

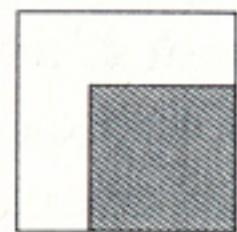
# Machine Vision

*M. Anthony Kapolka III*

*Wilkes University*

*CS 340 AI Fall 2019*

**Digitization**

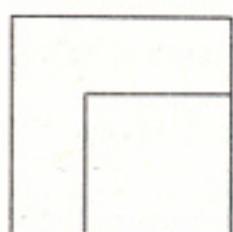


*Original scene*

**Low-level processing**

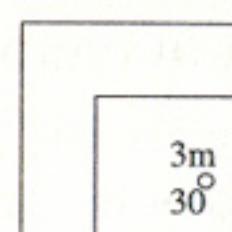
9	9	9	9
9	5	3	3
9	4	2	2
9	4	2	2

*Digitized image*



*Edges and regions*

**Medium-level processing**

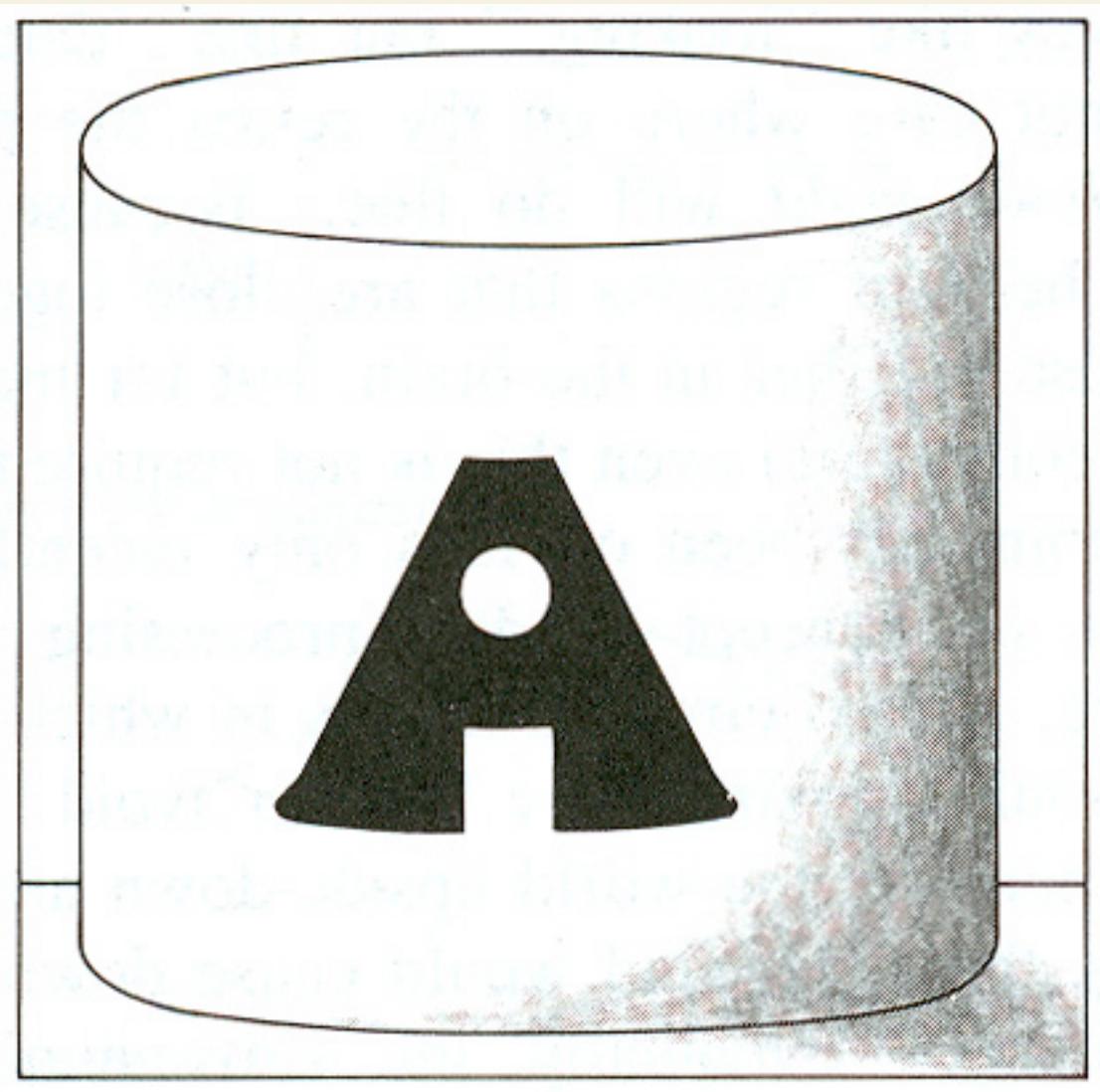


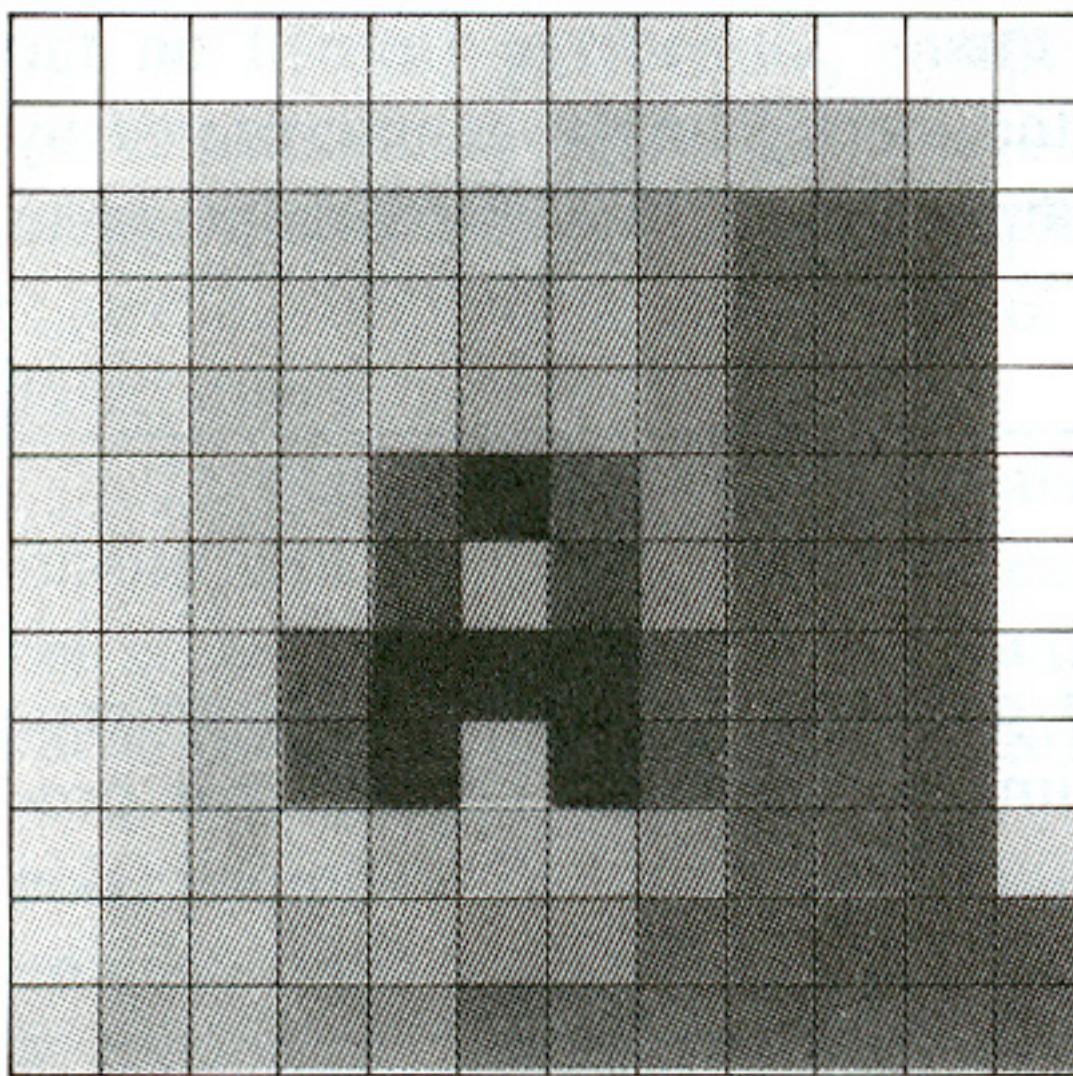
*Depth/orientation*

**High-level processing**

*Recognized object*

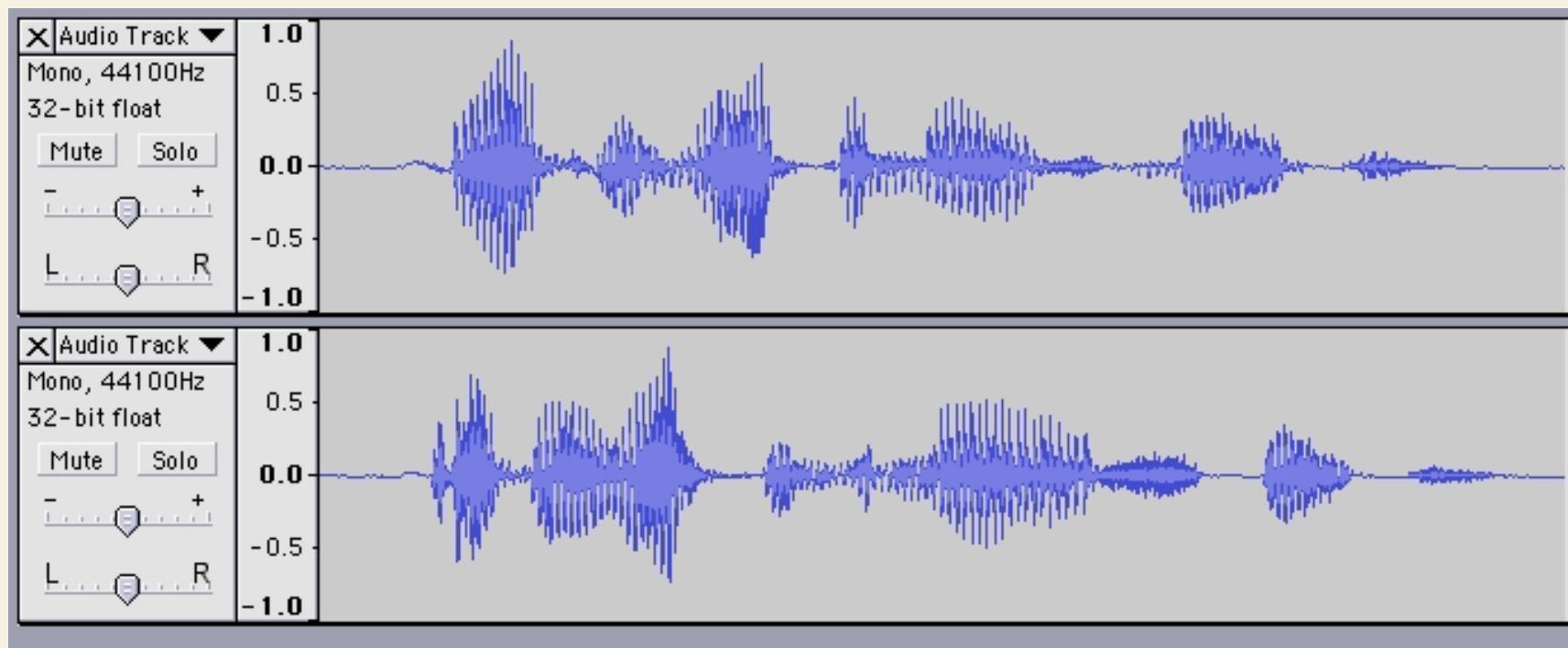
"Widget"



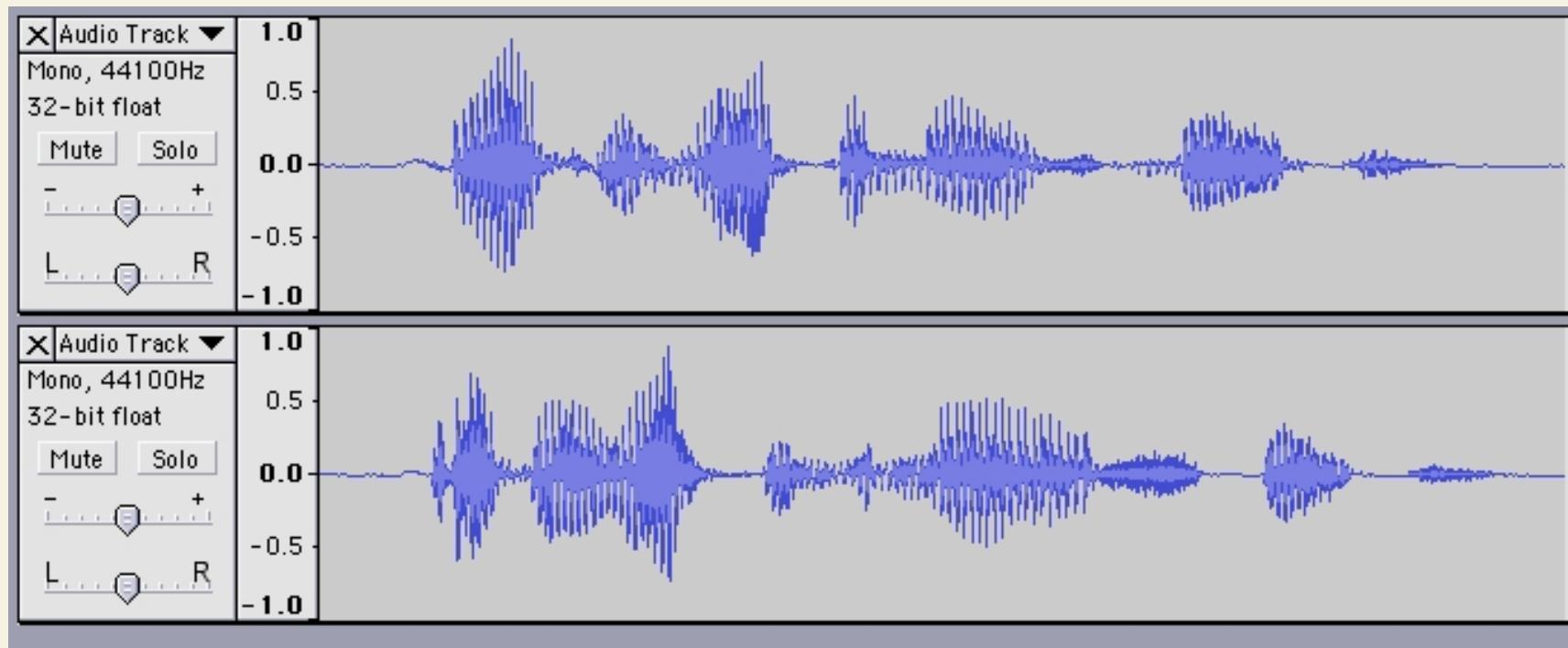


	1	2	3	4	5	6	7	8	9	10	11	12
1	.9	.9	.9	.8	.8	.7	.7	.7	.8	.9	.9	.9
2	.9	.8	.7	.7	.7	.7	.6	.6	.6	.5	.5	.9
3	.8	.7	.6	.6	.6	.6	.5	.4	.3	.2	.1	.9
4	.8	.7	.6	.6	.6	.5	.5	.4	.3	.2	.1	.9
5	.8	.7	.6	.6	.6	.5	.5	.4	.3	.2	.1	.9
6	.8	.7	.6	.6	.3	.0	.3	.4	.3	.2	.1	.9
7	.8	.7	.6	.6	.1	.5	.1	.4	.3	.2	.1	.9
8	.8	.7	.6	.3	0	0	0	.3	.3	.2	.1	.9
9	.8	.7	.6	.2	0	.5	0	.2	.3	.2	.1	.9
10	.8	.7	.6	.6	.5	.5	.5	.4	.3	.2	.1	.7
11	.8	.6	.6	.5	.5	.4	.4	.3	.2	.1	.1	.1
12	.7	.5	.5	.4	.4	.3	.3	.2	.1	.1	.1	.1

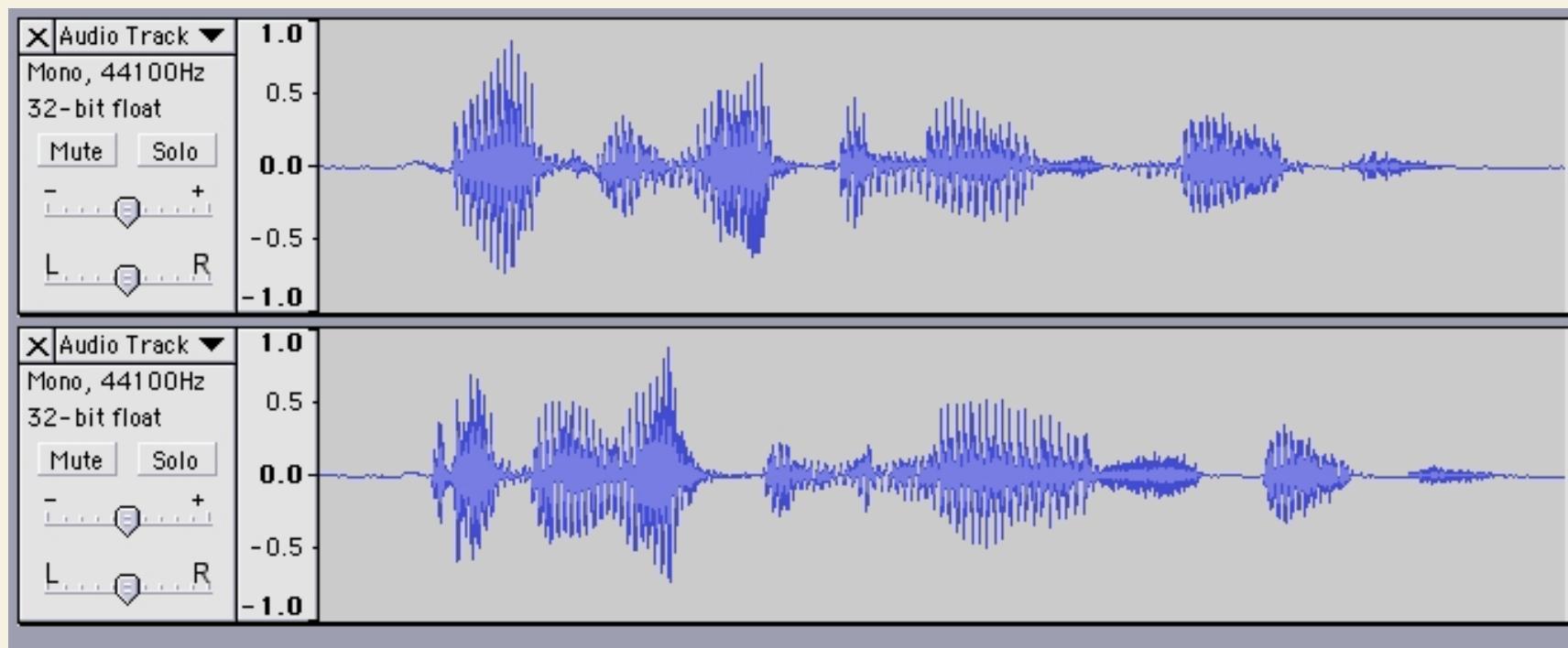
# speech recognition?

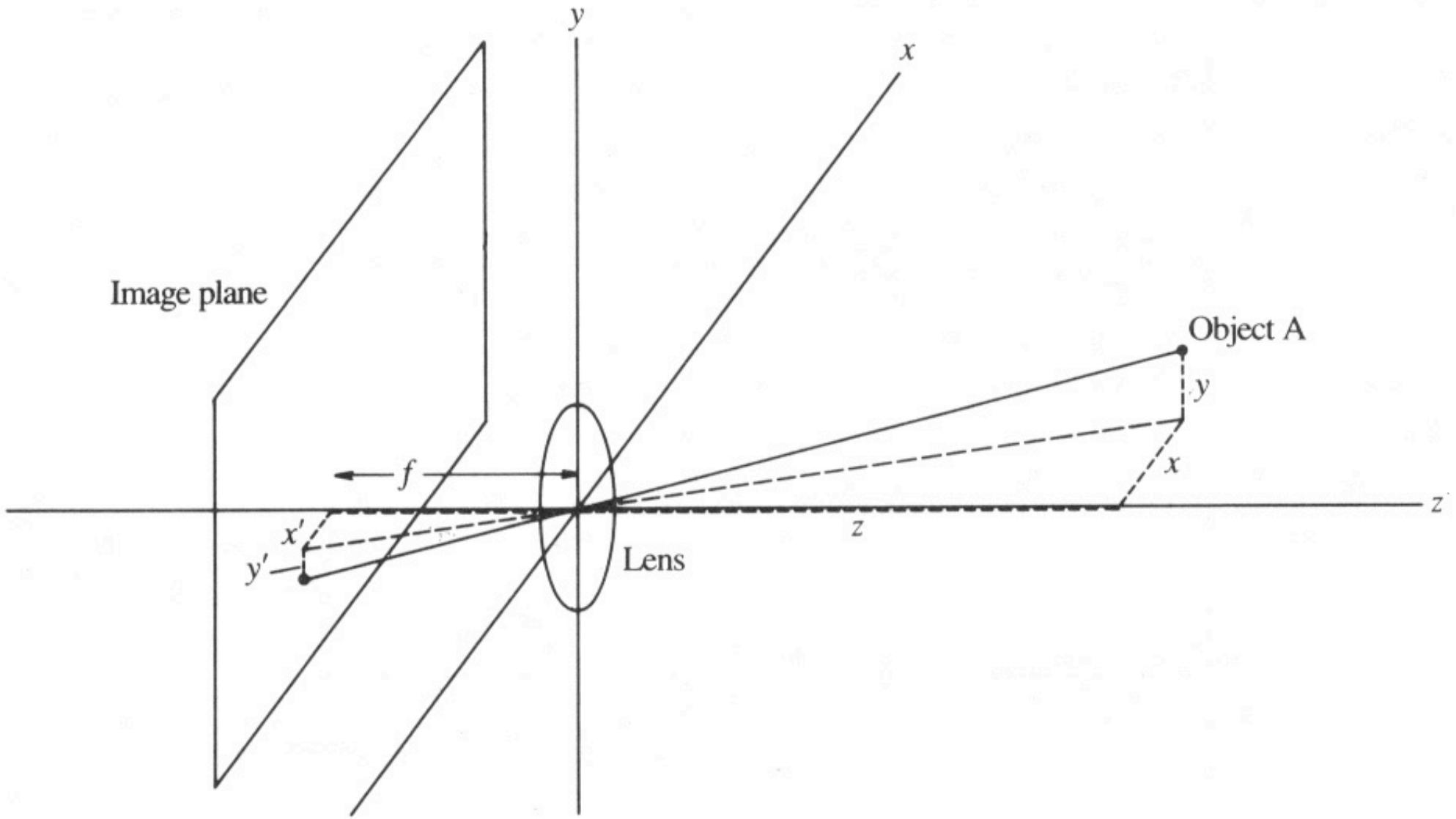


# speech recognition?



# speech recognition?





Input image

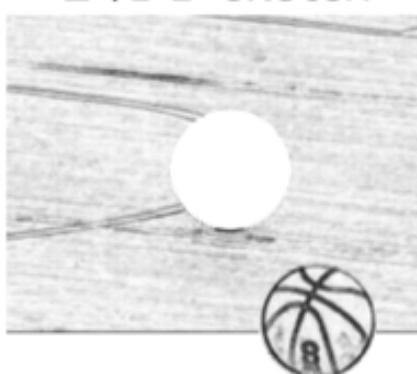


[This image is CC0 1.0 public domain](#)

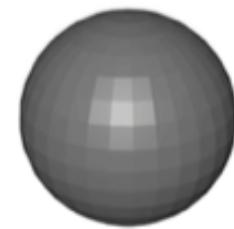
Edge image



2 ½-D sketch



3-D model



[This image is CC0 1.0 public domain](#)

Input  
Image

Perceived  
intensities

Primal  
Sketch

Zero crossings,  
blobs, edges,  
bars, ends,  
virtual lines,  
groups, curves  
boundaries

