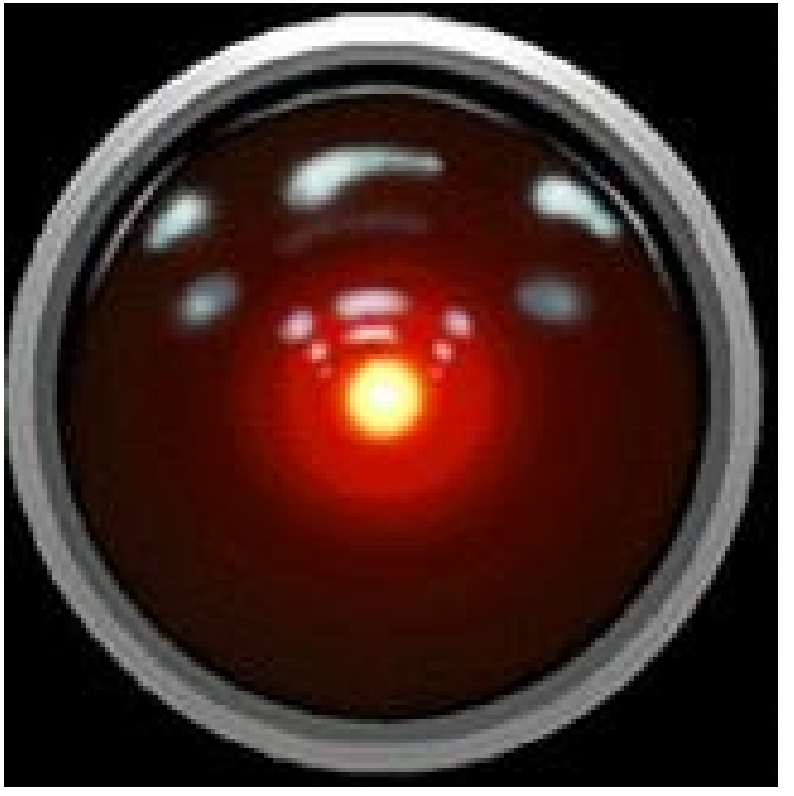


CS 378: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs378/

Computer Vision: 2D Images

Announcements

- Homework 5 deadline extended to this Thursday

Announcements

Volunteers needed for another study!

As before, there will be extra credit

To sign up, email:

– Rodolfo Rodriguez <rcorona@utexas.edu>

– Jesse Thomason <thomason.jesse@gmail.com>

Final Project Timeline

- Project Proposal due: ~~Mar. 29th~~ Apr. 1st
- Project Presentations / Demos: Last Week of Class (May 3rd and 5th)
- Final Report due: May 11th

Project Proposal Guidelines

- Work in groups of 2-3 (it's OK to work on your own if you really want to)
- Preferably, team up with people with different skills than yours
- Purpose of the proposal is to give you an outline / roadmap

Project Proposal Guidelines

- Each proposal should be about 2-3 pages

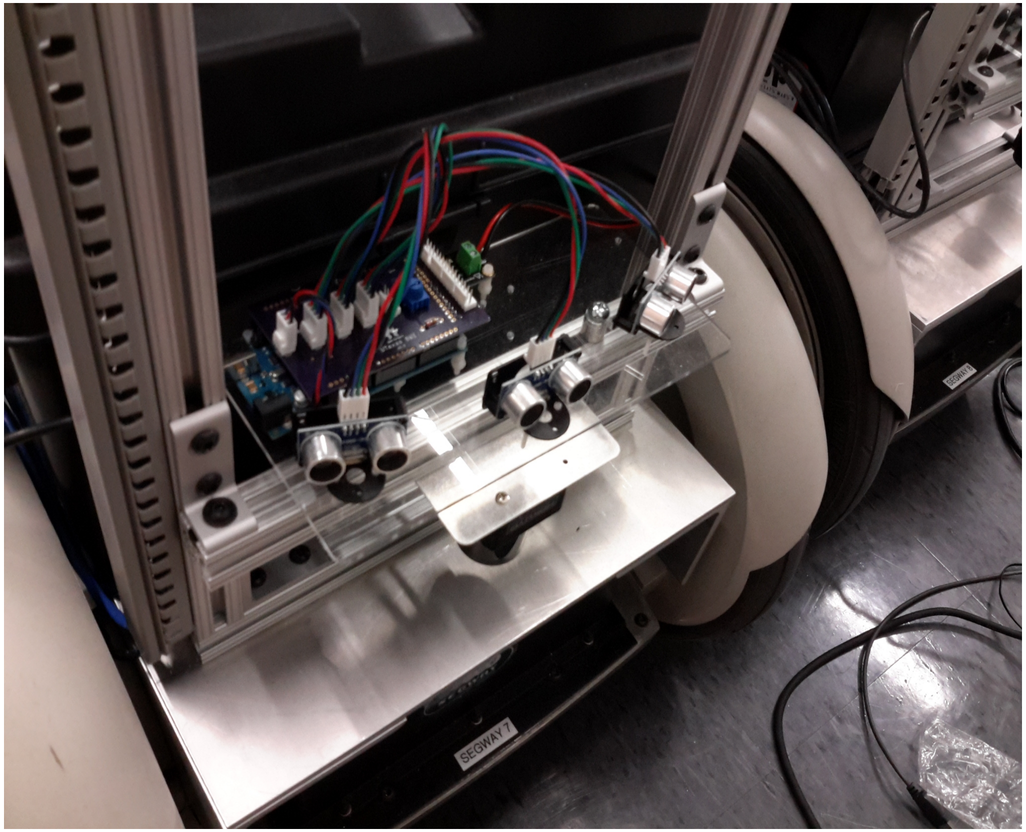
- Each proposal should include:
  - What is the application / task / problem?
  - Any previous experience you may have in that area
  - What do you expect to achieve by the end of the semester?
  - How do you plan to evaluate whether it works or not?
  - A timeline / schedule of progress and milestones

Project Proposal Guidelines

- Organization: your proposal should have sections and headings (don't just submit one long essay)
- For example:
  - Introduction / problem formulation
  - Proposed approach / software
  - Proposed evaluation
  - Summary of anticipated end result

Project Ideas

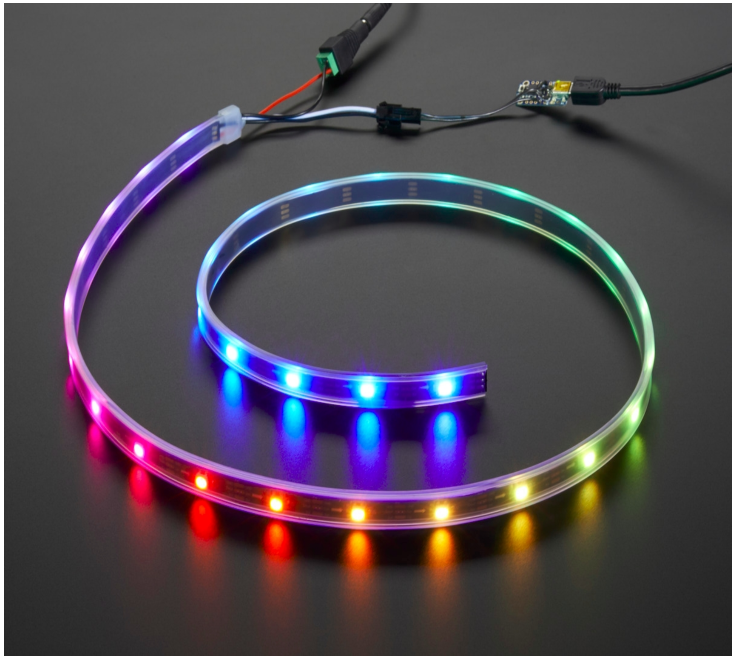
Vending Machine



Sonar Sensor

Project Ideas

Write ROS code to allow the robot to use an LED light strip



Project Ideas

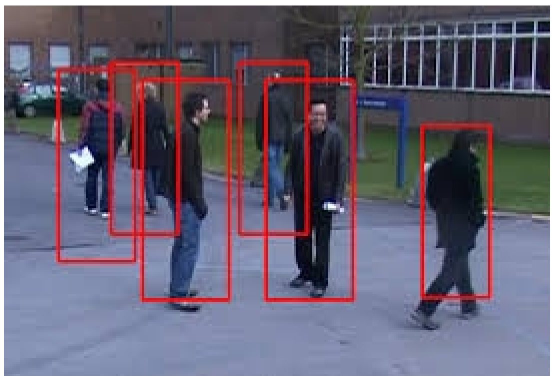
Help the robot “see” something it currently cannot

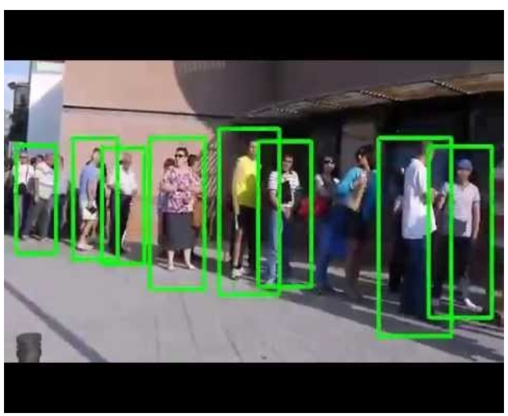
Help the robot “hear” something (e.g., the elevator sound)

Help the robot “do” something (e.g., follow a person)

