CSC496: Deep learning in computer vision

Prof. Bei Xiao

Lecture 2: A simple vision system, Numpy Primer

Slides mostly from
MIT 6.819 Lecture 1

Overview

- Current status of deep learning? Discuss with Michael Jordan
- Why is vision so hard?
- A simple vision system
- Take-home: Python/Numpy tutorial
- Take-home: Virtual box installation
- Project 1 will be out end of the week

Office hours (DMTI 204)

- Monday: 4pm-5pm (Away October 28th). The office hour on that day will be moved to the previous Friday.

- Wed: 3:30pm-5pm (Traveling September 18th). On that day the office hour will be moved to the morning.

- If you can't make the office hours, you can schedule with me another time to meet up. I need at least 48 hours notice to respond to your email for the scheduling.

Grading

- 65% homework projects (5-6 projects)
- 10% mid-term (in-class) closed book exam
- 15% Final project
- 5% in-class quiz
- 5% attendances. Missing one class will result in 2% reduction of the total grade (reasons accepted, sickness, pre-registered sports, and religious holidays).

Skeptical view of current status of AI revolution (assigned reading)

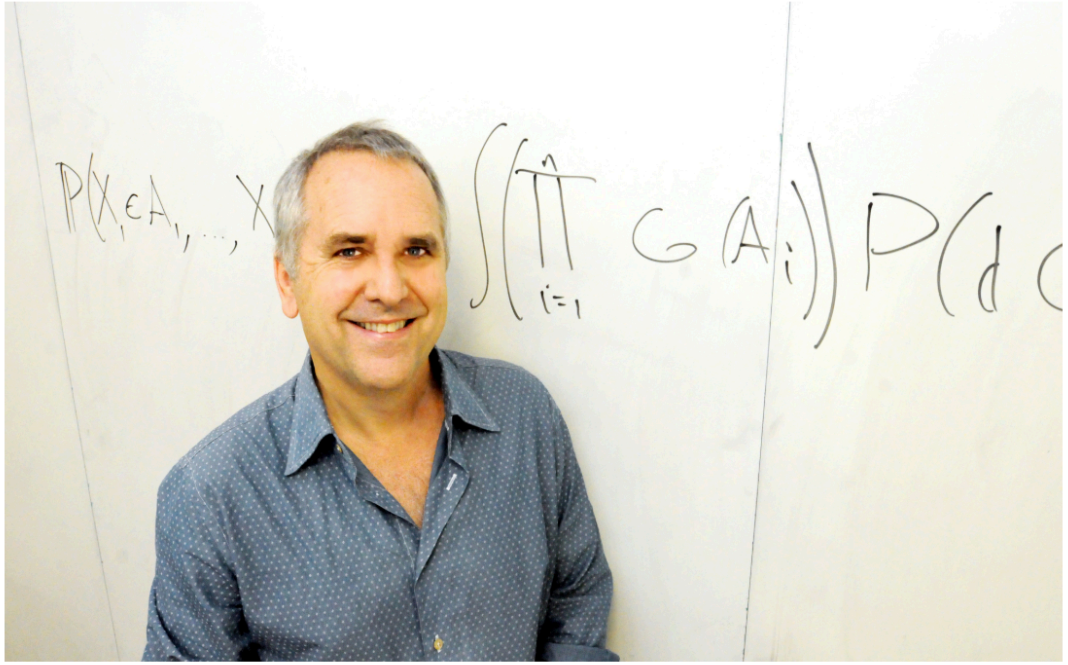


Photo credit: Peg Skorpinski

Artificial Intelligence — The Revolution Hasn't Happened Yet

Discussion:

What is the difference between ”AI” and “Machine learning”?

According to the author, what aspects of the “AI” are most successful so far?

What are some of the skepticism the author raised for the current “AI” revolution?

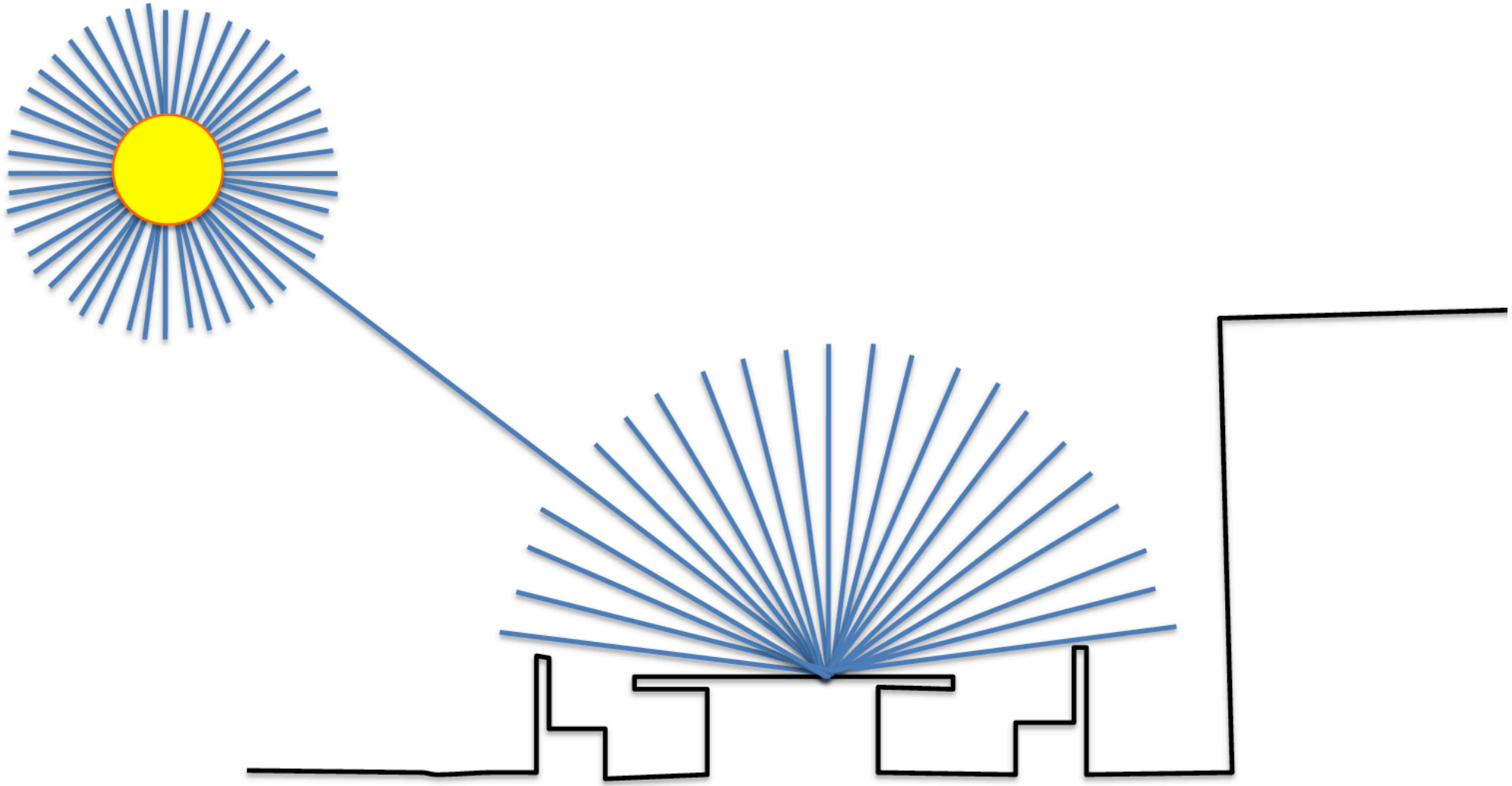
What algorithm is considered the ”core” of AI revolution?

What is IA? Why is it different from AI?

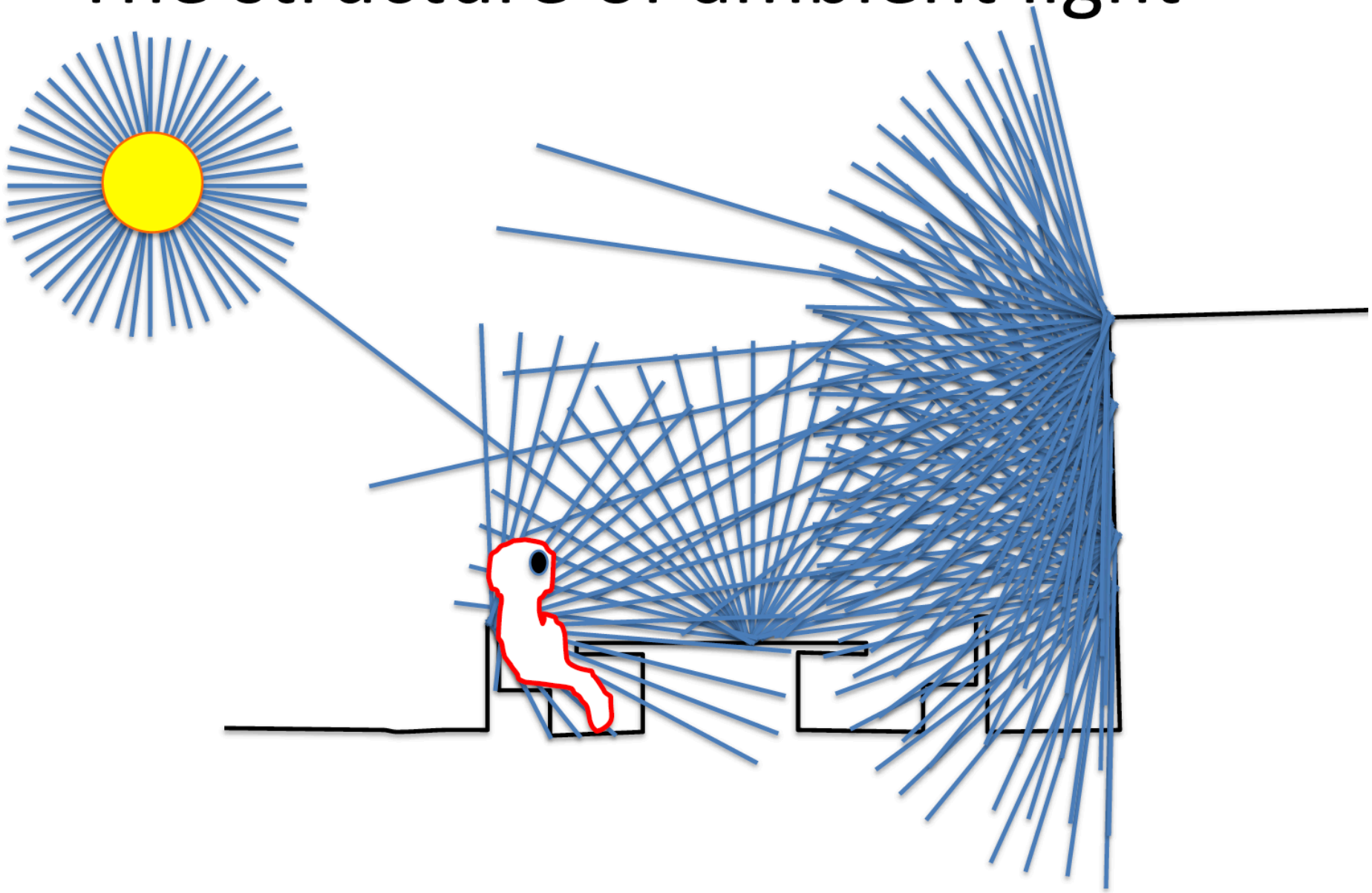
https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7

Why vision is so hard?

