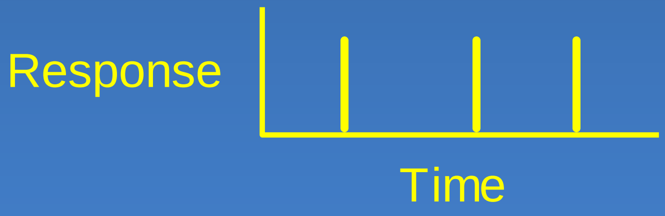
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





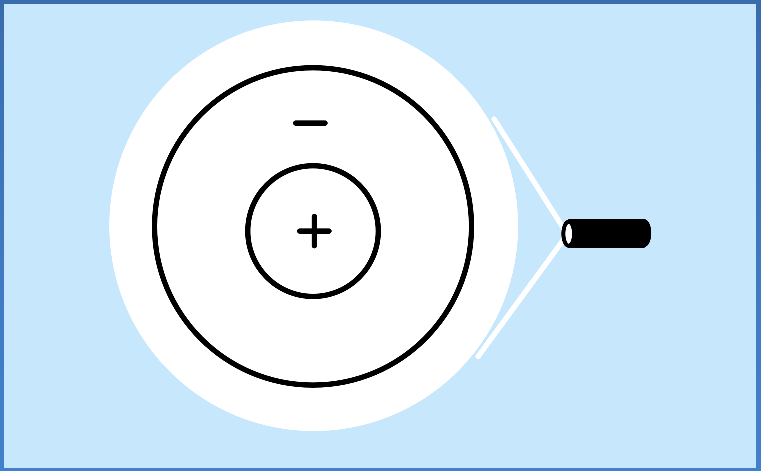
Stimulus condition

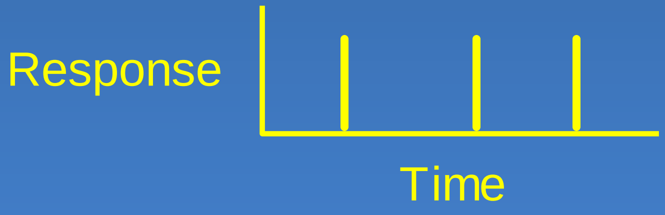
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





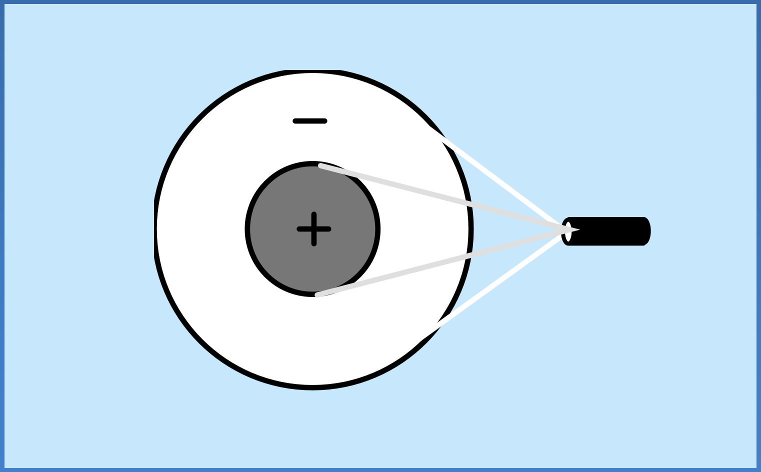
Stimulus condition

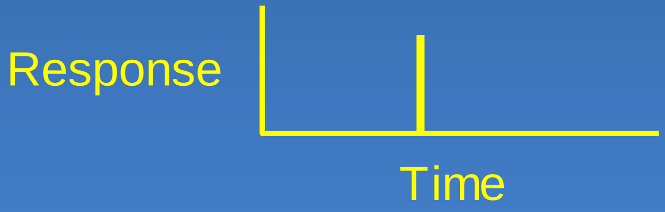
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround