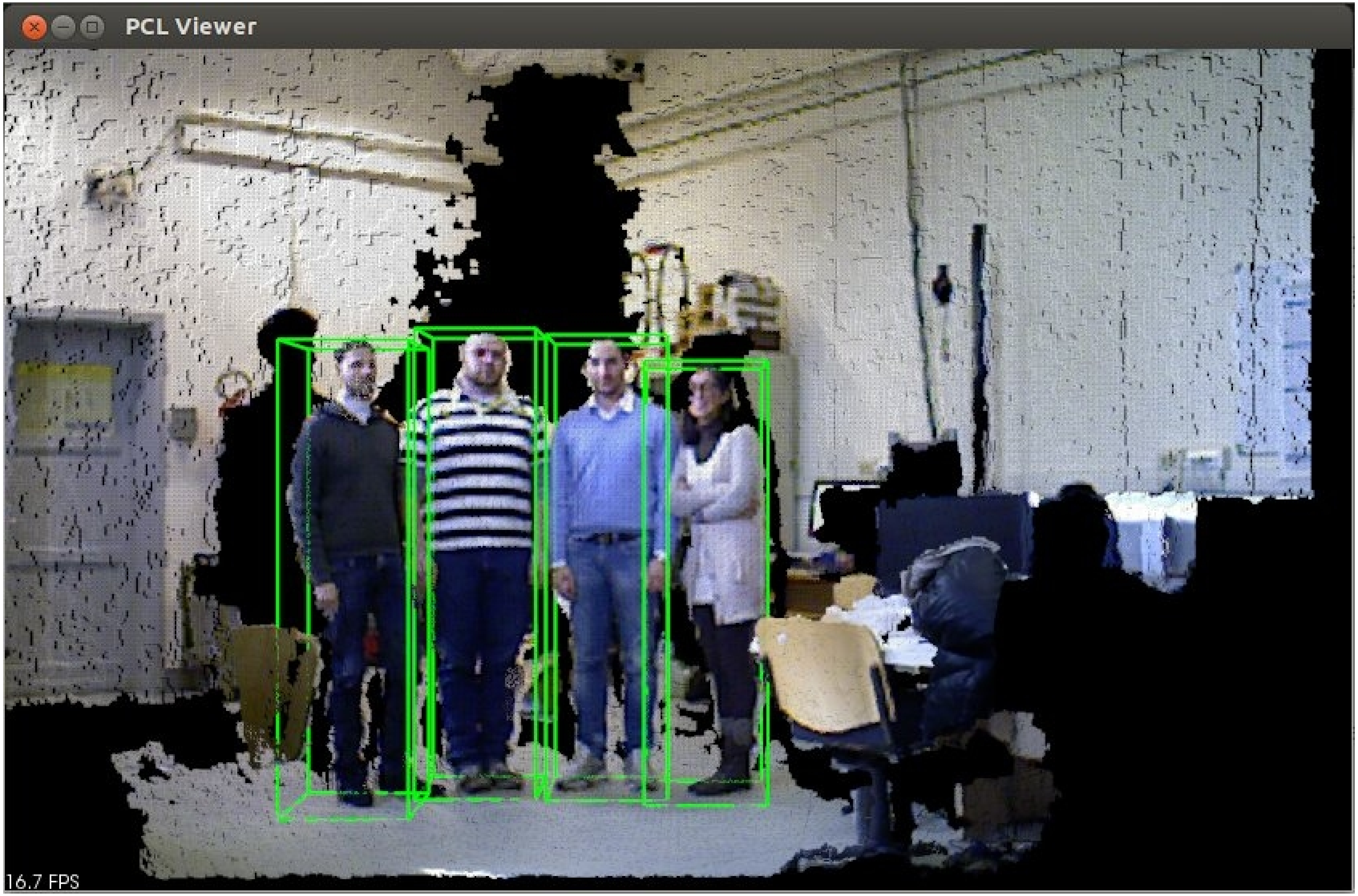
Project Ideas

Project Ideas

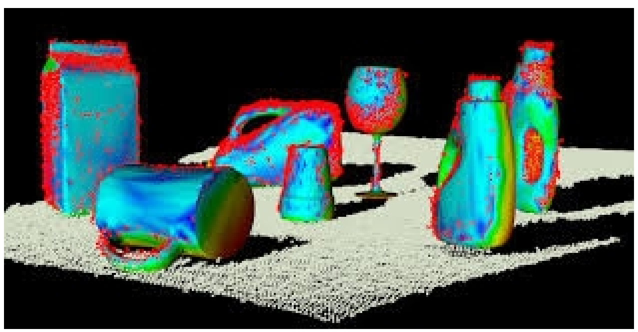


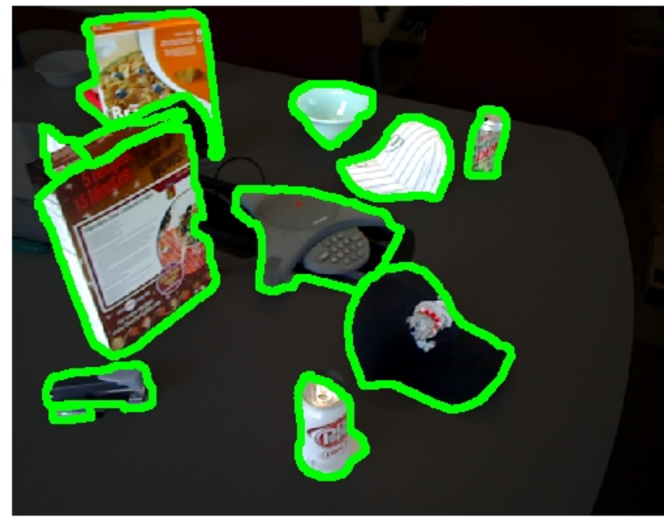


Project Ideas



Project Ideas





Final Project Timeline

The most important thing is to start early, and discuss your ideas with the TA, mentors and myself. We'll point you to a starting point, describe functionality that already exists, and help refine your ideas.

Final Project Timeline

- Project Proposal due: ~~Mar. 29th~~ Apr. 1st
- Project Presentations / Demos: Last Week of Class (May 3rd and 5th)
- Final Report due: May 11th

Computer Vision: 2D Images

Readings

- Jain, Kasturi, and Schunck (1995). Machine Vision, ``Chapter 1: Introduction,'' McGraw-Hill, pp. 1-24.

- Jain, Kasturi, and Schunck (1995). Machine Vision, ``Chapter 2: Binary Image Processing,'' McGraw-Hill, pp. 25-72.

Readings (con't)

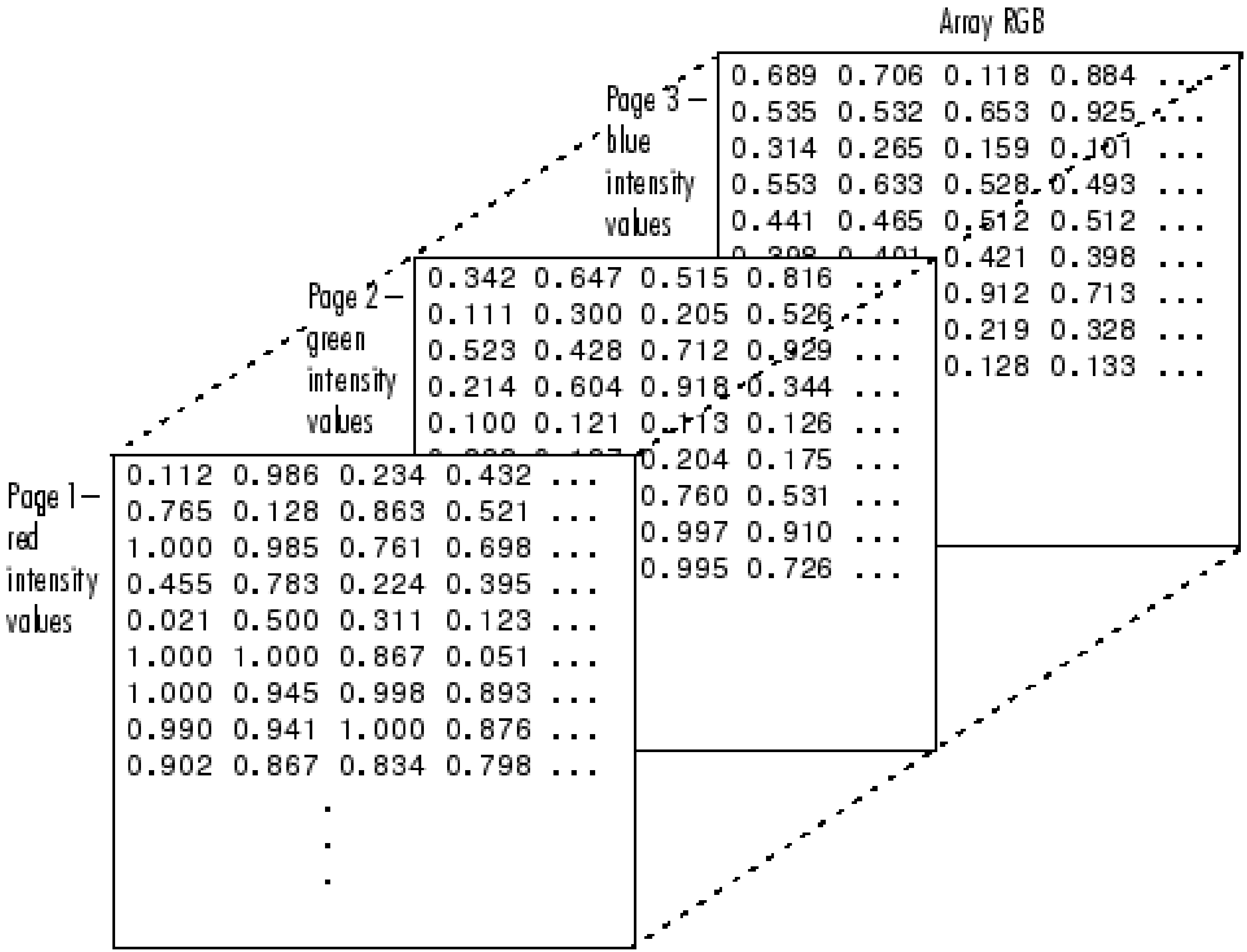
- J. K. O'Regan and A. Noe, (2001). ``A sensorimotor account of vision and visual consciousness'' , *Behavioral and Brain Sciences*, 24(5), 939- 1011.

What is an image?

A grayscale image

Index	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	21	22	23	24	25	26	27	28	29	30
3	31	32	33	34	35	36	37	38	39	40
4	41	42	43	44	45	46	47	48	49	50
5	51	52	53	54	55	56	57	$\boxed{58}$	59	60
6	61	62	63	64	65	66	67	68	69	70
7	71	72	73	74	75	76	77	78	79	80
8	81	82	83	84	85	86	87	88	89	90
9	91	92	93	94	95	96	97	98	99	100

An RGB image



Array RGB
Page 3 — blue intensity values
0.689 0.706 0.118 0.884 ...
0.535 0.532 0.653 0.925 ...
0.314 0.265 0.159 0.101 ...
0.553 0.633 0.528 0.493 ...
0.441 0.465 0.512 0.512 ...
0.421 0.398 ...
0.912 0.713 ...
0.219 0.328 ...
0.128 0.133 ...
Page 2 — green intensity values
0.342 0.647 0.515 0.816 ...
0.111 0.300 0.205 0.526 ...
0.523 0.428 0.712 0.929 ...
0.214 0.604 0.918 0.344 ...
0.100 0.121 0.113 0.126 ...
0.204 0.175 ...
0.760 0.531 ...
0.997 0.910 ...
0.995 0.726 ...
Page 1 — red intensity values
0.112 0.986 0.234 0.432 ...
0.765 0.128 0.863 0.521 ...
1.000 0.985 0.761 0.698 ...
0.455 0.783 0.224 0.395 ...
0.021 0.500 0.311 0.123 ...
1.000 1.000 0.867 0.051 ...
1.000 0.945 0.998 0.893 ...
0.990 0.941 1.000 0.876 ...
0.902 0.867 0.834 0.798 ...

How did computer vision start?

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that!