The structure of ambient light



The structure of ambient light



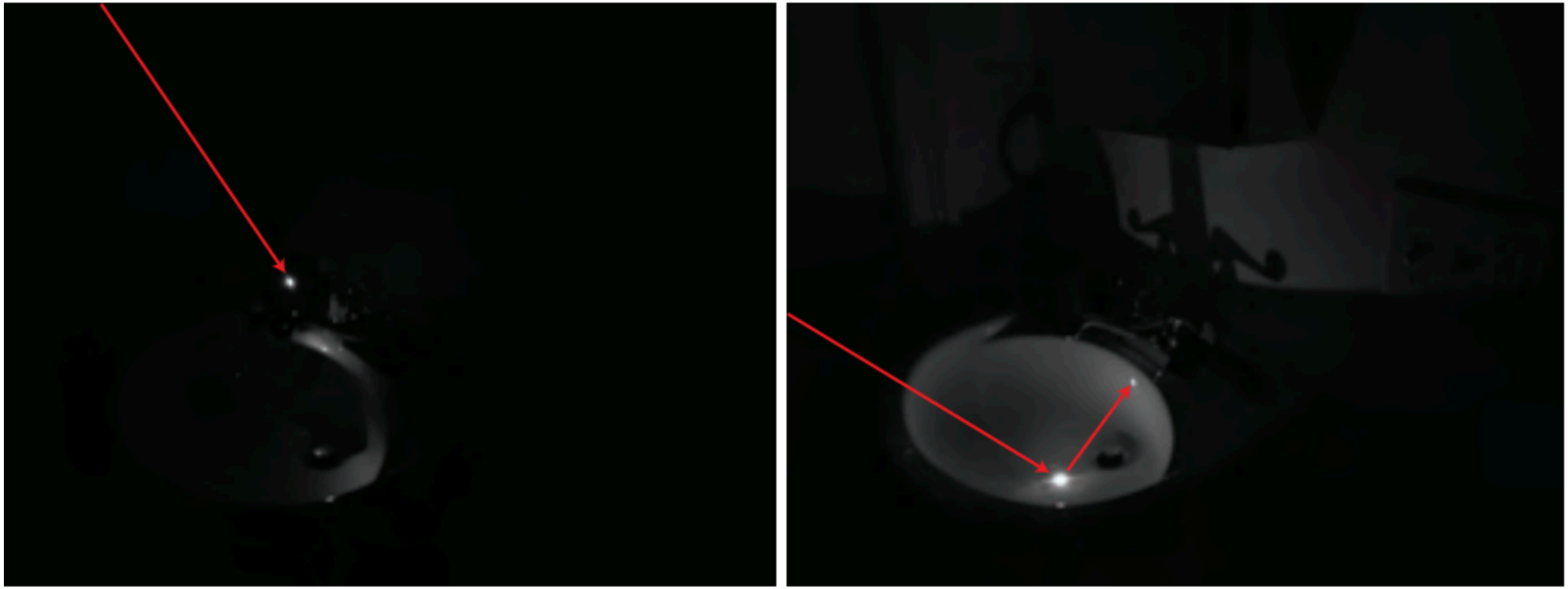


Photo: Antonio Torralba

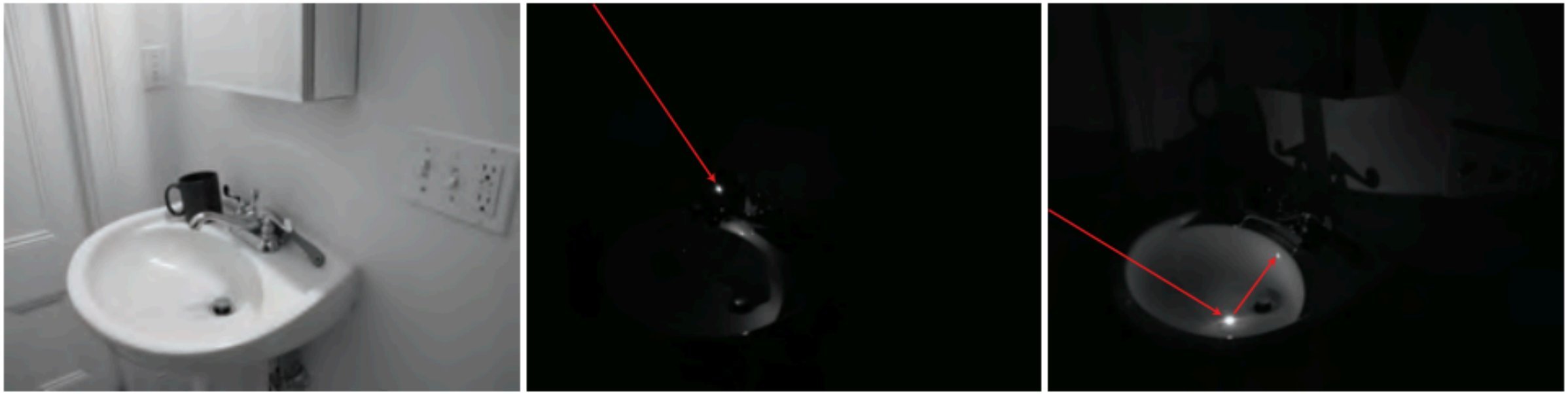


Photo: Antonio Torralba

Plenoptic function

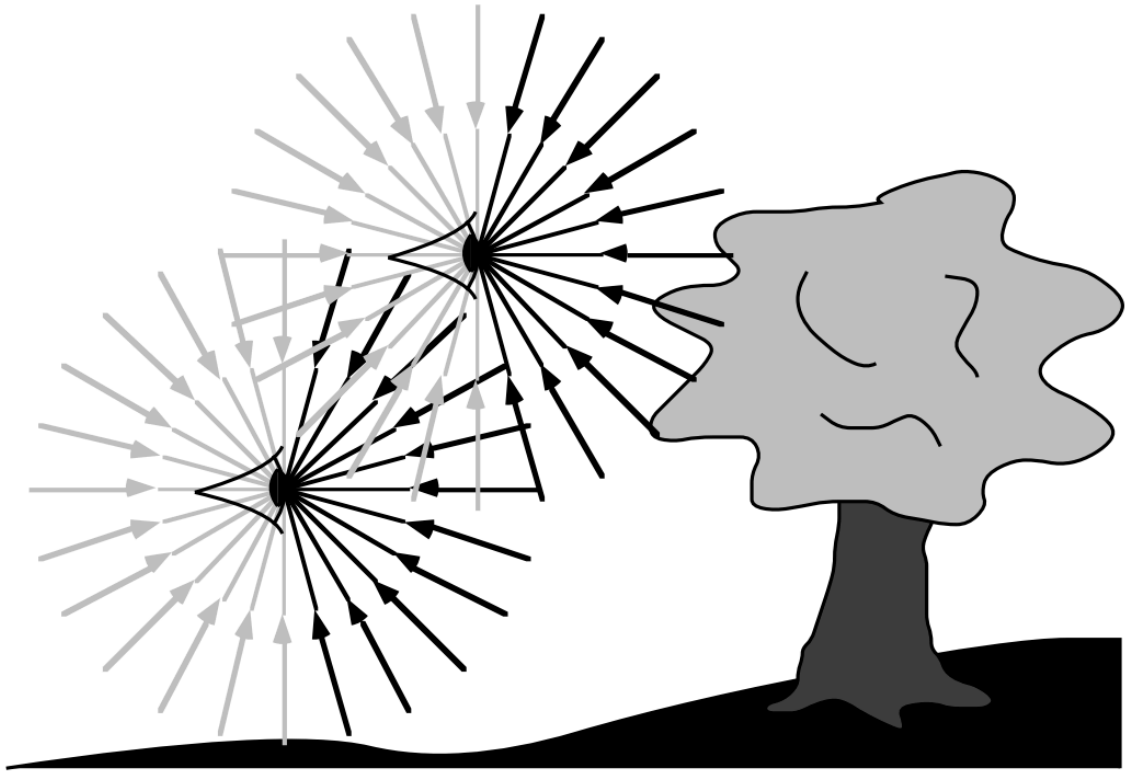
$$P = P(\theta,\phi,\lambda,t,V_x,V_y,V_z). \qquad (1)$$

The Plenoptic Function and the Elements of Early Vision

The world location (X,Y,Z) in the direction given by the angle (θ,8), and with wavelength λ (see later lecture on color and light)

http://persci.mit.edu/pub_pdfs/elements91.pdf

Plenoptic function describes the information of an observer at any point in space and time



For a given observer, most of the light rays are occluded.

Question: is the goal of vison to recover this function?

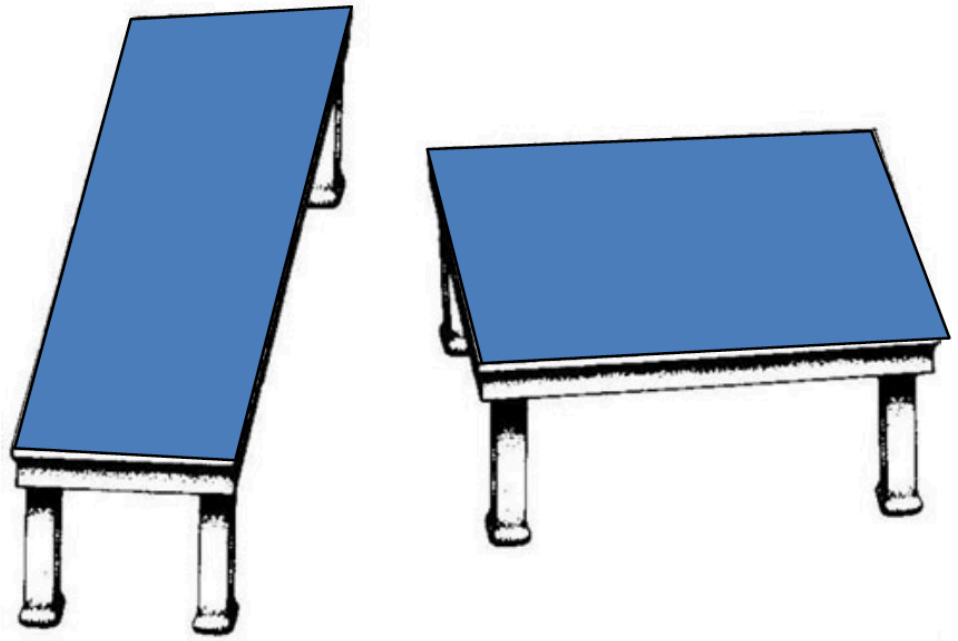
Adelson and Bergen 1991

Measuring light vs. measuring scene properties

Why do we interpret the scene like this?

We perceive two squares, one on top of each other.

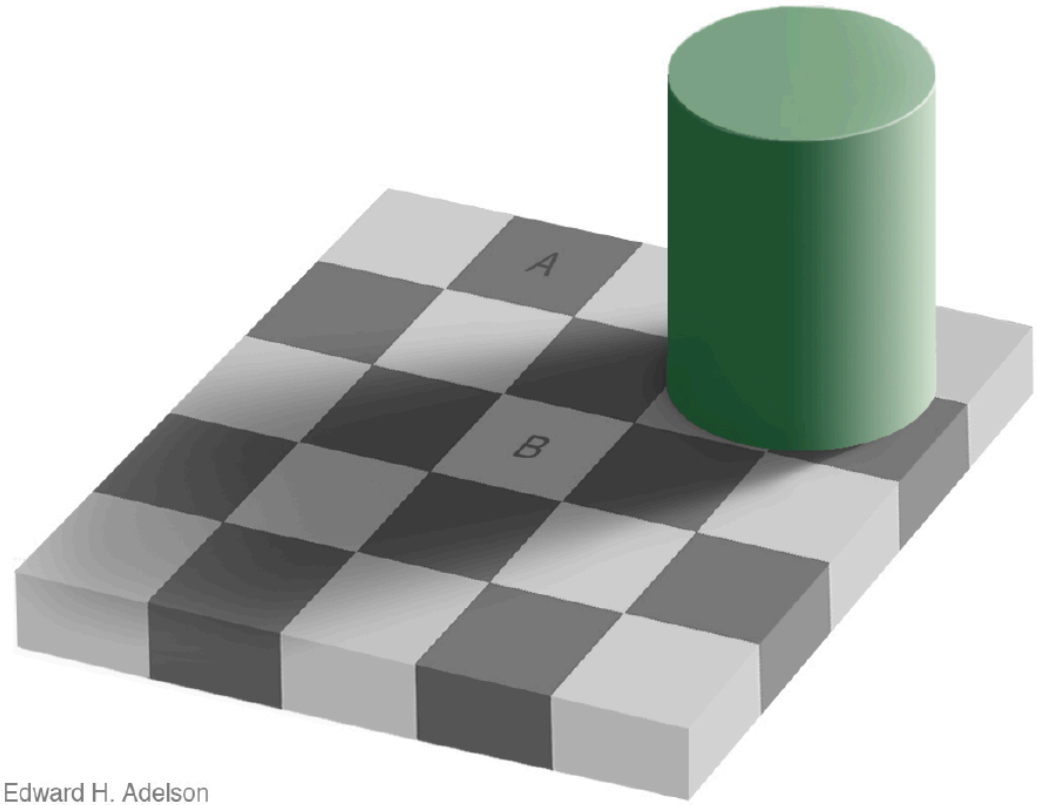
Measuring light vs. measuring scene properties



by Roger Shepard ("Turning the Tables")

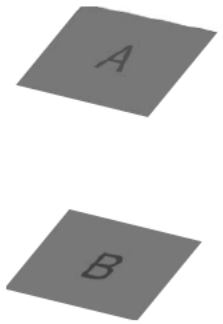
Depth processing is automatic, and we can not shut it down…

Measuring light vs. measuring scene properties

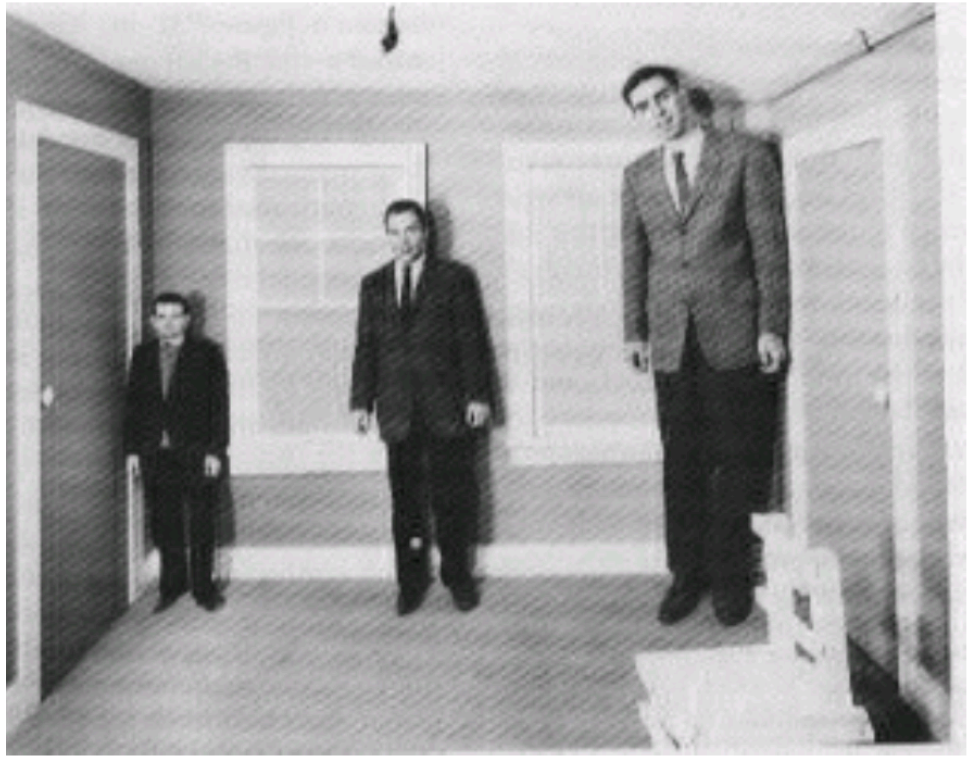


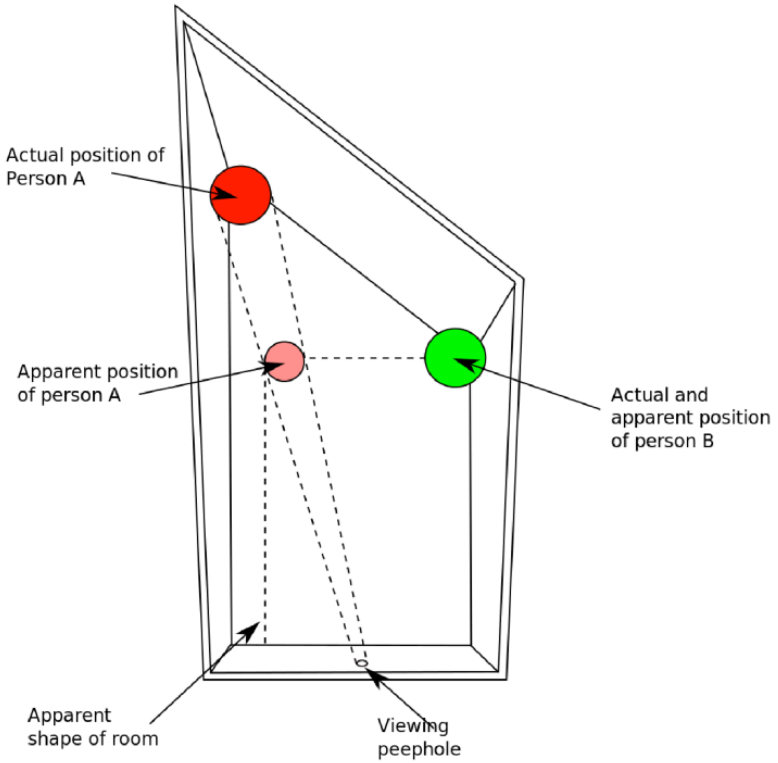
Edward H. Adelson

Measuring light vs. measuring scene properties



Assumptions can be wrong





Ames room (1934)

#TheDress

The same image, different percept?

What is going on?

A simple visual system

A simple world

A simple image formation

A simple goal

A simple visual system

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

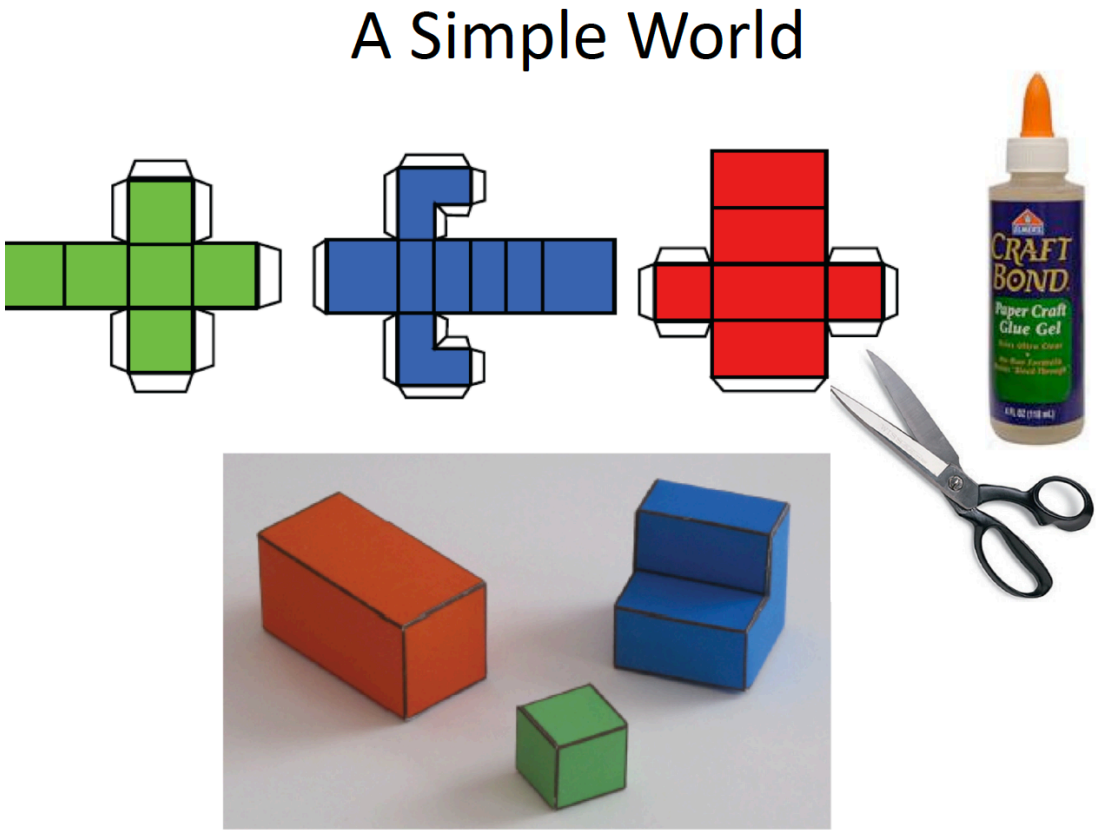
Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