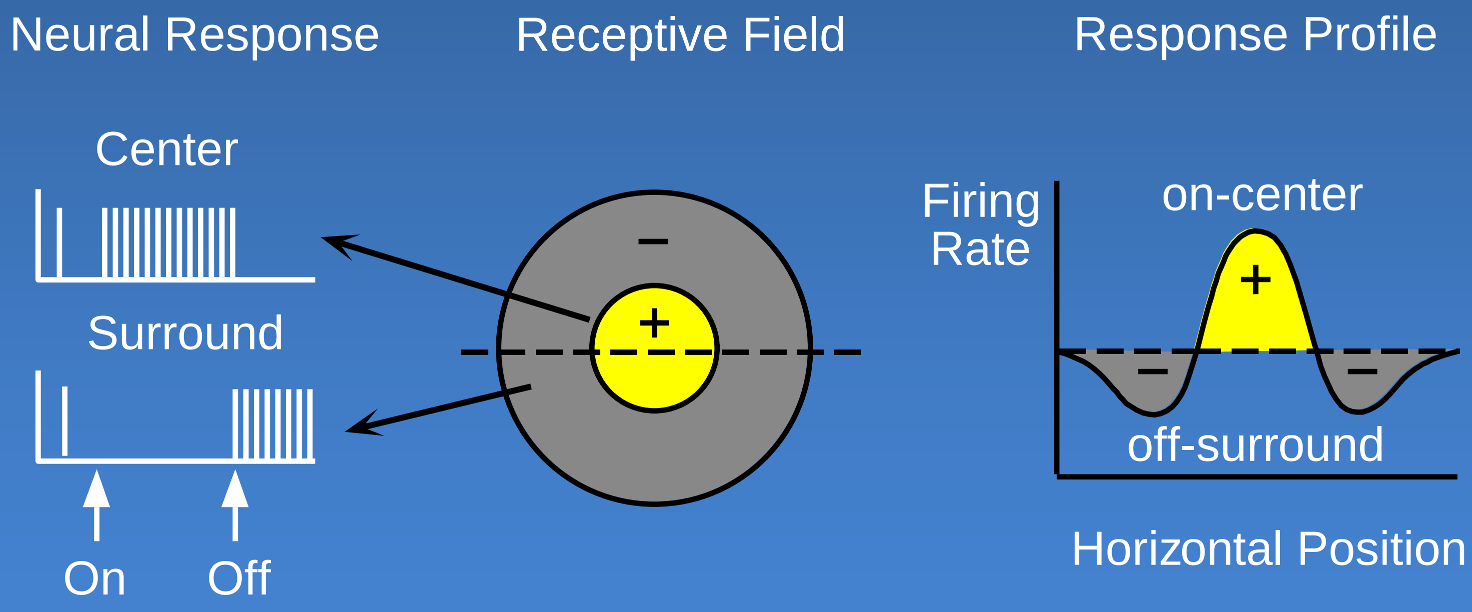


Stimulus condition

Electrical response

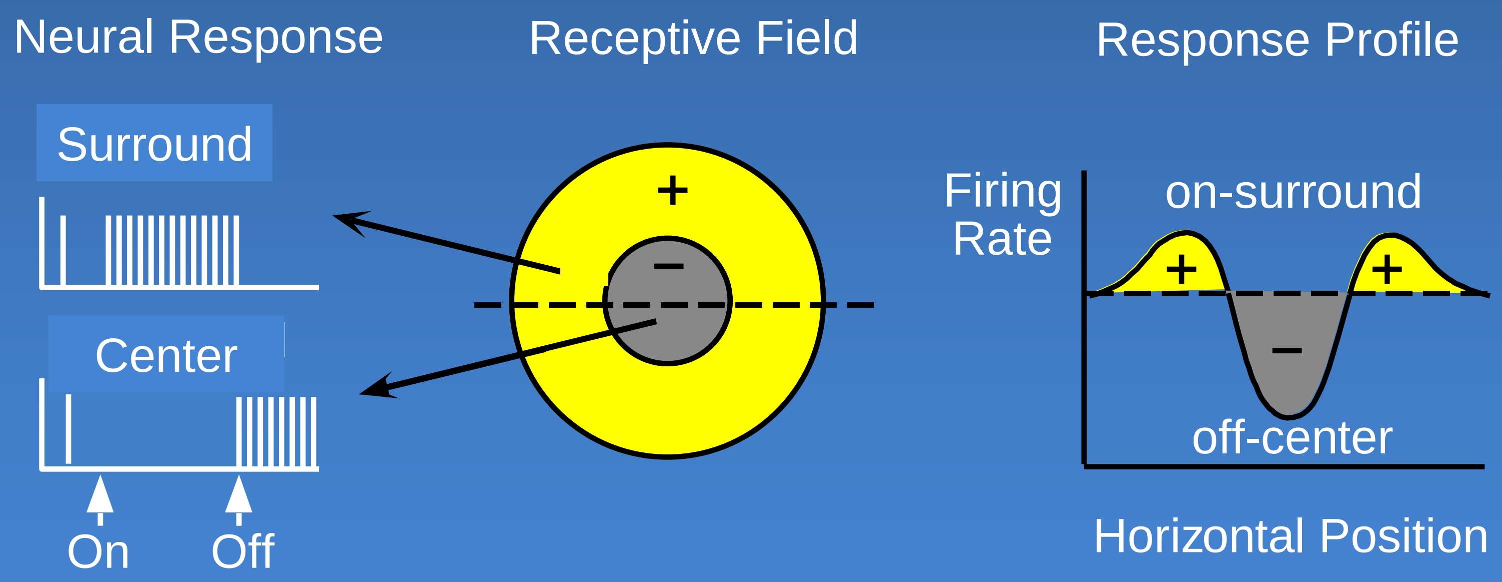