Computer vision vs human vision

What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

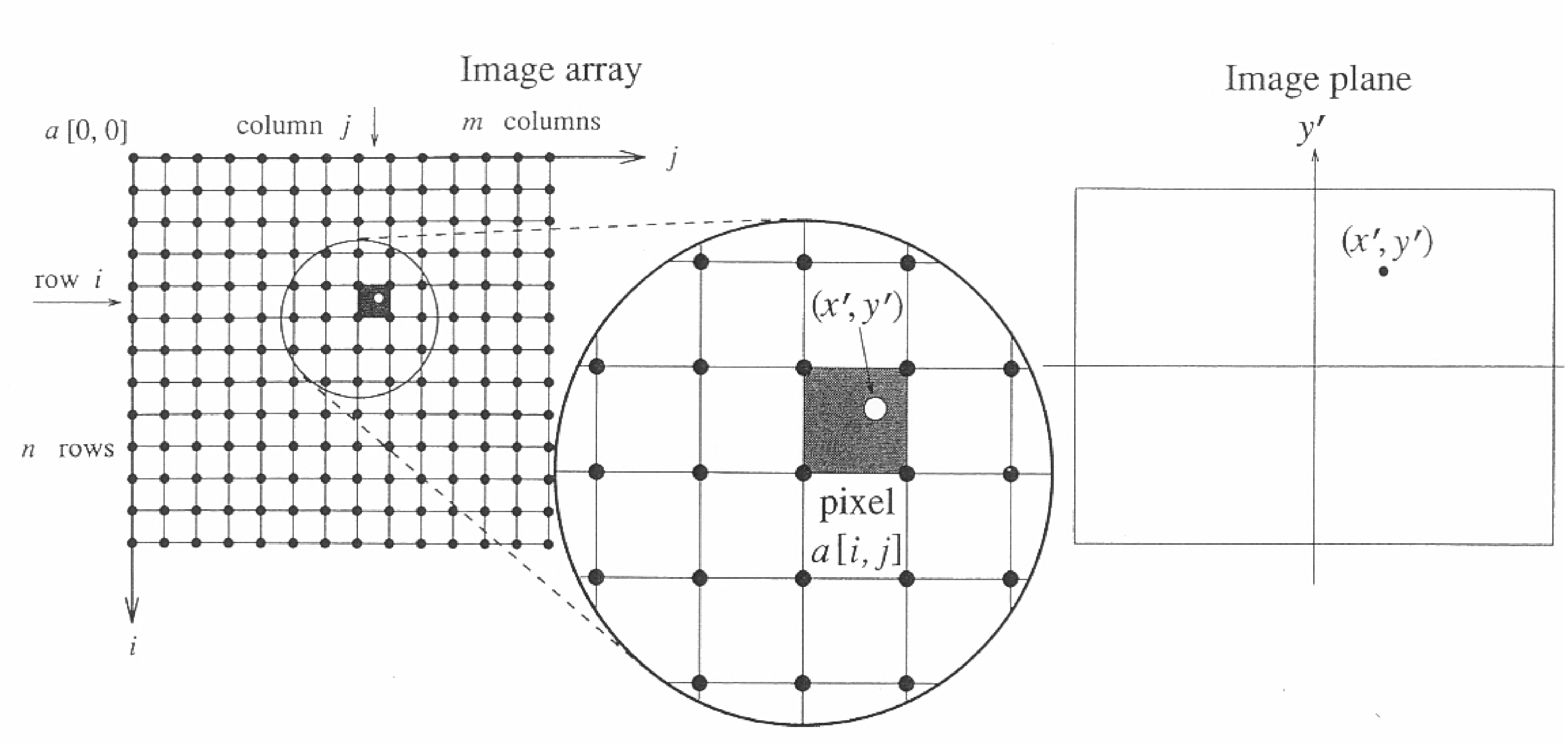
Intensity Levels

- 2
- 32
- 64
- 128
- 256 (8 bits)
- 512
- …
- 4096 (12 bits)

Intensity Levels

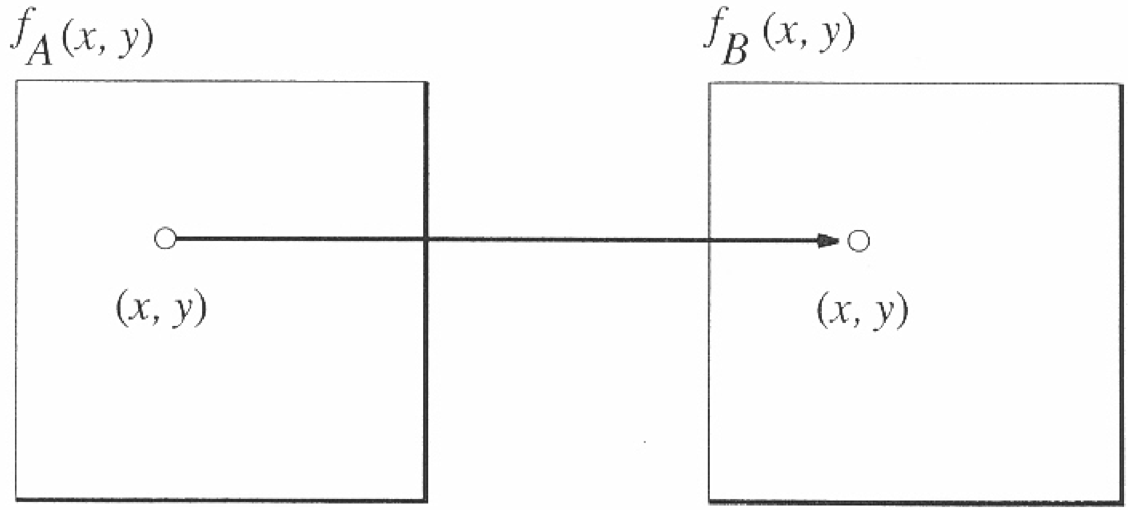
- 2
- 32
- 64
- 128
- **256 (8 bits)**
- 512
- …
- 4096 (12 bits)

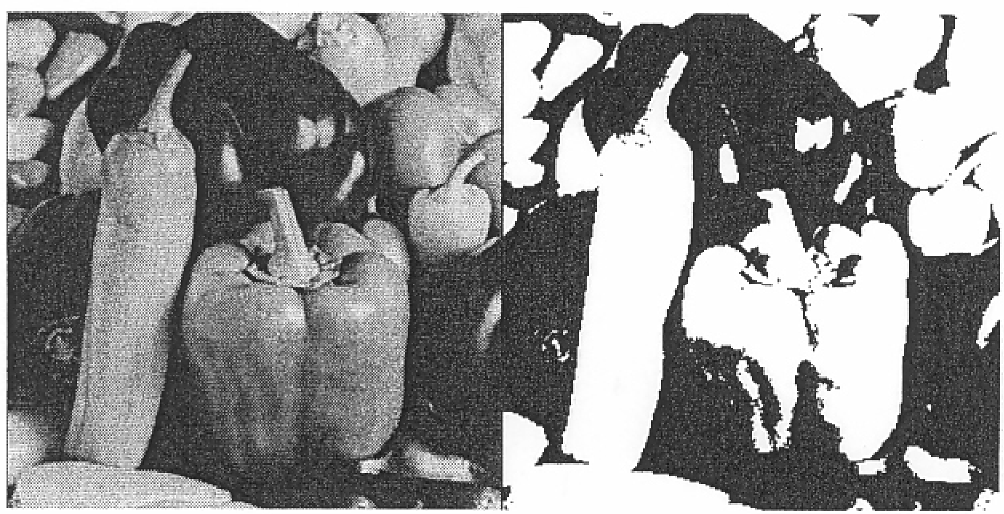
Image Plane v.s. Image Array



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

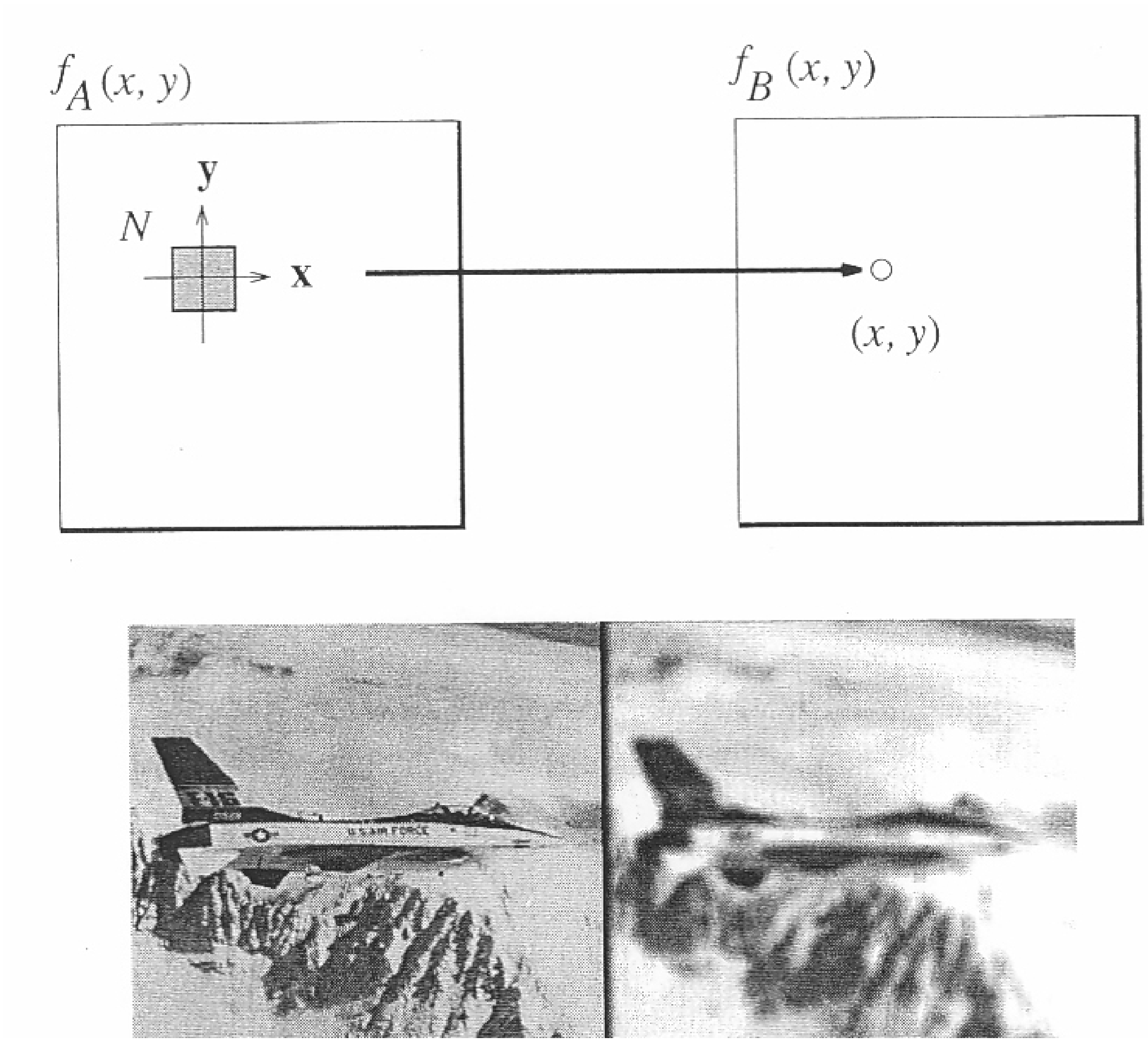
Point Operations





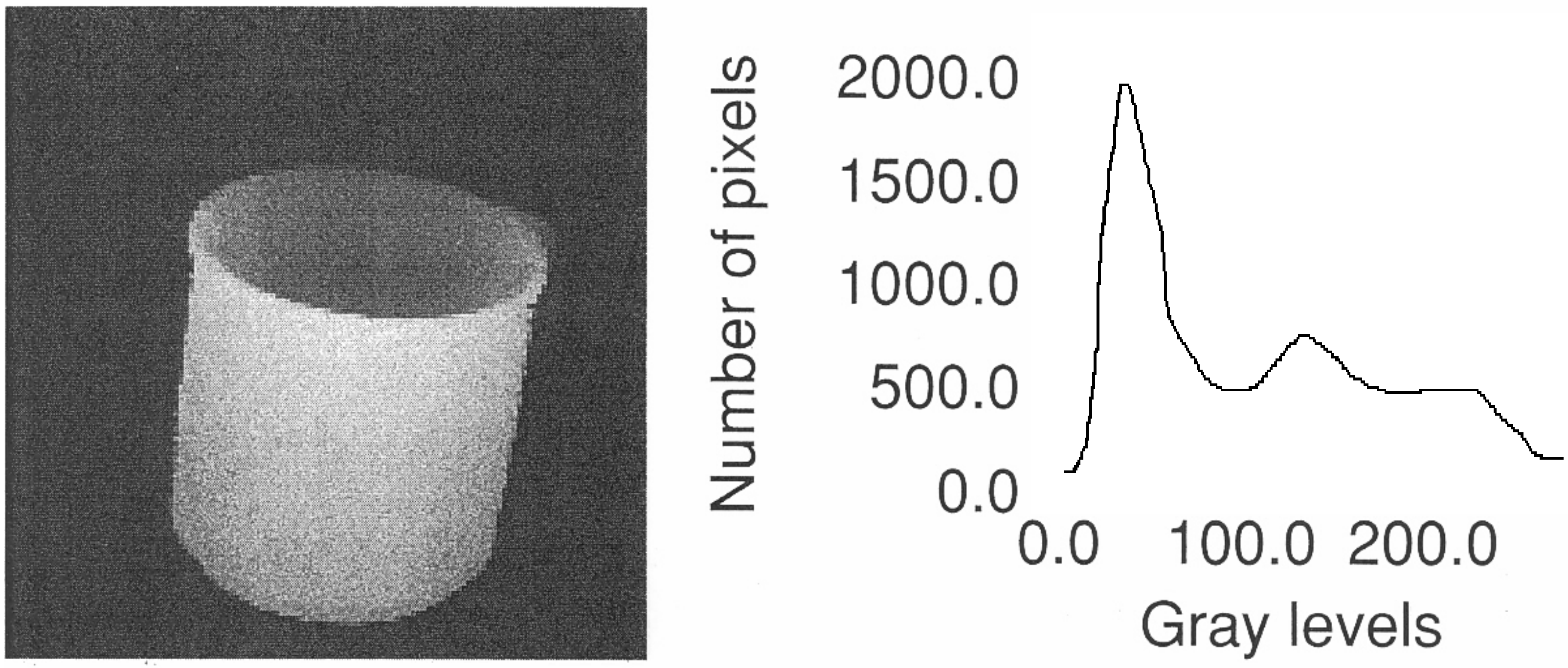
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Local Operations