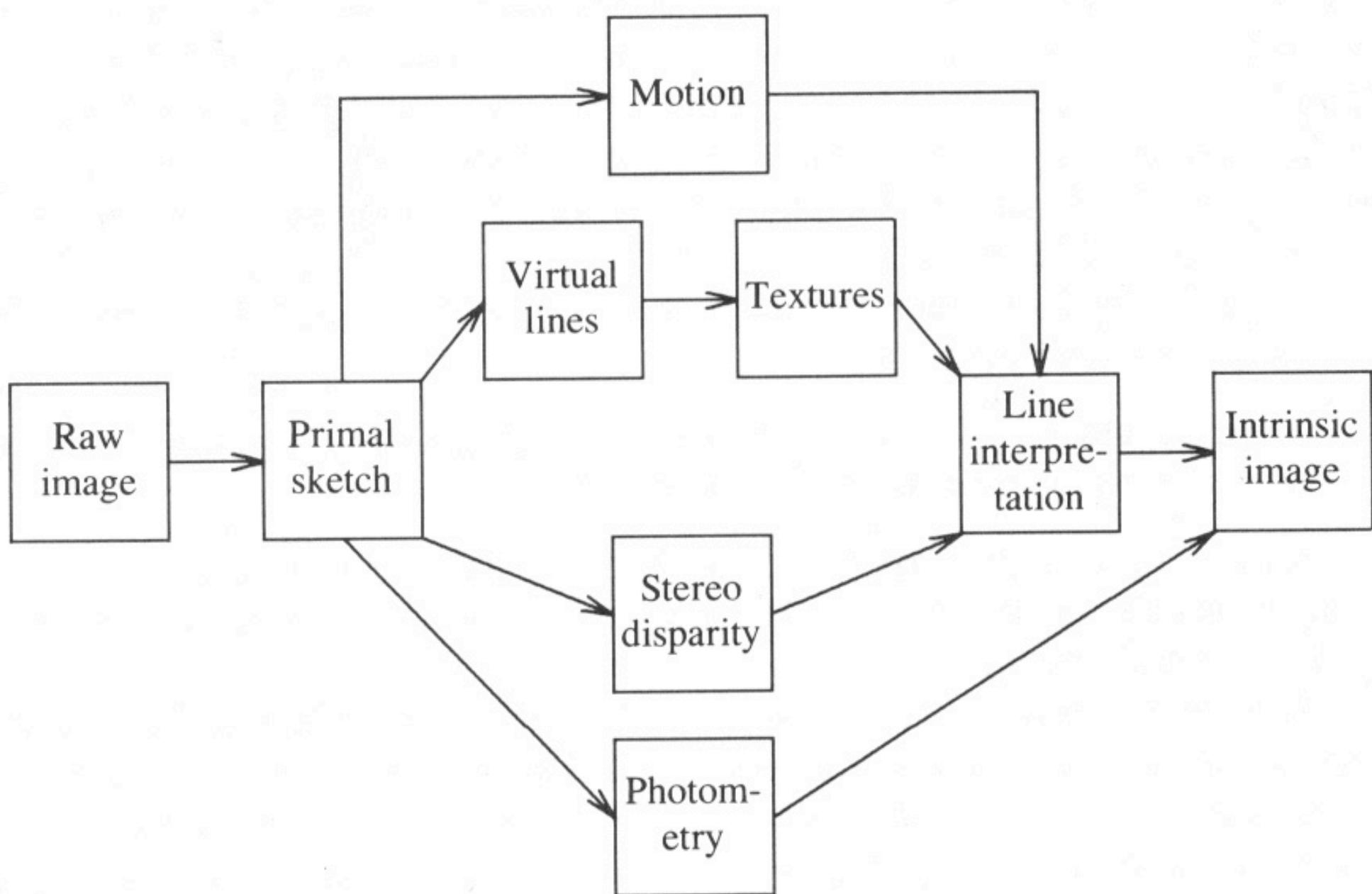
2 ½-D  
Sketch

Local surface  
orientation and  
discontinuities  
in depth and in  
surface  
orientation

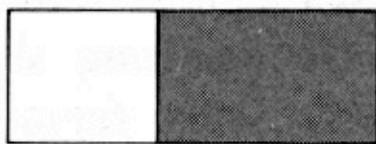
3-D Model  
Representation

3-D models  
hierarchically  
organized in  
terms of surface  
and volumetric  
primitives

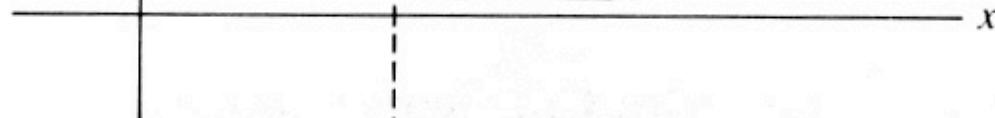
Stages of Visual Representation, David Marr, 1970s



Edge in image:



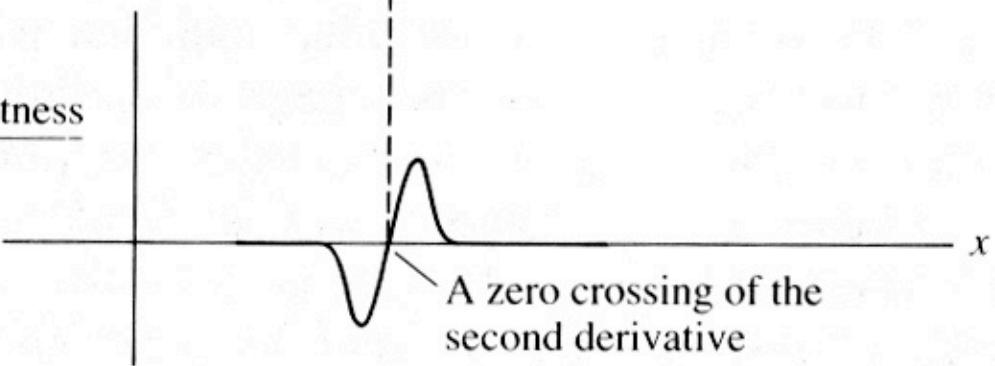
Brightness



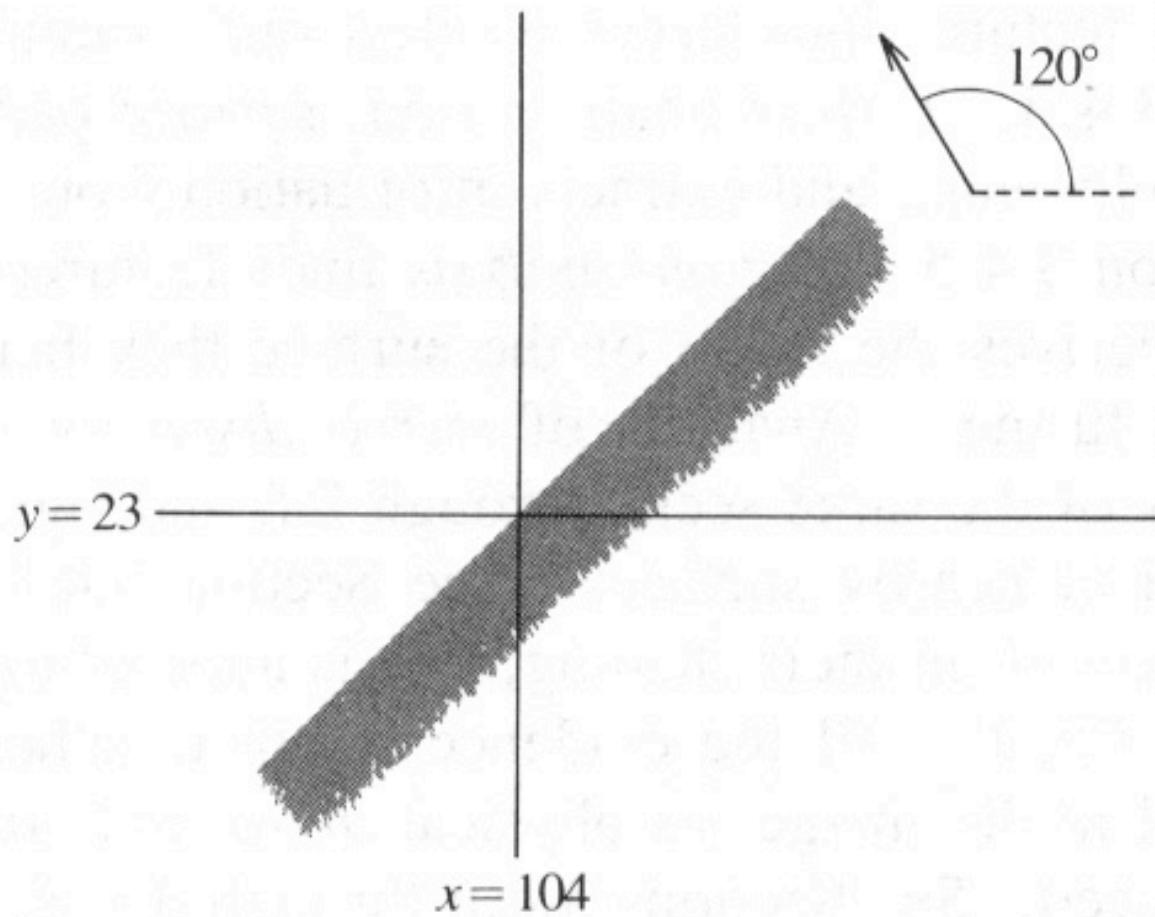
$\frac{d \text{ brightness}}{dx}$



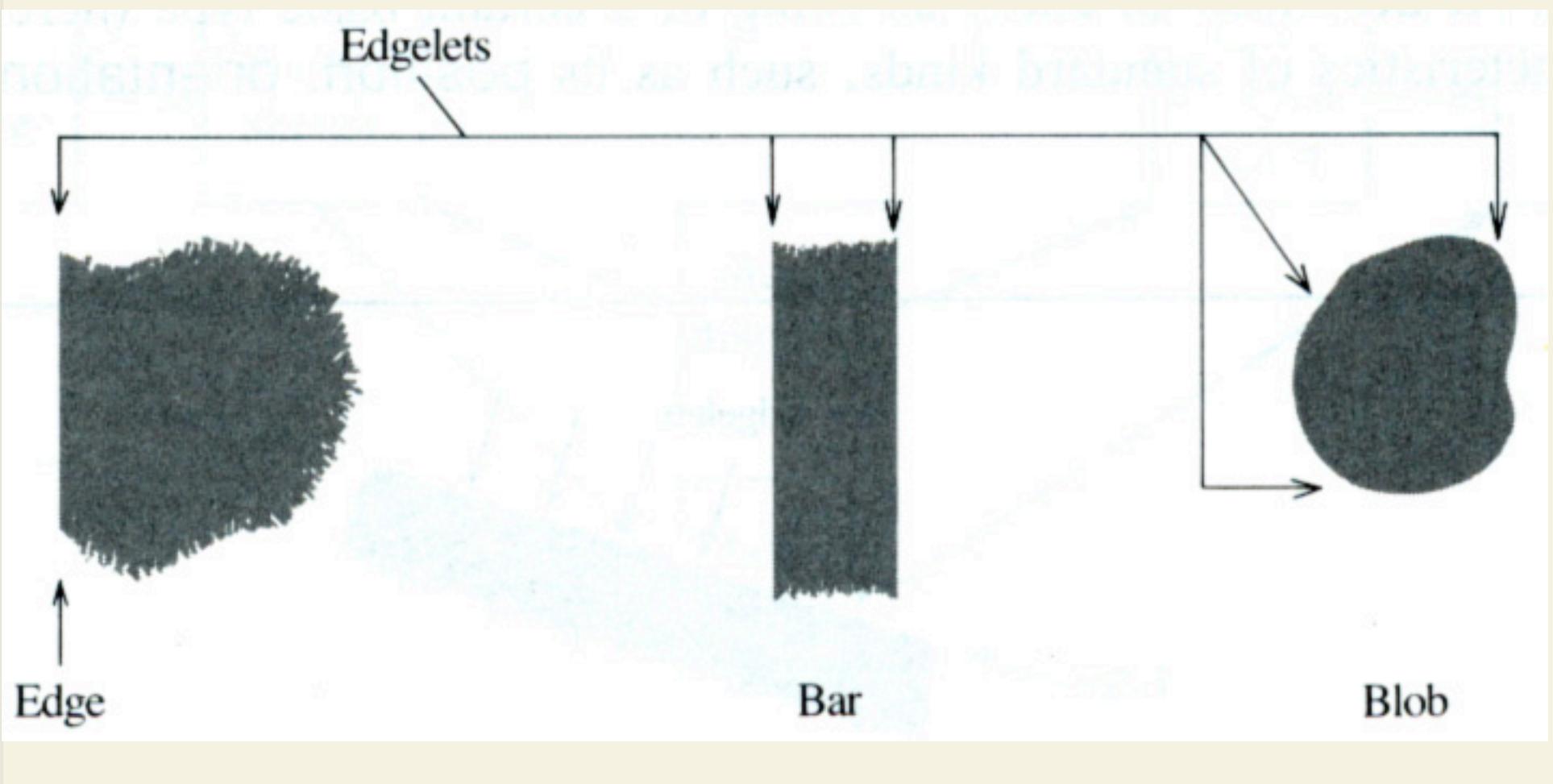
$\frac{d^2 \text{ brightness}}{dx^2}$



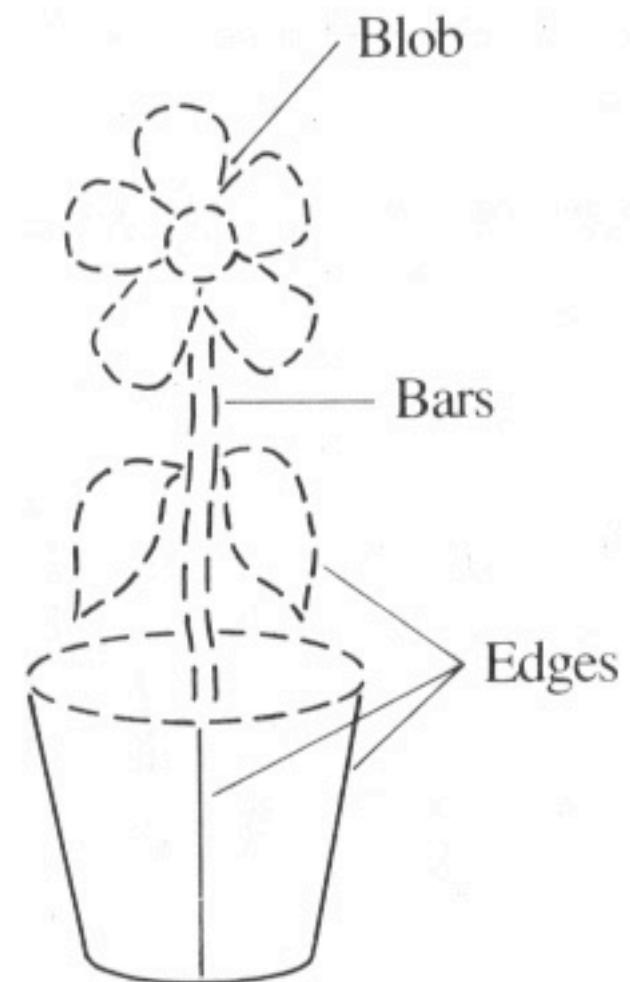
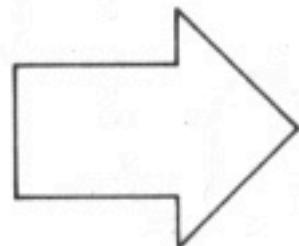
(edge edge2104  
  (position 104 23)  
  (orientation 120)  
  (contrast 25)  
  (length 25))



# Primitives



# Decomposition



# SIFT - Scale Invariant Feature Transform

extracts features, regardless of image scale,  
orientation, illumination, affine distortions