Retinal Receptive Fields

RF of On-center Off-surround cells



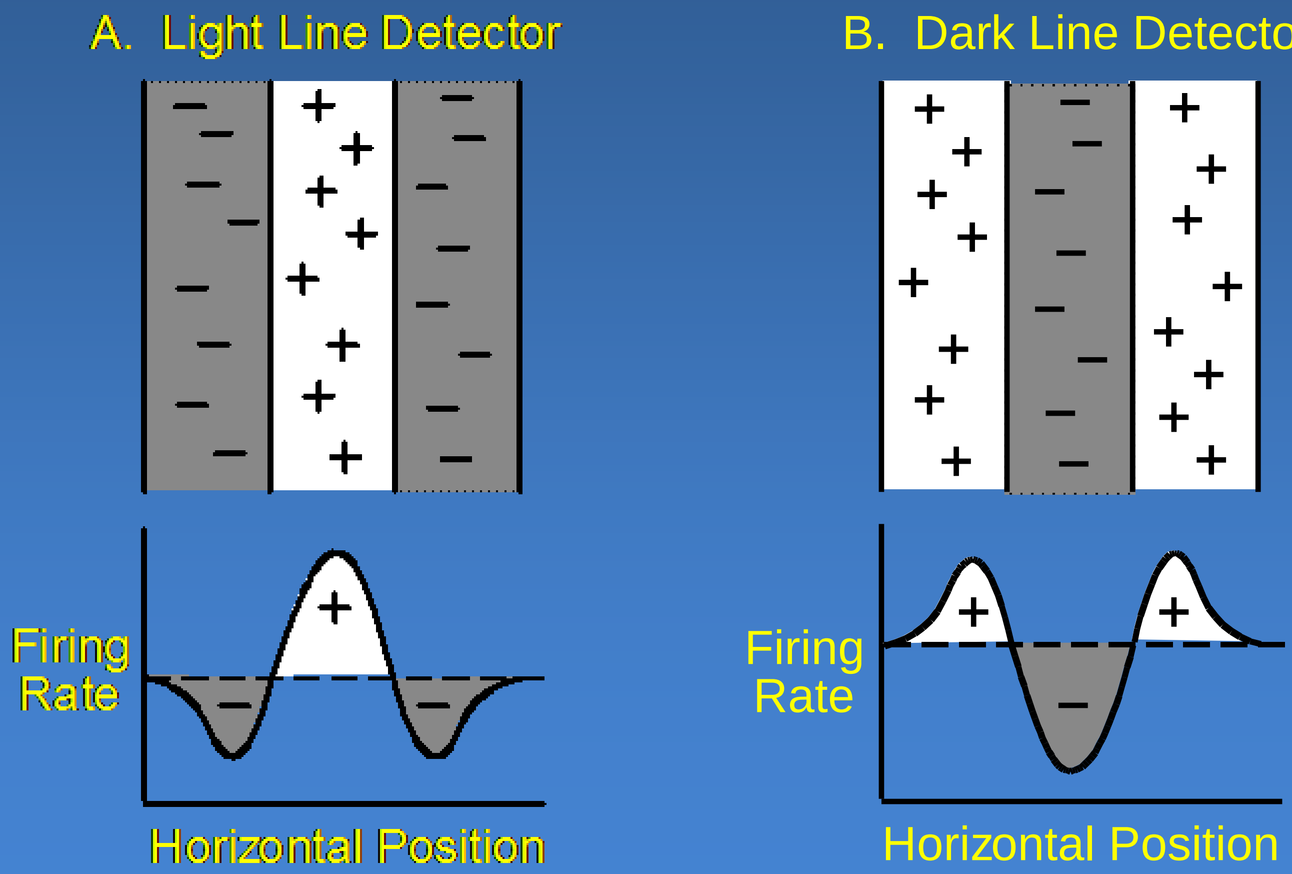
Retinal Receptive Fields