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

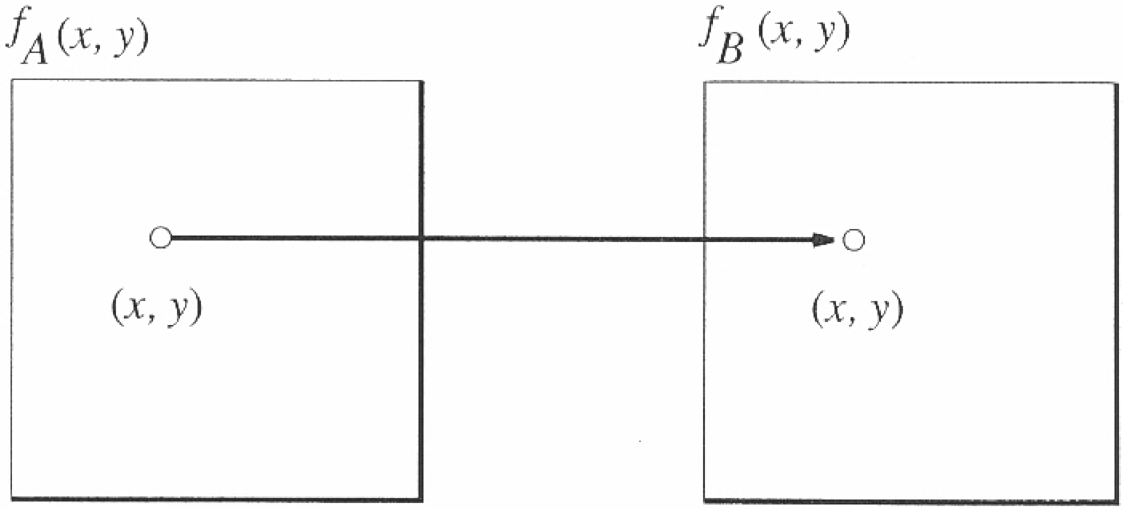
Global Operations



$$P = O_{\mathrm{global}}\{f[i, j]\}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Thresholding an Image



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Dark Image on a Light Background

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \le T \\ 0 & \text{otherwise.} \end{cases}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Selecting a range of intensity values

$$F_T[i,j] = \begin{cases} 1 & \text{if } T_1 \leq F[i,j] \leq T_2 \\ 0 & \text{otherwise.} \end{cases}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

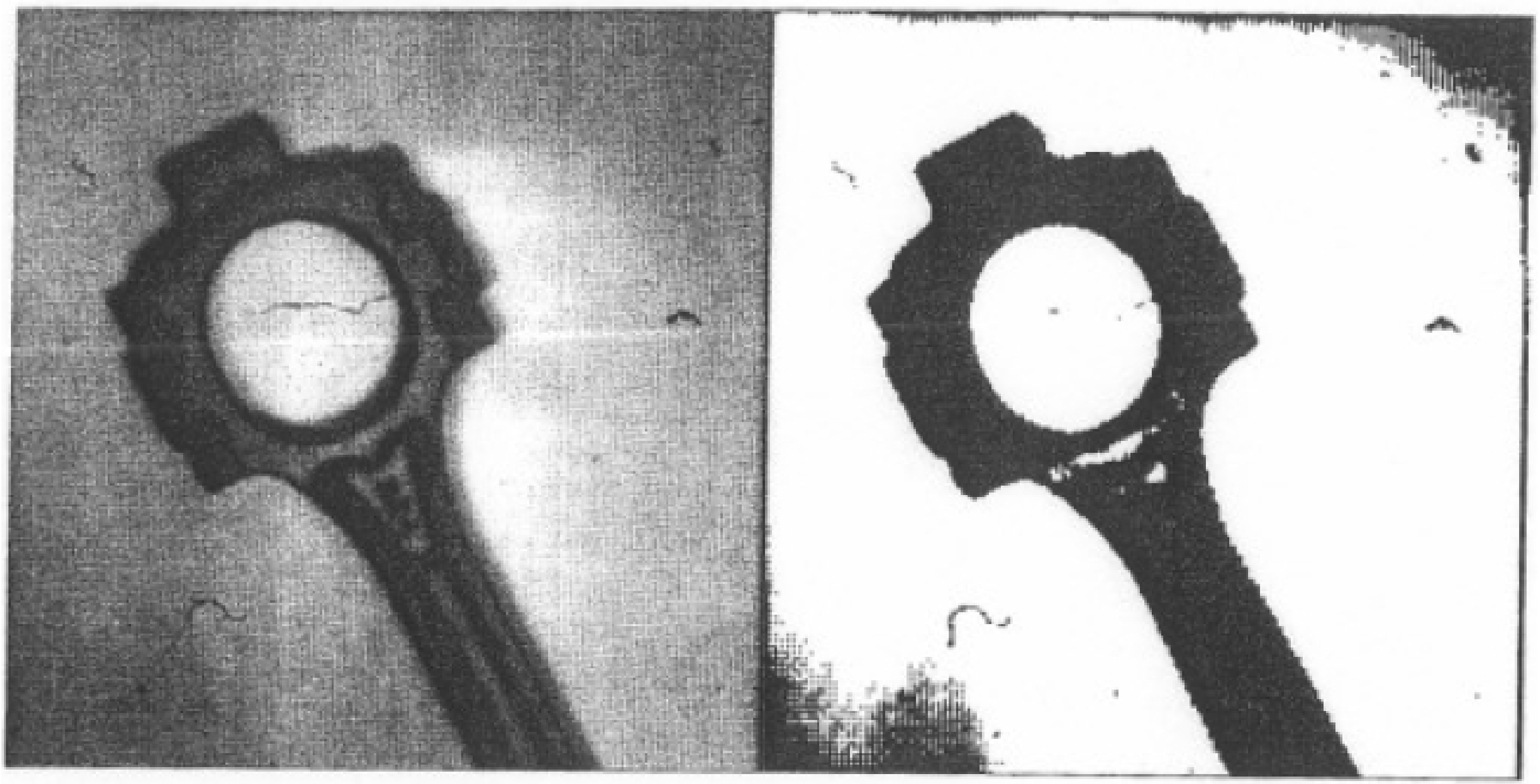
Generalized Thresholding

A general thresholding scheme in which the intensity levels for an object may come from several disjoint intervals may be represented as

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \in Z \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

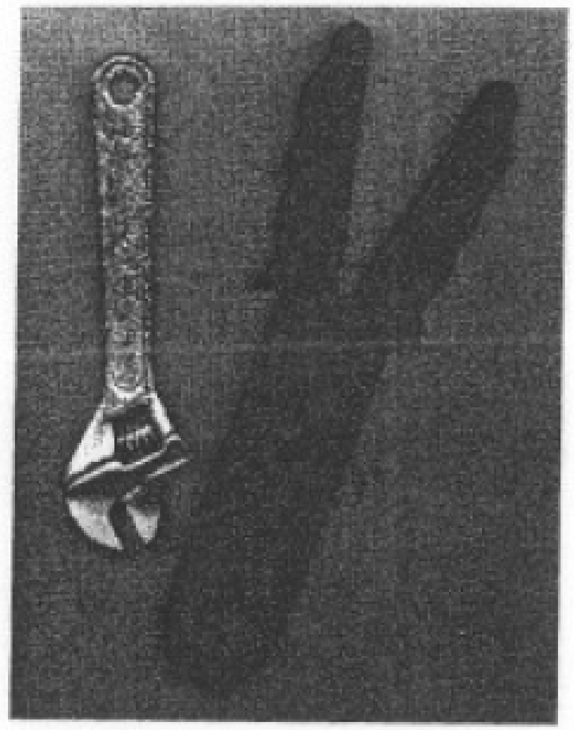
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thresholding Example (1)



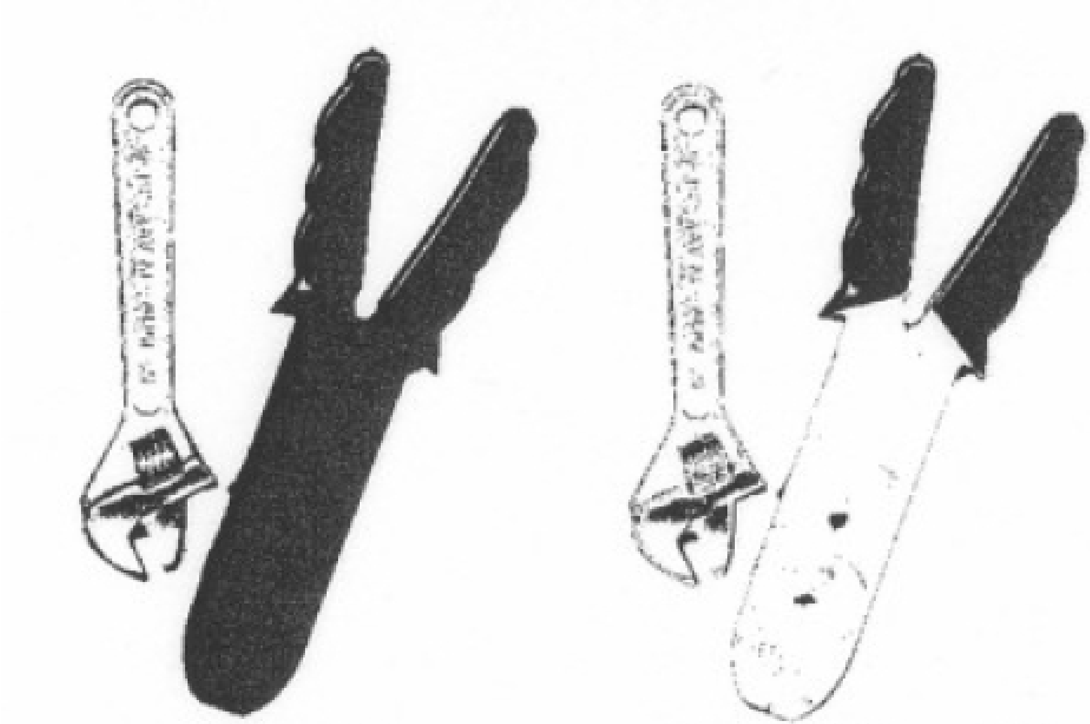
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thresholding Example (2)