A simple visual system

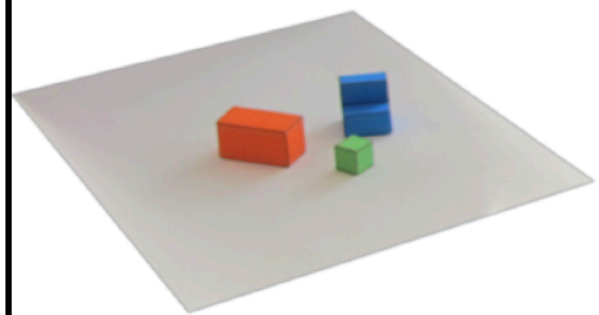


We assume we don’t know the exactly 3D geometry of the objects in advance

L.G. Roberts 1963

A simple image formation model

Simple world rules:
- Surfaces can be horizontal or vertical.
- Objects will be resting on a white horizontal ground plane



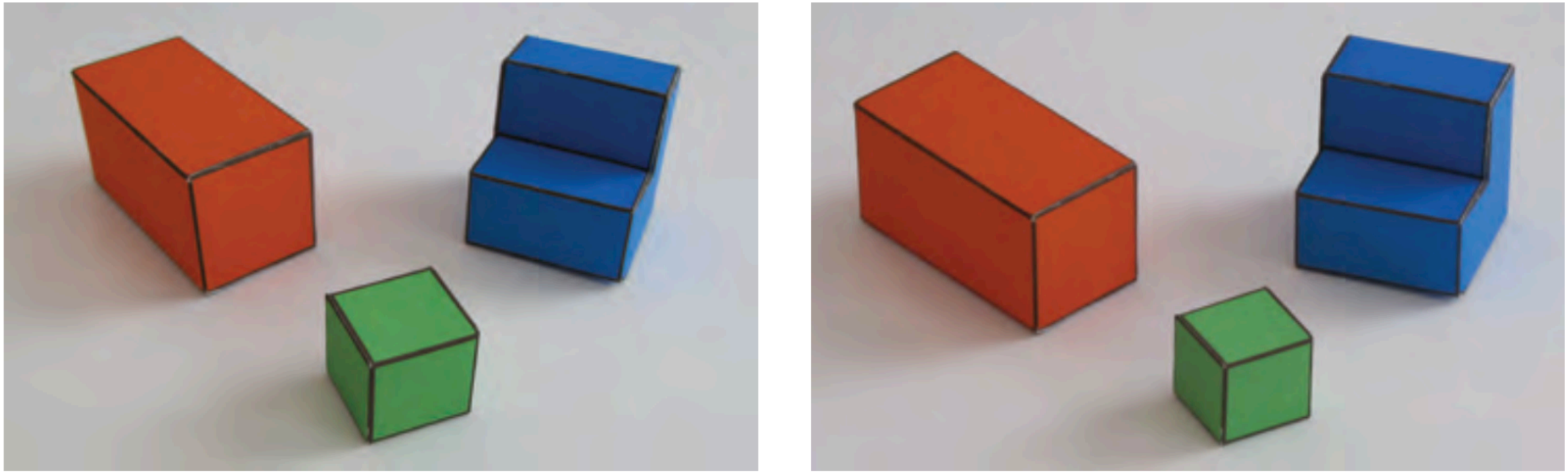
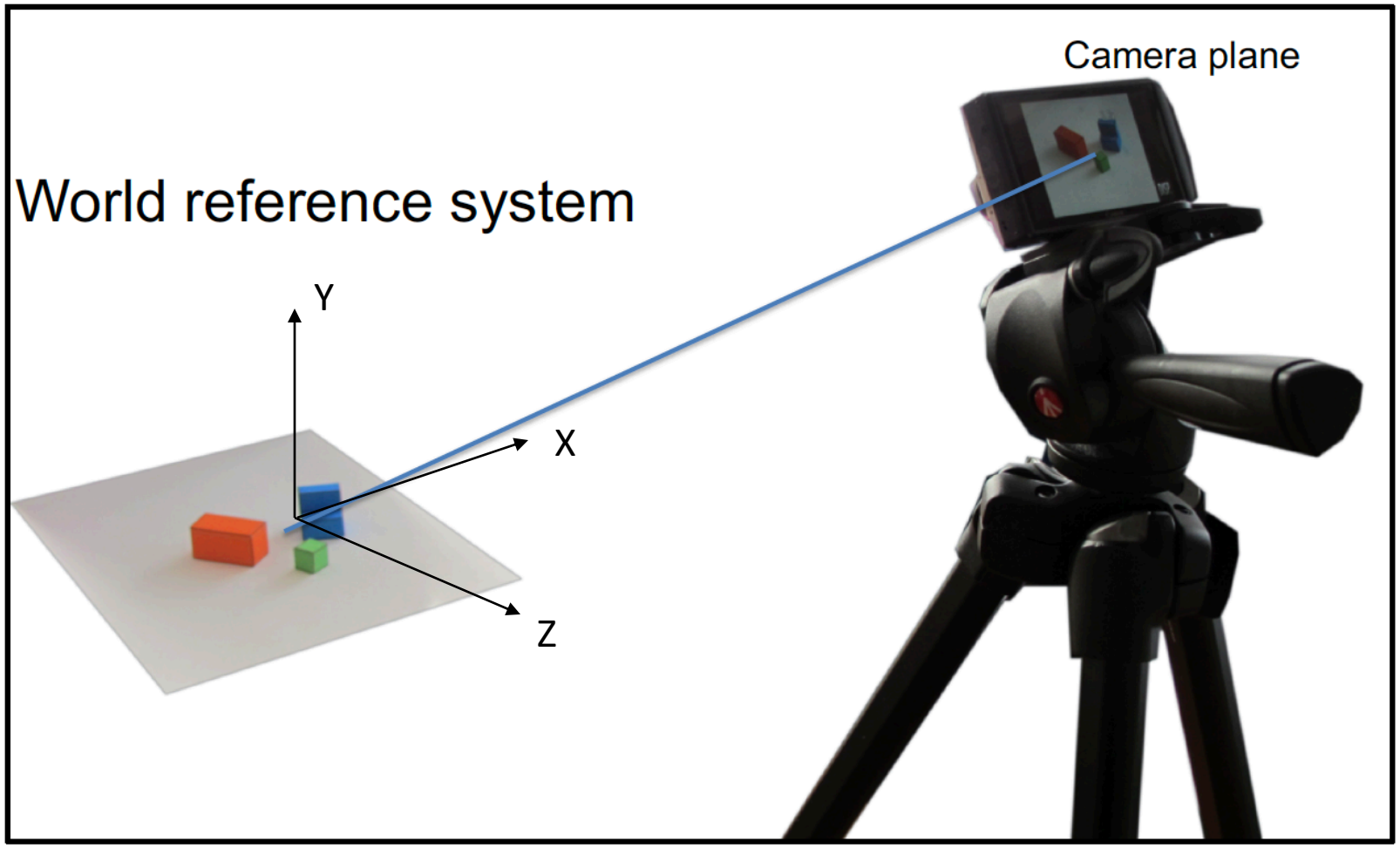


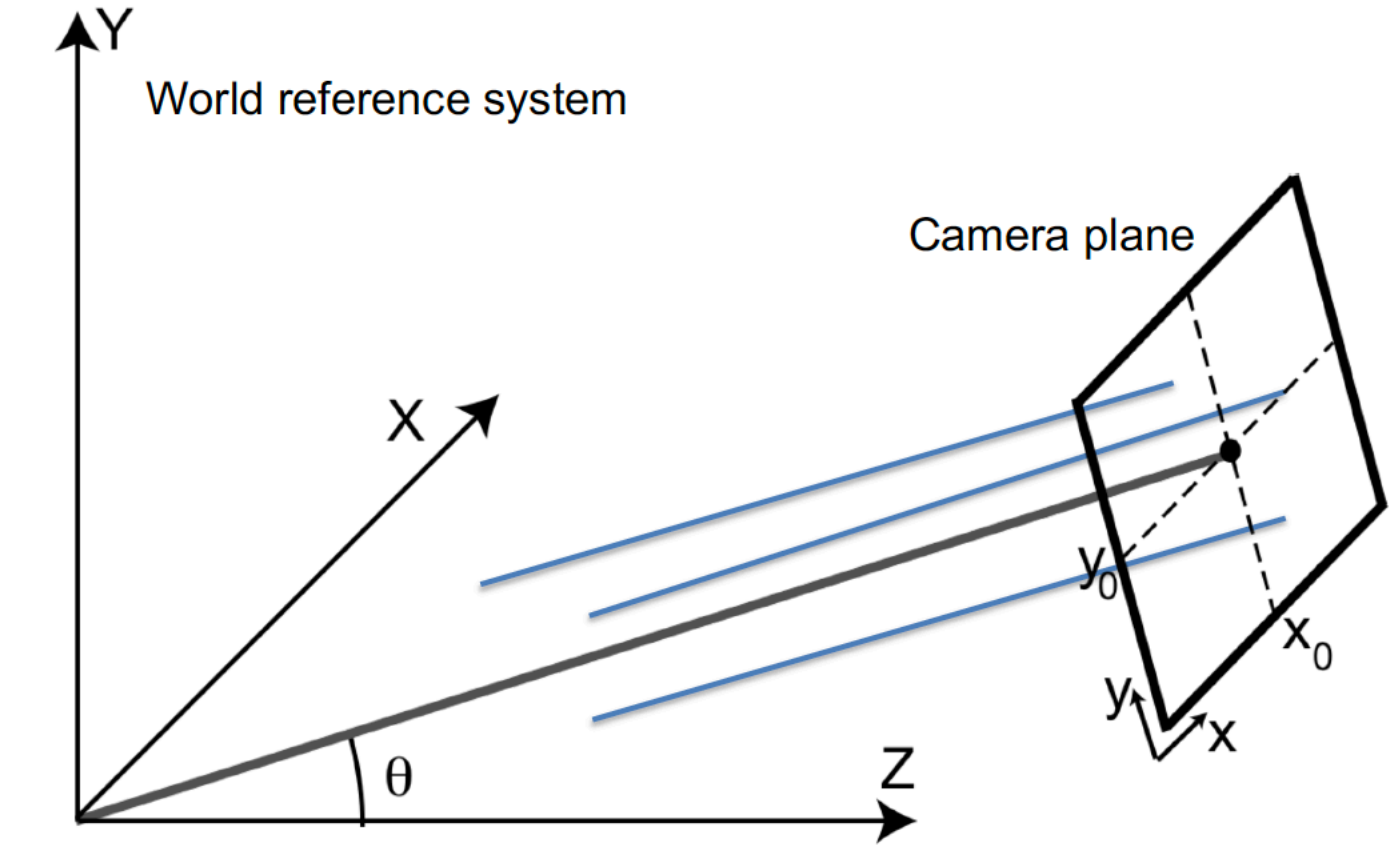
Figure 1.4

a) Close up picture without zoom. Note that near edges are larger than far edges, and parallel lines in 3D are not parallel in the image, b) Picture taken from far away but using zoom. This creates an image that can be approximately described by parallel projection.

A simple image formation model



Relating world coordinates (X,Y,Z) to image coordinates (x,y)



(right-handed reference system)

A simple image formation model

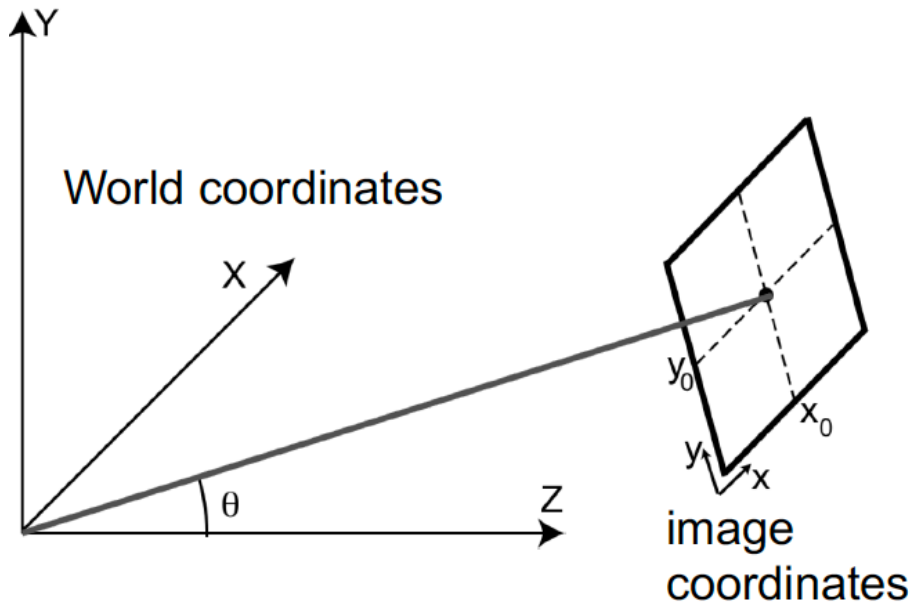
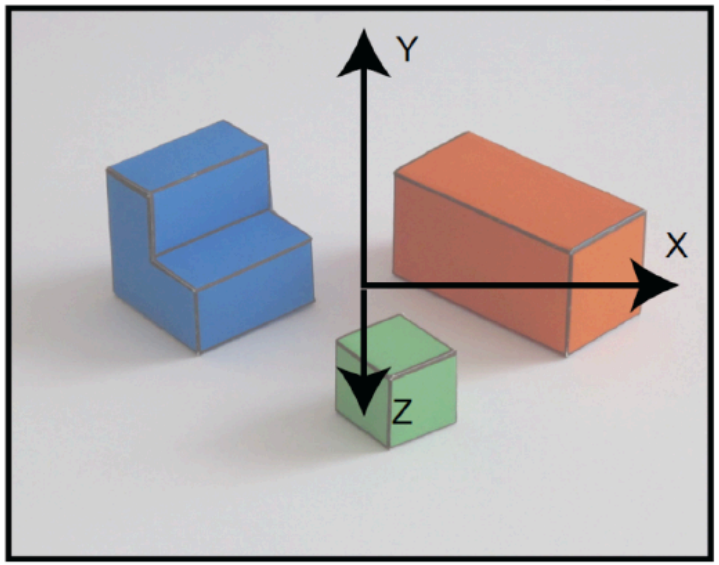