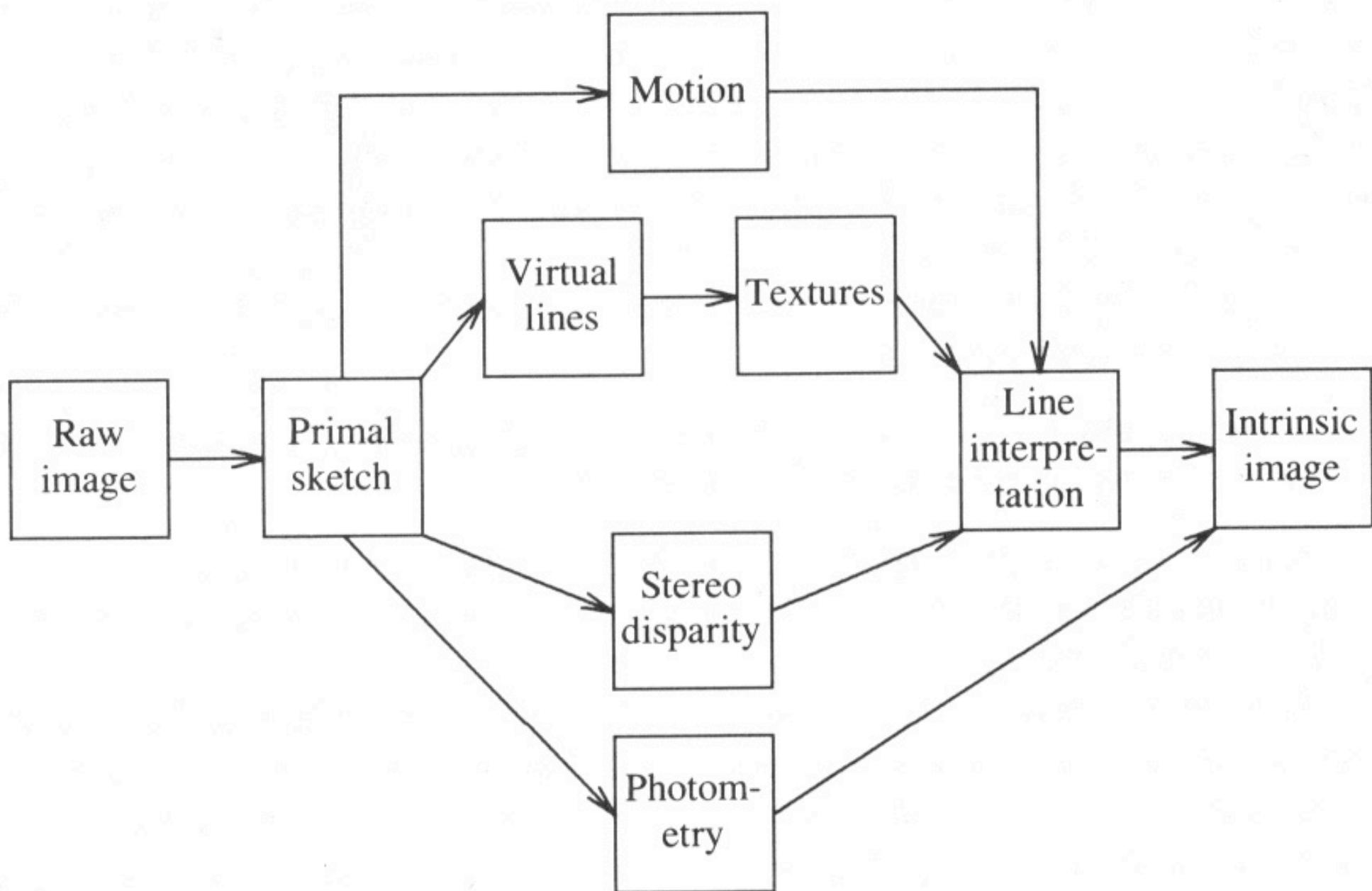


[Image](#) is public domain

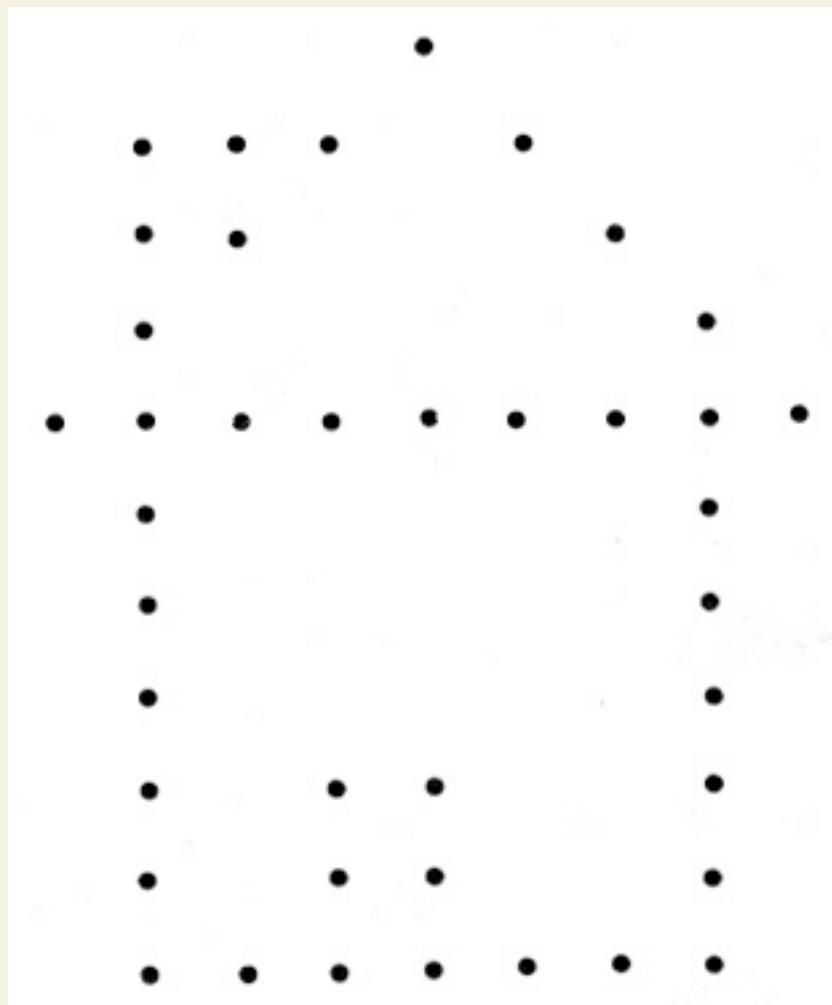


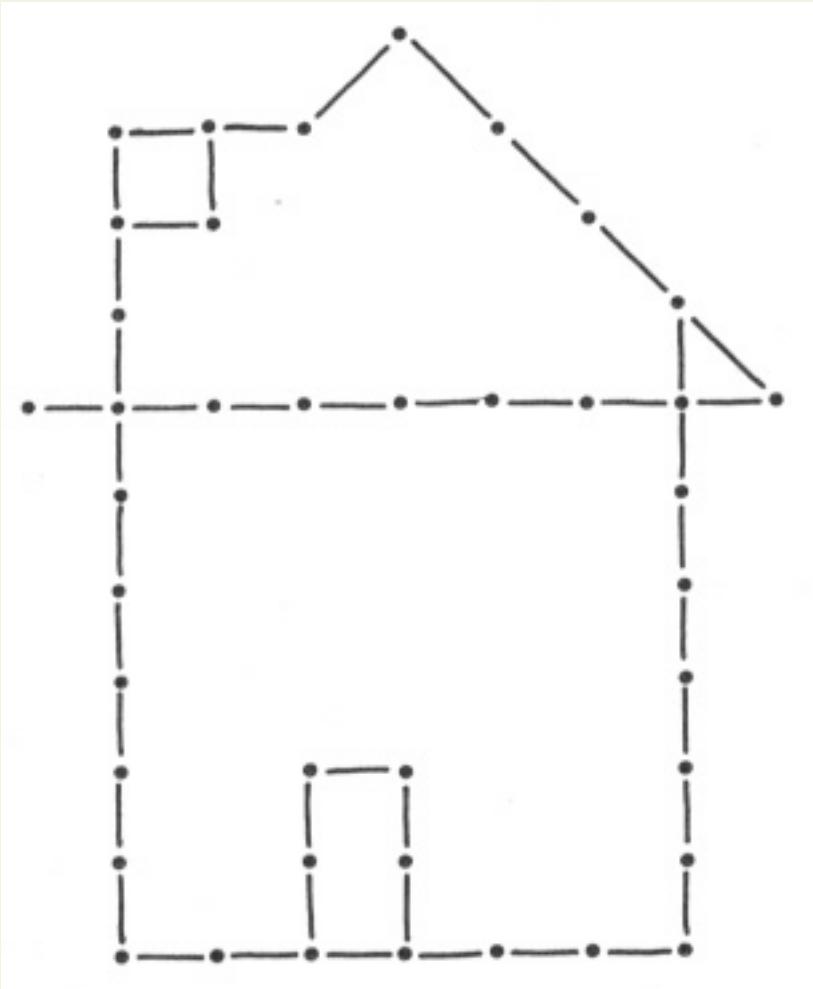
[Image](#) is CC BY-SA 2.0

"SIFT" & Object Recognition, David Lowe, 1999



# missing information

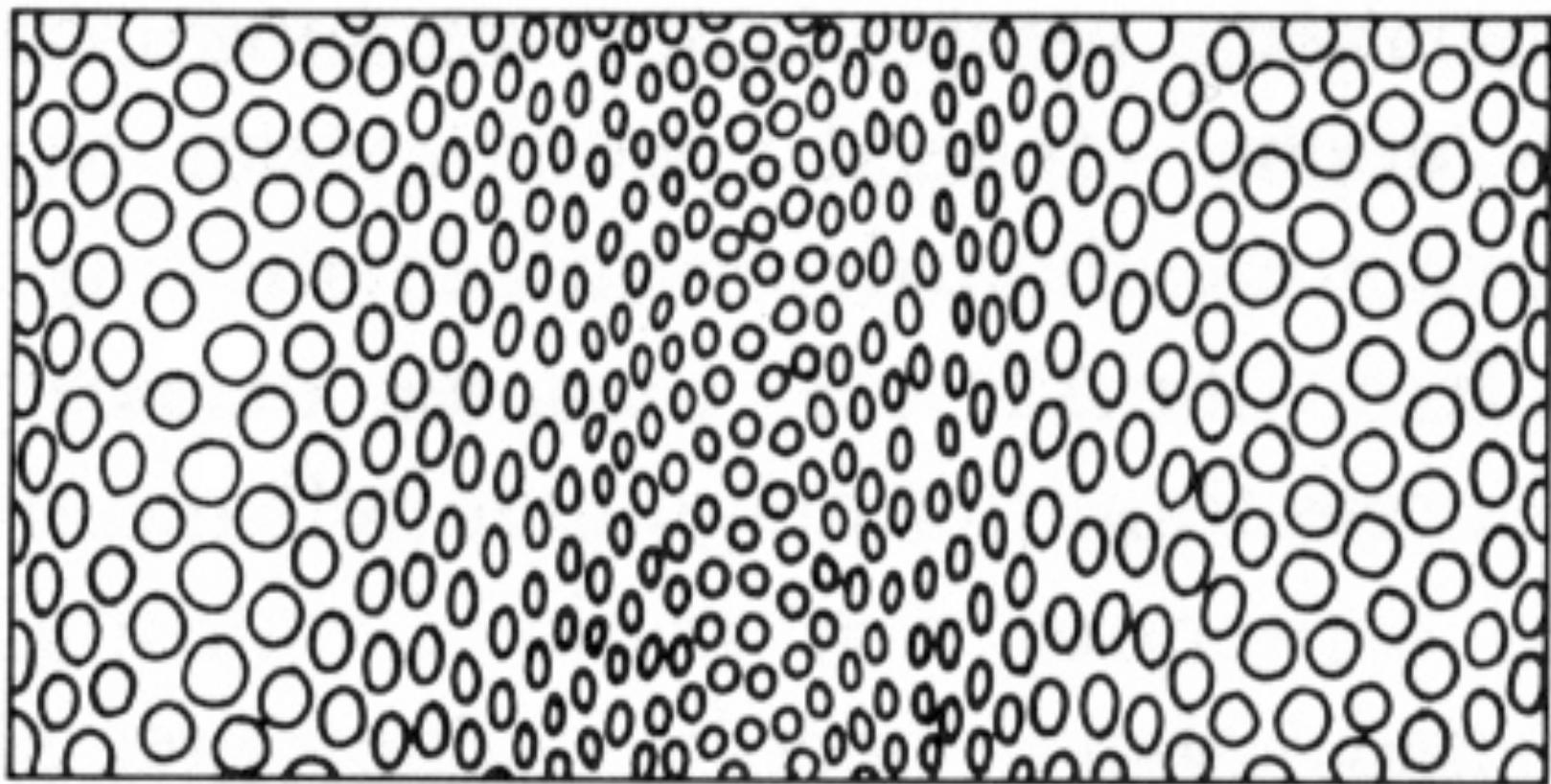


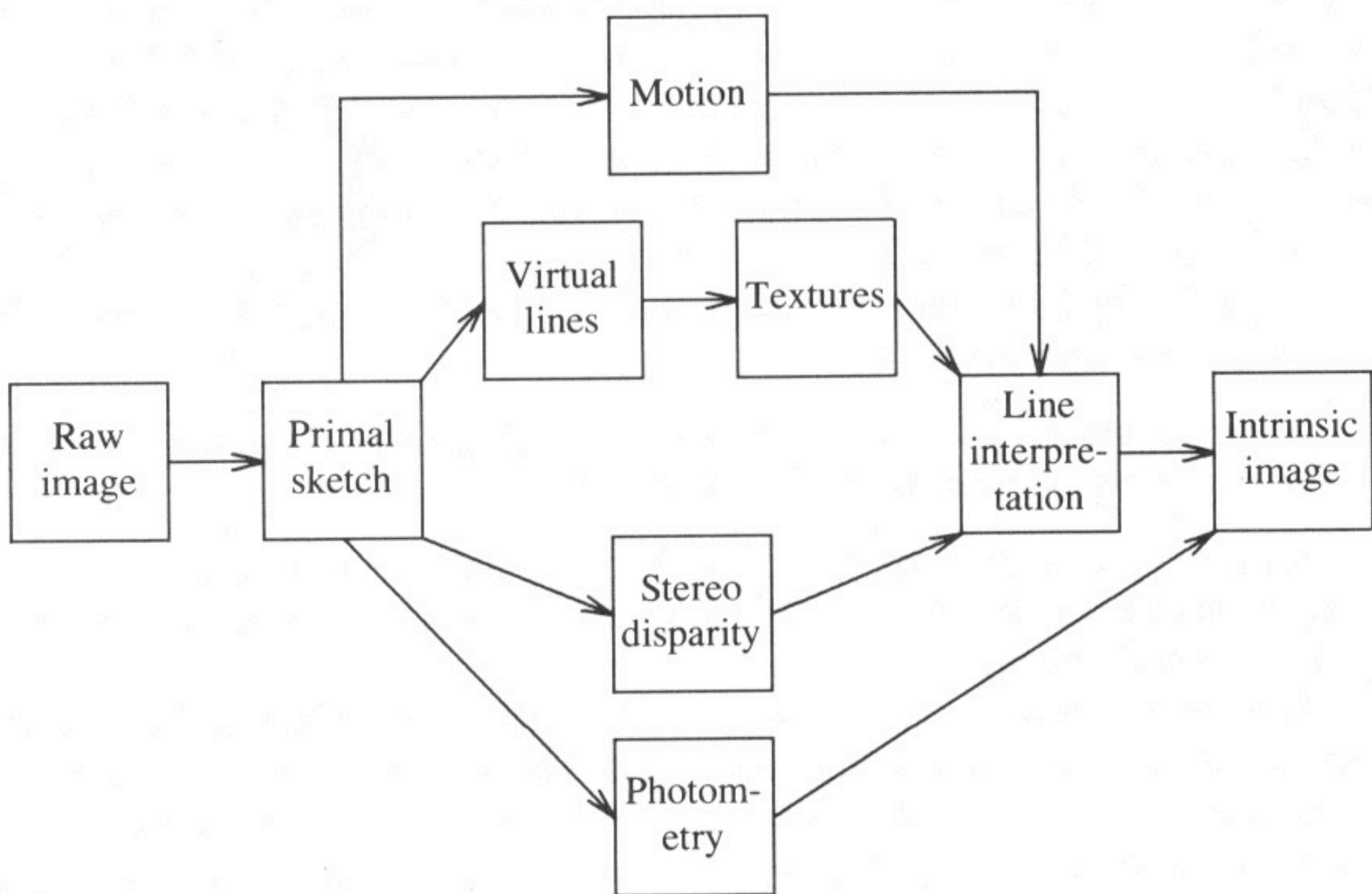


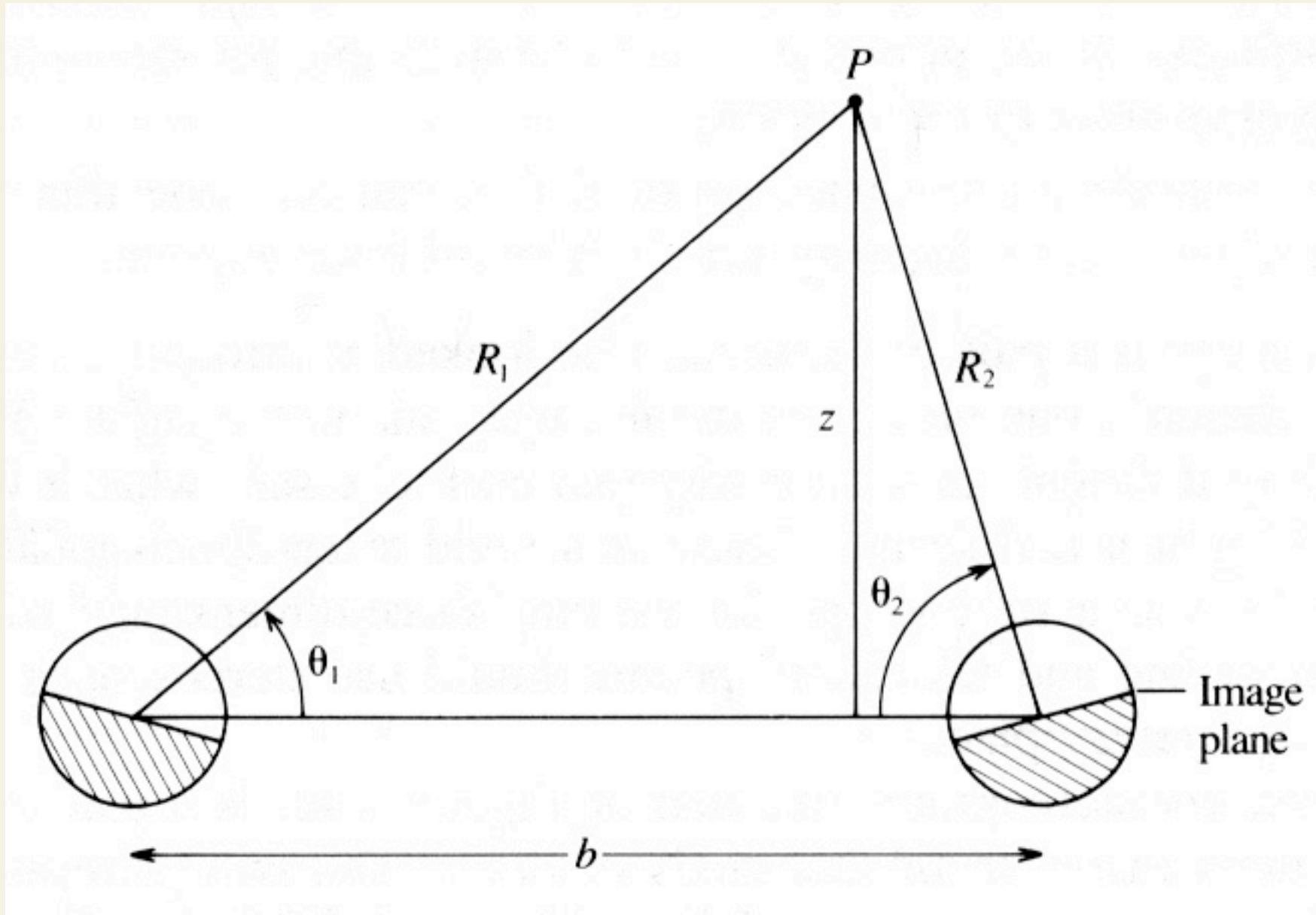
# Texture



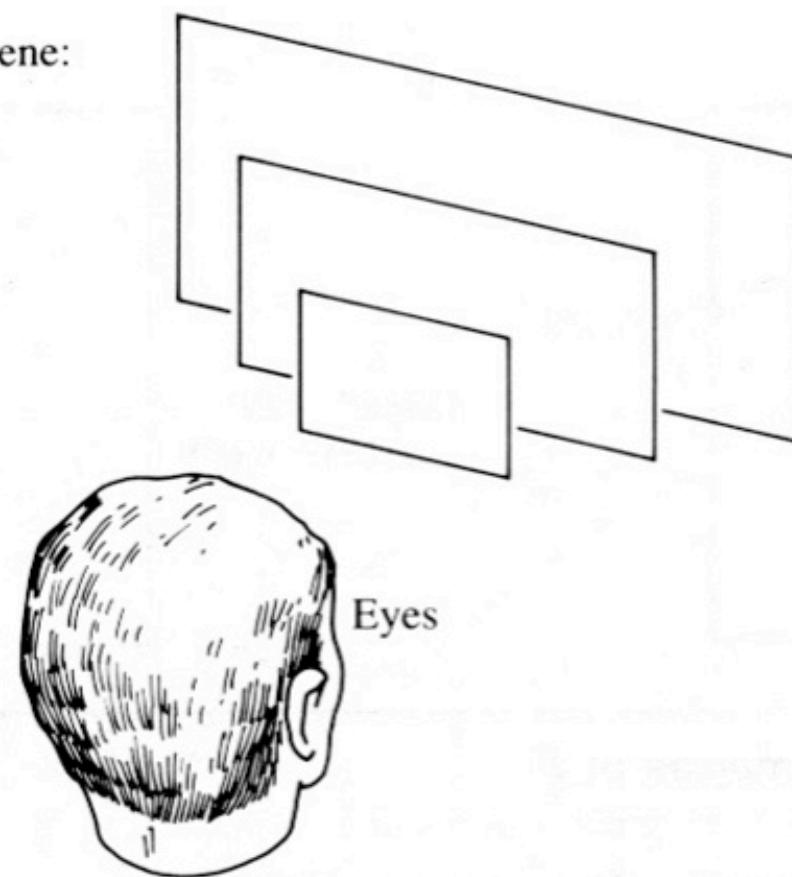
# Texture





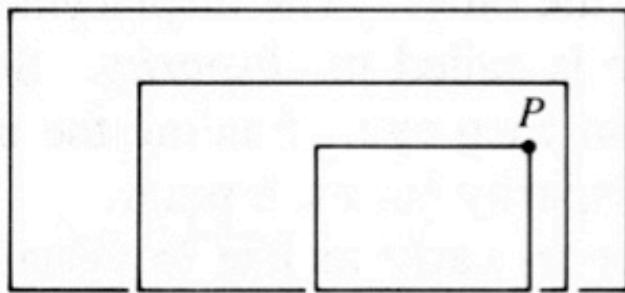


Scene:

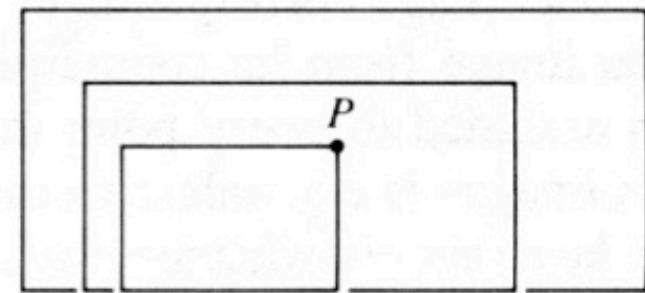


Eyes

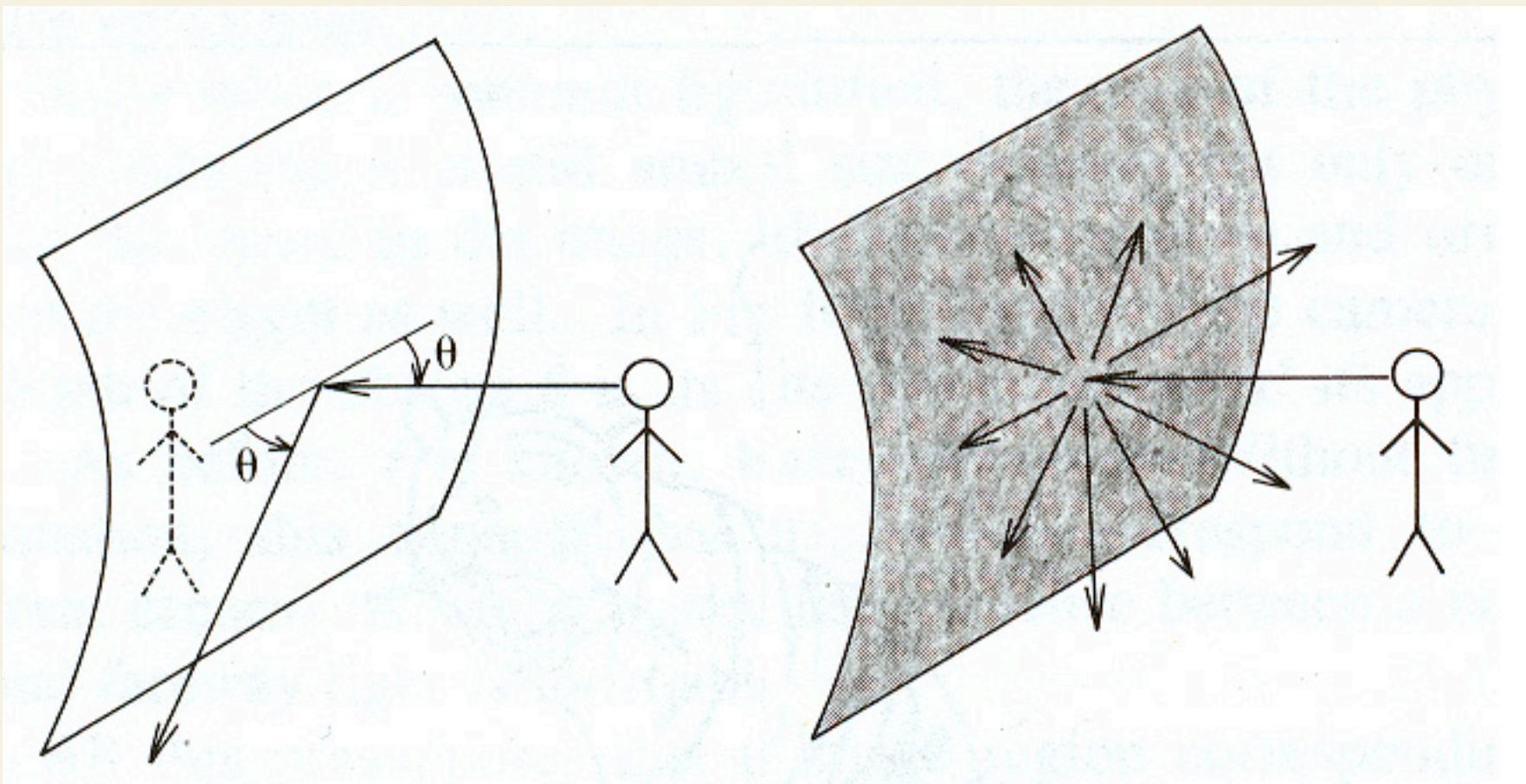
Left image

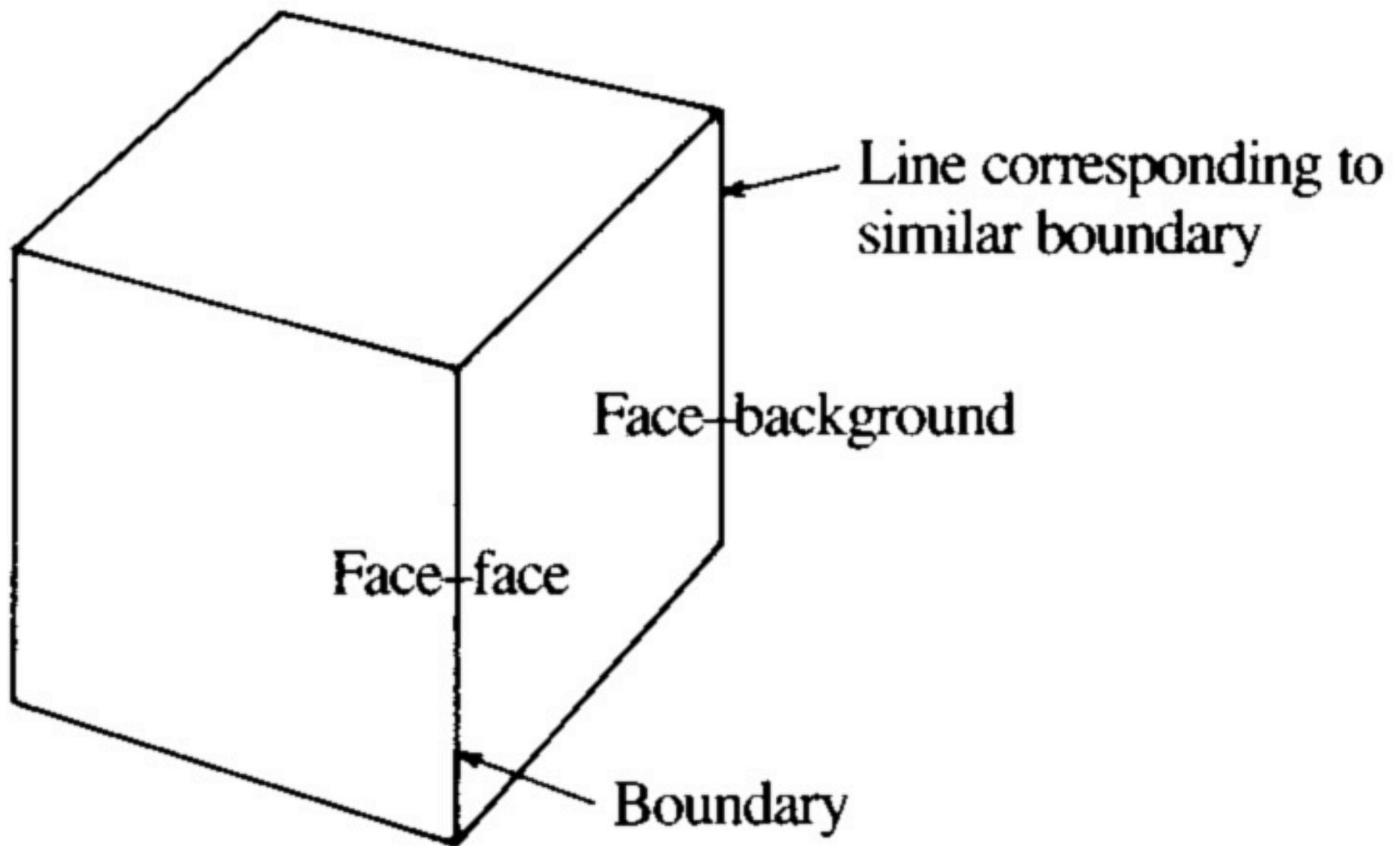


Right image



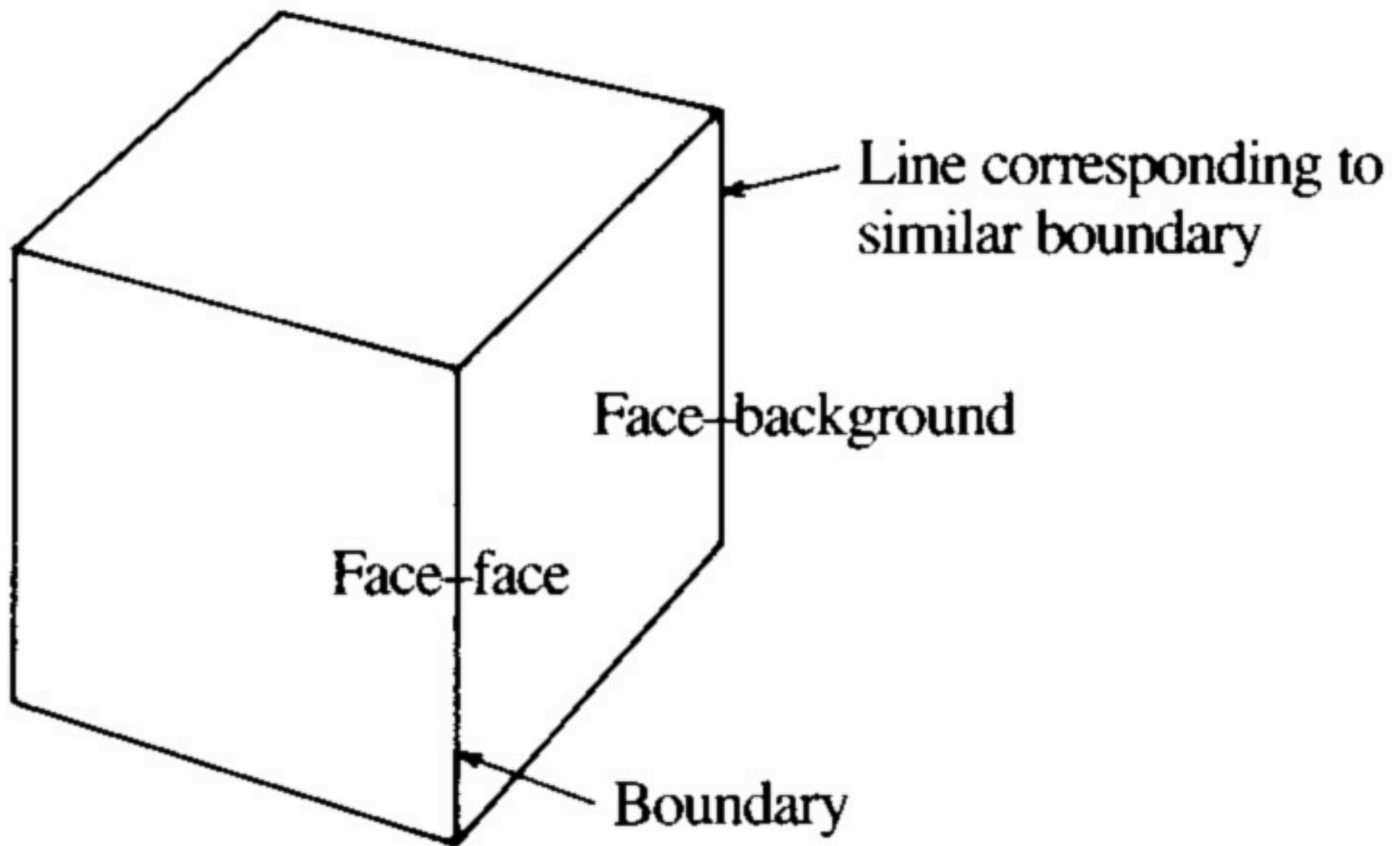
# Reflections





# Concepts

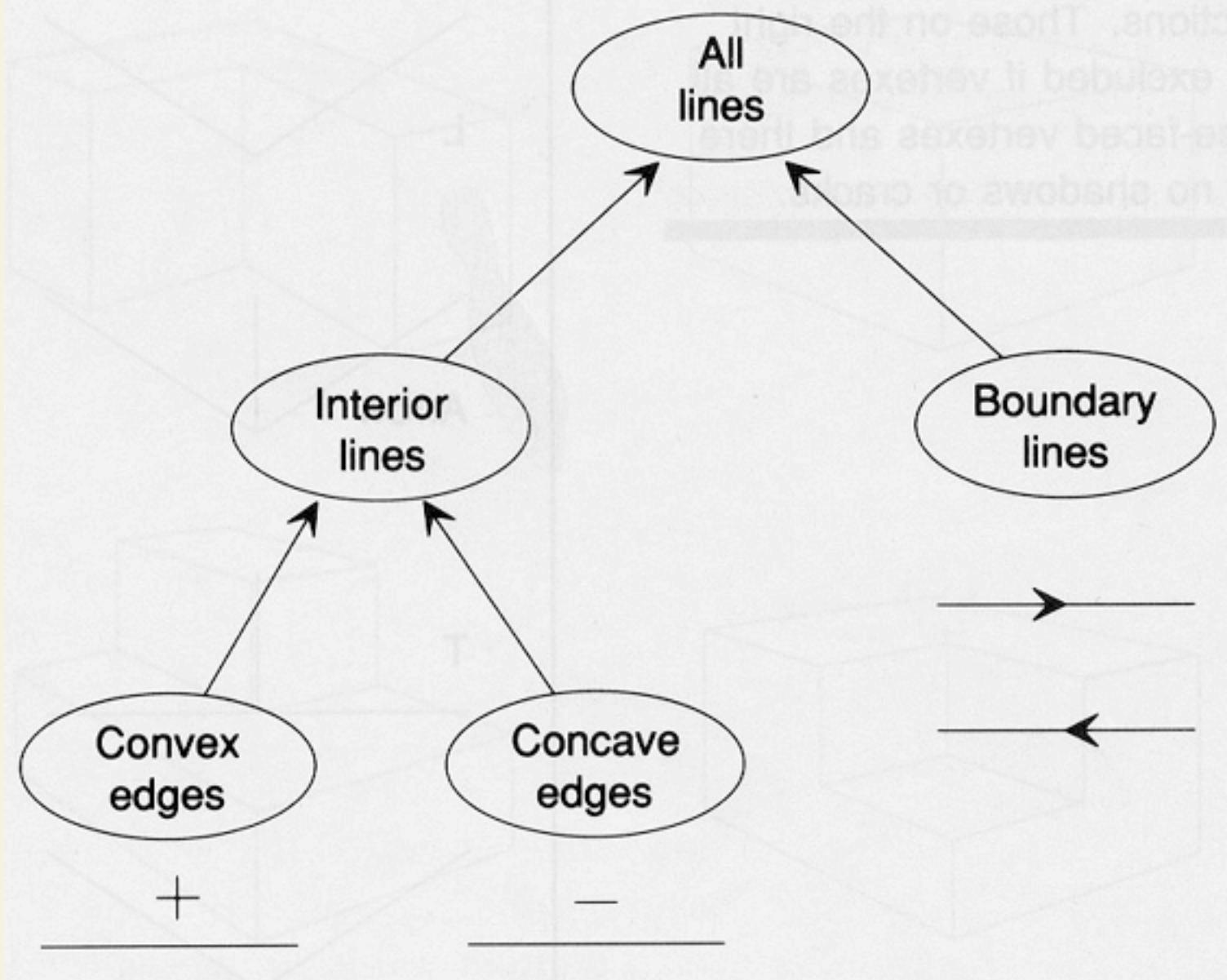
Where Defined	Concept	Definition
Image	Edgelet	Unit intensity change
Image	Edge	Primal-sketch feature, composed of several edgelets
Image	Line	Sequence of primal-sketch features, <i>or</i> texture-region boundary, <i>or</i> . . .
World	Boundary	Place where two surfaces meet, or where illumination or reflectance changes abruptly