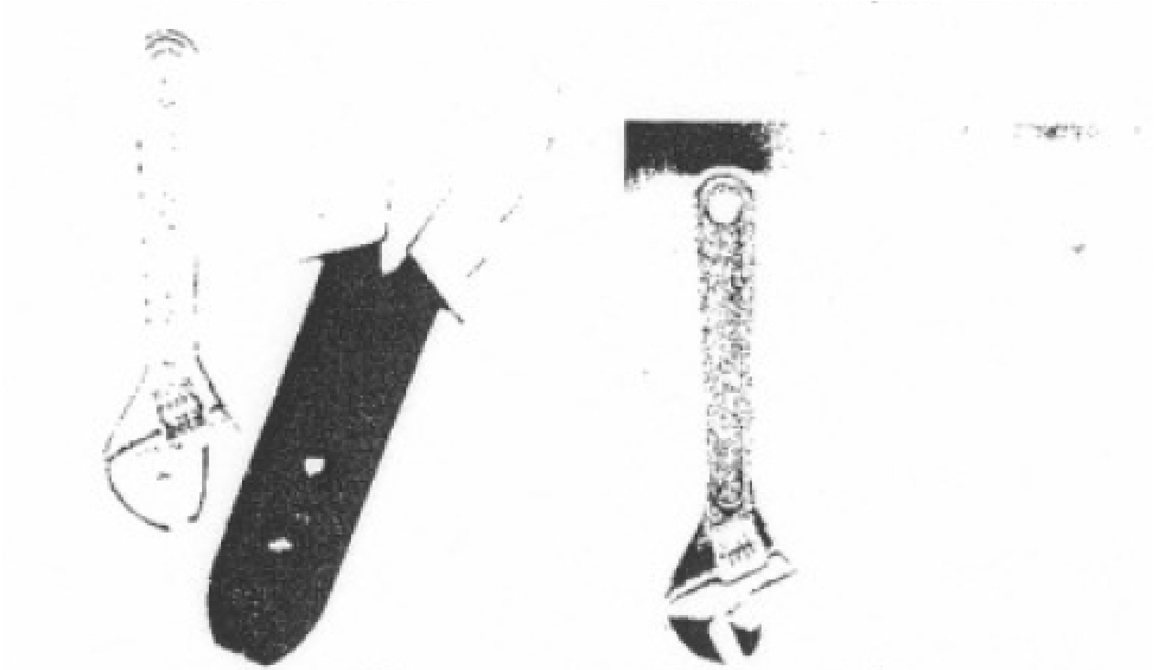
Original grayscale Image

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



$T = 48$ $T_1 = 2 | T_2 = 48$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



$T = 21$ $\quad$ $T_1 = 135 | T_2 = 255$

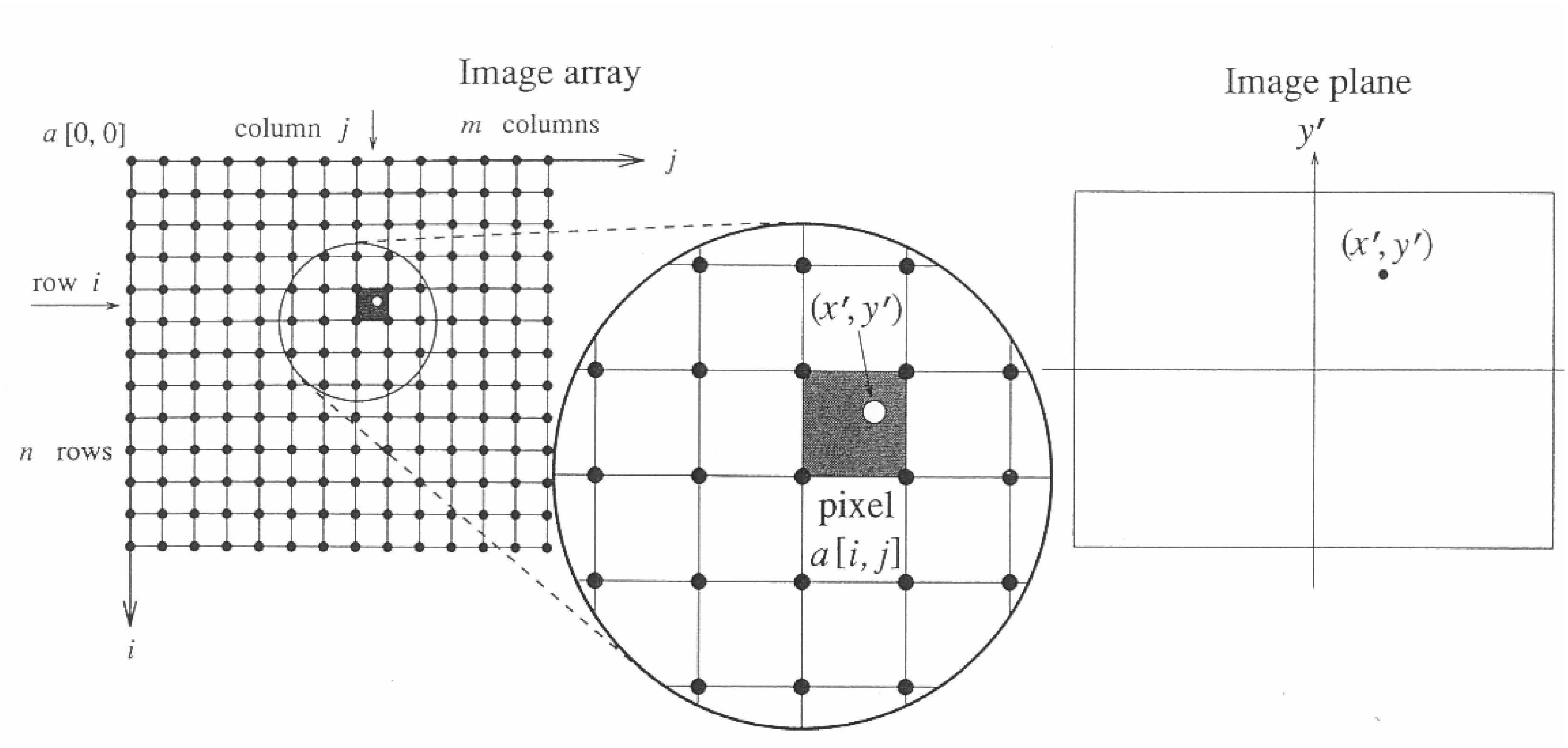
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Area of a Binary Image

$$A = \sum_{i=1}^{n} \sum_{j=1}^{m} B[i, j].$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

This figure now becomes important



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Calculating the Position of an Object

$$\bar{x} \sum_{i=1}^{n} \sum_{j=1}^{m} B[i,j] = \sum_{i=1}^{n} \sum_{j=1}^{m} j B[i,j]$$

$$\bar{y} \sum_{i=1}^{n} \sum_{j=1}^{m} B[i,j] = \sum_{i=1}^{n} \sum_{j=1}^{m} i B[i,j]$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