Image and projection of the world coordinate axes into the image plane



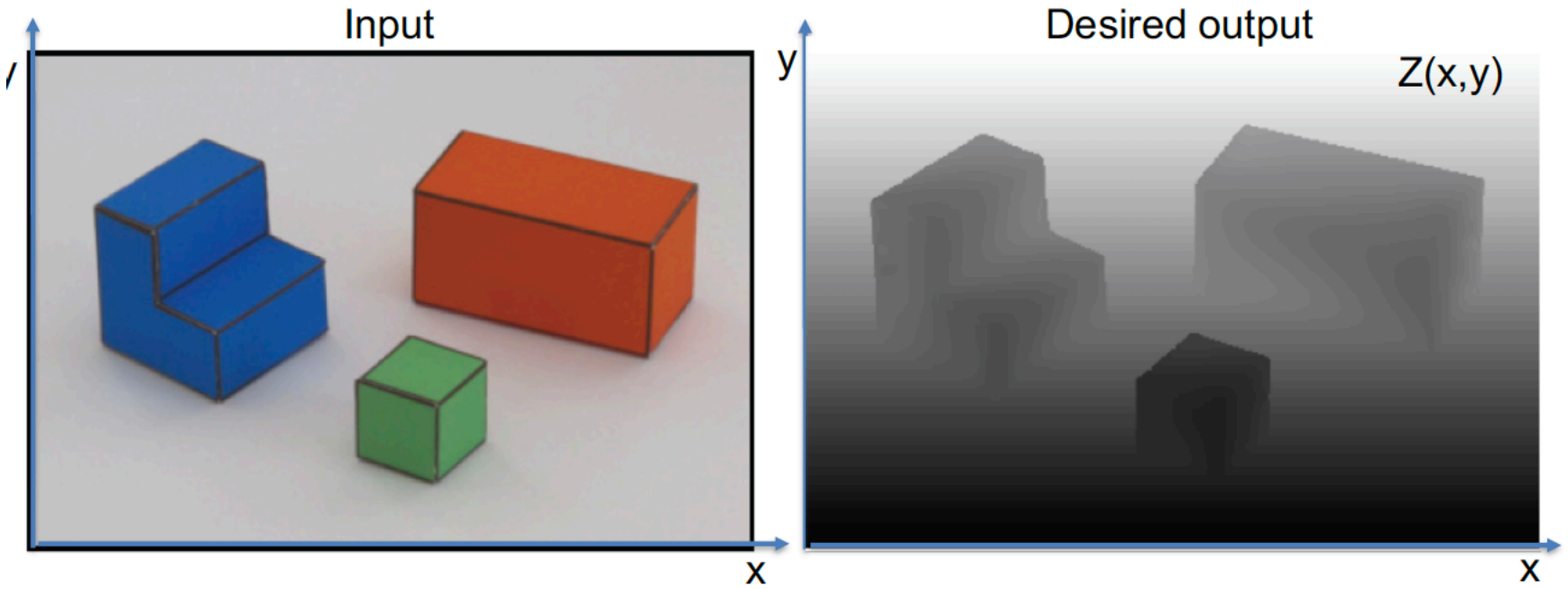
World coordinates

$x = X + x_0$

$y = \cos(\theta)\, Y - \sin(\theta)\, Z + y_0$

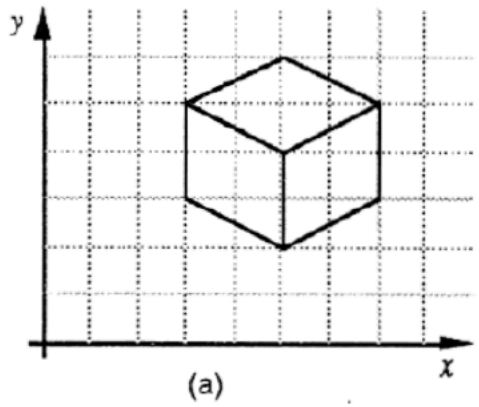
image coordinates

A simple goal:
To recover the 3D structure of the world



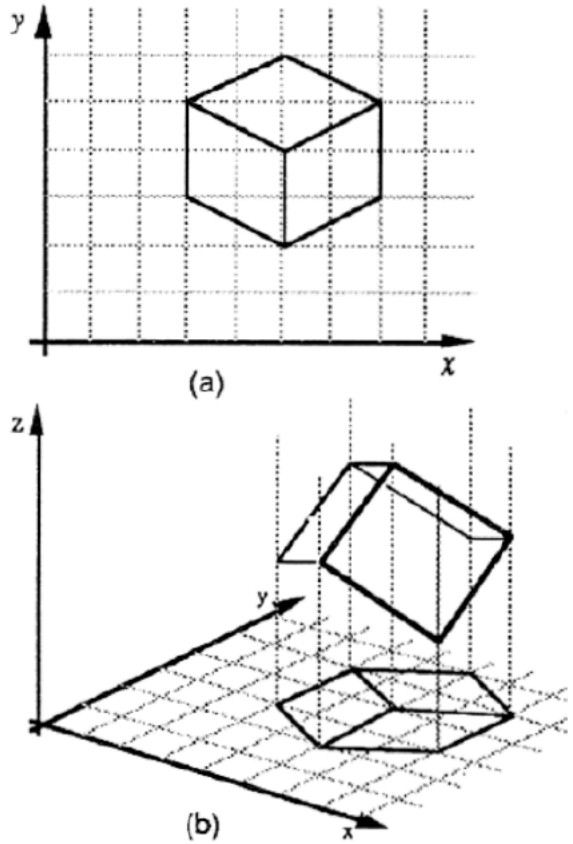
We want to recover world coordinates for each image pixel:
X(x,y), Y(x,y), Z(x,y)

Why is this hard?



Sinha & Adelson 93

Why is this hard?



Sinha & Adelson 93

Why is this hard?

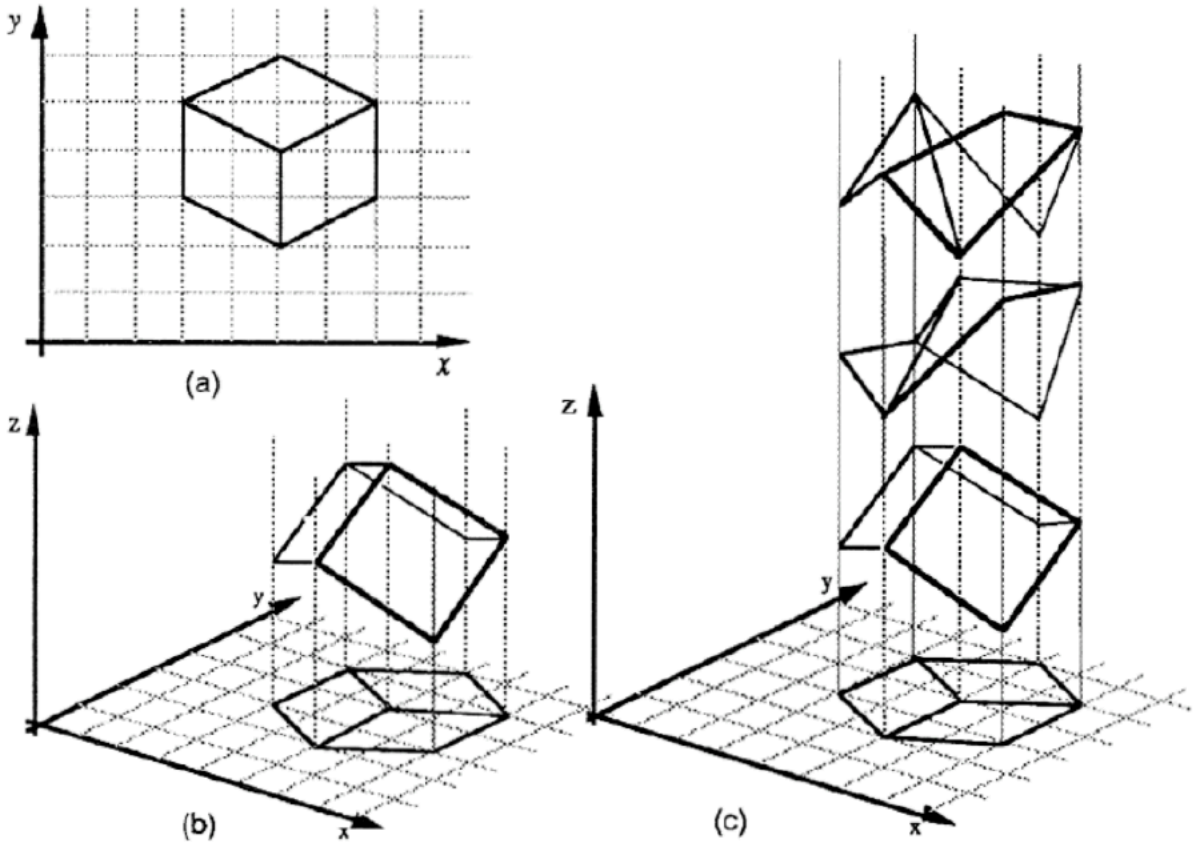
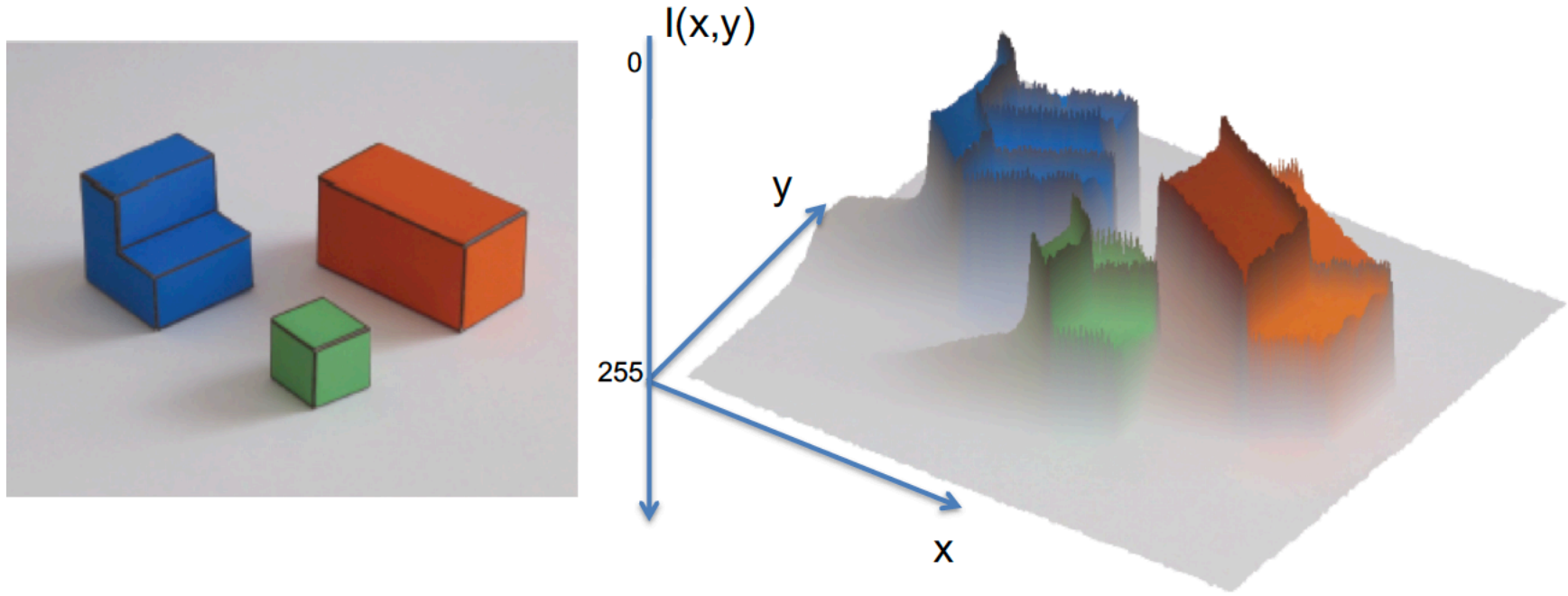


Figure 1. (a) A line drawing provides information only about the x, y coordinates of points lying along the object contours. (b) The human visual system is usually able to reconstruct an object in three dimensions given only a single 2D projection (c) Any planar line-drawing is geometrically consistent with infinitely many 3D structures.

Sinha & Adelson 93

A simple visual system
The input image



Question:

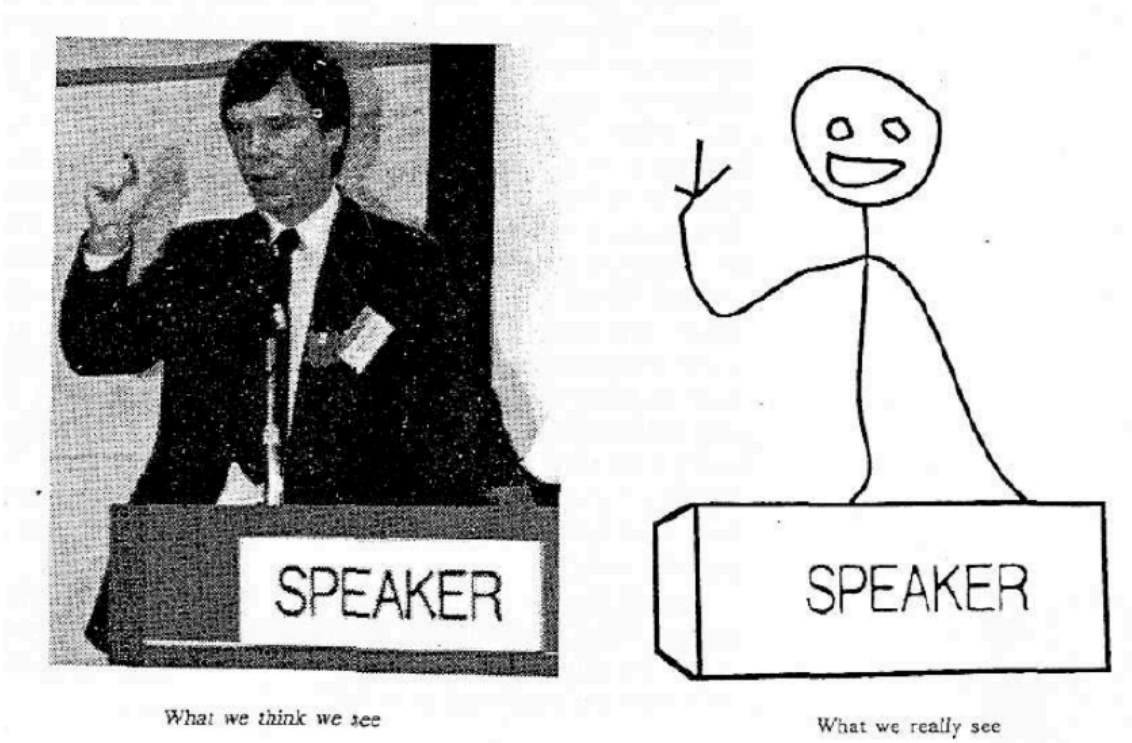
The observed image is I(x,y), how do we represent the pixel intensities if we care about the 3D structure?

• Proposition 1. The primary task of early vision is to deliver a small set of useful measurements about each observable location.

• Proposition 2. The elemental operations of early vision involve the measurement of local change along various directions.