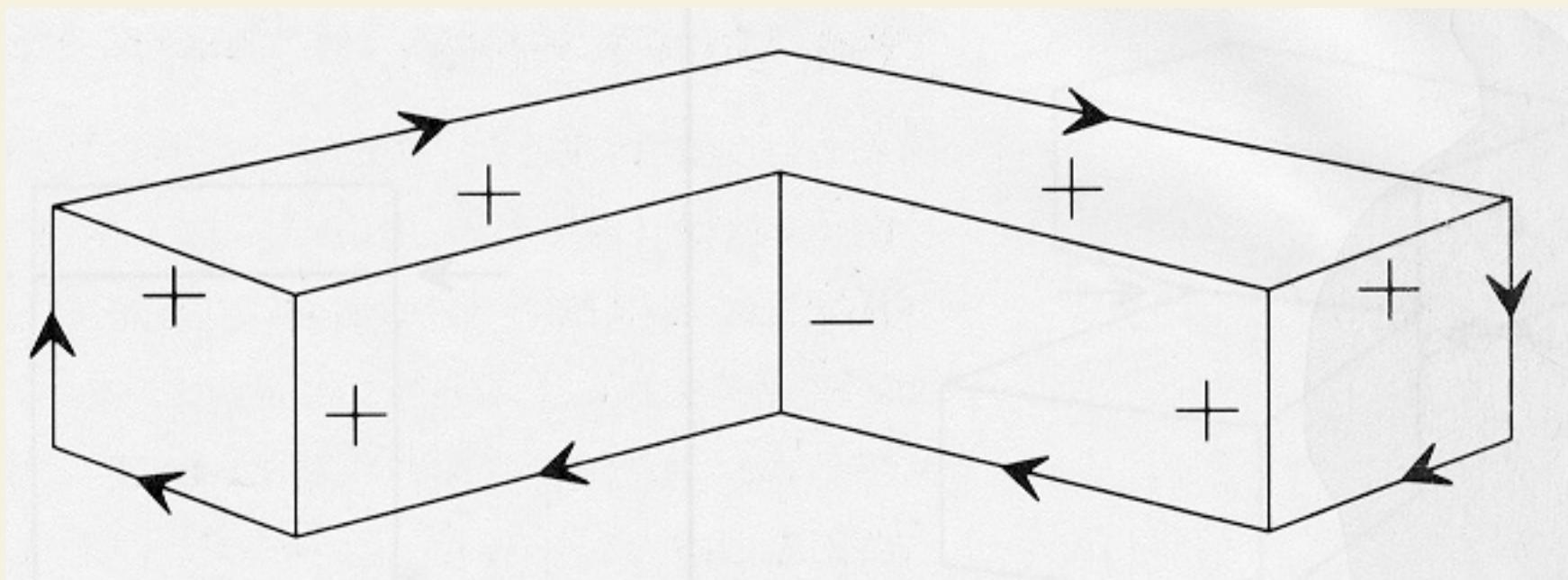
# Waltz Algorithm:

## Only 18 ways to Label a Three-Faced Junction

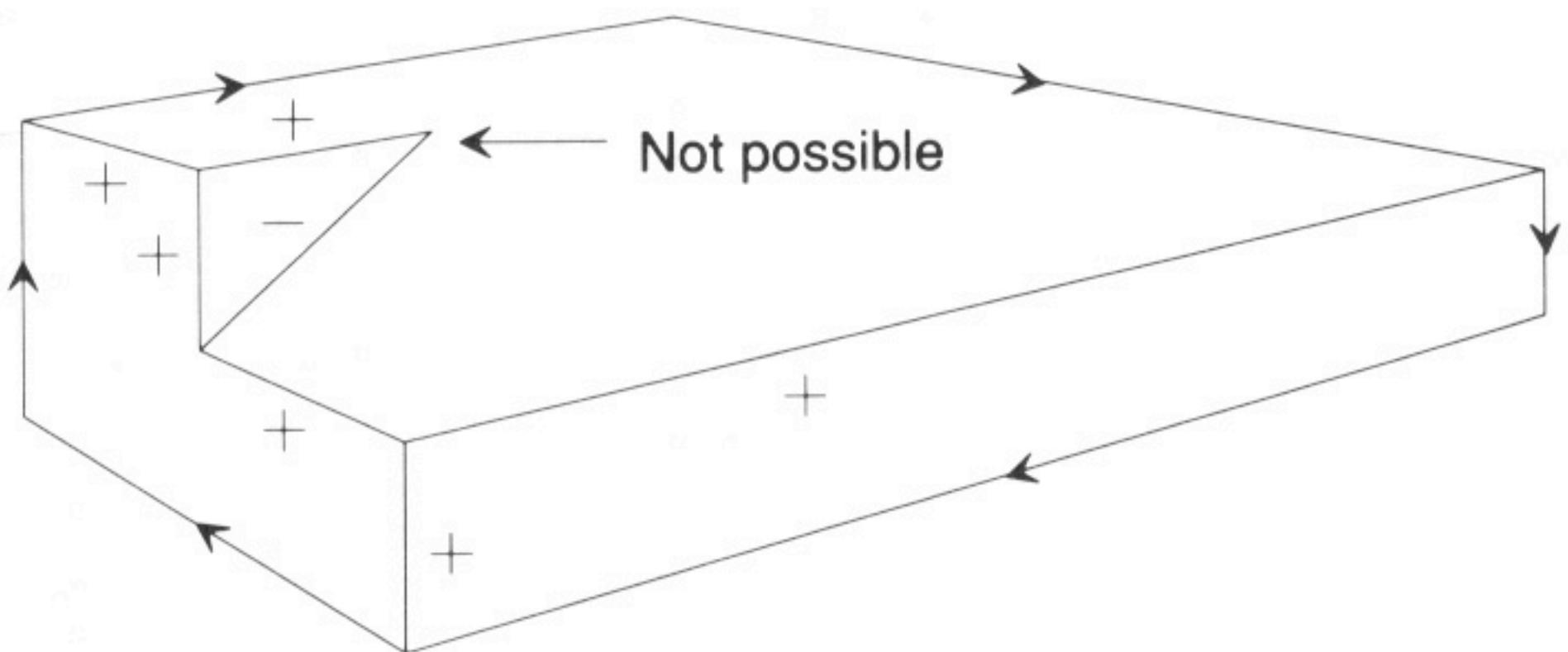
MIT's Lecture <https://www.youtube.com/watch?v=l-tzjenXrvI>

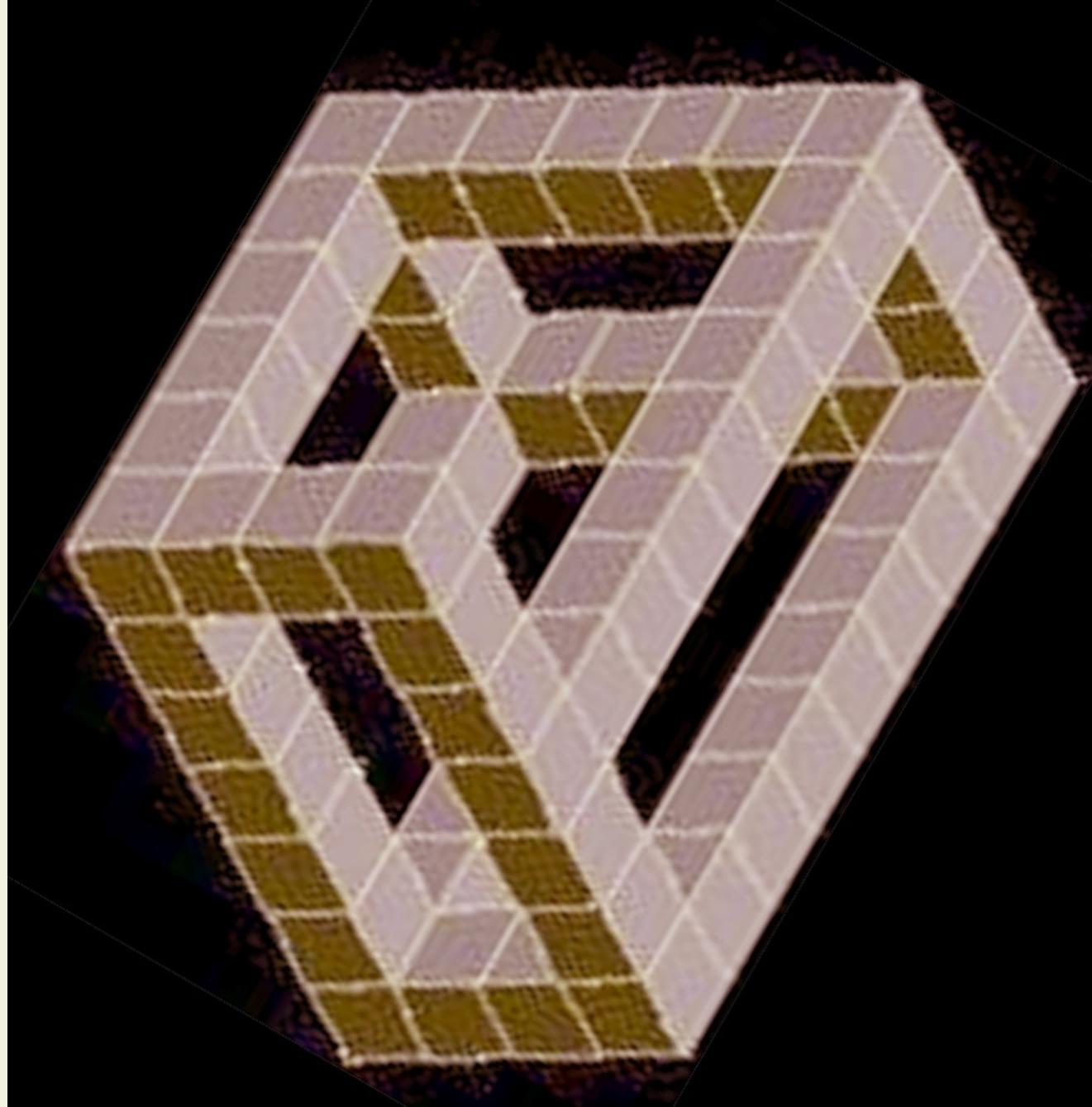


# “Label”



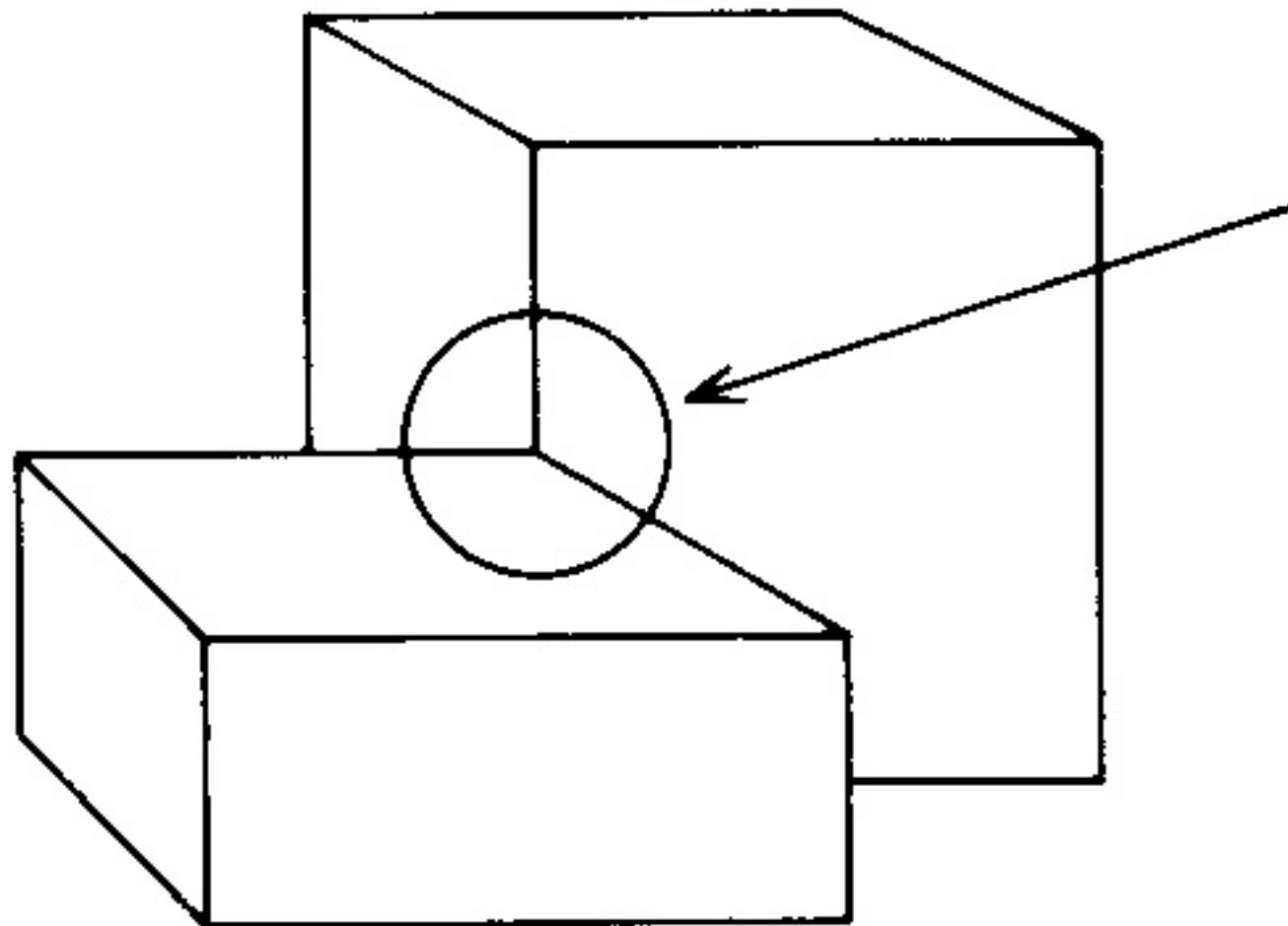
# Discover Problems

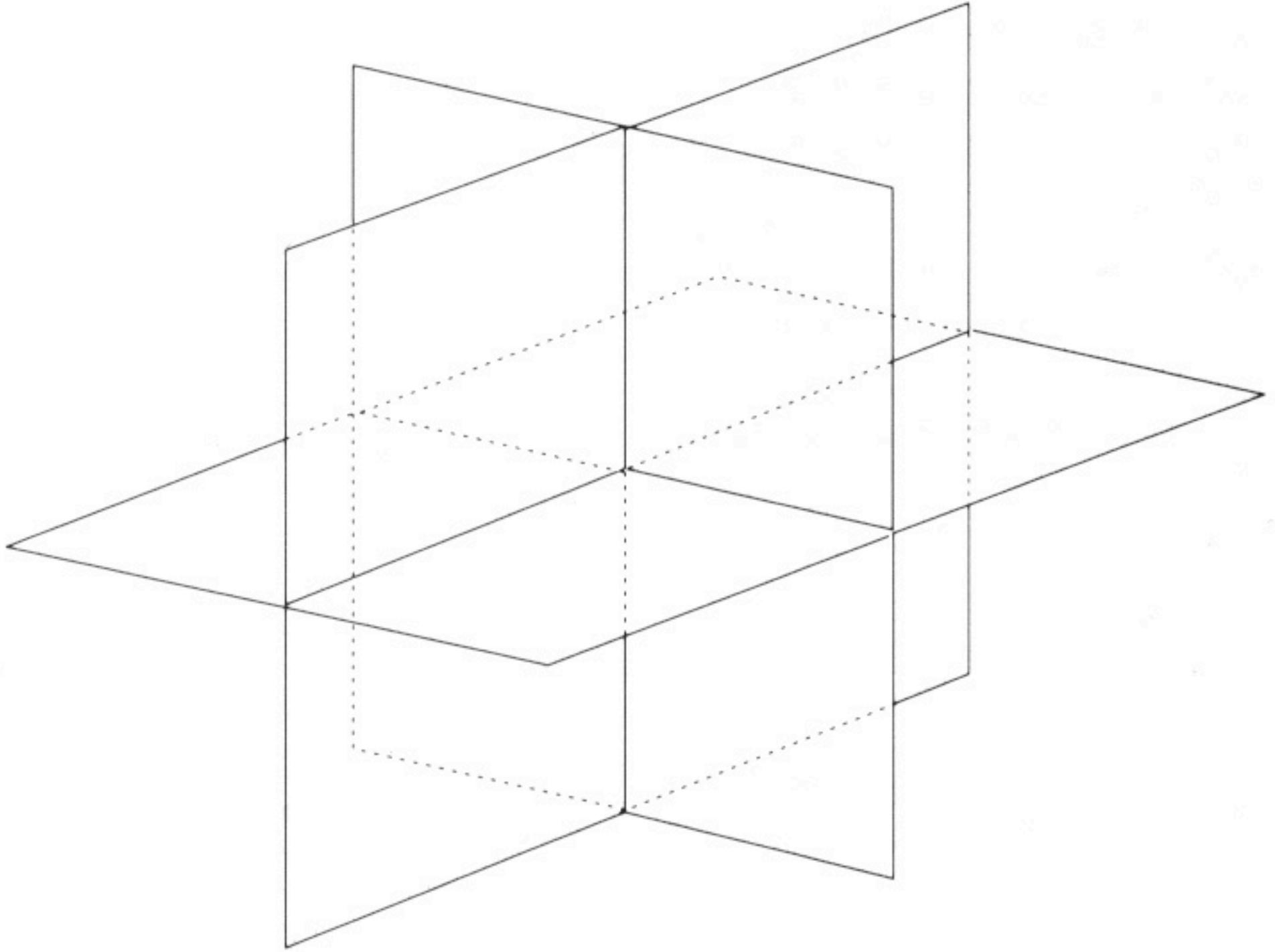




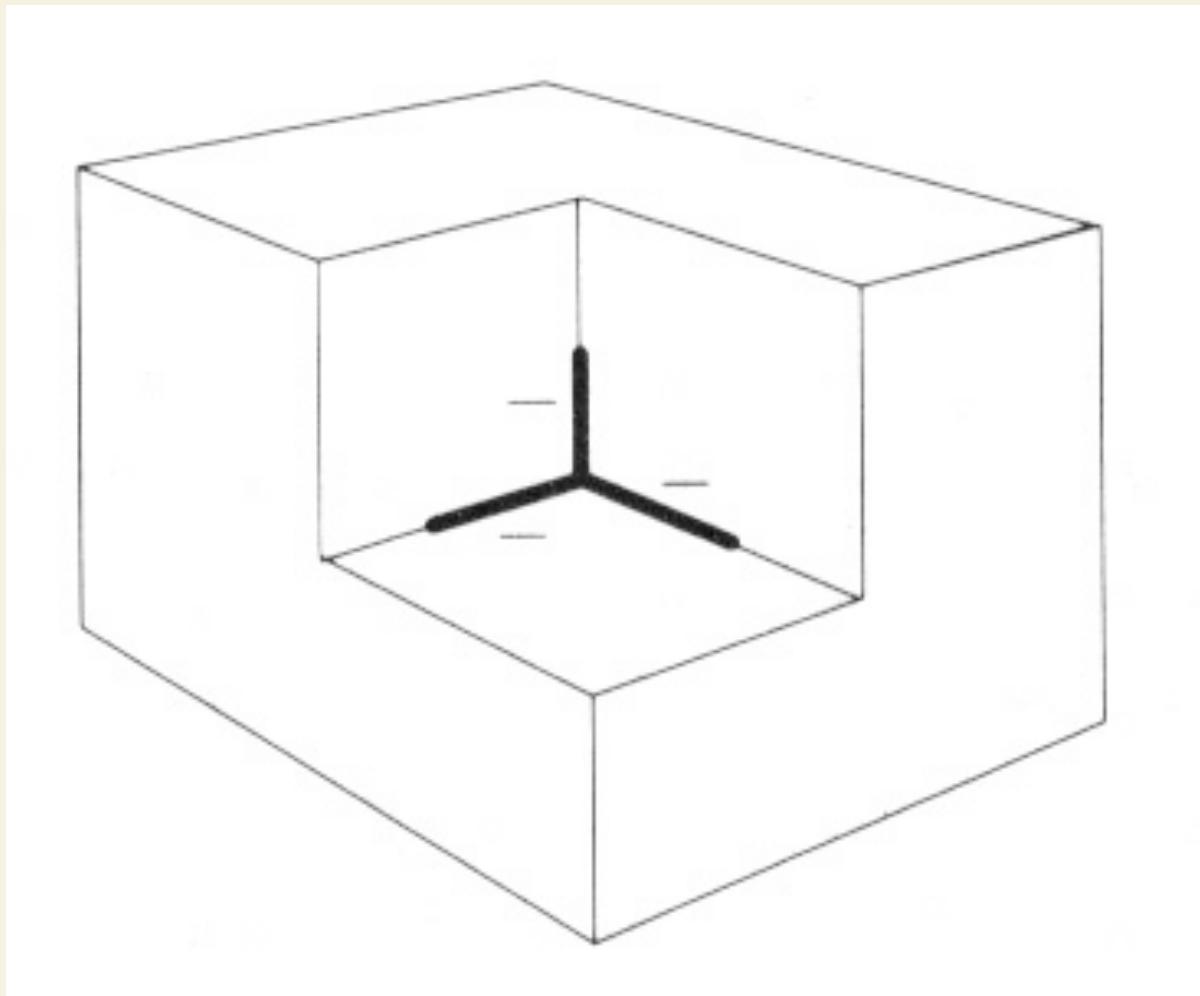
<http://home.uchicago.edu/~janie/escher/EscherBoxbrown.jpg>