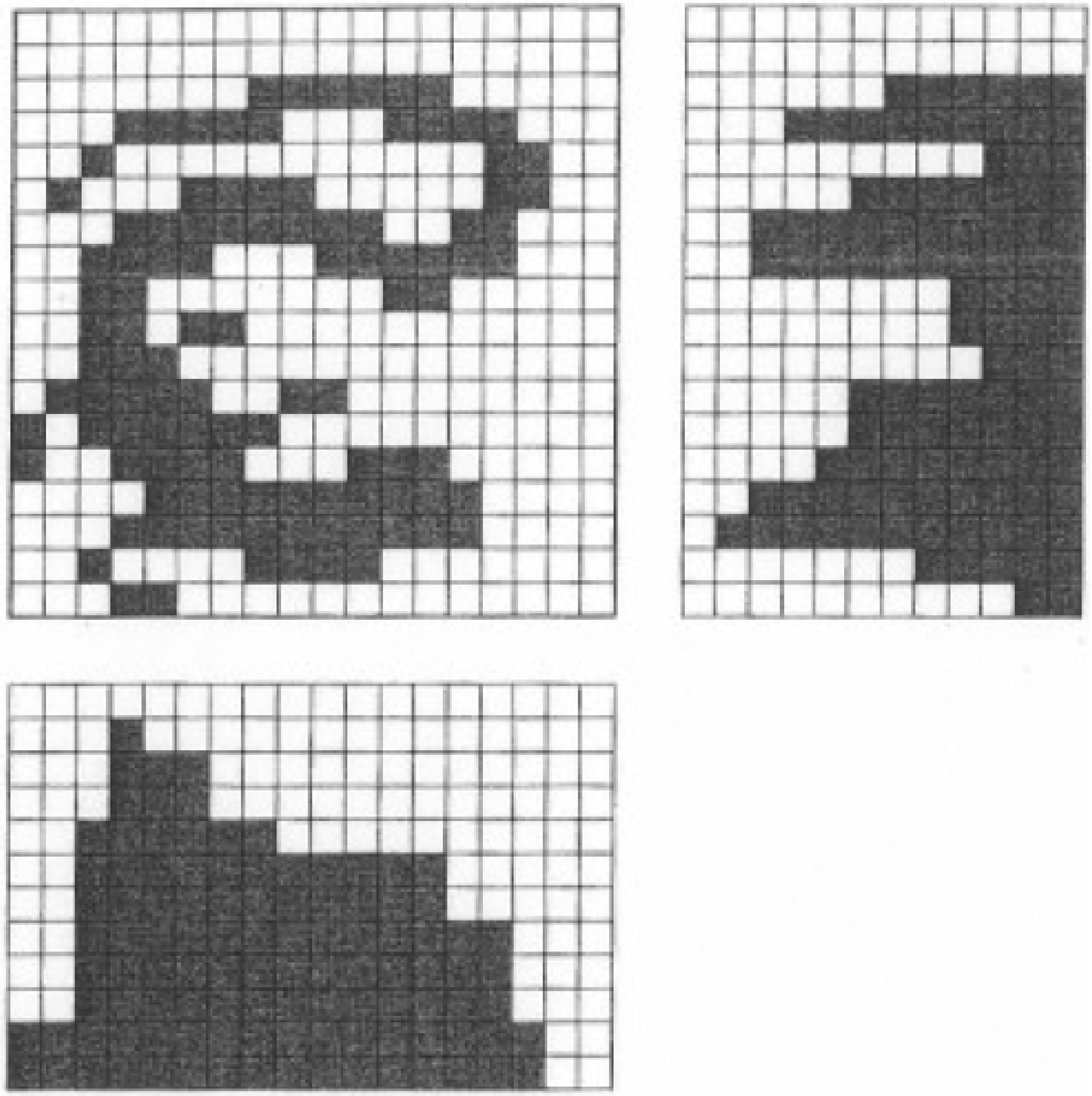
The center is given by

$$\bar{x} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} jB[i,j]}{A}$$

$$\bar{y} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} iB[i,j]}{A}.$$

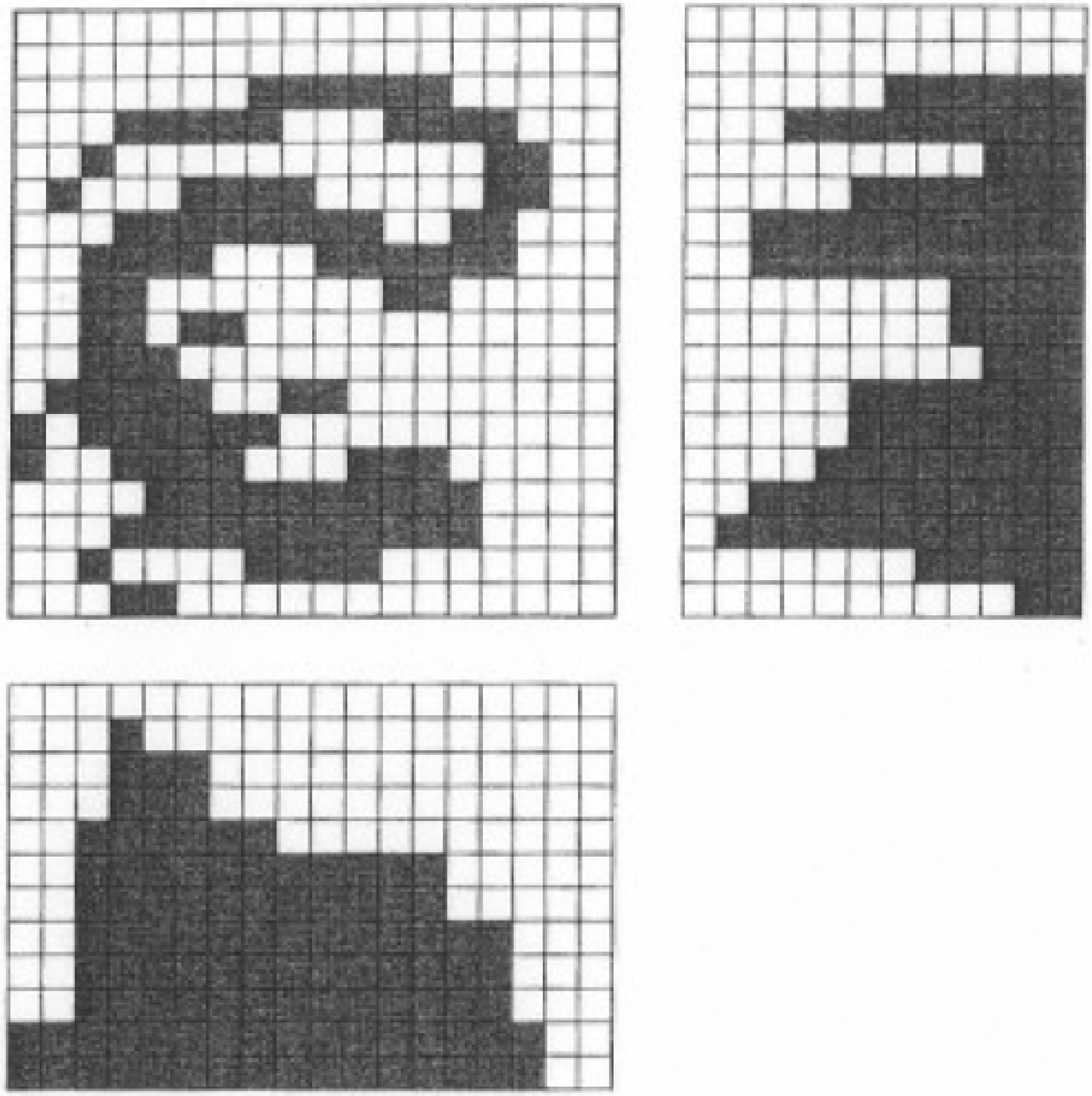
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Horizontal and Vertical Projections



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Horizontal and Vertical Projections



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

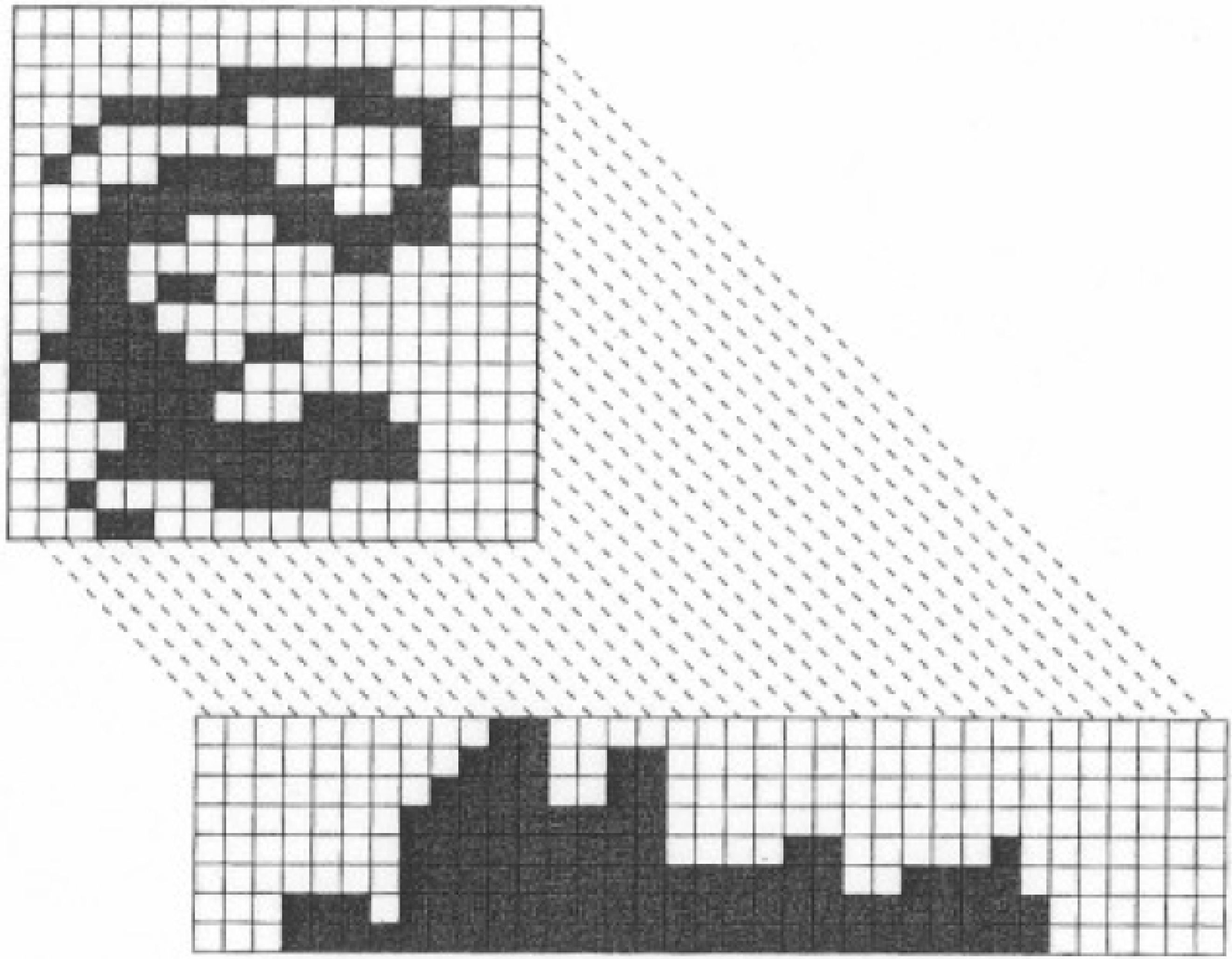
Projection Formulas

$$H[i] = \sum_{j=1}^{m} B[i,j]$$

$$V[j] = \sum_{i=1}^{n} B[i,j].$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Diagonal Projection



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

The area and the position can be computed from the H and V projections

$$A = \sum_{j=1}^{m} V[j] = \sum_{i=1}^{n} H[i]$$

$$\bar{y} = \frac{\sum_{i=1}^{n} iH[i]}{A}$$

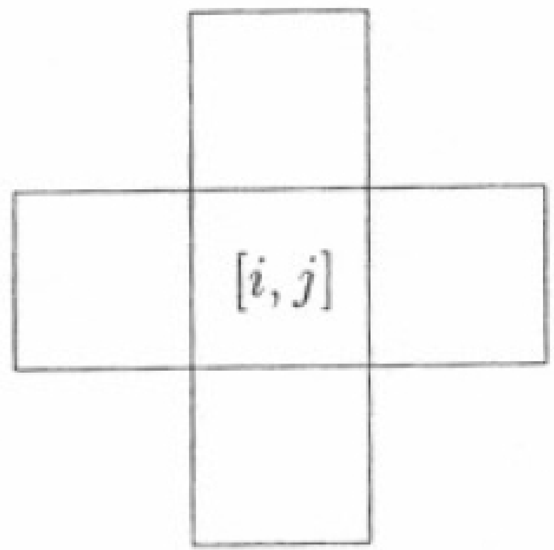
$$\bar{x} = \frac{\sum_{j=1}^{m} jV[j]}{A}.$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Neighbors and Connectivity

4-Connected

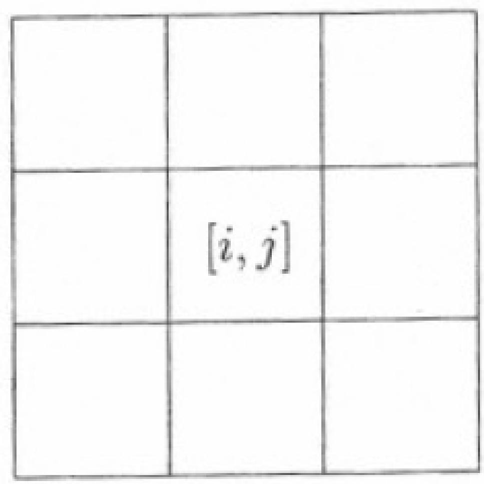
4-neighbors $[i+1,j]$, $[i-1,j]$, $[i,j+1]$, $[i,j-1]$



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

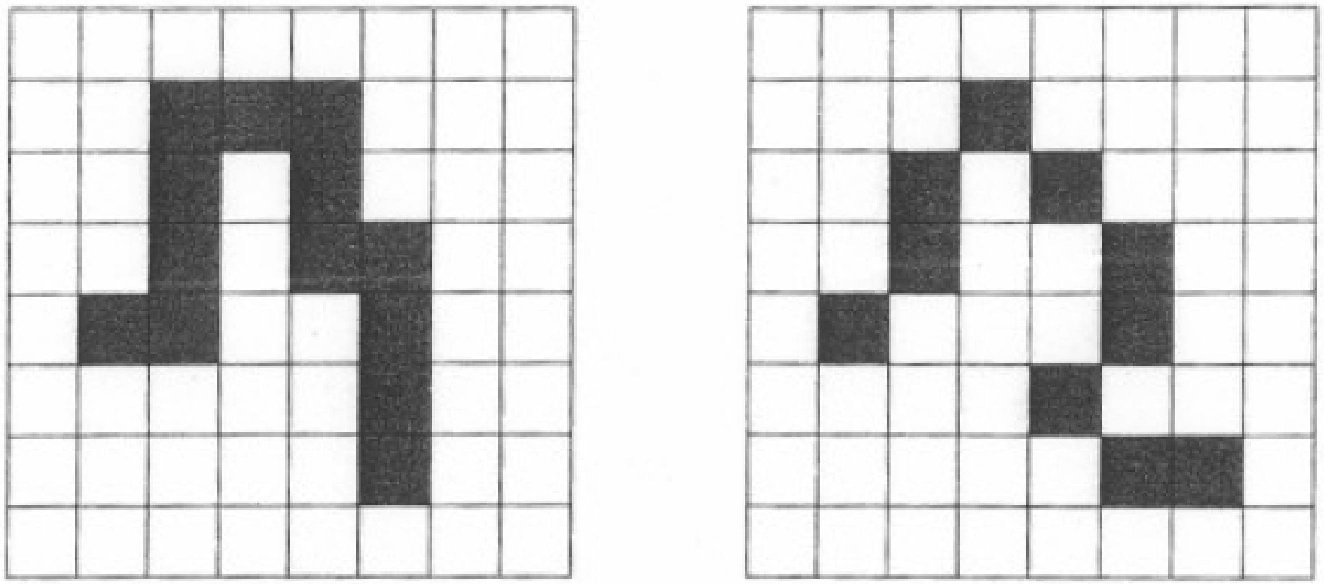
8-connected

8-neighbors $[i+1, j+1]$, $[i+1, j-1]$, $[i-1, j+1]$, $[i-1, j-1]$ plus all of the 4-neighbors