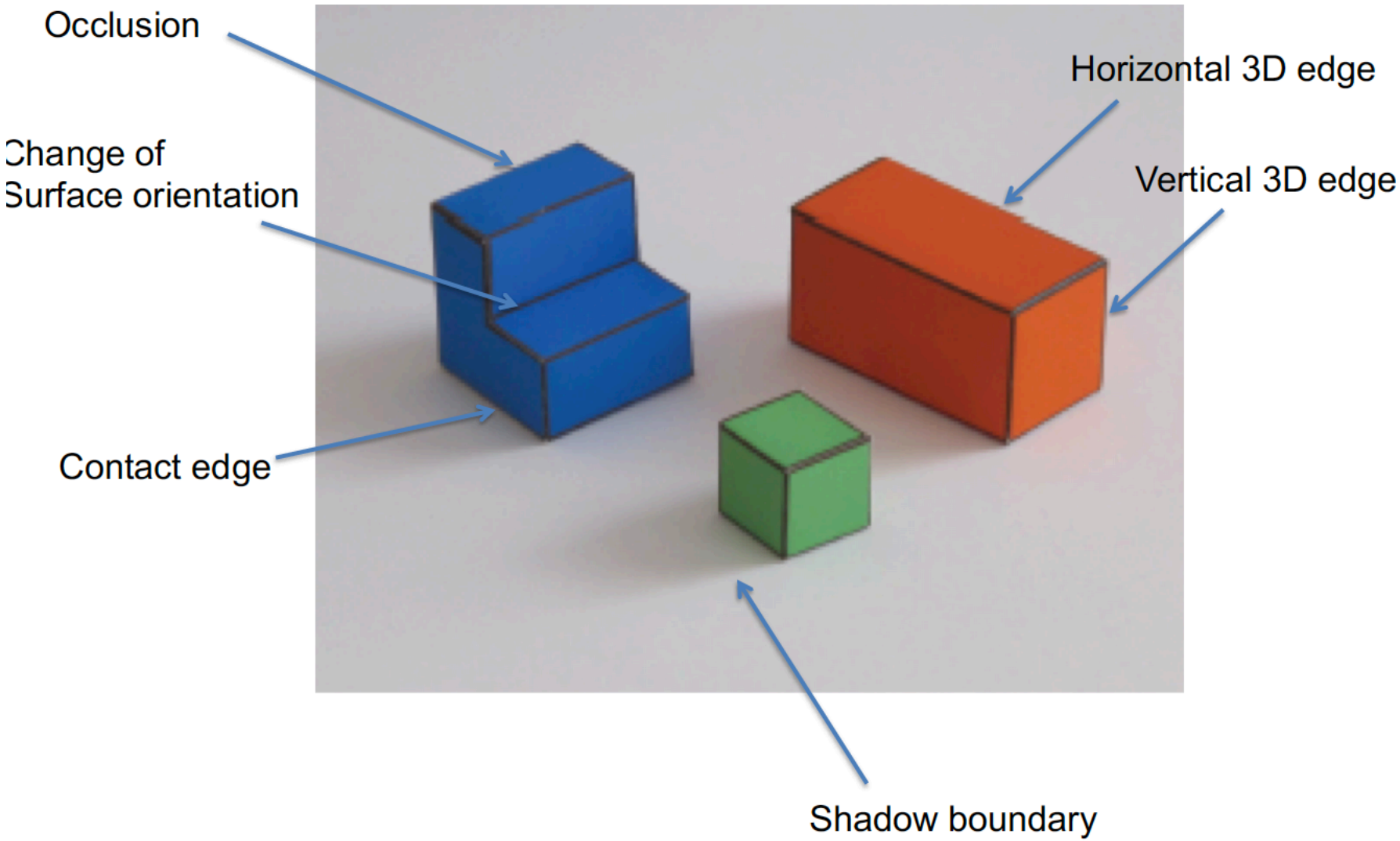
Adelson, Bergen. 91

• Goal: to transform the image into other representations (rather than pixel values) that makes scene information more explicit

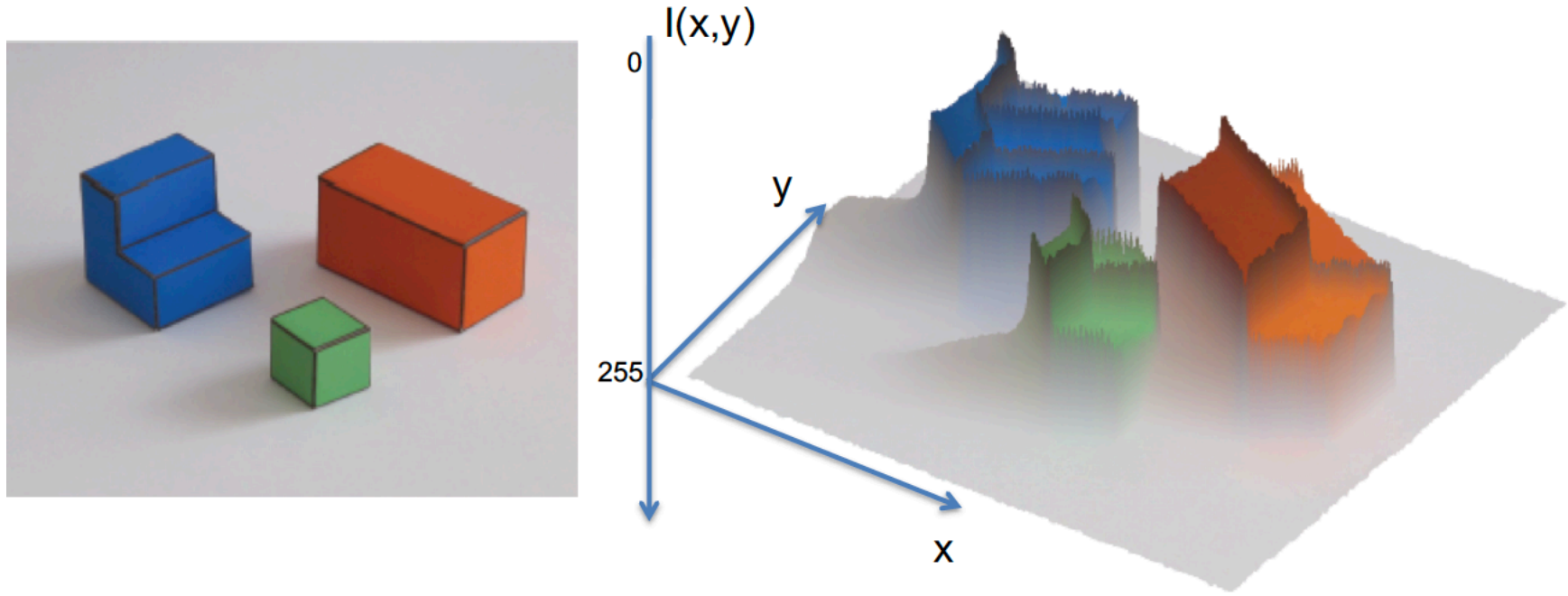


Cavanagh, Perception 95

Edges


Occlusion
Change of
Surface orientation
Contact edge
Horizontal 3D edge
Vertical 3D edge
Shadow boundary

A simple visual system
The input image



Finding edges in the image

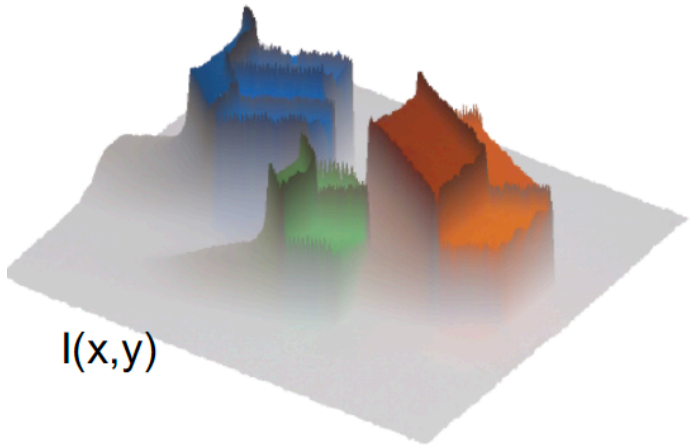


Image gradient:

$$\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y} \right)$$

Approximation image derivative:

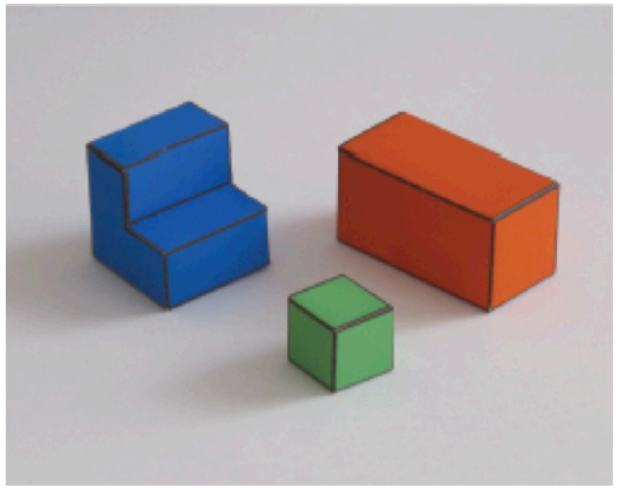
$$\frac{\partial \mathbf{I}}{\partial x} \simeq \mathbf{I}(x, y) - \mathbf{I}(x-1, y)$$

Edge strength $E(x, y) = |\nabla \mathbf{I}(x, y)|$

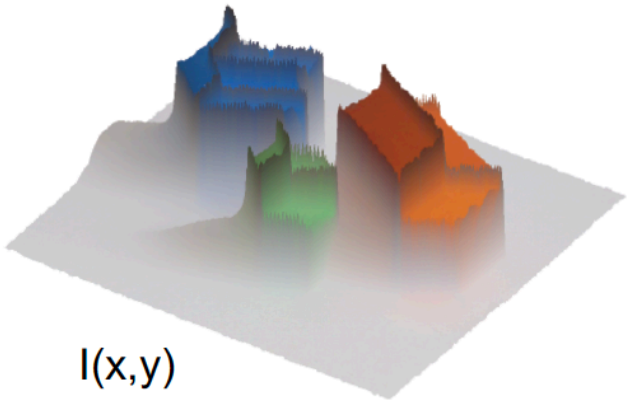
Edge orientation: $\theta(x, y) = \angle \nabla \mathbf{I} = \arctan \frac{\partial \mathbf{I}/\partial y}{\partial \mathbf{I}/\partial x}$

Edge normal: $\mathbf{n} = \frac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|}$

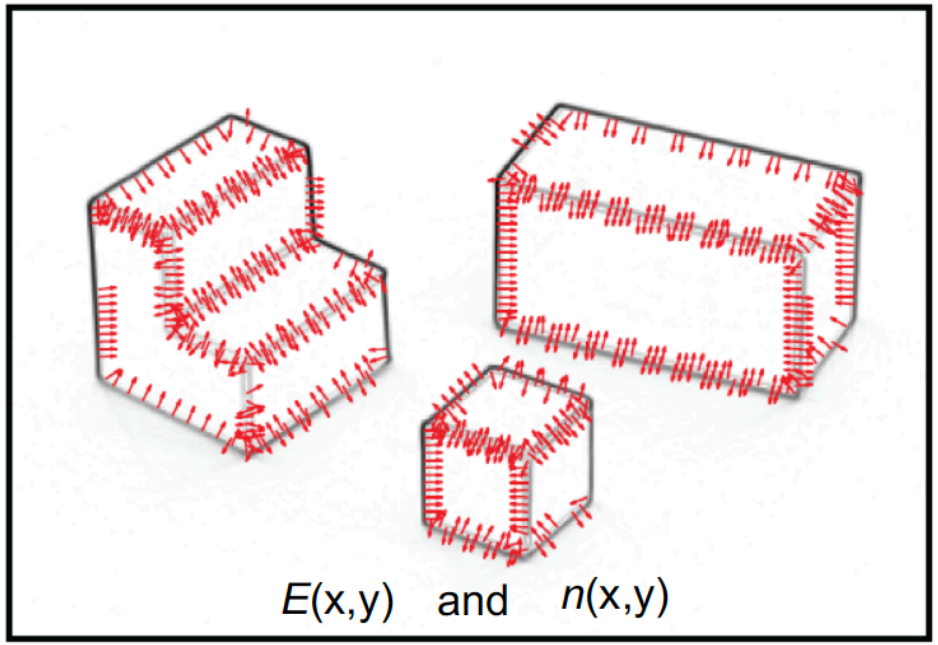
Finding edges in the image



$$\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y} \right) \begin{cases} \mathbf{n} = \dfrac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|} \\ E(x,y) = |\nabla \mathbf{I}(x,y)| \end{cases}$$

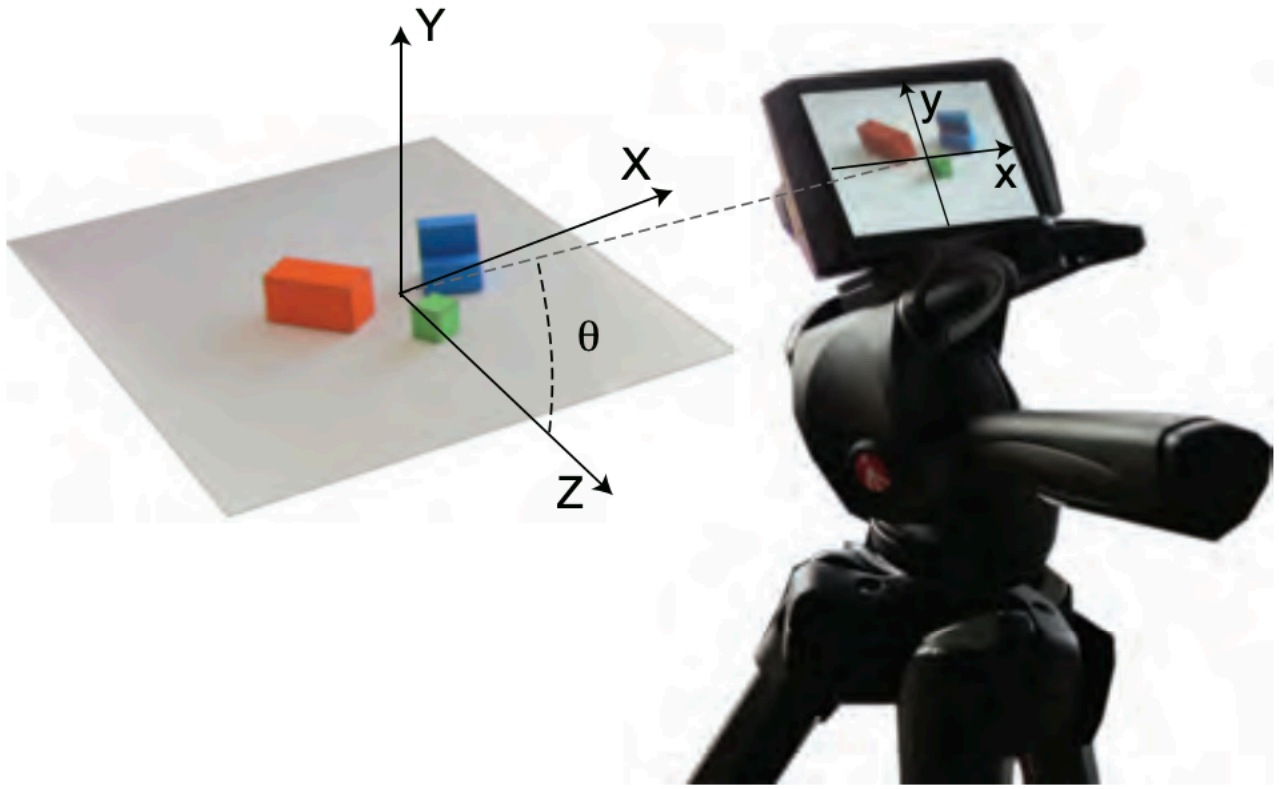


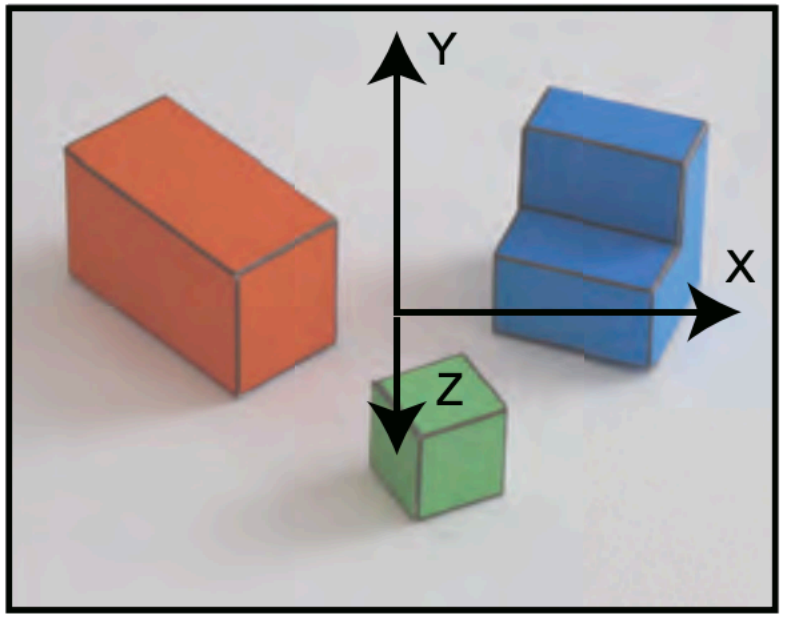
I(x,y)



E(x,y) and *n*(x,y)