# No Coincidences

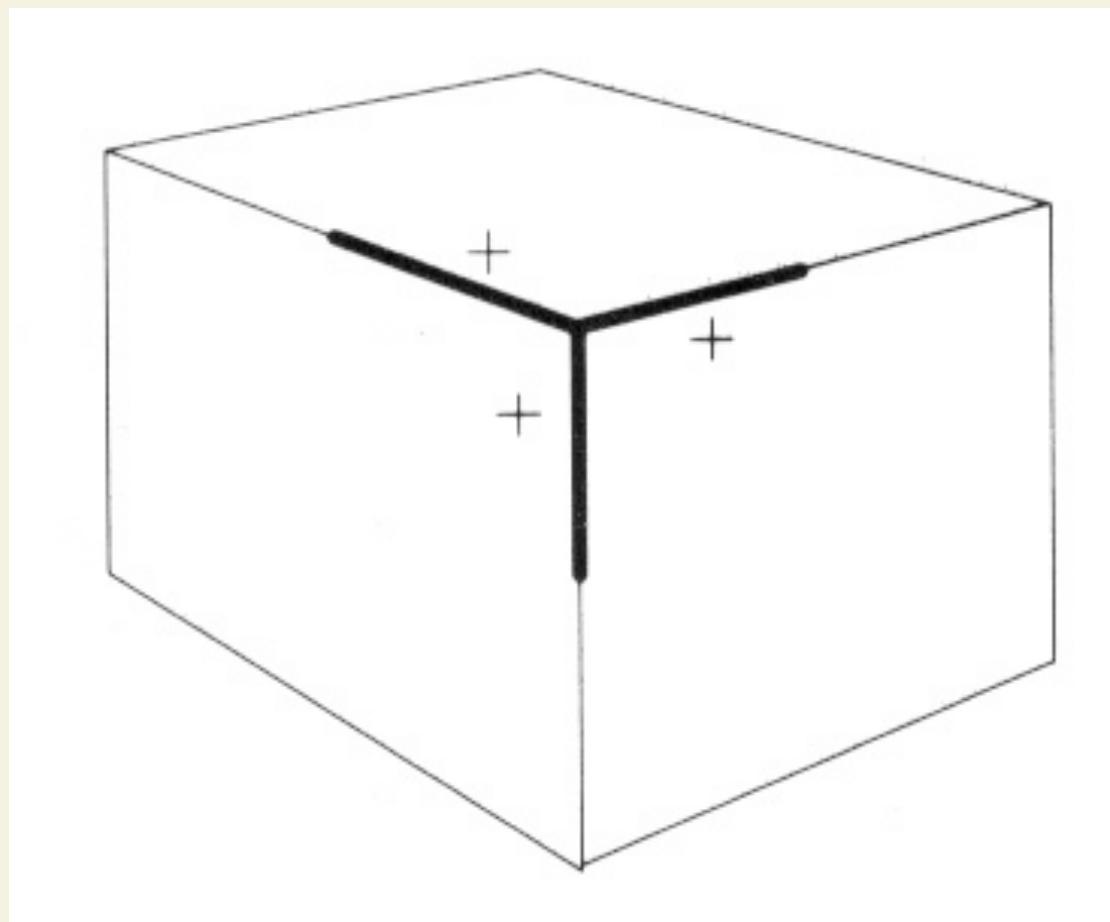




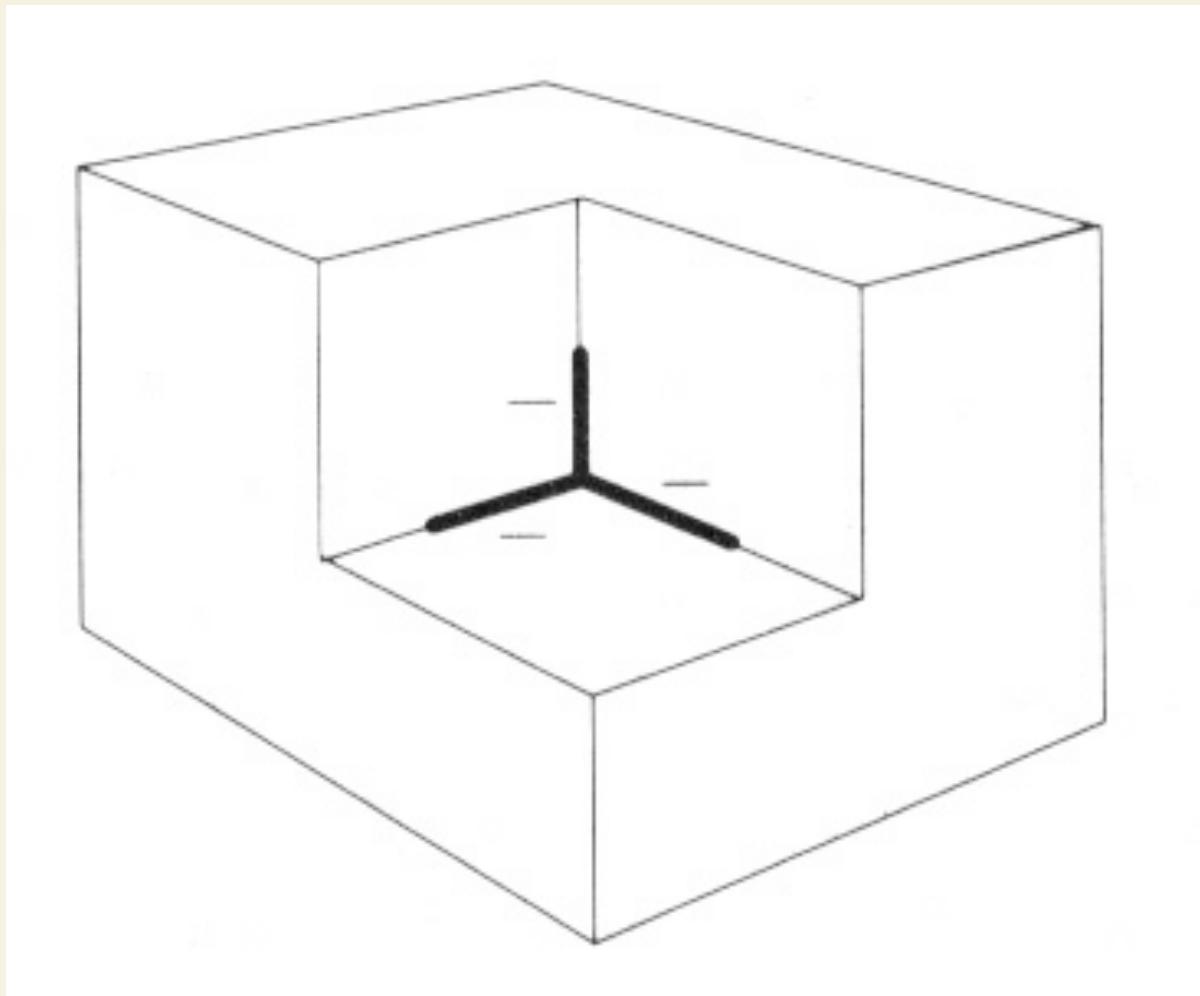
# 7 quadrants



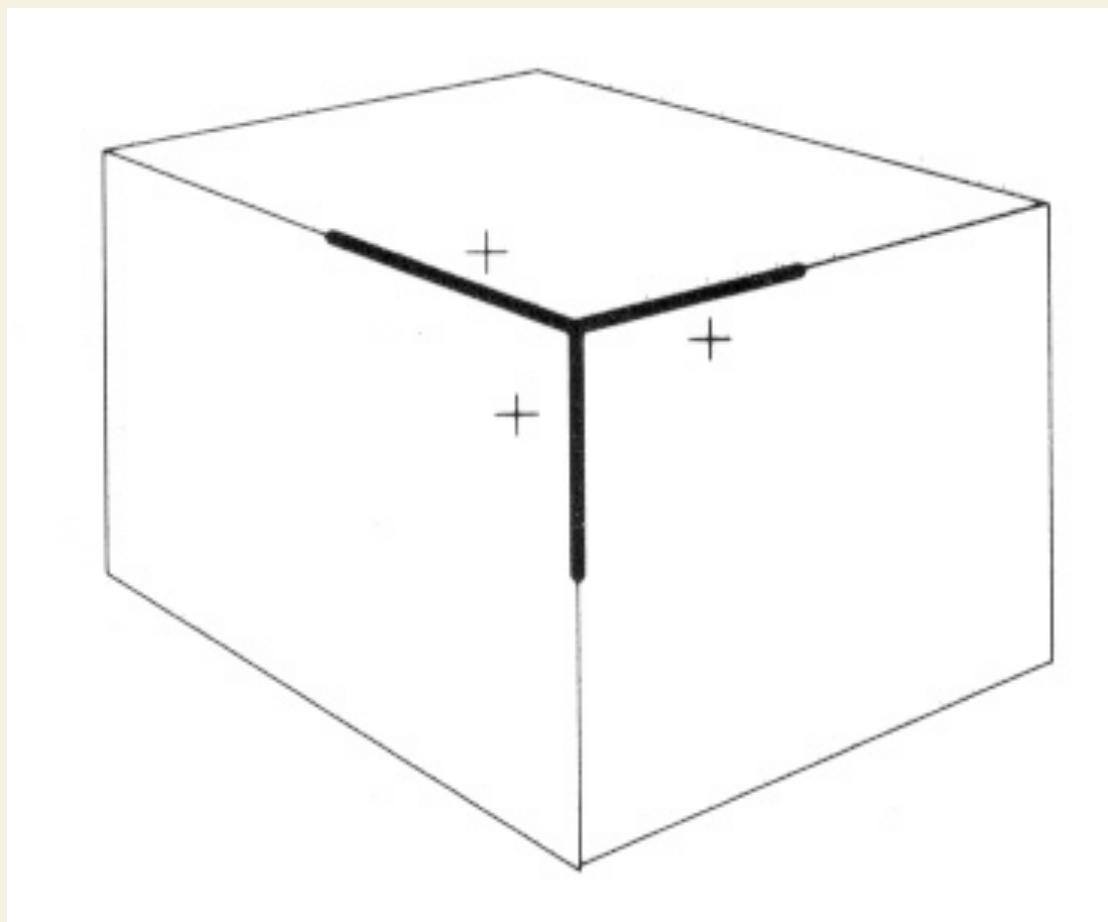
# 1 quadrant

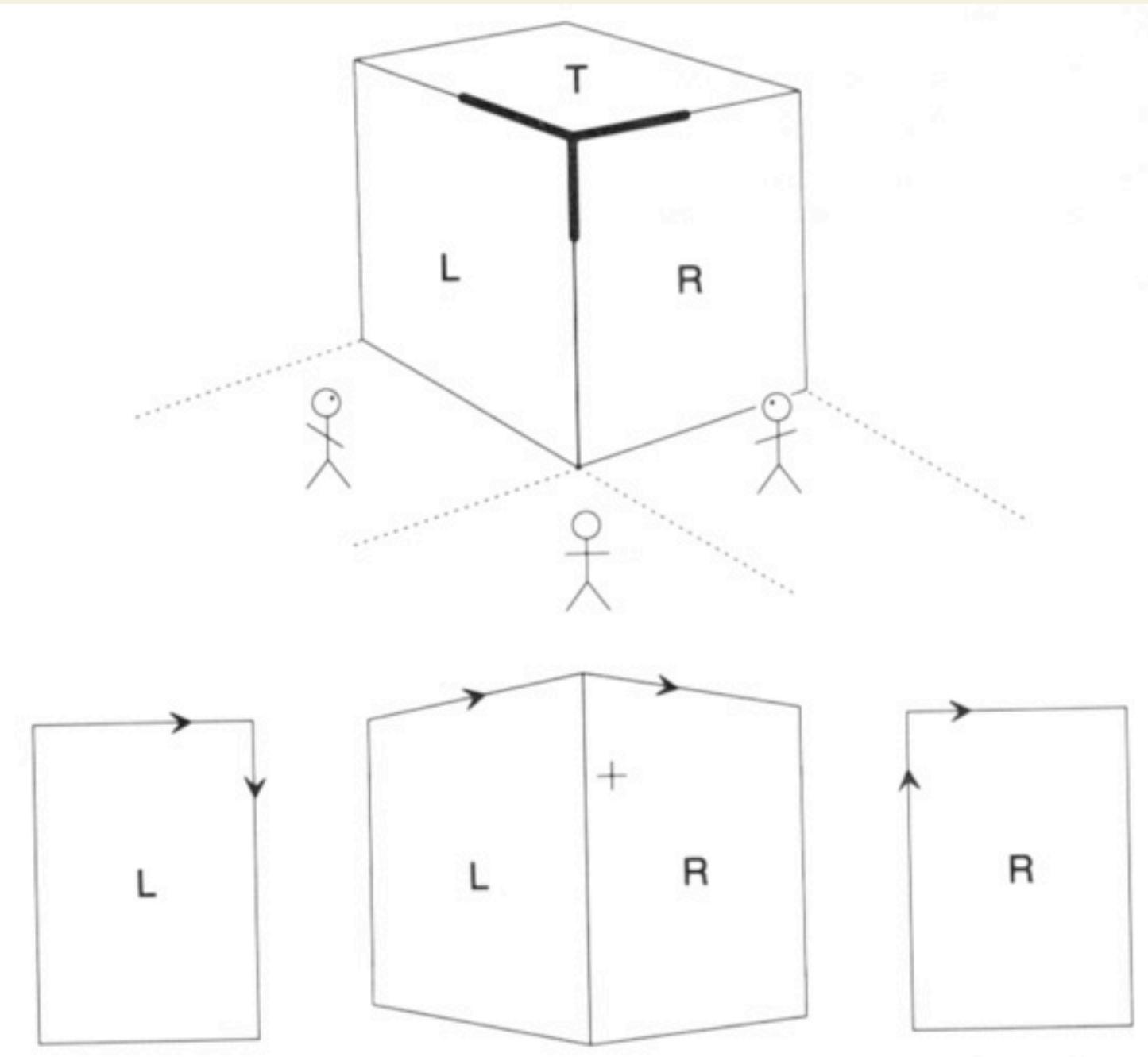


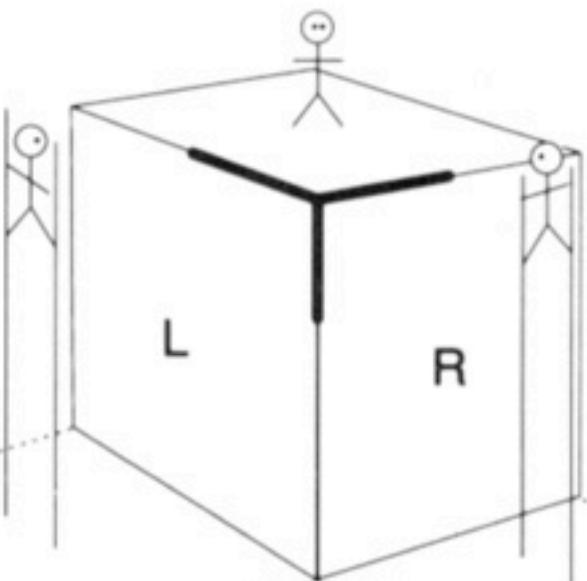
# 7 quadrants



# 1 quadrant

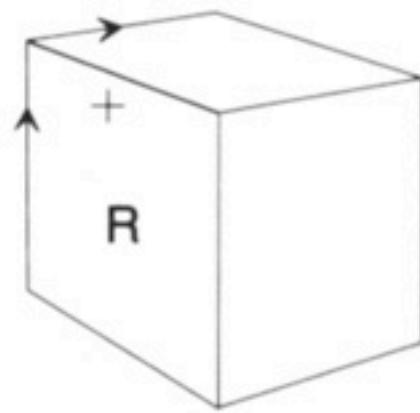
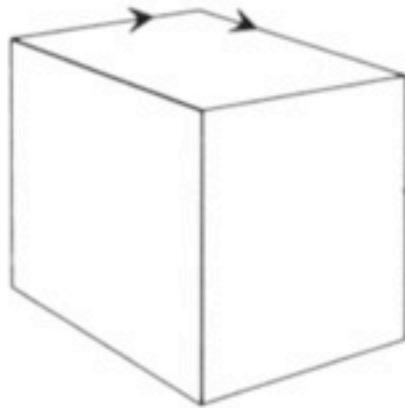
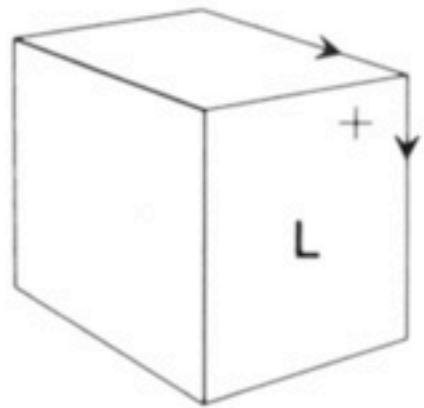




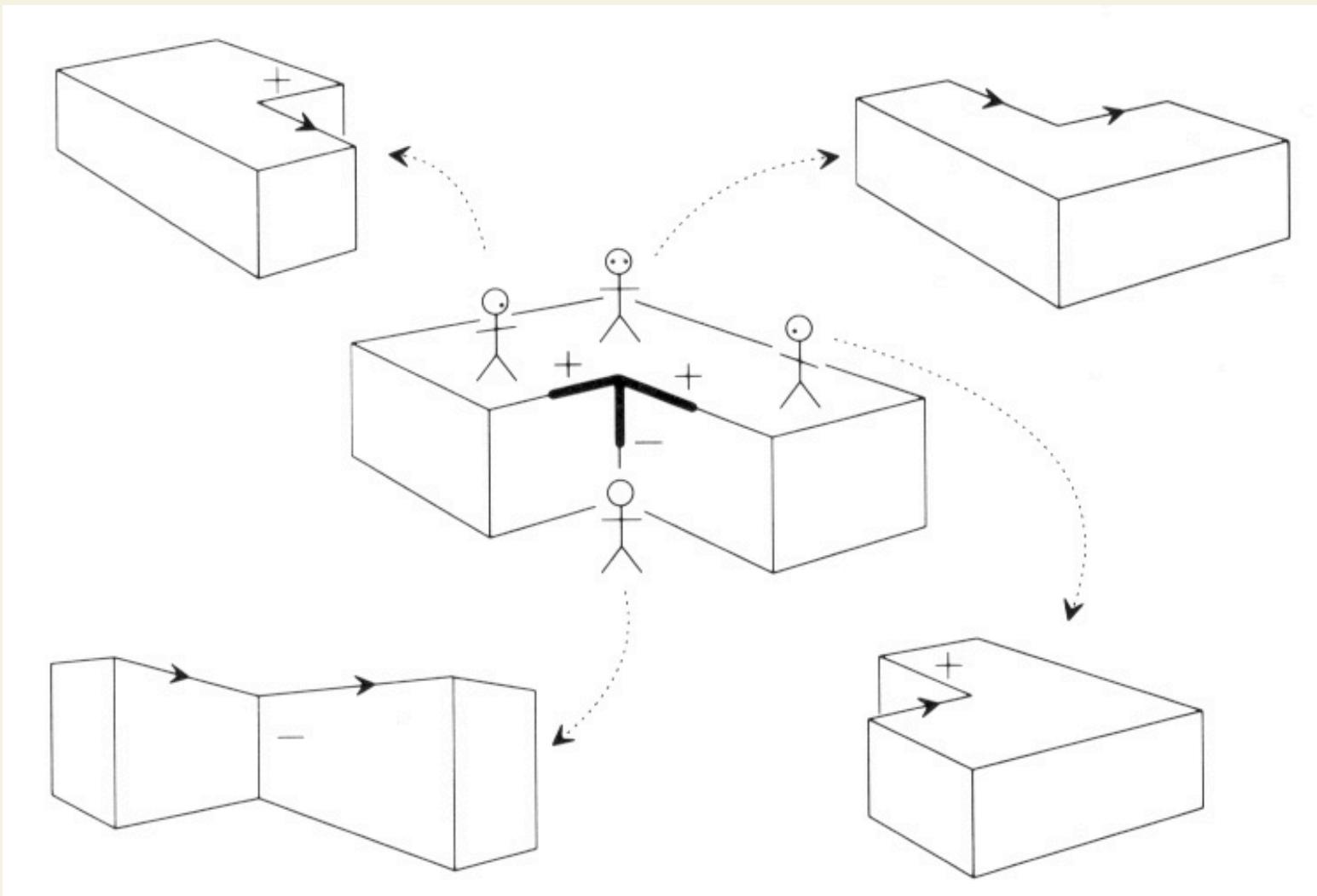


20  
10  
0  
-10  
-20

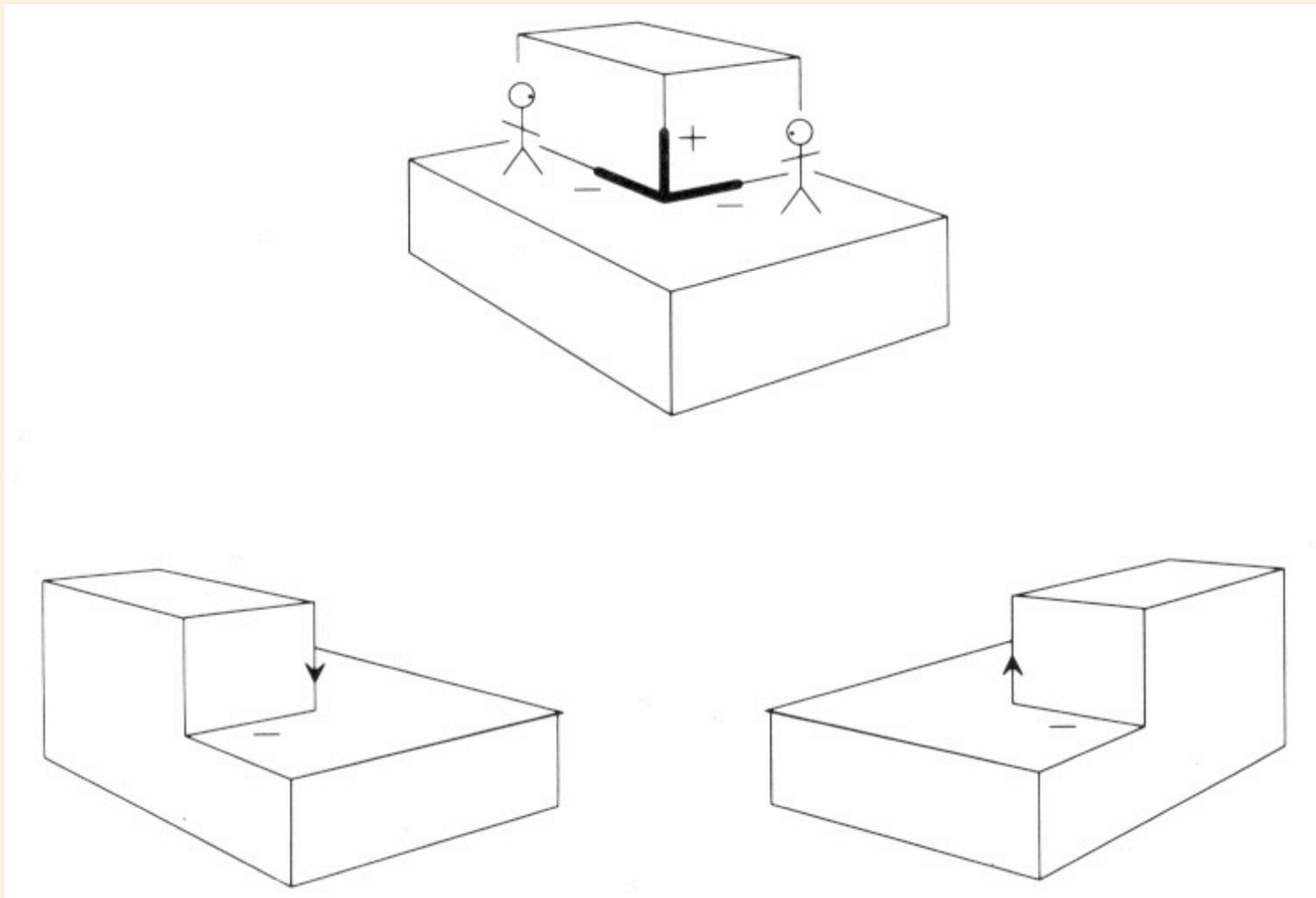
Y  
X  
Z  
W  
V  
U  
T  
S  
R  
Q  
P  
O  
N  
M  
L  
K  
J  
I  
H  
G  
F  
E  
D  
C  
B  
A