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

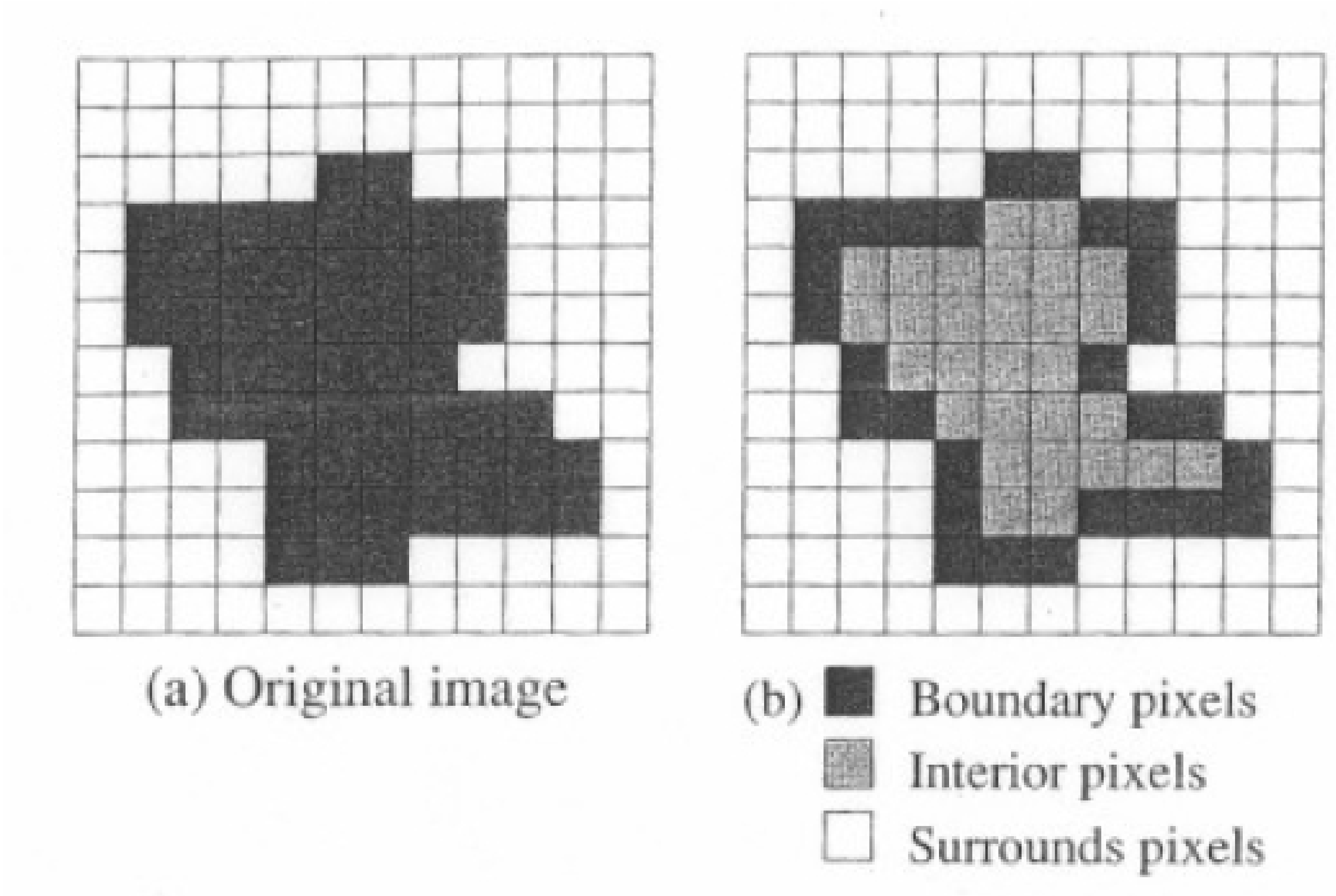
Examples of Paths



(a) 4-path (b) 8-path

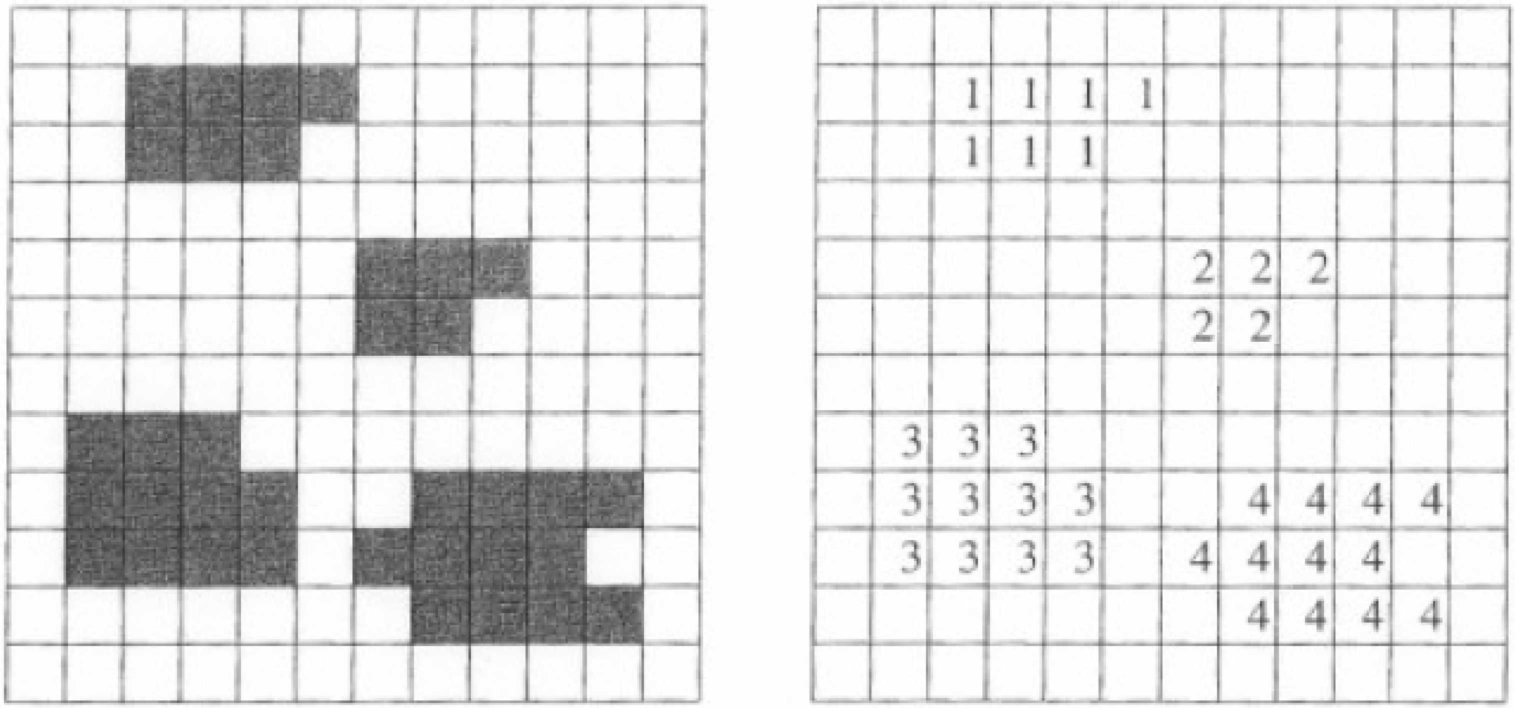
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Boundary, Interior, and Background



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

An Image (a) and Its Connected Components (b)

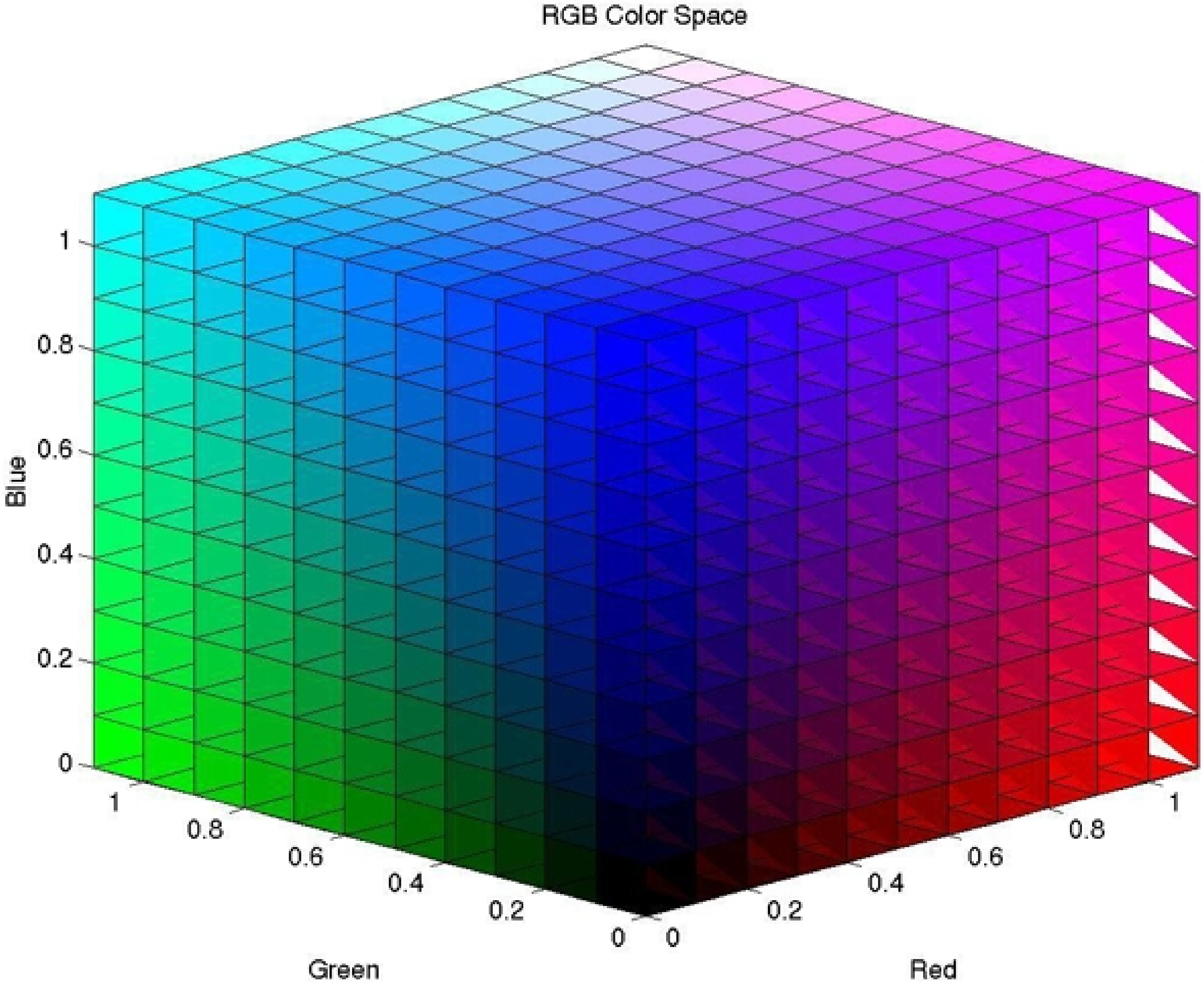


(a) (b)