RF of Off-center On-surround cells



Cortical Receptive Fields

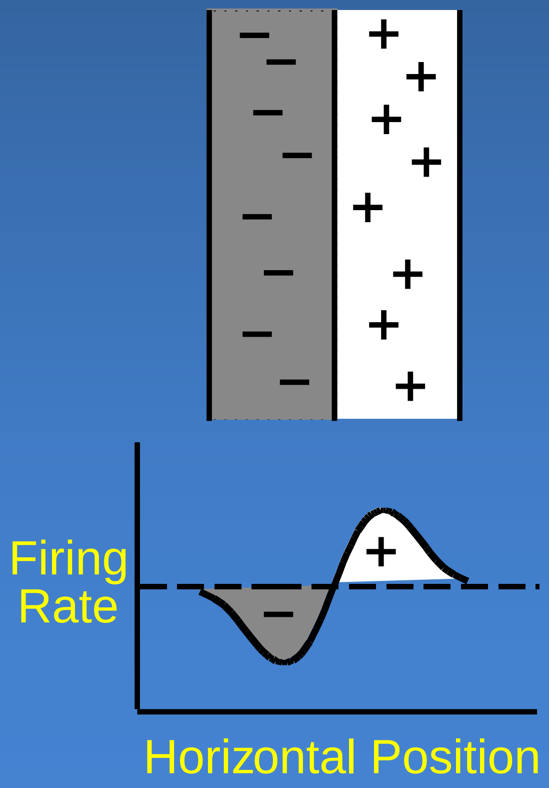
Simple Cells: "Line Detectors"