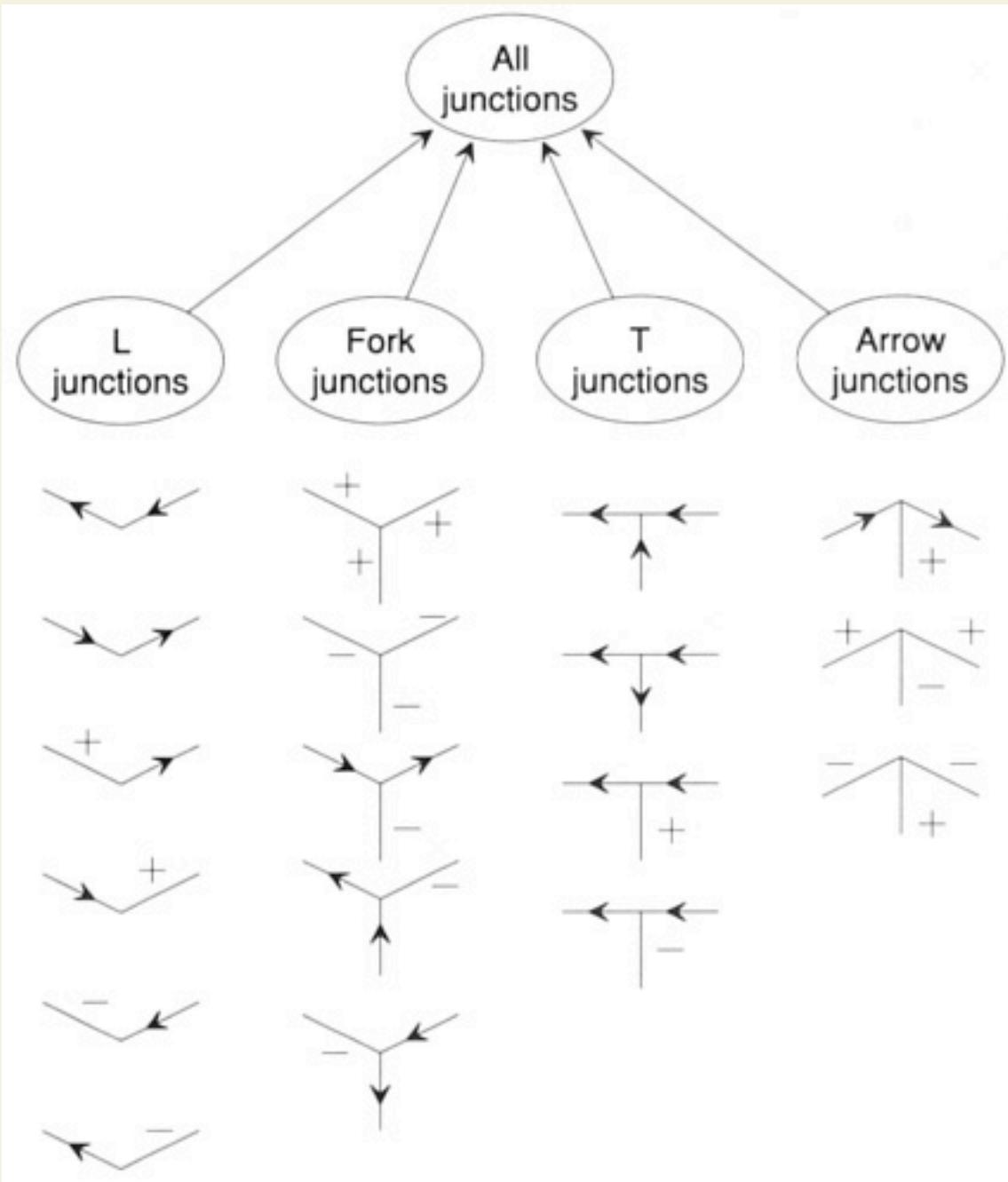
# 3 quadrants



# 5 quadrants



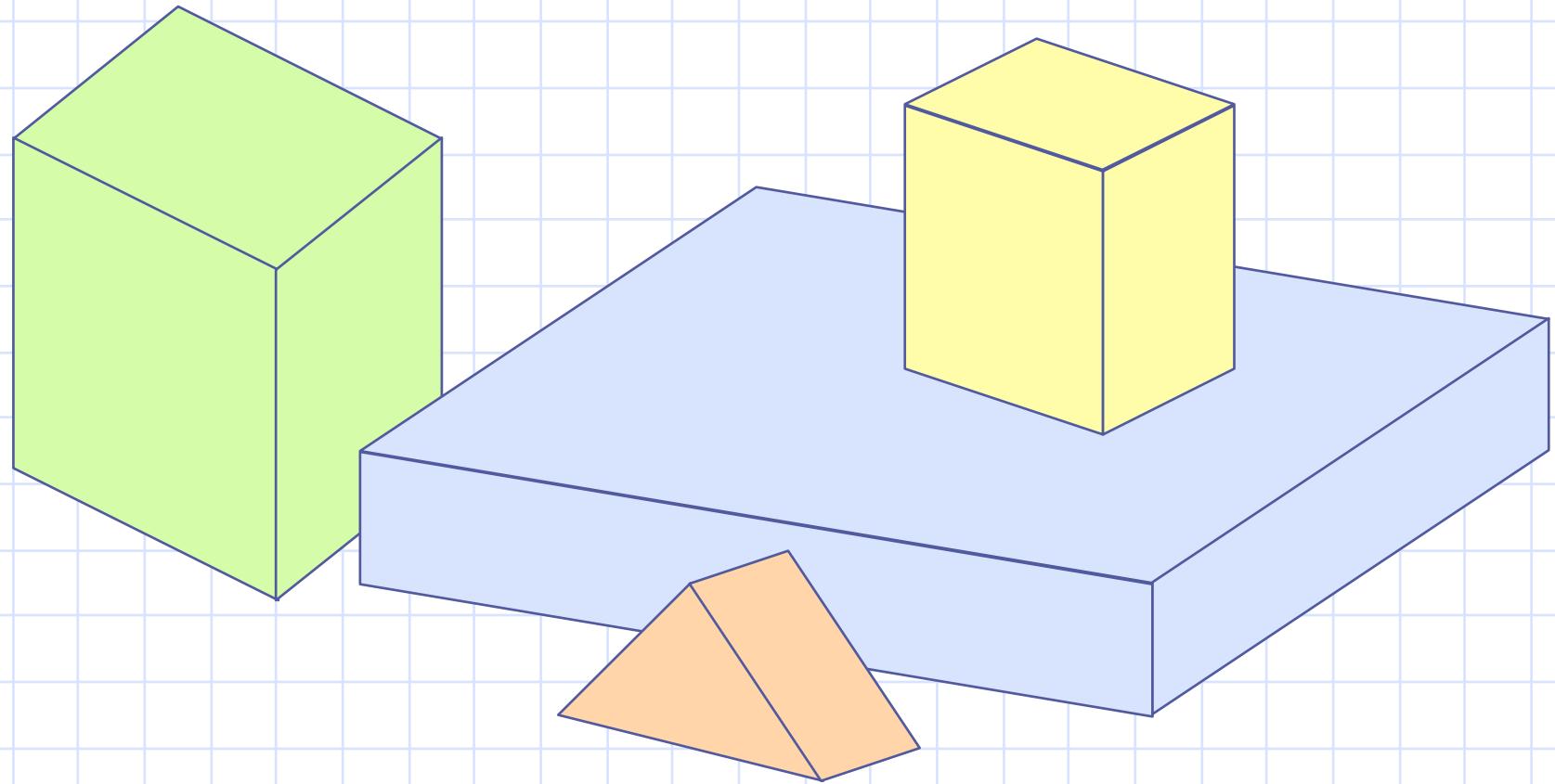
filling an even number of quadrants does not produce a three-faced junction.



# Edge Labeling in Computer Vision

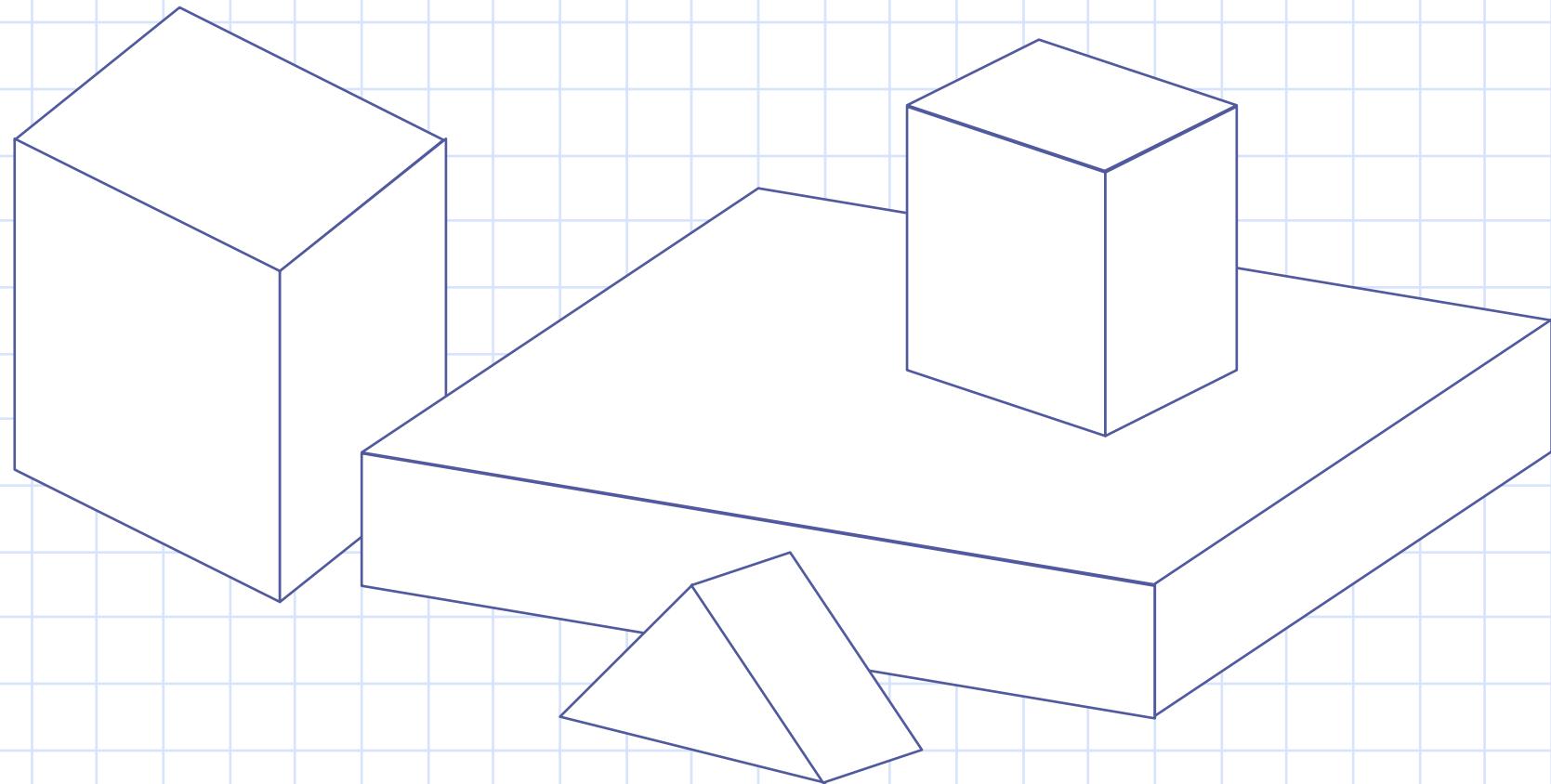
Russell and Norvig:  
Chapter 24, pages 745-749

# Edge Labeling

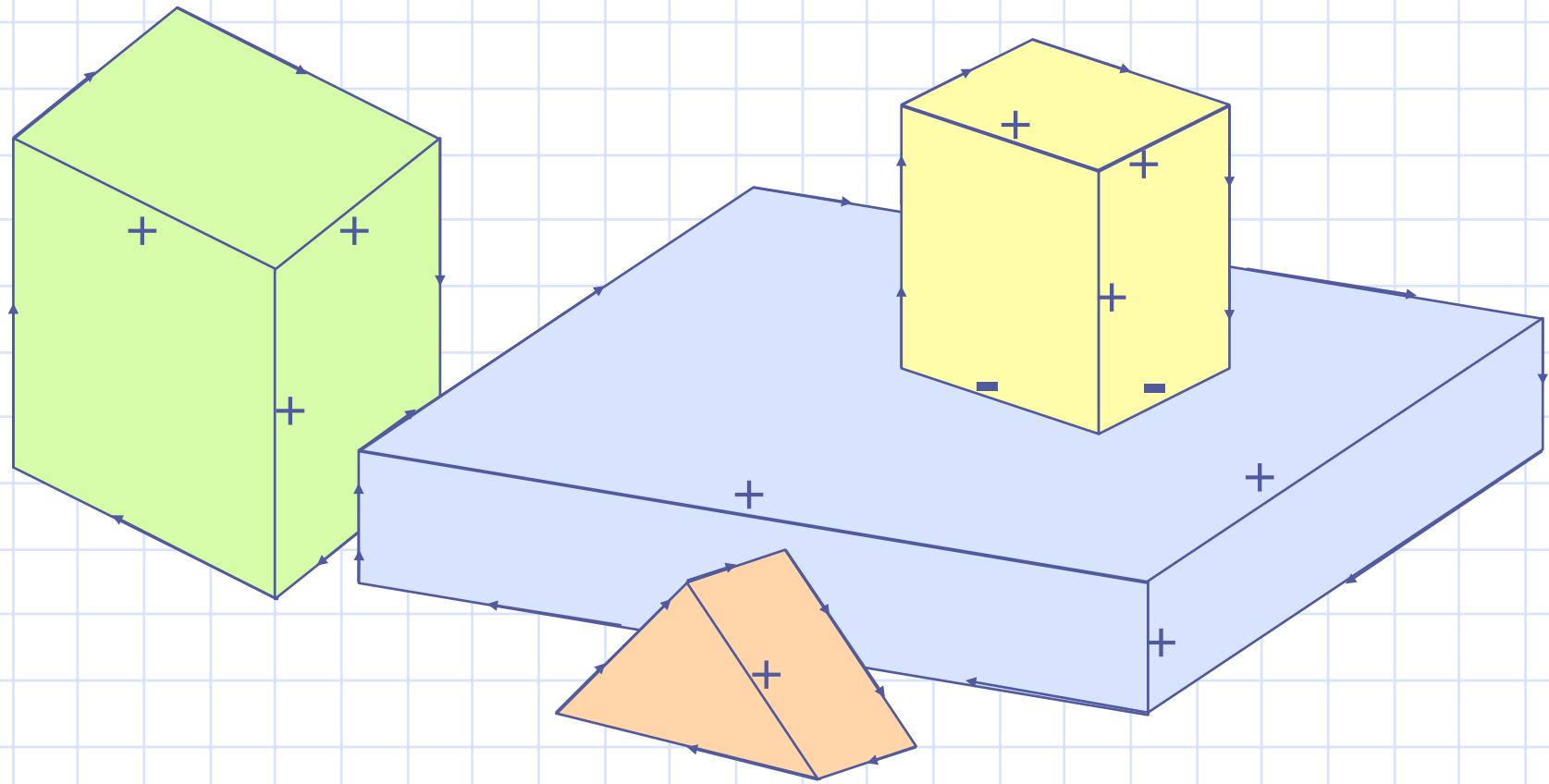


# Edge Labeling

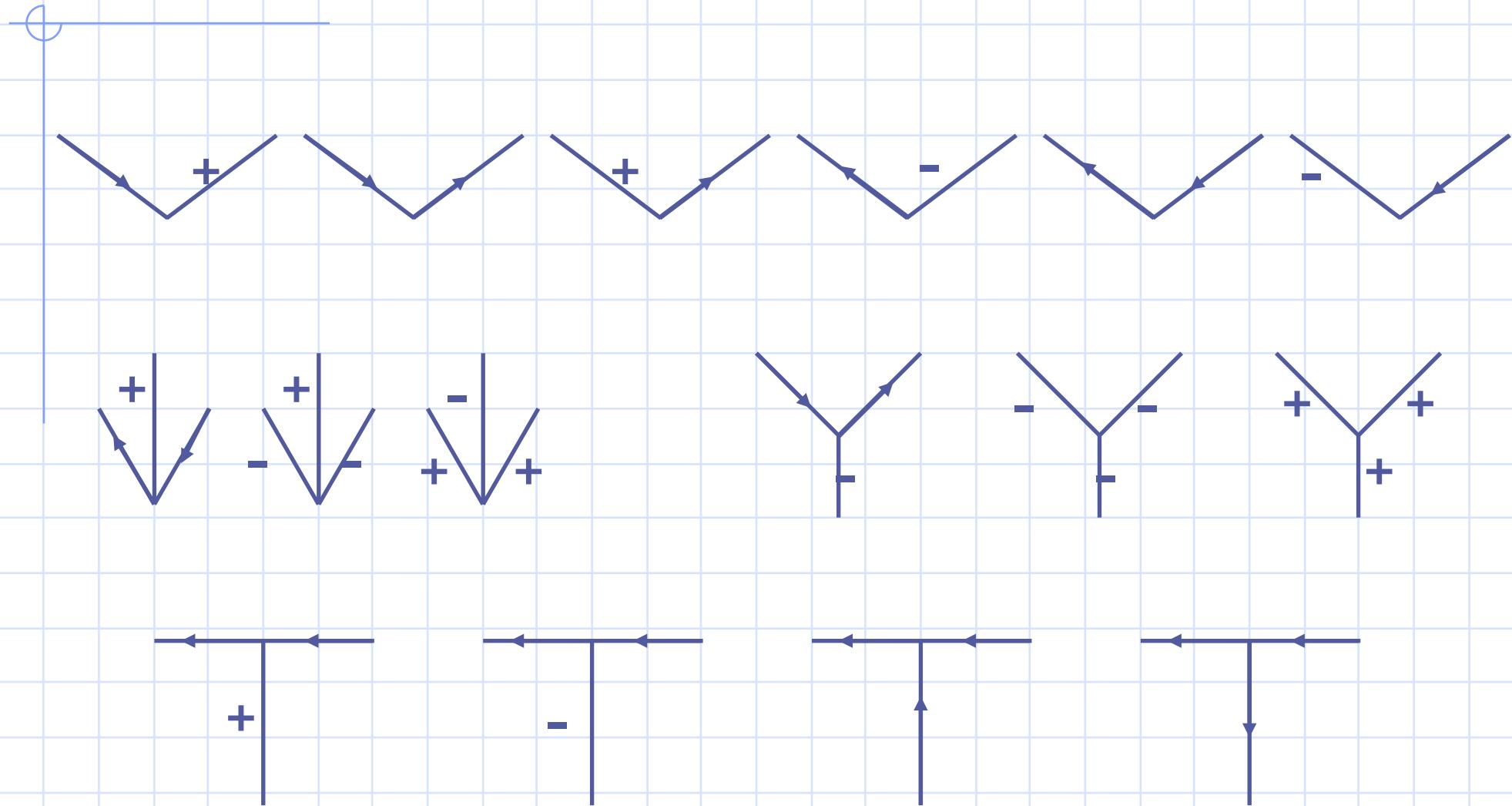
Diagram illustrating Edge Labeling:



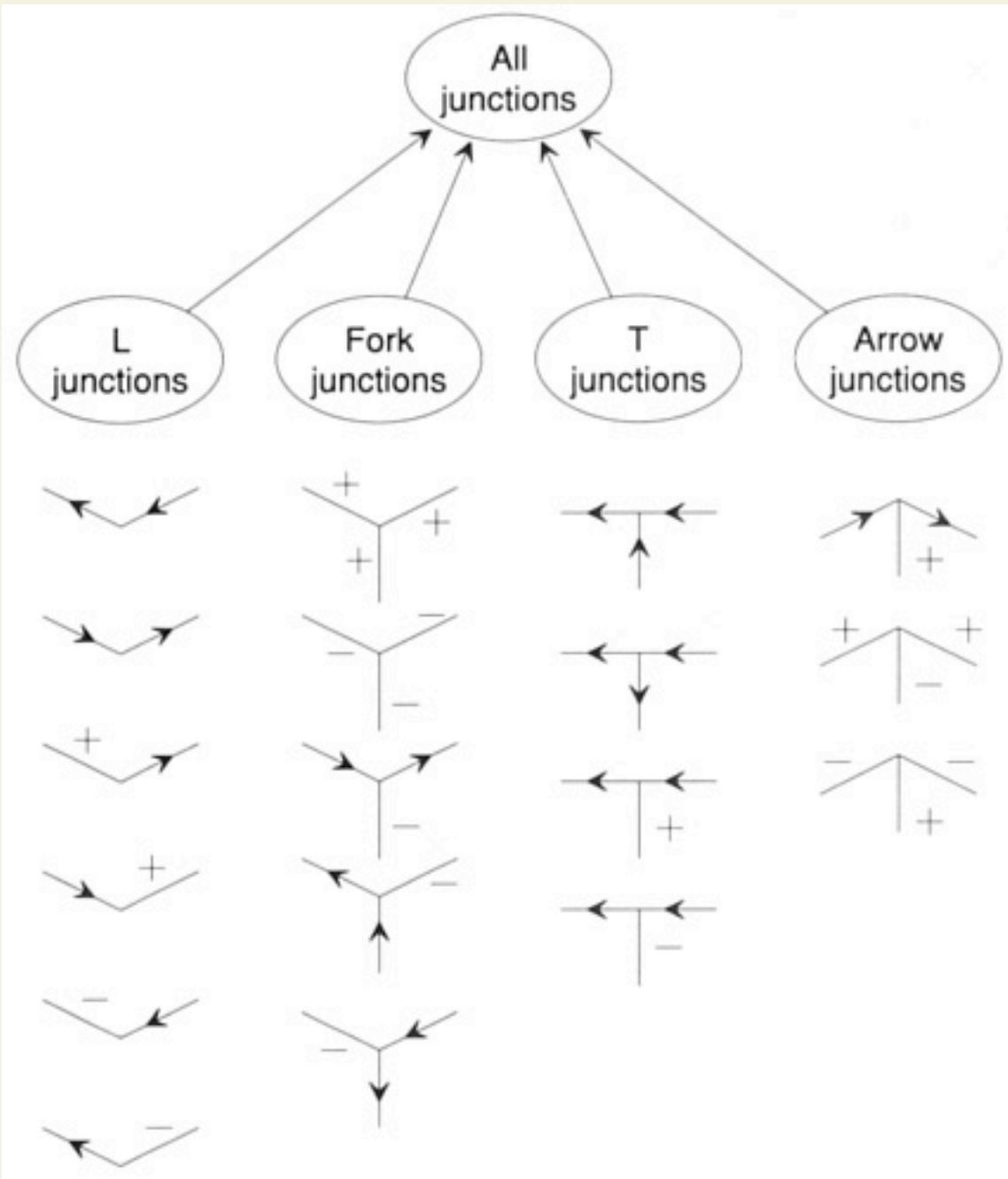
# Edge Labeling



# Junction Label Sets



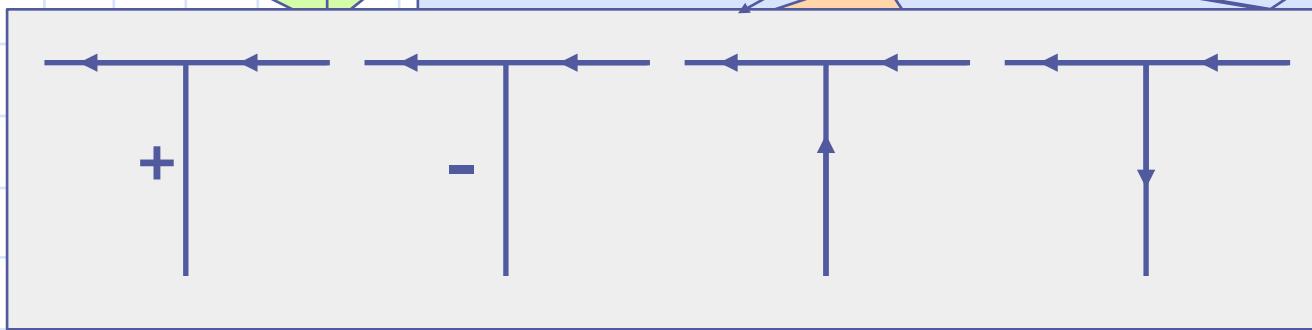
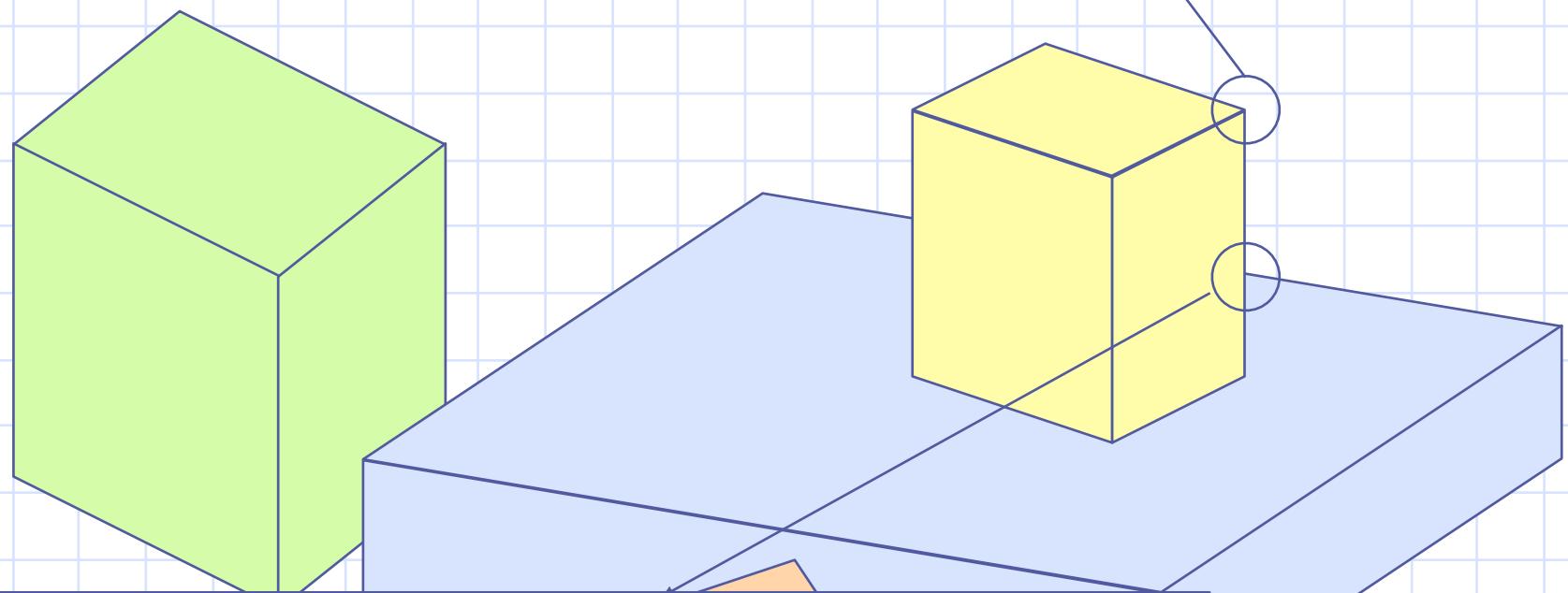
(Waltz, 1975; Mackworth, 1977)