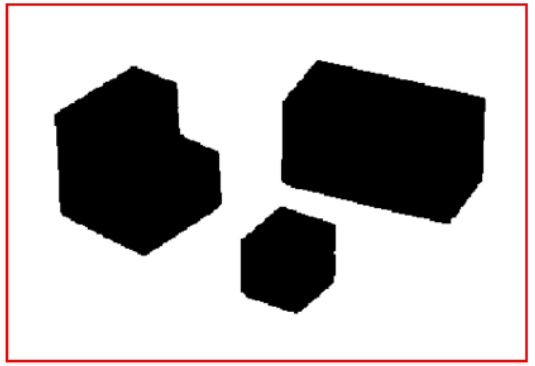
For each pixel with image coordinates (x, y), the corresponding world coordinate is X(x, y) = x

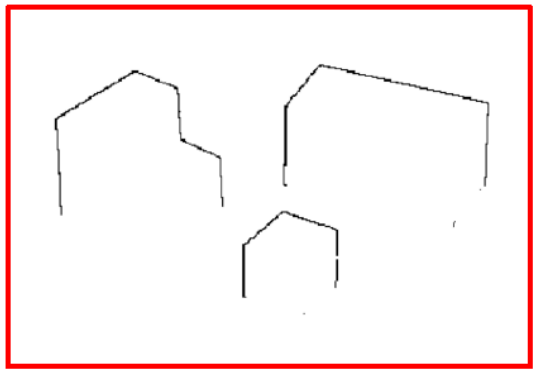


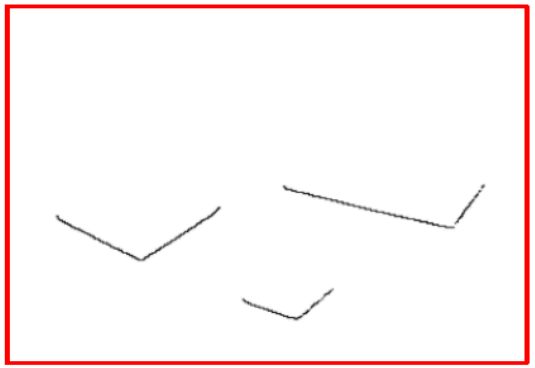


Edge classification

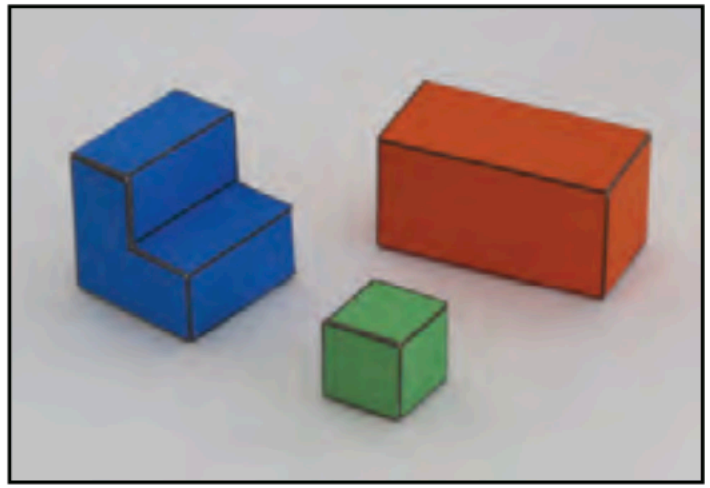
- Figure/ground segmentation
  - Using the fact that objects have color
- Occlusion edges
  - Occlusion edges are owned by the foreground
- Contact edges



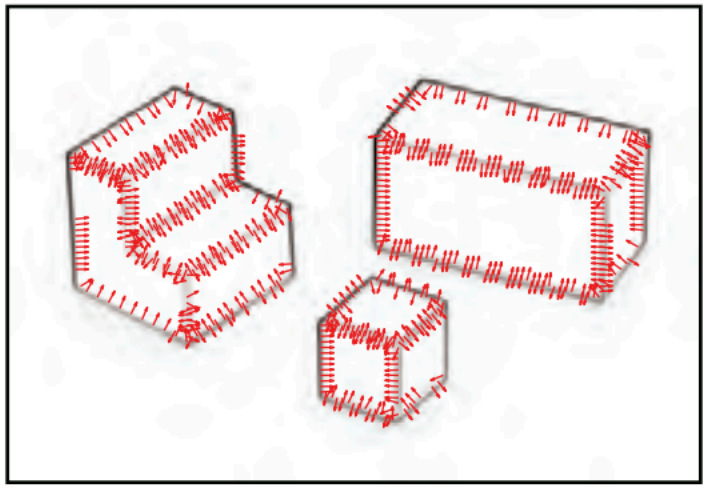




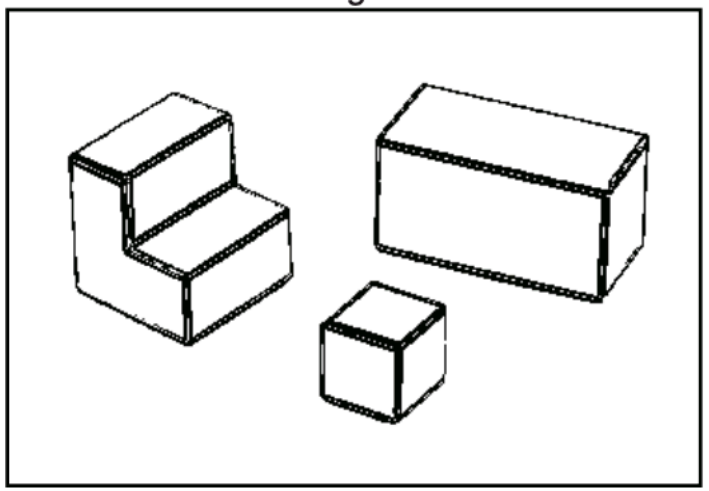
Input image



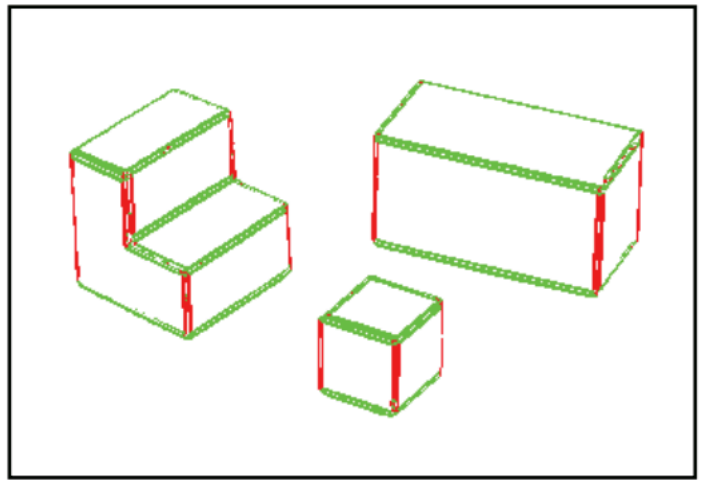
Gradient (magnitude and orientation)`



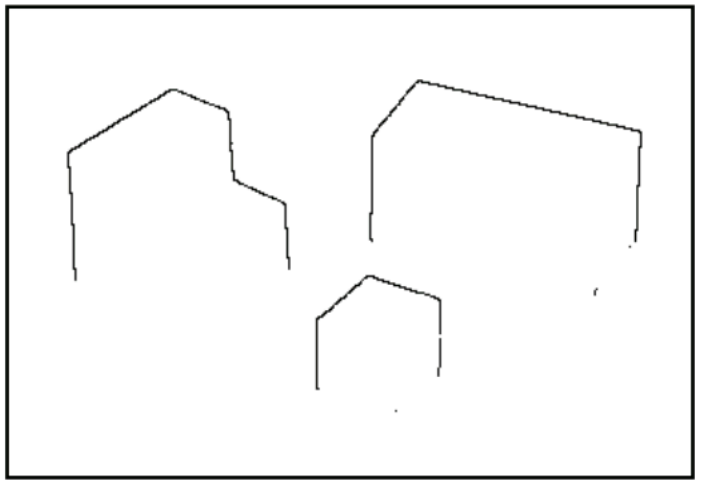
Edges



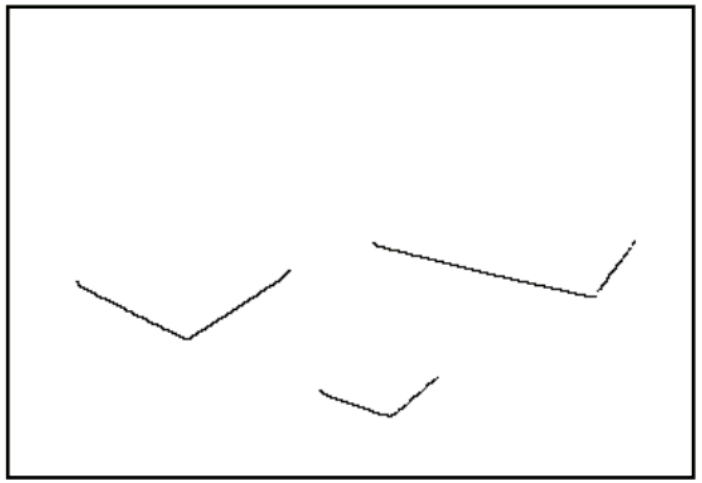
3D orientation



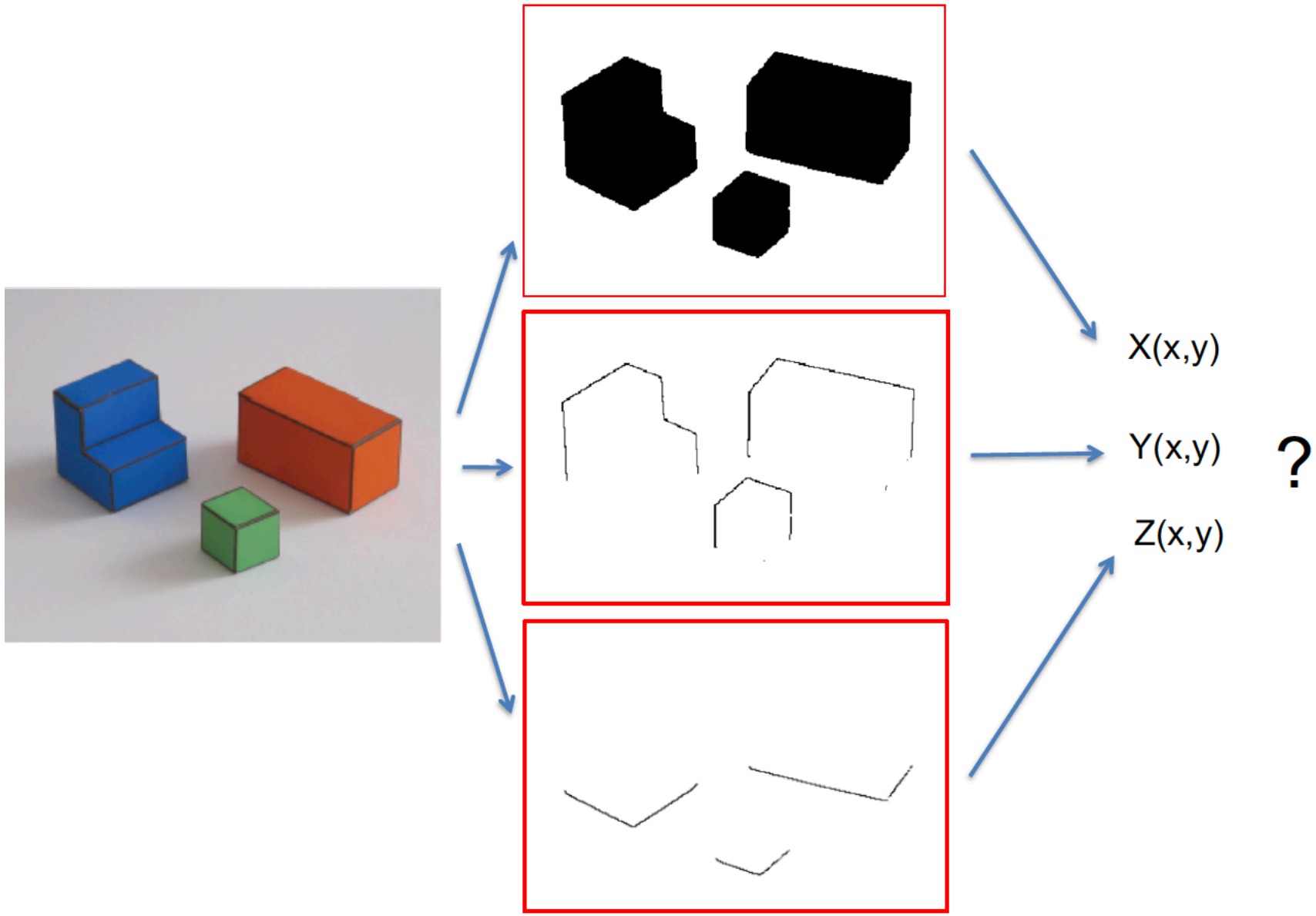
Depth discontinuities



Contact edges

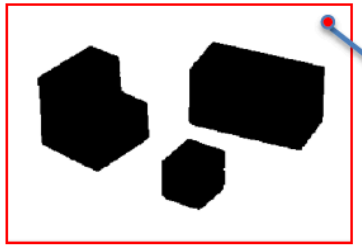


From edges to surface constraints



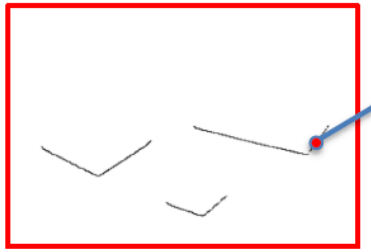
From edges to surface constraints

- Ground



Y(x,y) = 0 if (x,y) belongs to a ground pixel

- Contact edge

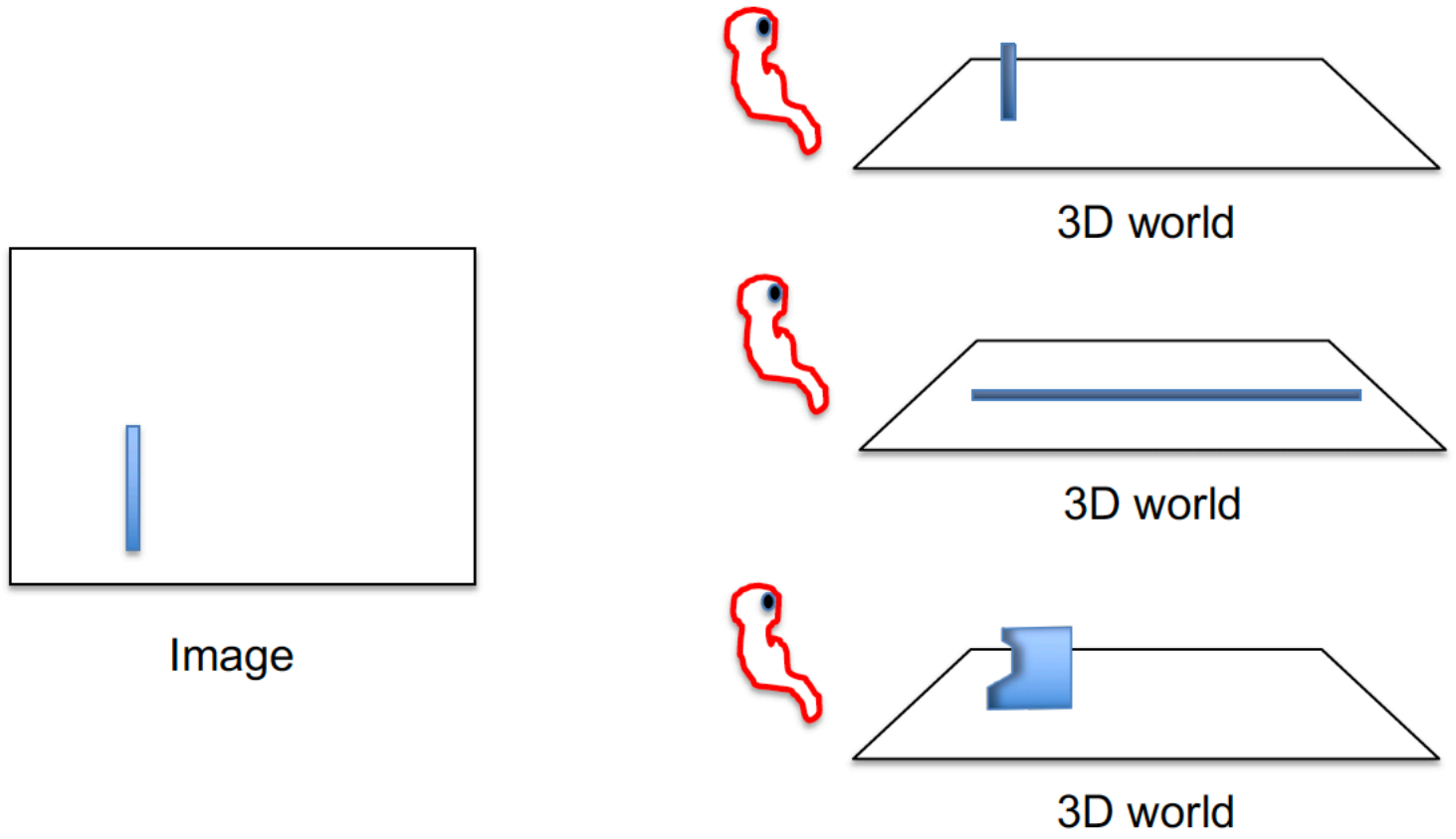


Y(x,y) = 0 if (x,y) belongs to foreground and is a contact edge

- What happens inside the objects?