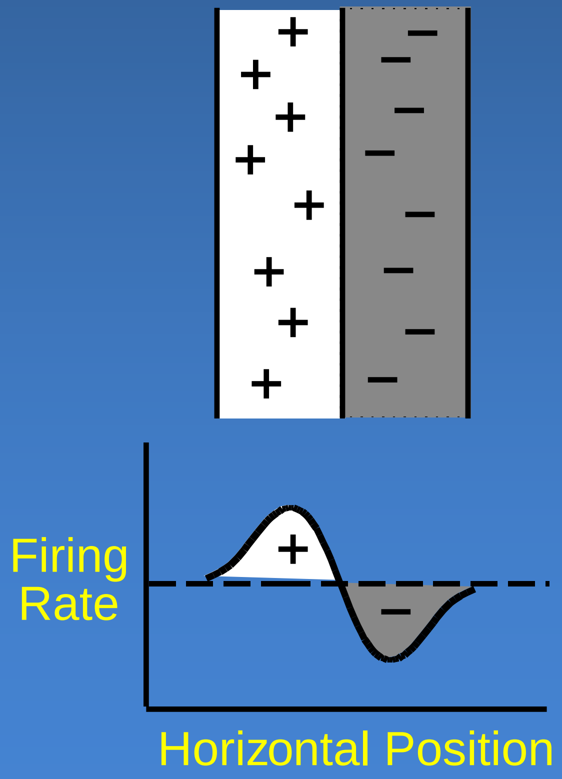
Cortical Receptive Fields

Simple Cells: “Edge Detectors”

C. Dark-to-light Edge Detector



D. Light-to-dark Edge Detector



So, what does it say?

* The early ‘low-level filtering’ achieves efficient encoding by finding intensity changes.
  * With Gabor-like filters.
* This is one of the important evidence for the hypothesis that our visual system is tuned to the statistical regularities in the environment.

Summary of this week

Topics

* Cameras, projective geometry, calibration
* Filtering
* Edges

Problems

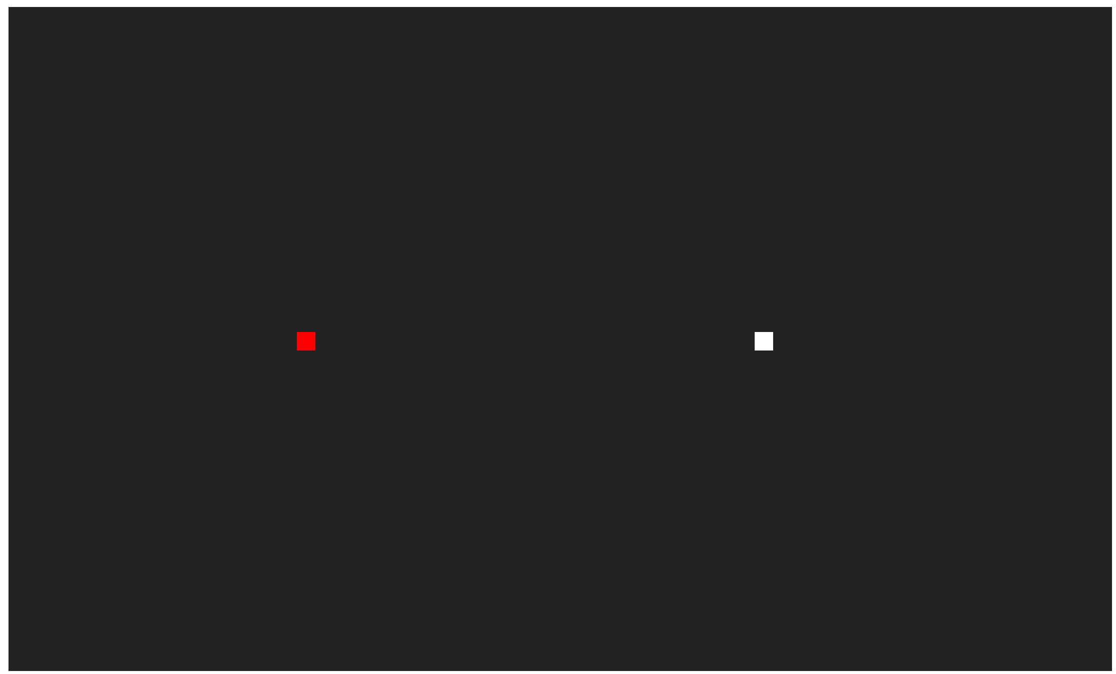
* Scale for filtering
* Thresholding
* Noisy, incomplete, ambiguous information

Reading

* Ch1 & Ch2 from “Computer Vision: A Modern Approach”

Illusion of the week

1. Close your left eye.
2. Stare at the red spot.
3. Move towards the image until the white spot disappears.



http://www.vonrechenberg.ch/blindspots.html