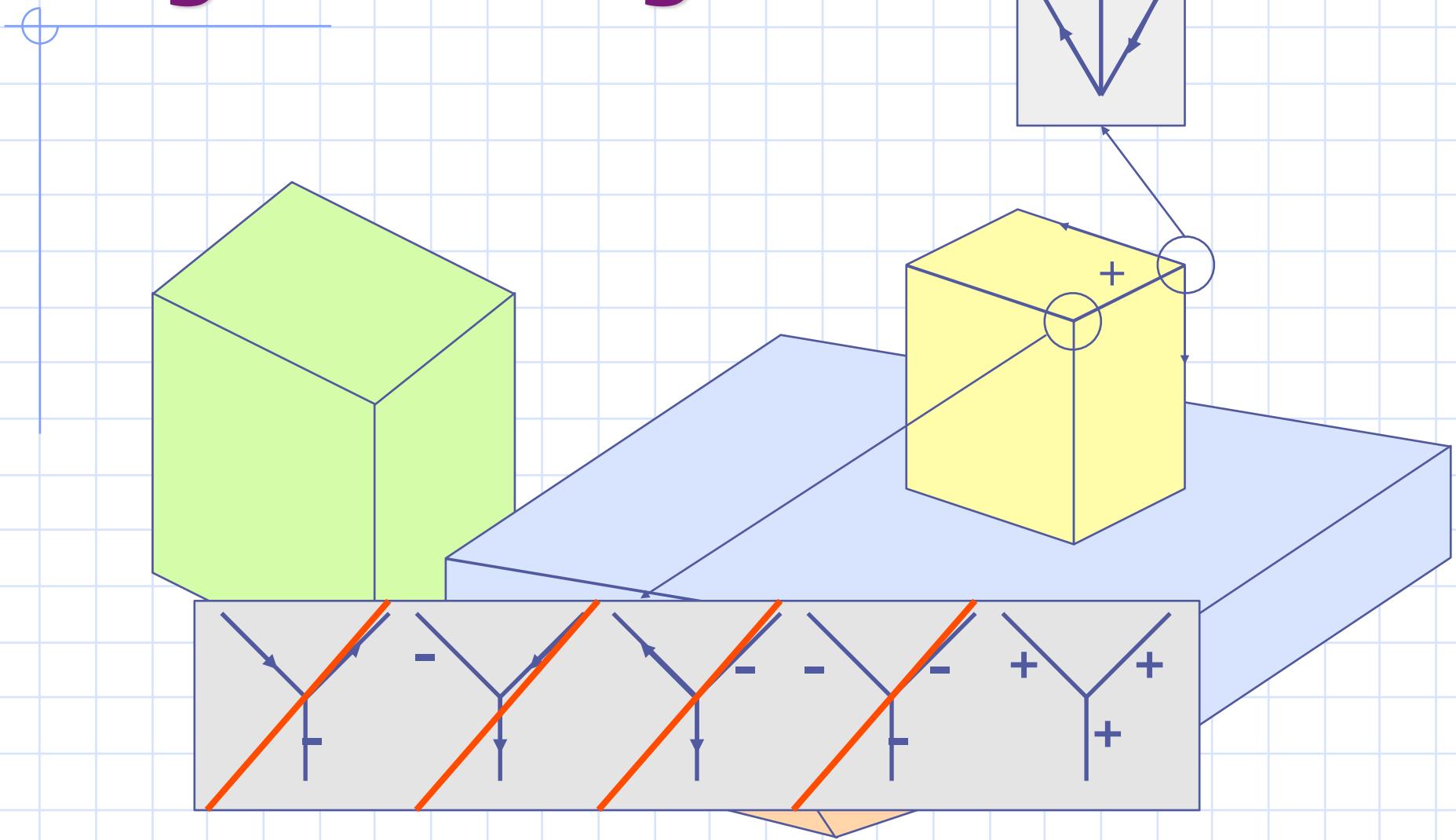
# Edge Labeling as a CSP

- ◆ A **variable** is associated with each junction
- ◆ The **domain** of a variable is the label set of the corresponding junction
- ◆ Each **constraint** imposes that the values given to two adjacent junctions give the same label to the joining edge

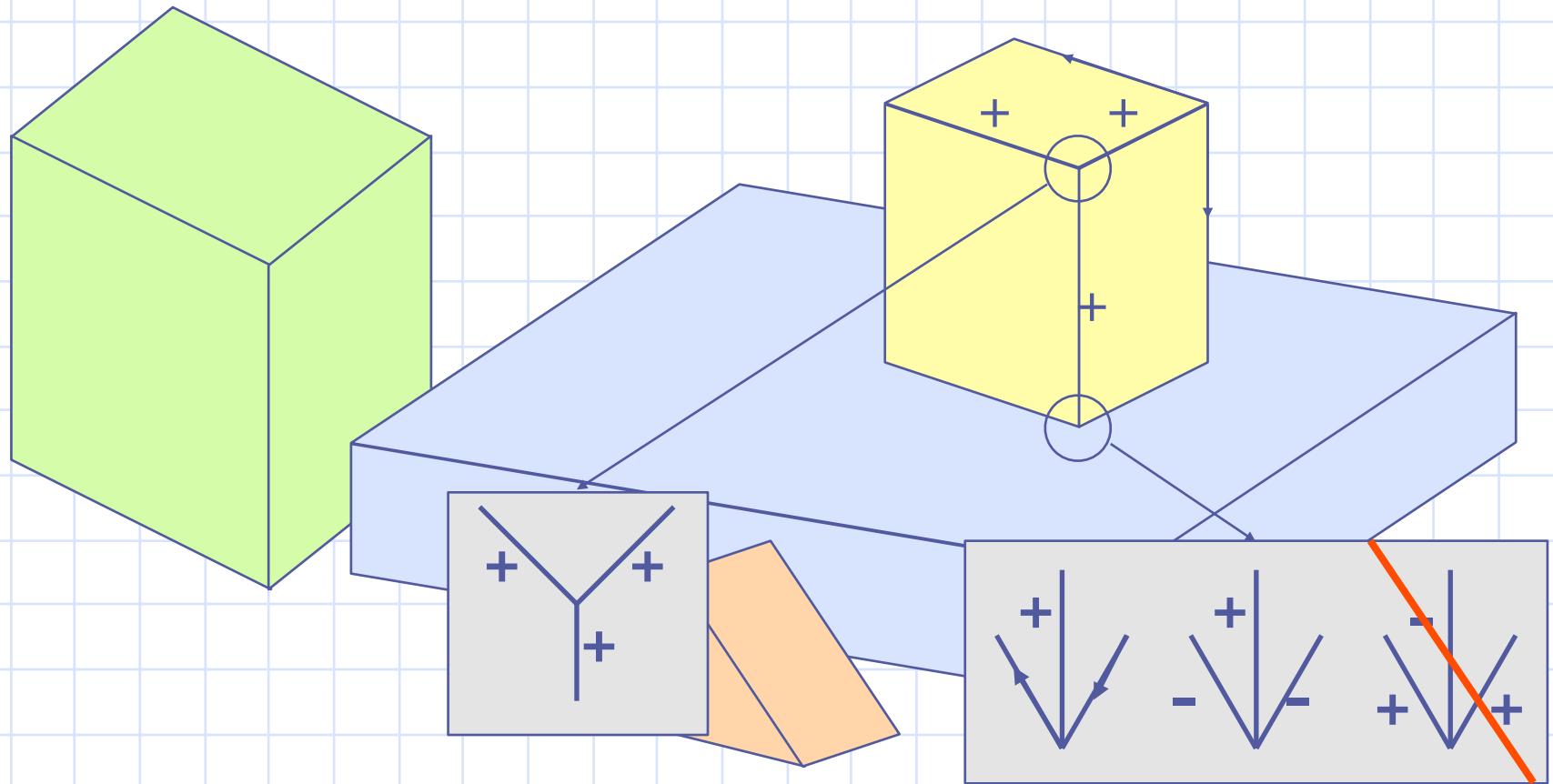
# Edge Labeling



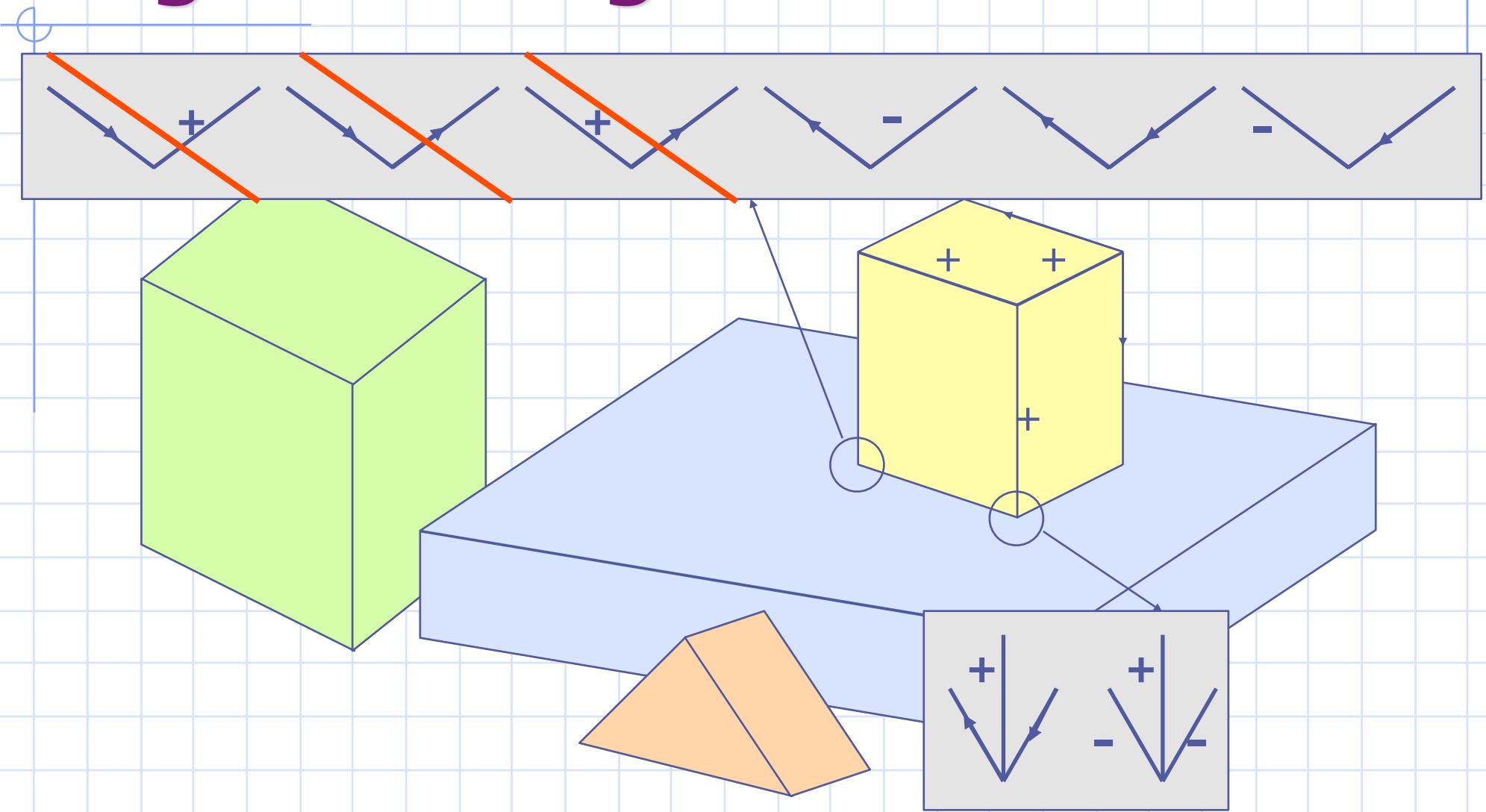
# Edge Labeling



# Edge Labeling



# Edge Labeling



# Removal of Arc Inconsistencies

REMOVE-ARC-INCONSISTENCIES( $J, K$ )

- ◆ removed  $\leftarrow$  false
- ◆  $X \leftarrow$  label set of  $J$
- ◆  $Y \leftarrow$  label set of  $K$
- ◆ For every label  $y$  in  $Y$  do
  - If there exists no label  $x$  in  $X$  such that the constraint  $(x, y)$  is satisfied then
    - ◆ Remove  $y$  from  $Y$
    - ◆ If  $Y$  is empty then contradiction  $\leftarrow$  true
    - ◆ removed  $\leftarrow$  true
- ◆ Label set of  $K \leftarrow Y$
- ◆ Return removed

# CP Algorithm for Edge Labeling

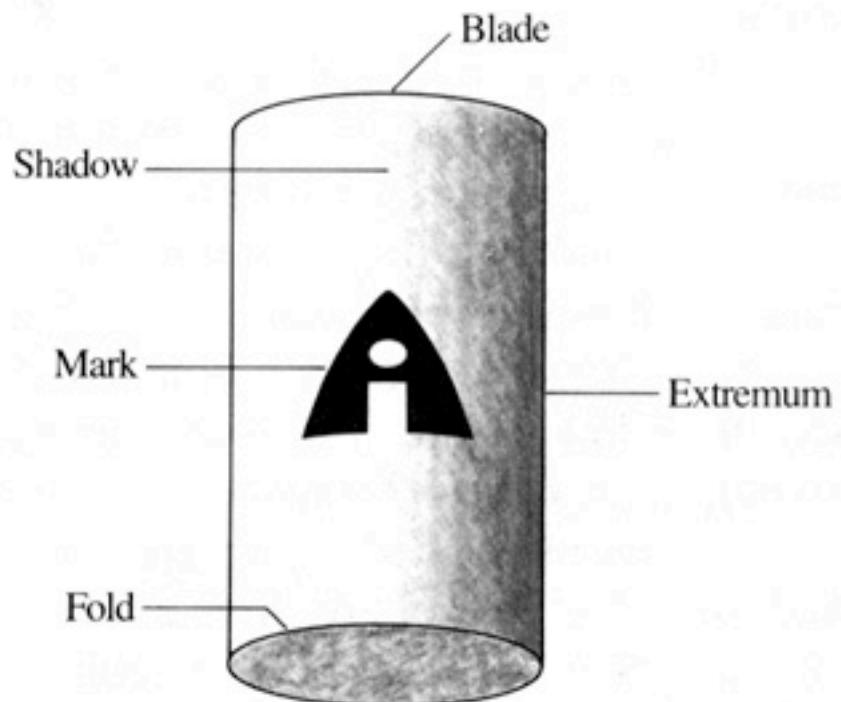
- ◆ Associate with every junction its label set
- ◆  $Q \leftarrow$  stack of all junctions
- ◆ while  $Q$  is not empty do
  - $J \leftarrow \text{UNSTACK}(Q)$
  - For every junction  $K$  adjacent to  $J$  do
    - ◆ If  $\text{REMOVE-ARC-INCONSISTENCIES}(J, K)$  then
      - If  $K$ 's domain is non-empty then  $\text{STACK}(K, Q)$
      - Else return false

(Waltz, 1975; Mackworth, 1977)

# Visual Clues

* Fold	two surfaces meet, both visible
Blade	two surfaces meet, just one visible
Extremum	self-occlusion by a curved object, like a cylinder or sphere
* Shadow	an abrupt change in illumination
* Mark	an abrupt change in reflectance or texture

\*Also a physical boundary type



# Visual Semantics

	Must Stay the same	May change
Fold	Depth	Orientation, reflectance, Illumination
Extremum	Nothing	Depth, illumination, orientation, reflectance
Blade	Nothing	Depth, illumination, orientation, reflectance
Shadow	Depth, orientation, reflectance	Illumination
Mark	Depth, orientation, illumination	Reflectance

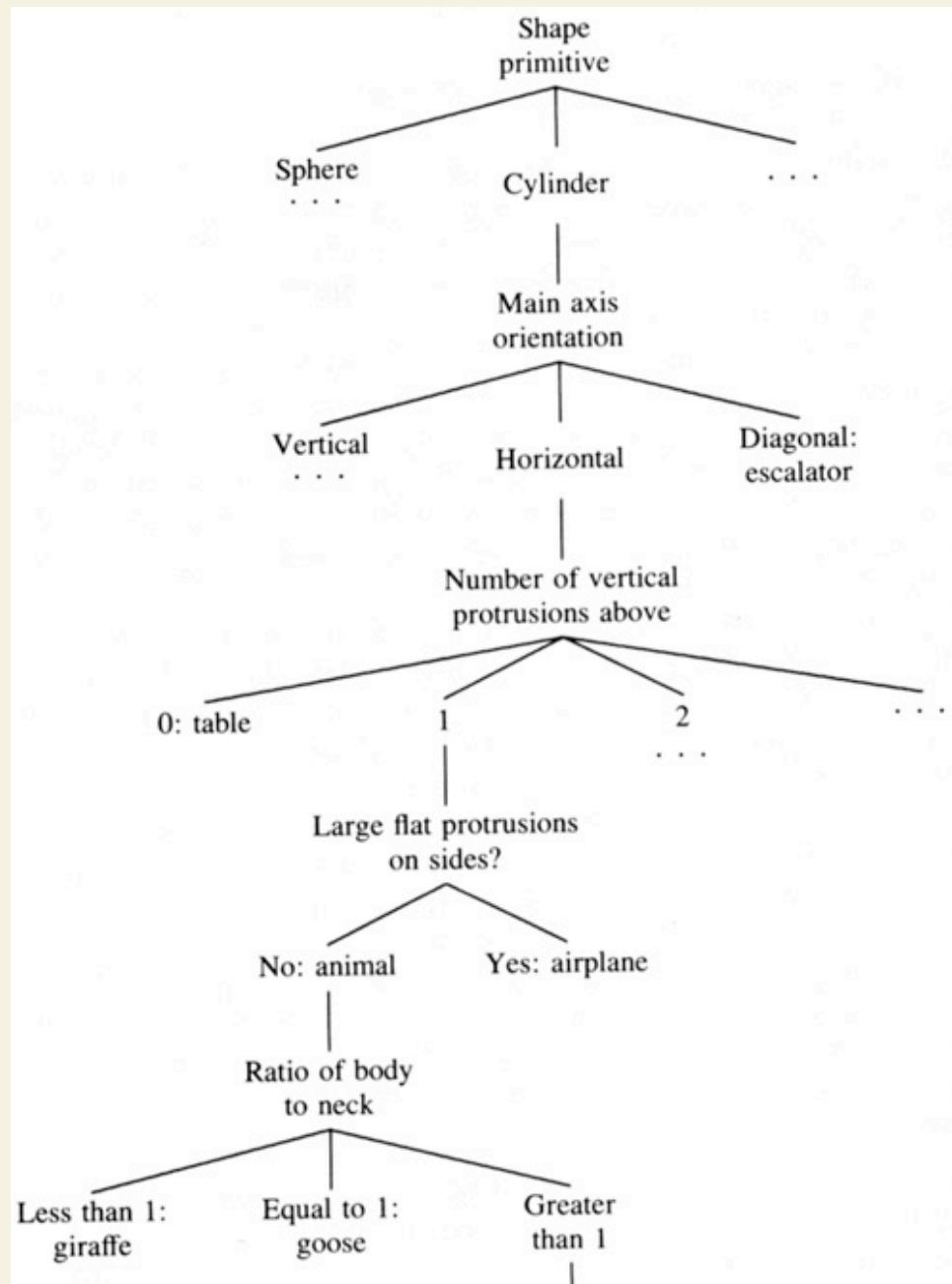
# frame representation

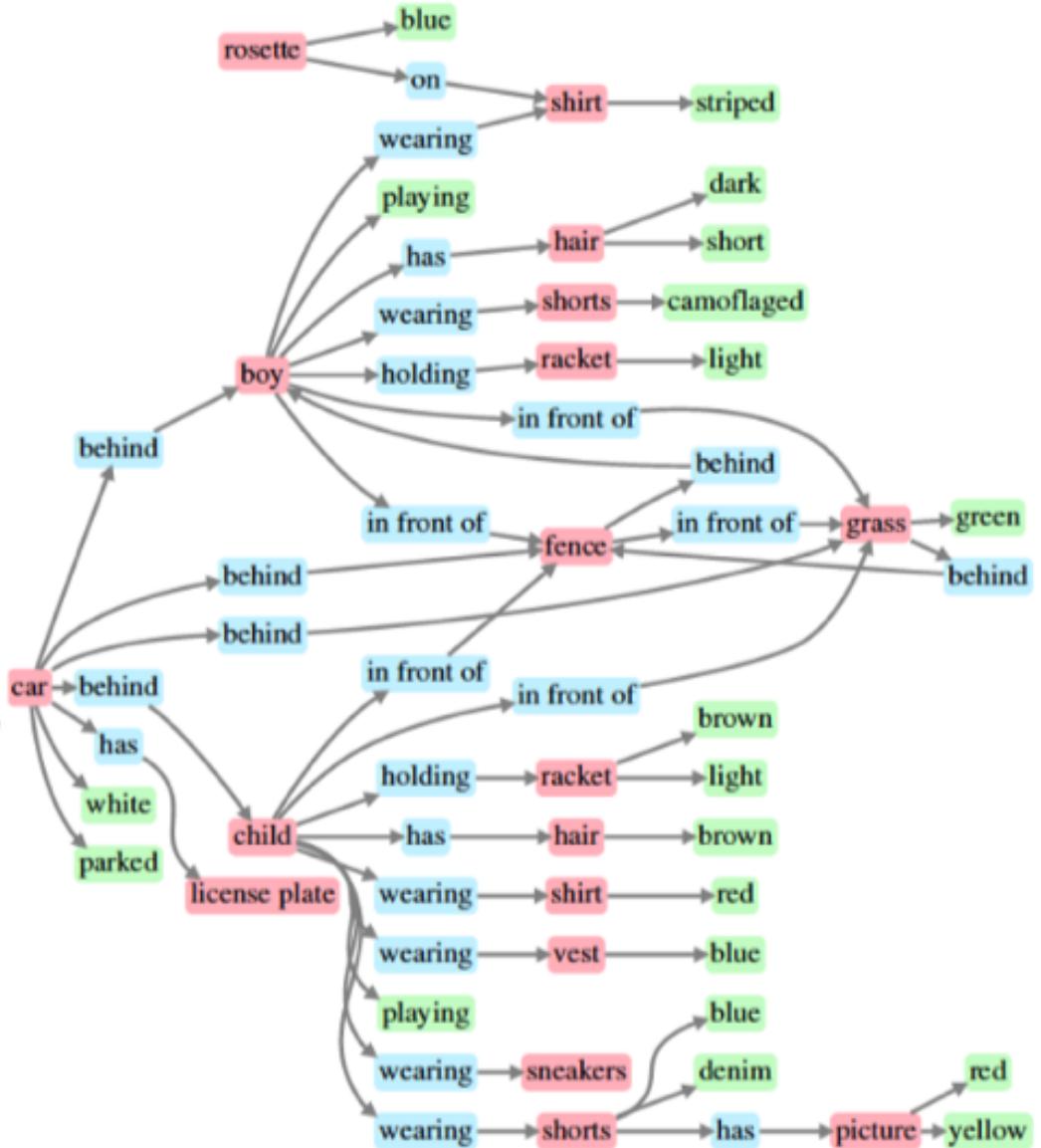
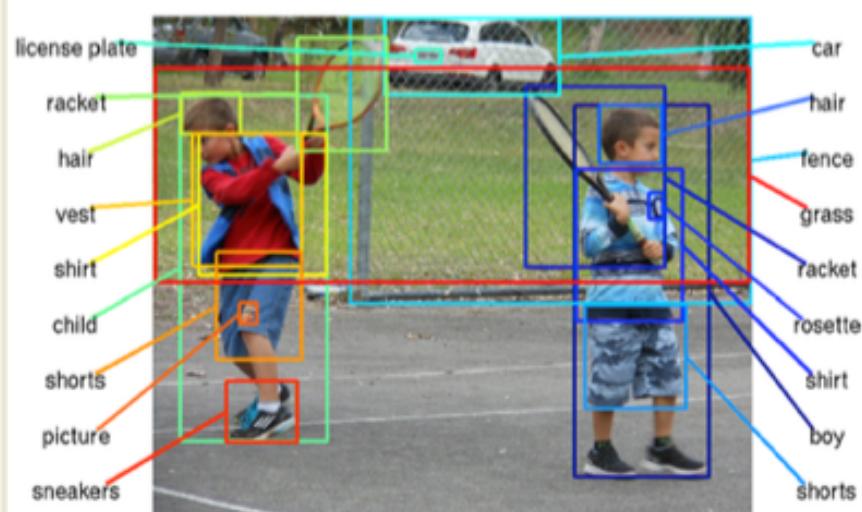
```
(sphere shape595
  (diameter . . . )
  ...
  (indentation (cylinder thumb-hole
    (length (3 cm))
    (width (2.5 cm))
    . . . )
    (location (0.1 0.1)))
  (indentation (cylinder ring-finger-hole
    (length (3 cm))
    (width (2.5 cm))
    . . . )
    (location (0 0)))
  (indentation (cylinder middle-finger-hole
    (length (3 cm))
    (width (2.5 cm))
    . . . )
    (location (-0.1 -0.1))))
```

# frame representation



```
(sphere shape595  
  (diameter . . . )  
  . . .  
  (indentation (cylinder thumb-hole  
    (length (3 cm))  
    (width (2.5 cm))  
    . . . ))  
  (location (0.1 0.1)))  
 (indentation (cylinder ring-finger-hole  
    (length (3 cm))  
    (width (2.5 cm))  
    . . . ))  
  (location (0 0)))  
 (indentation (cylinder middle-finger-hole  
    (length (3 cm))  
    (width (2.5 cm))  
    . . . ))  
  (location (-0.1 -0.1))))
```