… now things get a bit more complicated.

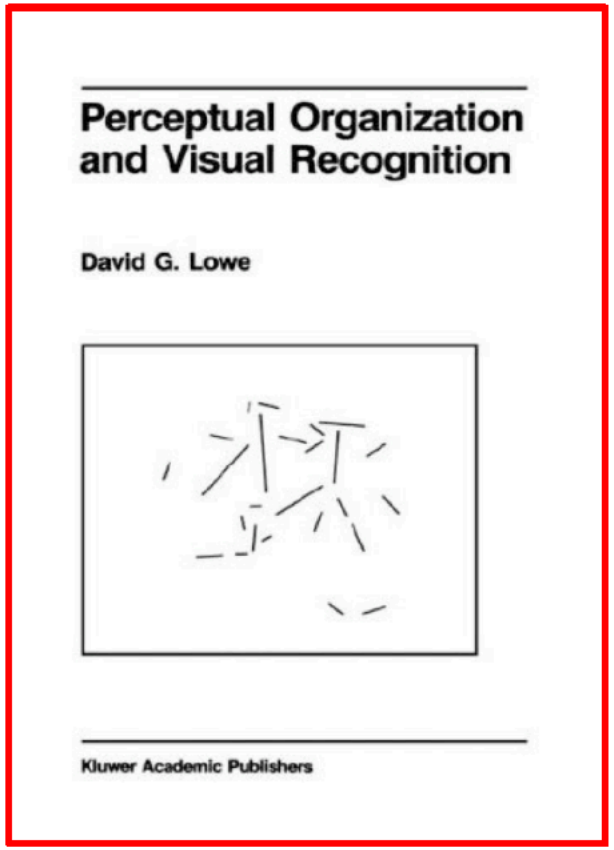
Generic view assumption



Generic view assumption: the observer should not assume that he has a special position in the world… The most generic interpretation is to see a vertical line as a vertical line in 3D.

Freeman, 93

Non-accidental properties



D. Lowe, 1985

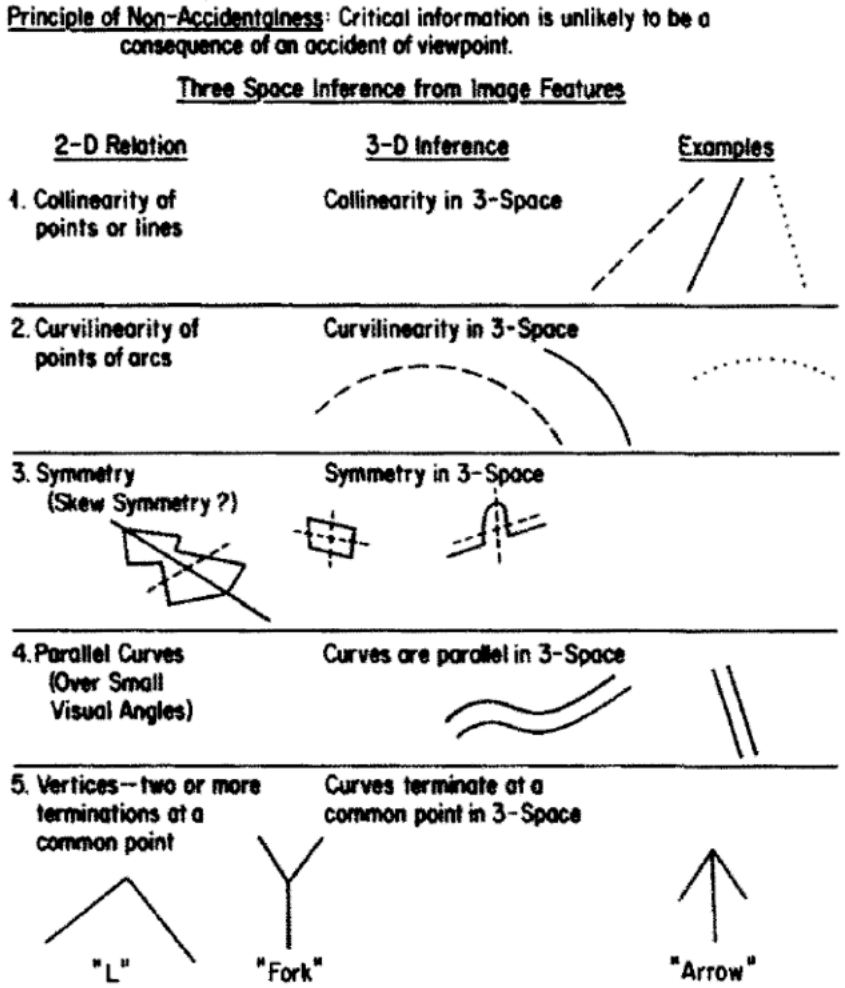


Figure 4. Five nonaccidental relations. (From Figure 5.2, *Perceptual organization and visual recognition* [p. 77] by David Lowe. Unpublished doctorial dissertation, Stanford University. Adapted by permission.)

Biederman_RBC_1987

Invariant properties:

Collinearity, Cotermination

Intersection

Parallelism

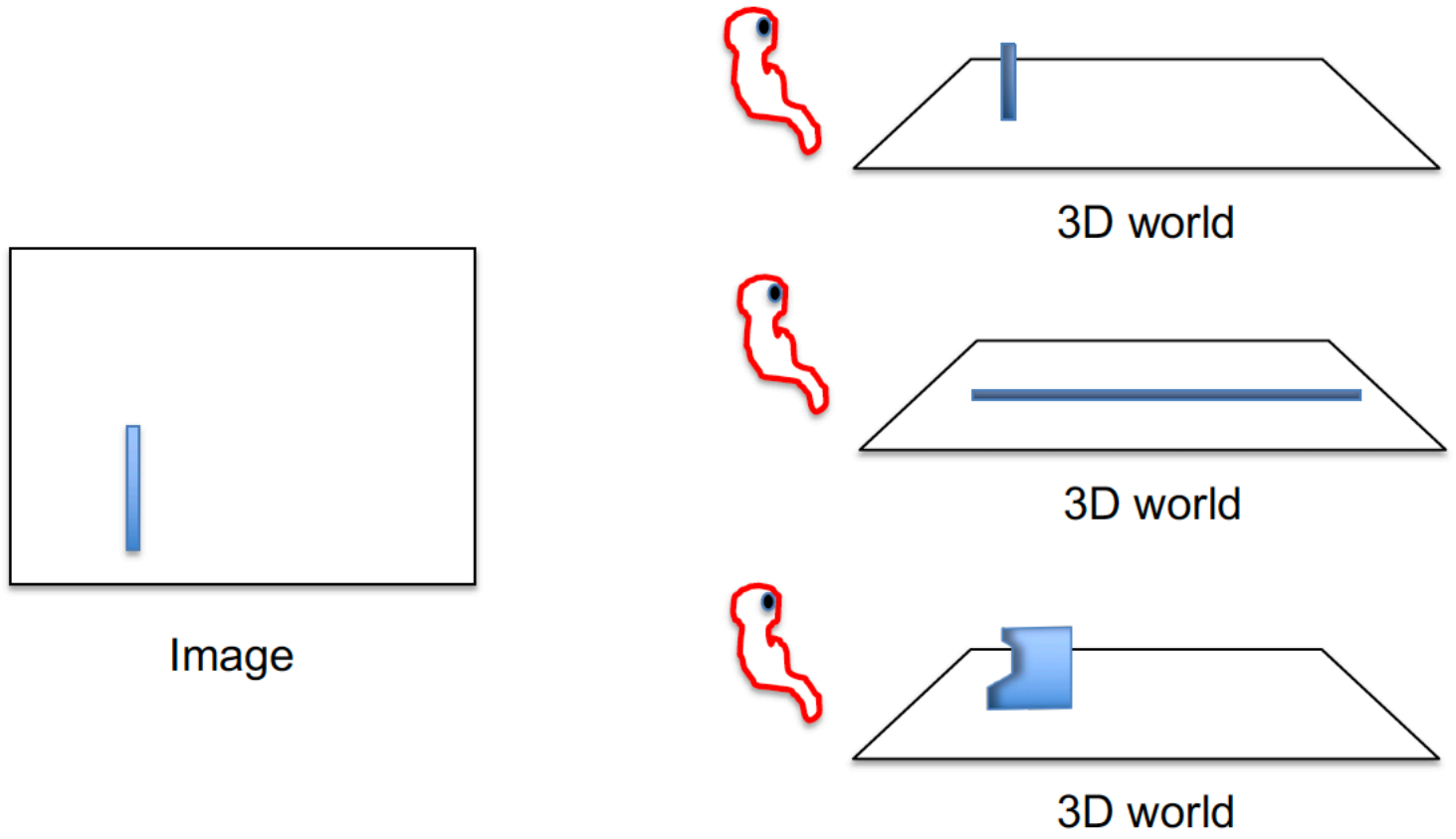
Symmetry

Smoothness

Using non-accidental scene properties to infer 3D structure from image

- For example: If two lines conterminate in the image, then, one can conclude that it is vey likely that they also touch each other in 3D. if the 3D lines do not touch each other, then it will require a very specific alignment between the observer and lines for them to appear conterminate in the image. Therefore, we can conclude the lines also touch in 3D.

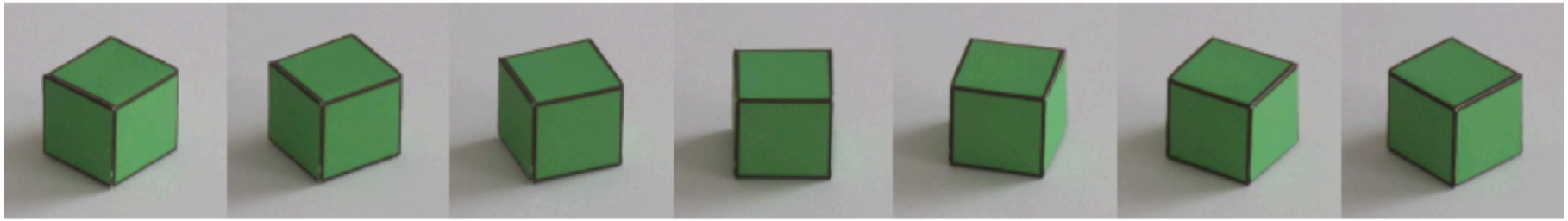
Generic view assumption

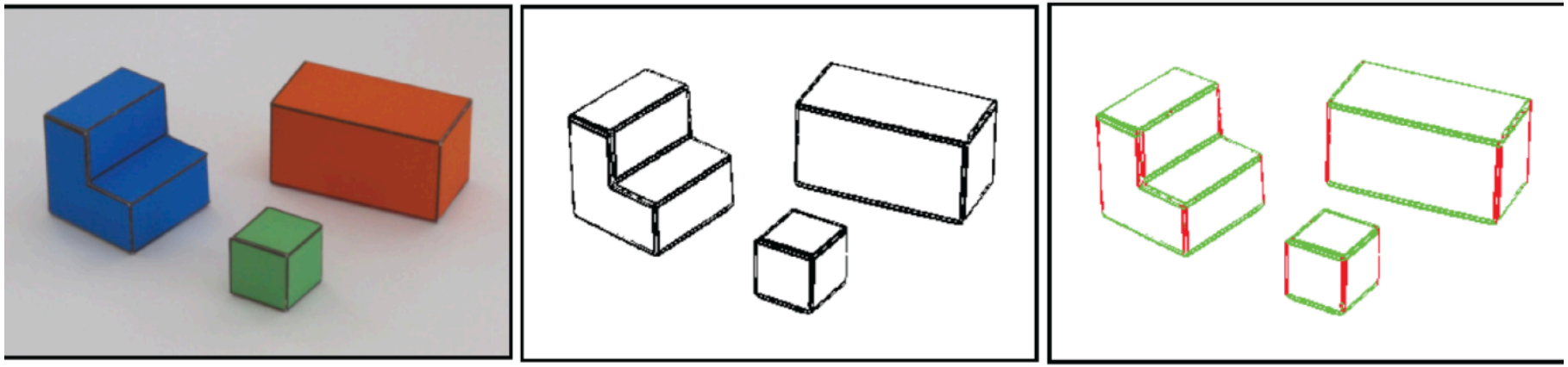


Generic view assumption: the observer should not assume that he has a special position in the world… The most generic interpretation is to see a vertical line as a vertical line in 3D.

Freeman, 93

Non-accidental properties in the simple world



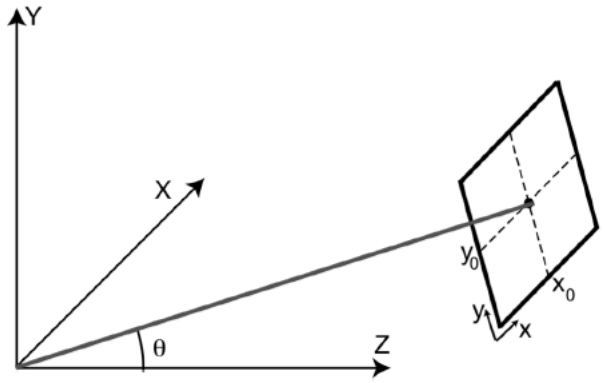


Using angle to classify Horizontal vs. vertical edges

From edges to surface constraints

How can we relate the information in the pixels with 3D surfaces in the world?

- Vertical edges



World coordinates

$x = X + x_0$

$y = \cos(\theta)\, Y - \sin(\theta)\, Z + y_0$

image coordinates

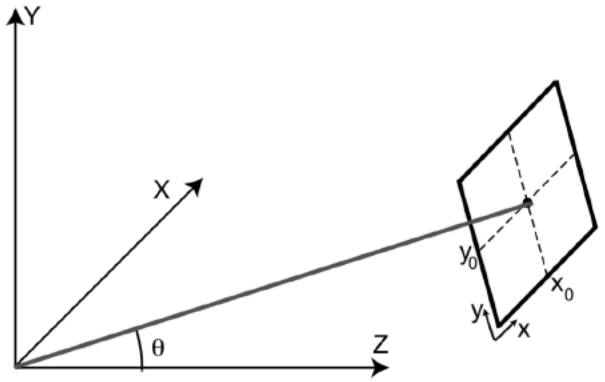
Given the image, what can we say about X, Y and Z in the pixels that belong to a vertical edge?

Z = constant along the edge

$$\partial Y / \partial y = 1 / \cos(\theta)$$

From edges to surface constraints

- Horizontal edges



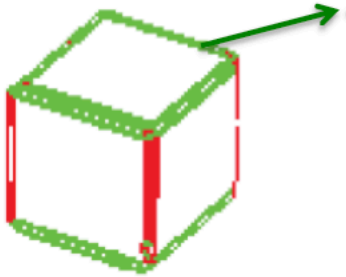
World coordinates

$x = X + x_0$

$y = \cos(\theta)\, Y - \sin(\theta)\, Z + y_0$

image coordinates

Given the image, what can we say about X, Y and Z in the pixels that belong to an horizontal 3D edge?



Y = constant along the edge

$$\partial Y/\partial \mathbf{t} = 0$$

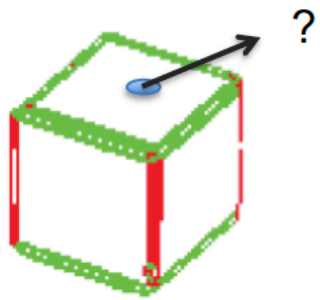
Where **t** is the vector parallel to the edge

$$\mathbf{t} = (-n_y, n_x)$$

$$\partial Y/\partial \mathbf{t} = -n_y \partial Y/\partial x + n_x \partial Y/\partial y$$

From edges to surface constraints

- What happens where there are no edges?



This approximation to the second derivative can be obtained by applying twice the first order derivative approximated by [-1 1]. The result is: $\begin{bmatrix}-1 & 2 & -1\end{bmatrix}$ which corresponds to $\partial^2 Y/\partial x^2 \simeq 2Y(x,y) - Y(x+1,y) - Y(x-1,y)$, and similarly for $\partial^2 Y/\partial y^2$

Assumption of planar faces:

$$\begin{aligned} \partial^2 Y/\partial x^2 &= 0 \\ \partial^2 Y/\partial y^2 &= 0 \\ \partial^2 Y/\partial y \partial x &= 0 \end{aligned}$$

Information has to be propagated from the edges

A simple inference scheme

All the constraints are linear

$Y(x,y) = 0$ if (x,y) belongs to a ground pixel

$$\partial Y/\partial y = 1/\cos(\theta)$$ if (x,y) belongs to a vertical edge

$$\partial Y/\partial \mathbf{t} = 0$$ if (x,y) belongs to an horizontal edge

$$\begin{aligned} \partial^2 Y/\partial x^2 &= 0 \\ \partial^2 Y/\partial y^2 &= 0 \\ \partial^2 Y/\partial y \partial x &= 0 \end{aligned}$$ if (x,y) is not on an edge

A similar set of constraints could be derived for Z

Discrete approximation

We can transform every differential constrain into a discrete linear constraint on Y(x,y)

Y(x,y)

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

$$\frac{dY}{dx} \approx Y(x,y) - Y(x\text{-}1,y)$$

-1	1

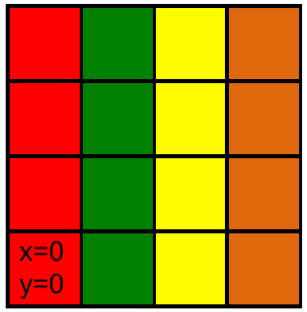
A slightly better approximation
(it is symmetric, and it averages horizontal derivatives over 3 vertical locations)

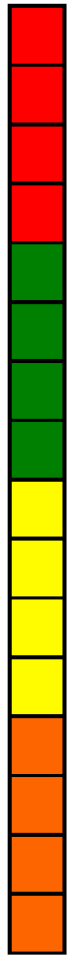
-1	0	1
-2	0	2
-1	0	1

Discrete approximation

Transform the “image” Y(x,y) into a column vector:

Y(x,y)

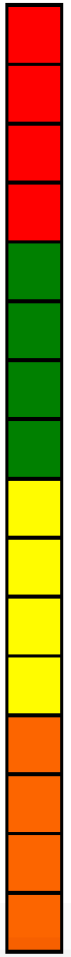




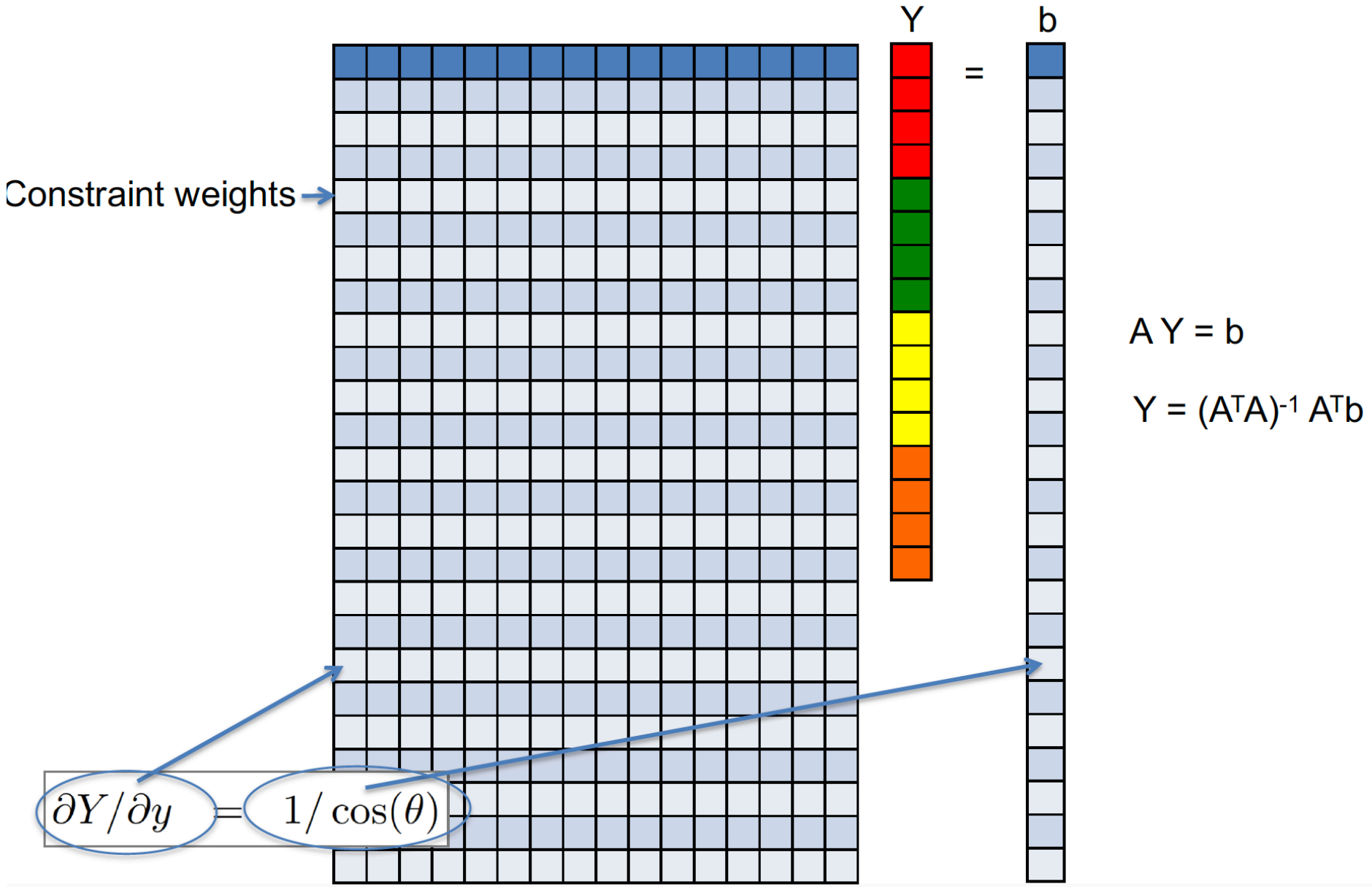
x=2, y=2

$$\frac{dY}{dx} \approx Y(x,y) - Y(x-1,y) = Y(2,2) - Y(1,2) =$$

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



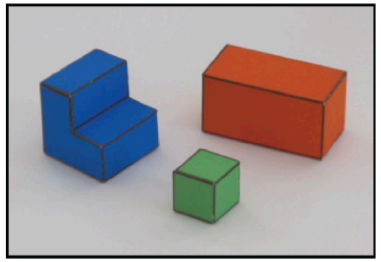
A simple inference scheme



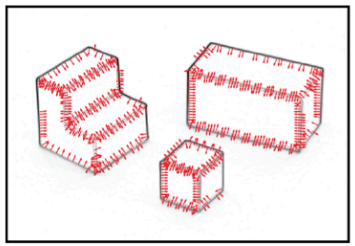
$A\,Y = b$

$Y = (A^T A)^{-1} A^T b$

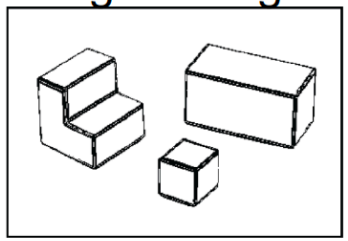
Results



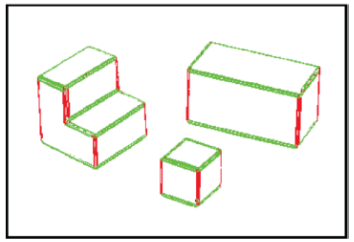
Edge normals



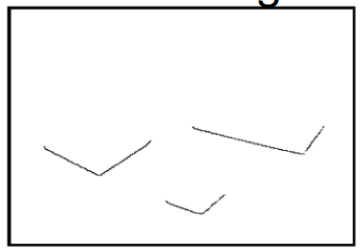
Edge strength



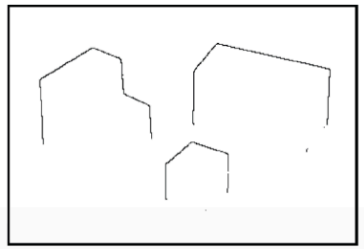
3D orientation

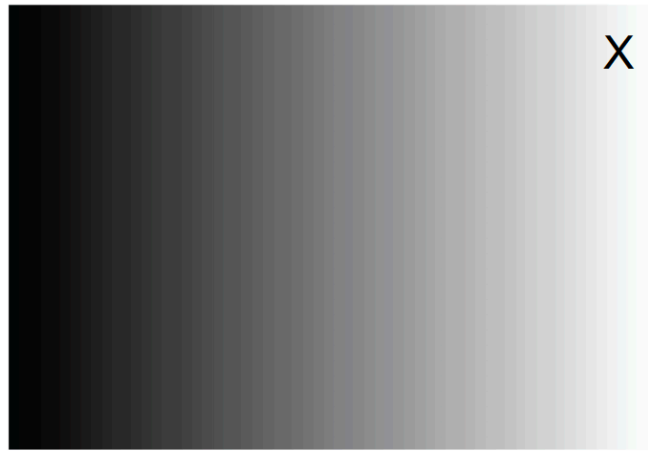


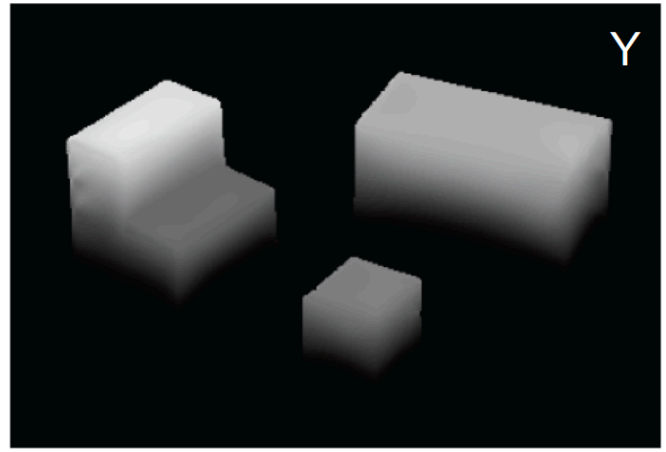
Contact edges



Depth discontinuities





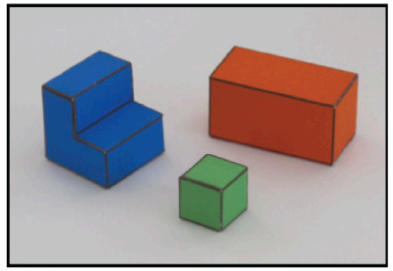


height

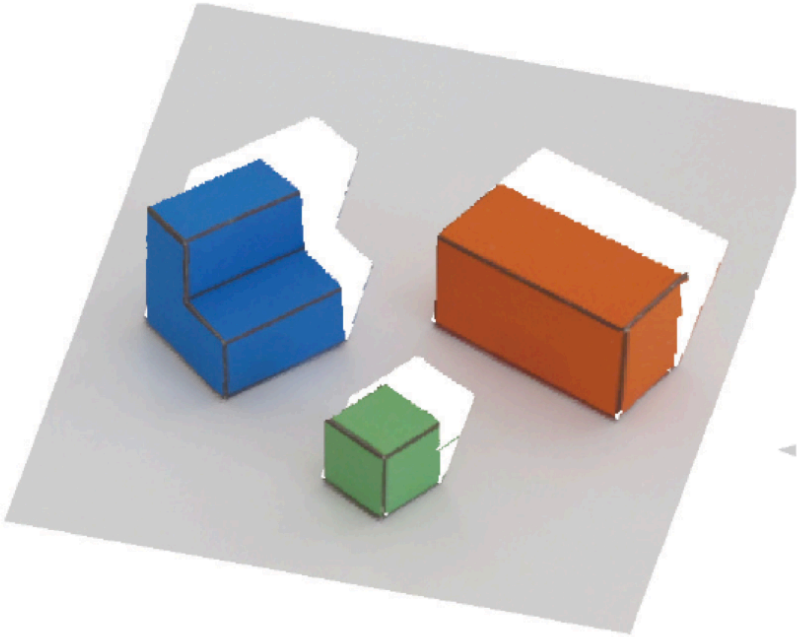
Depth

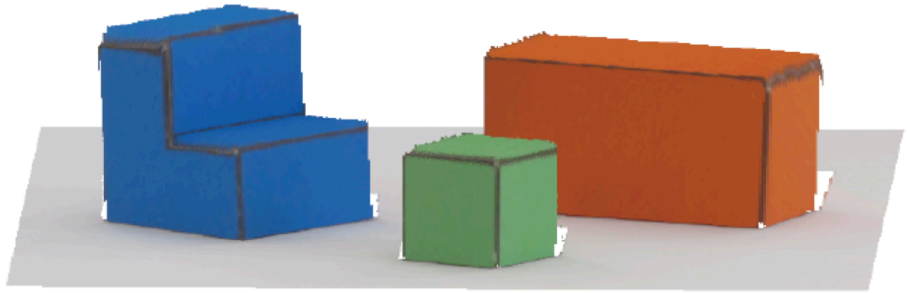
Changing view point

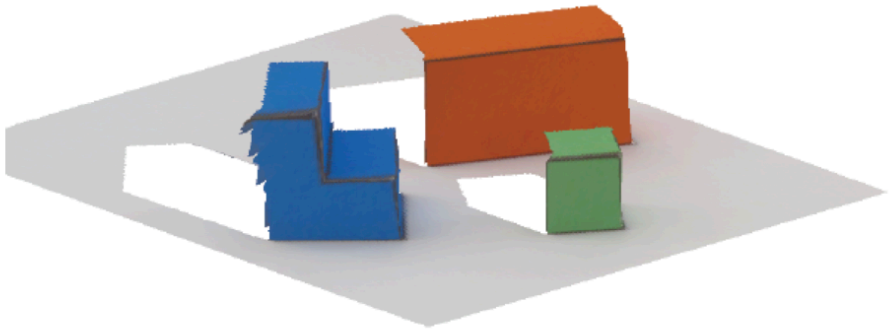
Input



New view points:







Next class:

- Virtual box installation (please do it at home)
- Python/Numpy primer (finish the exercises in the tutorial)
- Pin hole camera

- Homework 1: A simple vision problem

Take-home reading

- A simple vision system (pdf uploaded on blackboard).

- Linear Algebra review (GoodFellow Chapter 2).
- https://www.deeplearningbook.org/contents/linear_algebra.html

- https://www.deeplearningbook.org/slides/02_linear_algebra.pdf