Johnson *et al.*, “Image Retrieval using Scene Graphs”, CVPR 2015

Figures copyright IEEE, 2015. Reproduced for educational purposes

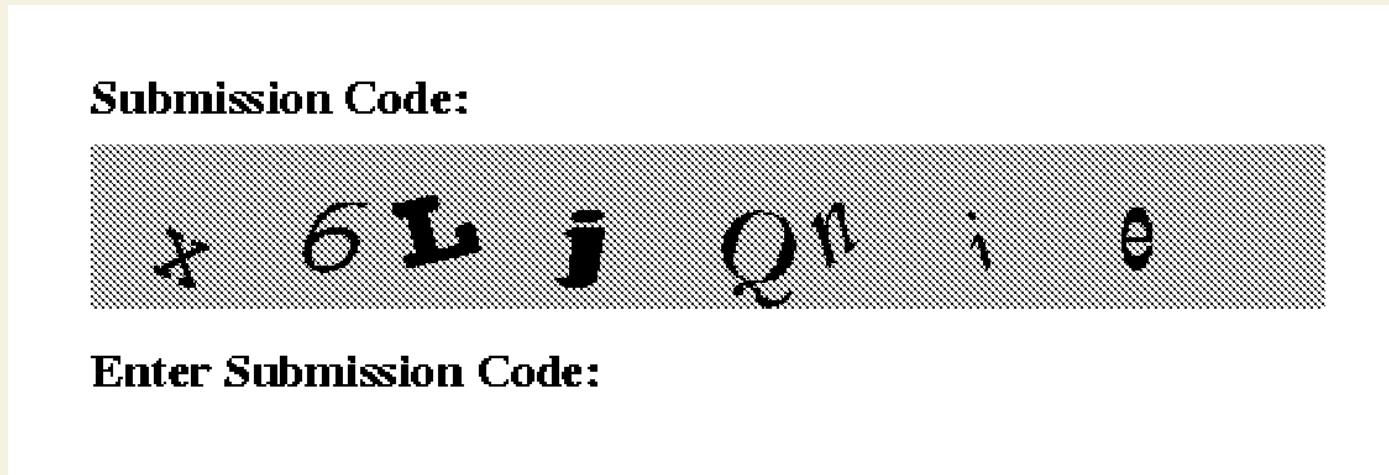
# CAPTCHAs

- ~ Completely Automated Public Turing Tests to tell Computers and Humans Apart.
- ~ Challenges generated and graded by computer
- ~ humans have no problems passing
- ~ rejects virtually all machines - resists automated attacks

# Used to prevent

- ~ skewing search engine rankings
- ~ monitoring chat rooms
- ~ access to financial accounts
- ~ comment spamming blogs
- ~ automating free account creation

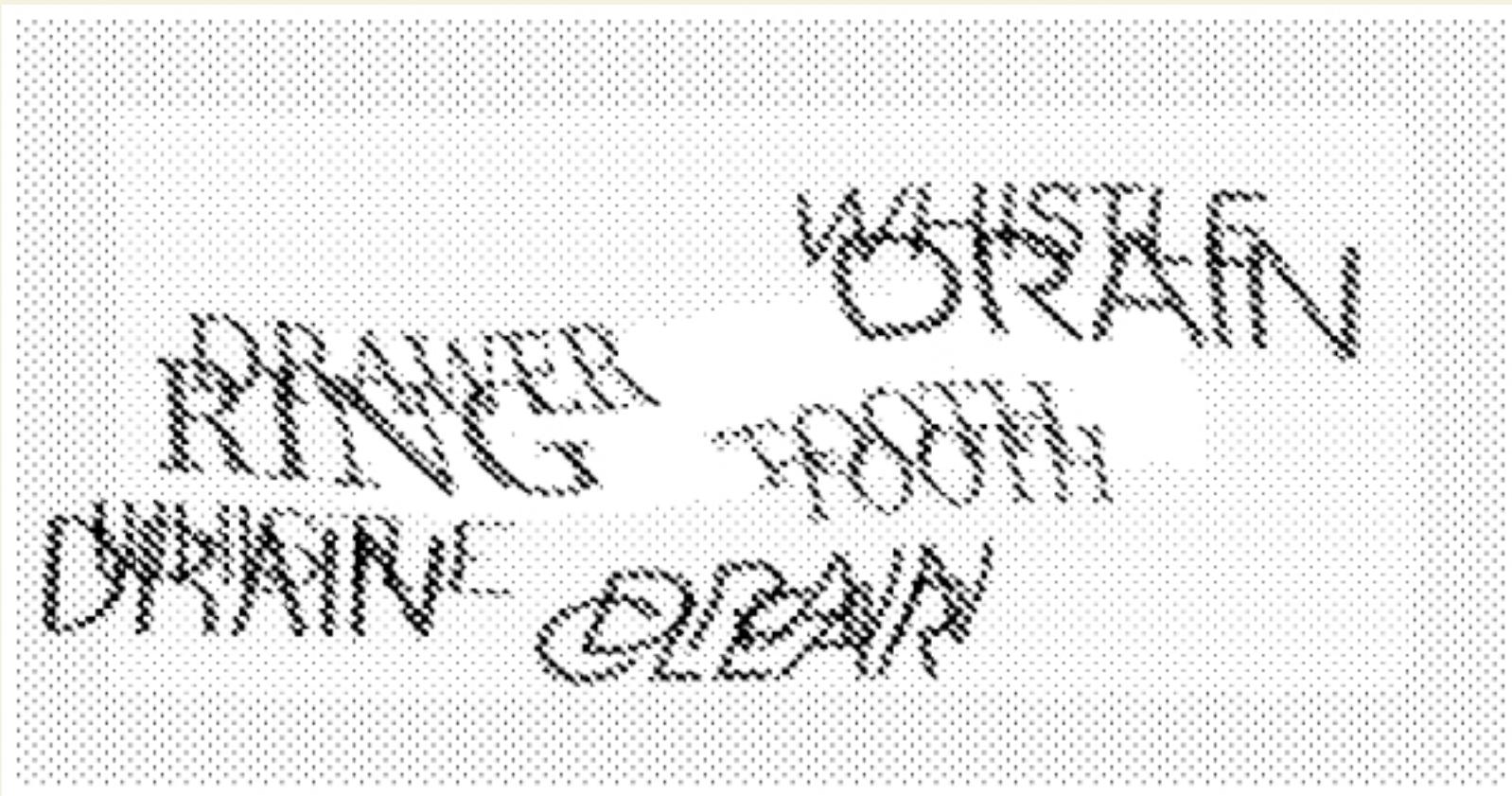
# Altavista's AddURL Filter



In 1999 reduced spam by over 95%

M. D. Lillibridge, M. Abadi, K. Bharat, & A. Z. Broder, "Method for Selectively Restricting Access to Computer Systems," U.S. Patent No. 6,195,698, Filed April 13, 1998, Issued February 27, 2001.

# CMU's CAPTCHA



English words, deformations, occlusions, etc.

User types any three words. Tried by Yahoo! ppl hated it.

L. Von Ahn, M. Blum, N. J. Hopper, J. Langford, The CAPTCHA Web Page, <http://www.captcha.net>.

# PayPal's CAPTCHA

As a security measure, please enter the characters you see in the box on the right into the box on the left. (The characters are not case sensitive.) Help?



# Microsoft's CAPTCHA



Random strings, space-warping,  
meaningless strokes added and subtracted

Used to during Hotmail account registration

# Blogger's CAPTCHA



No use of dictionary to generate words

# AI Hard

- ~ large ability gap between human and machine vision systems
- ~ 7 year old children consistently outperform OCR machines

# Image Degradation



Blurring  
Thresholding  
Sensitivity

Can mix degradation types together

# Machines can't manage:

**Artistic**

**Human**

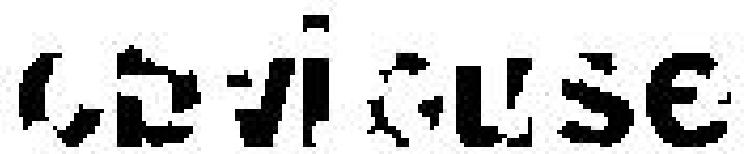
**corner**

**conscious**

**emotional**

# BaffleText

- ~ Nonsense words which are pronounceable - look familiar.
- ~ Use occlusion to force inference of whole word image



A blurred, illegible word composed of black dots, representing a baffle text.

# Masking

BaffleText type	word image	mask image
	<b>kanies</b>	
addition	<b>kanies.</b>	
subtraction	<b>kanies</b>	
xor	<b>kanies.</b>	

# Discuss!

# Image Classification



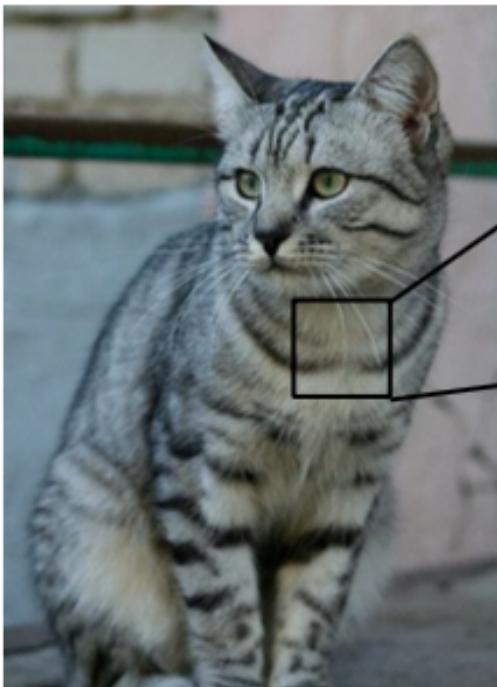
This image by Nikita is  
licensed under CC-BY 2.0

(assume given set of discrete labels)  
{dog, cat, truck, plane, ...}



cat

# The Problem: Semantic Gap



This image by Nikita is  
licensed under CC-BY 2.0

```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 100 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 183 65 81 88 65 52 54 74 84 102 93 85 82]
 [128 137 144 148 189 95 86 70 62 65 63 63 60 73 86 181]
 [125 133 148 137 119 121 117 94 65 79 88 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 98 97 100 147 131 118 113 114 113 109 106 95 77 88]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
 [ 62 65 82 89 78 71 88 181 124 126 119 187 114 131 119]
 [ 63 65 75 88 89 71 62 81 120 130 135 105 81 98 118 118]
 [ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 88 82 120 124 184 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 88 107 112 99]
 [122 121 102 88 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

What the computer sees

An image is just a big grid of  
numbers between [0, 255]:

e.g. 800 x 600 x 3  
(3 channels RGB)

image: <http://cs231n.stanford.edu/>

## **Challenges:** Viewpoint variation

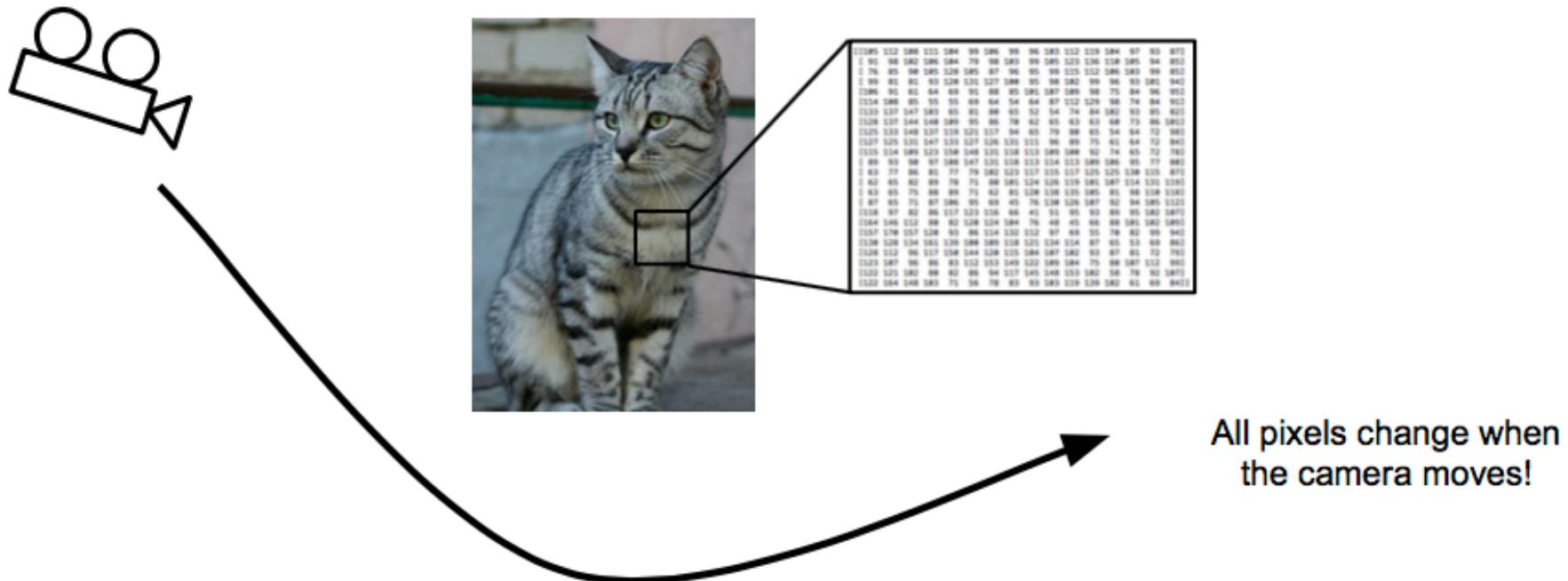
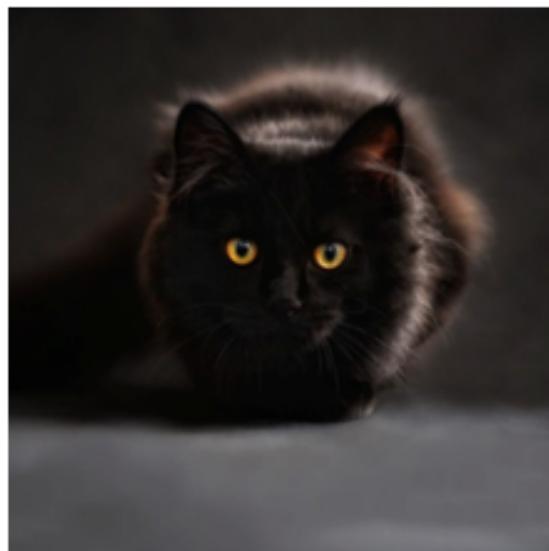


image: <http://cs231n.stanford.edu/>

# Challenges: Illumination



[This image is CC0 1.0 public domain](#)



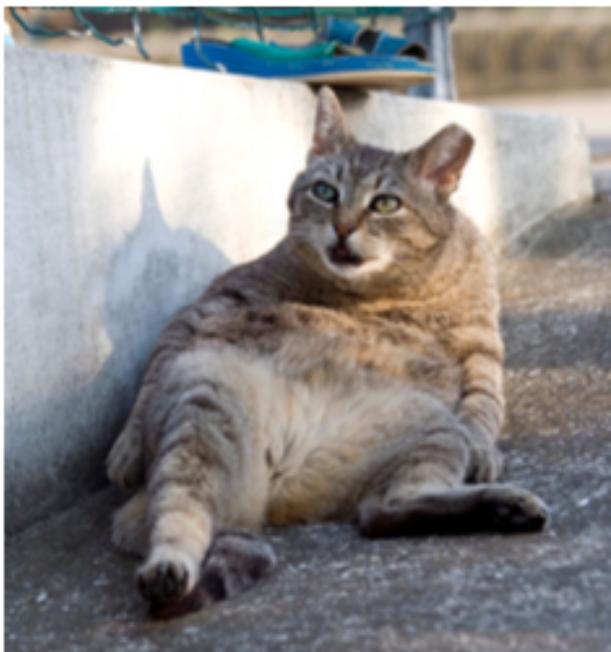
[This image is CC0 1.0 public domain](#)



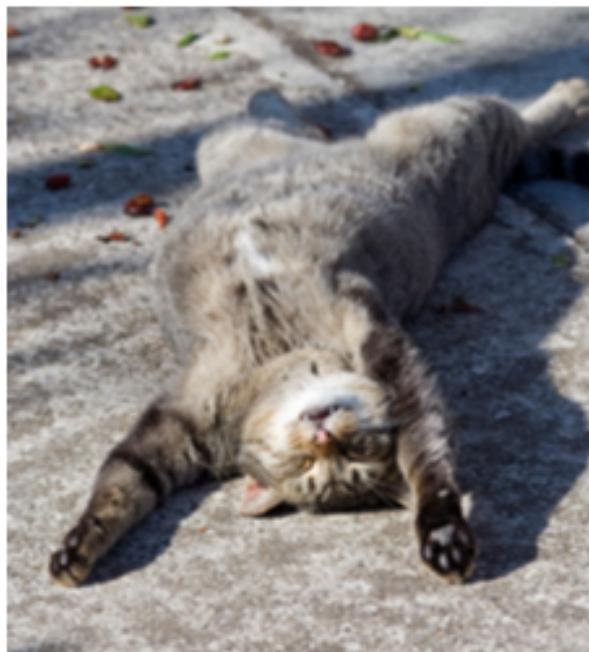
[This image is CC0 1.0 public domain](#)

image: <http://cs231n.stanford.edu/>

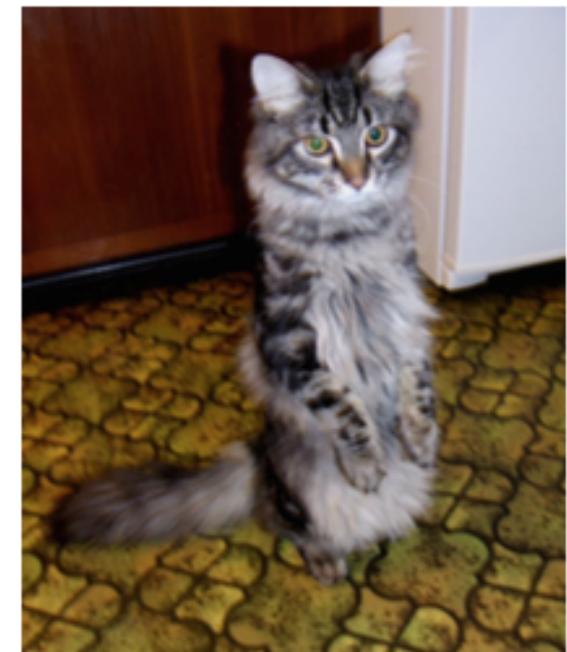
# Challenges: Deformation



[This image](#) by [Umberto Salvagnin](#)  
is licensed under [CC-BY 2.0](#)



[This image](#) by [Umberto Salvagnin](#)  
is licensed under [CC-BY 2.0](#)



[This image](#) by [sare\\_bear](#) is  
licensed under [CC-BY 2.0](#)

image: <http://cs231n.stanford.edu/>

## Challenges: Occlusion



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image by Jonsson is licensed under CC-BY 2.0](#)

image: <http://cs231n.stanford.edu/>

## Challenges: Background Clutter



[This image](#) is CC0 1.0 public domain



[This image](#) is CC0 1.0 public domain

image: <http://cs231n.stanford.edu/>

## Challenges: Background Clutter



[This image](#) is CC0 1.0 public domain



[This image](#) is CC0 1.0 public domain

image: <http://cs231n.stanford.edu/>

## Challenges: Intraclass variation



This image is CC0 1.0 public domain

image: <http://cs231n.stanford.edu/>

# Dog or Bagel?



# Dog or Mop?



# Dog or Muffin?



# Think this will work?

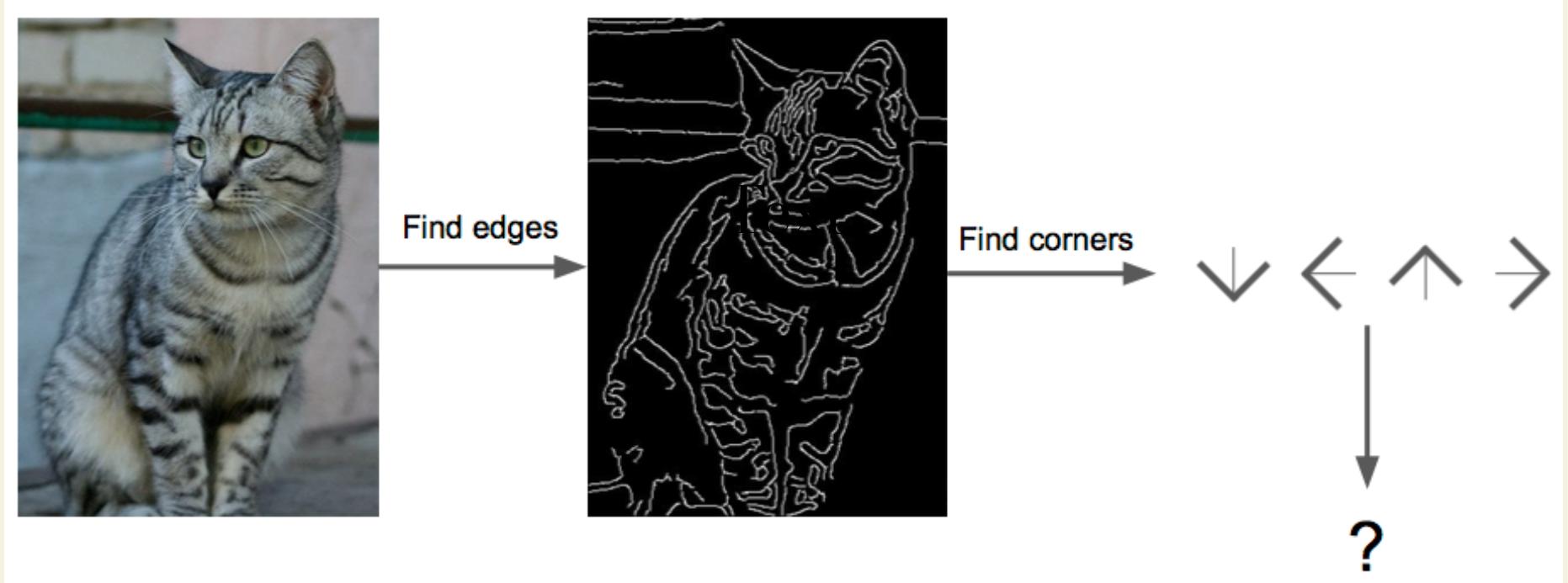


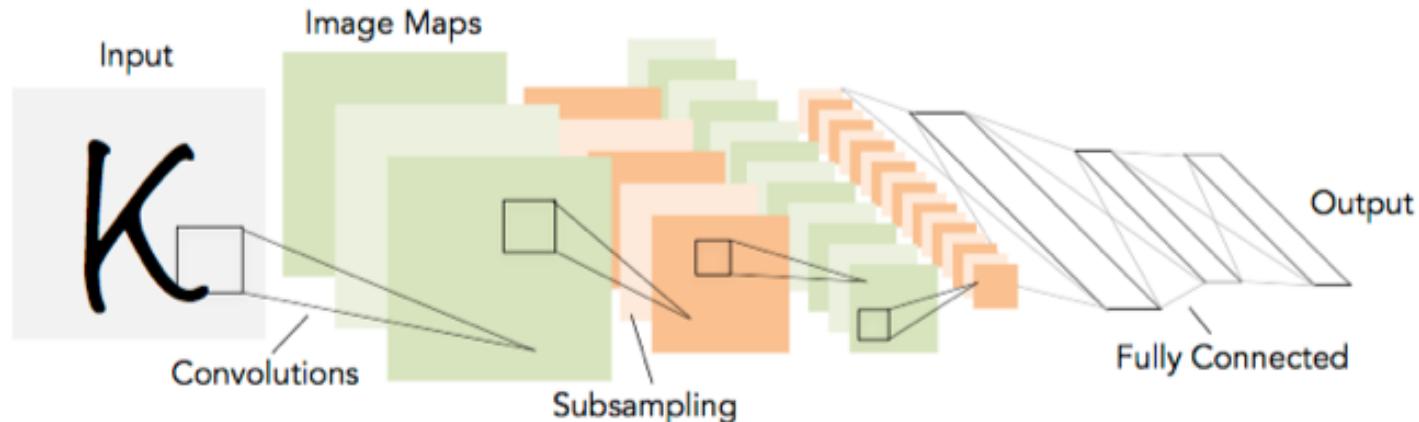
image: <http://cs231n.stanford.edu/>

NO!

- ~ But image recognition systems exist.
- ~ How?
  - ~ Knowledge-poor approaches
    - ~ Subsymbolic AI
    - ~ Math + Big Data

1998

LeCun et al.



# of transistors



$10^6$

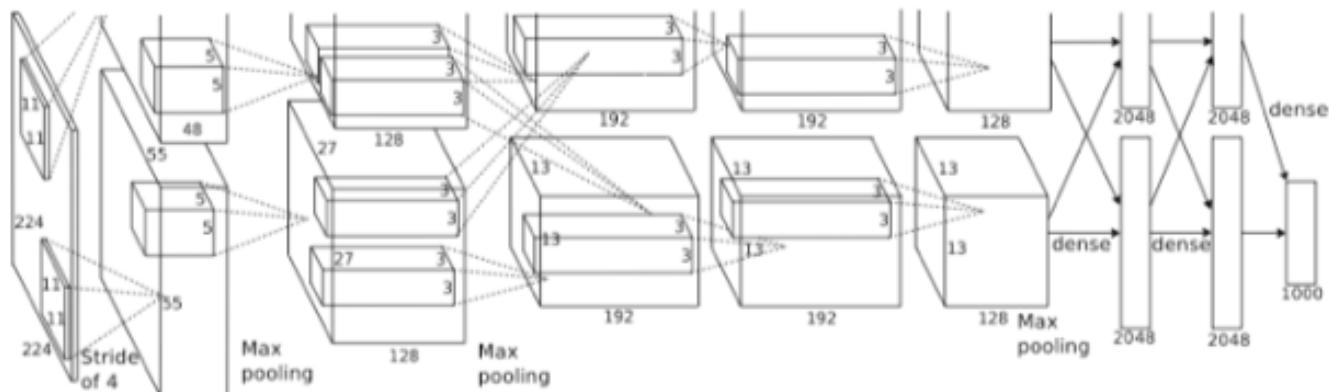
pentium® II

# of pixels used in training

$10^7$

2012

Krizhevsky et al.



# of transistors



$10^9$

GPUs



# of pixels used in training

$10^{14}$

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

# Study Problem: Label this image