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

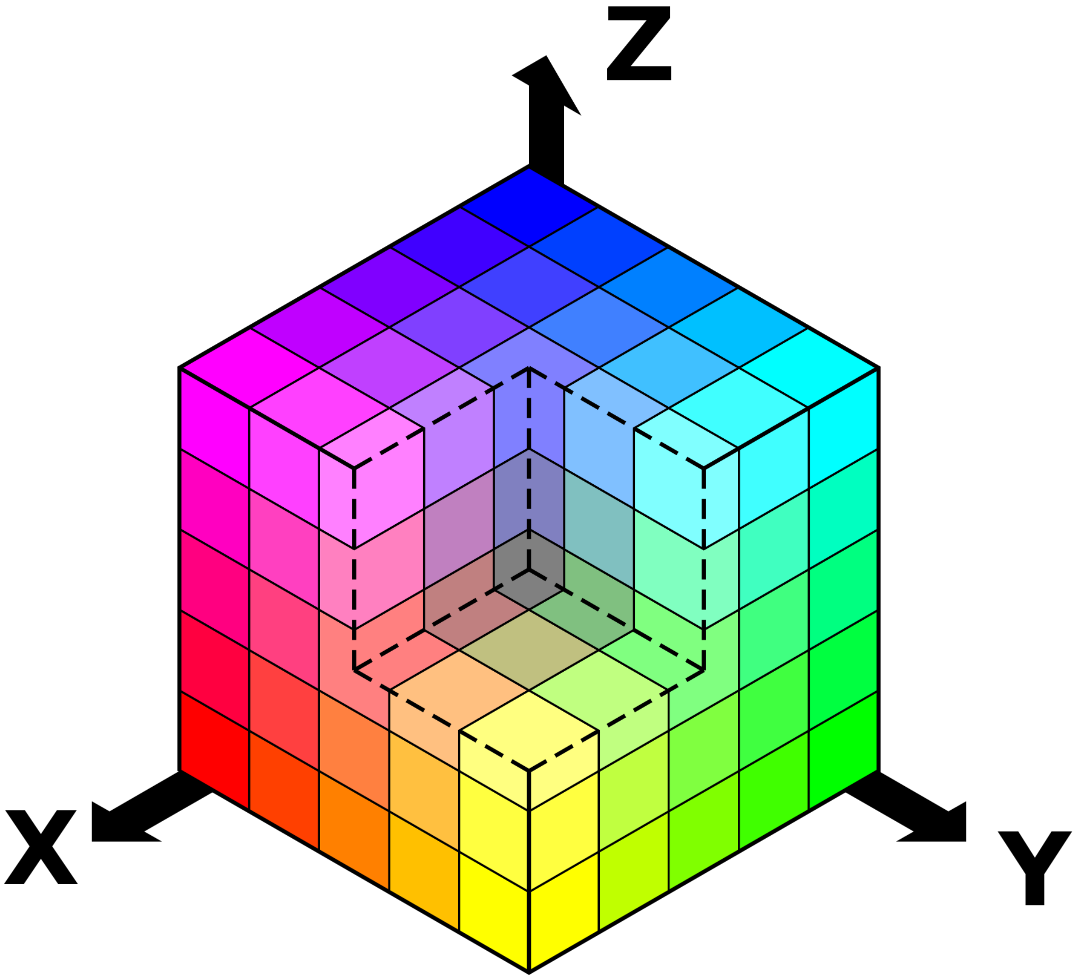
Color Perception

The RGB Color Space



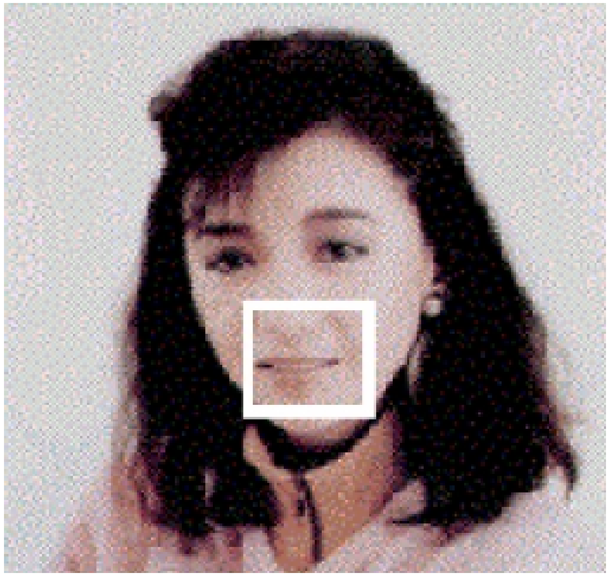
[http://www.arcsoft.com/images/topics/darkroom/what-is-color-space-RGB.jpg]

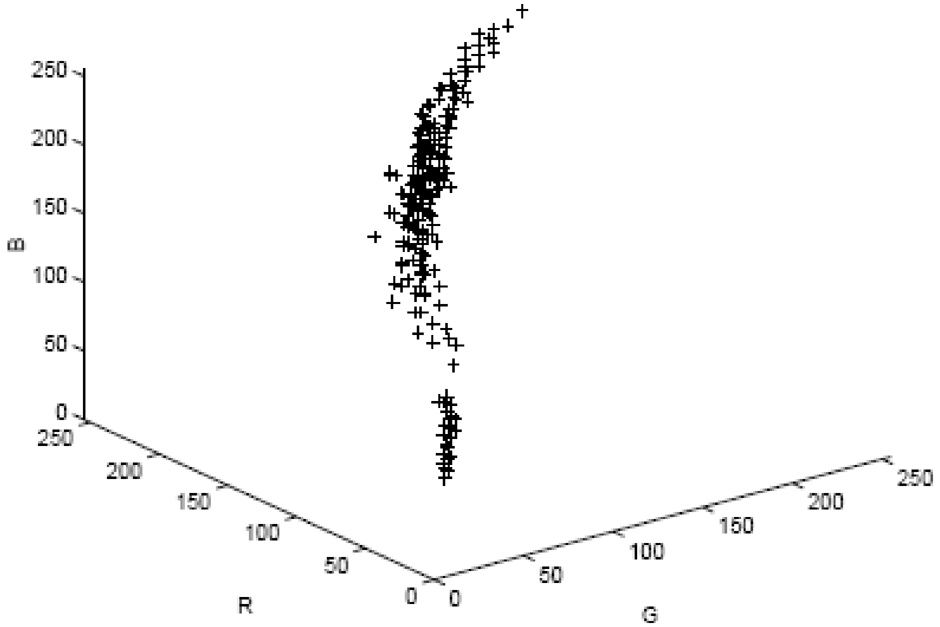
The RGB Color Space



https://upload.wikimedia.org/wikipedia/commons/thumb/1/11/RGBCube_b.svg/2000px-RGBCube_b.svg.png

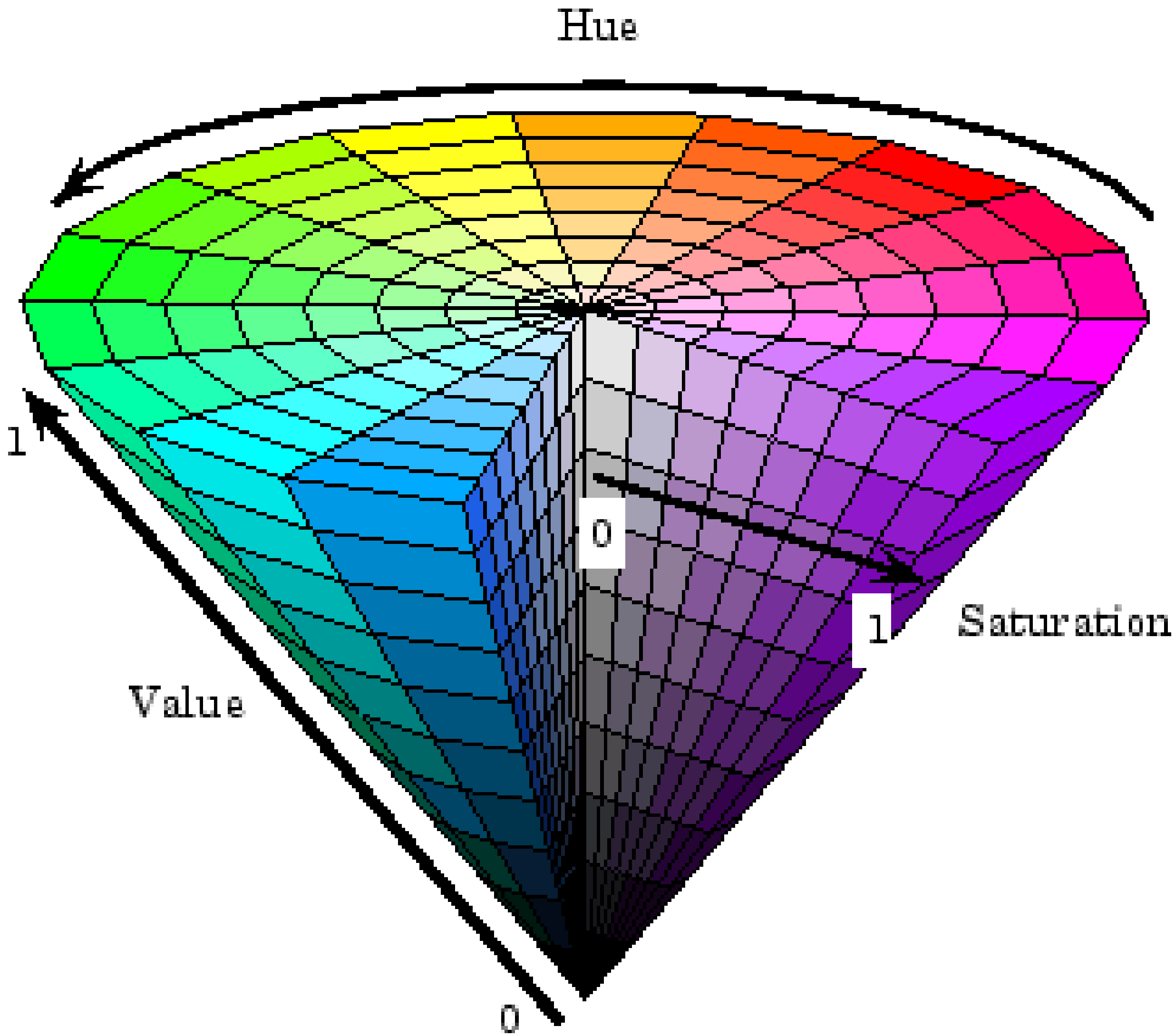
3D Scatter Plot for a patch of skin





3-D scatterplot

The HSV Color Space

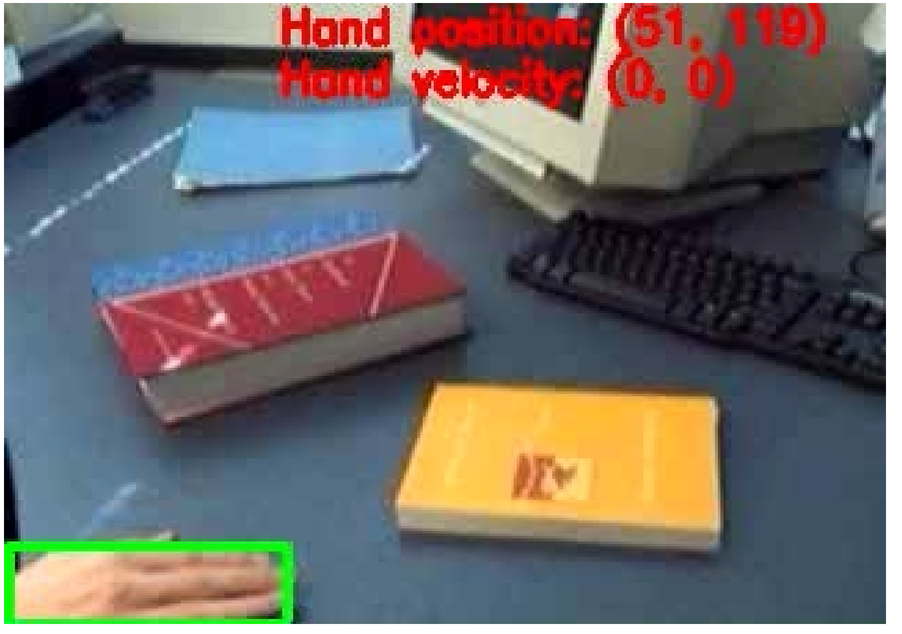


Color Detection and Segmentation

Color Detection and Segmentation

Discussion: how may we achieve this?

Example Hand Tracking using Color



Computer Vision in ROS

Computer Vision in ROS

1) Subscribing to an image topic
2) Converting a ROS image to an OpenCV image
3) Copy an image
4) Convert an image to grayscale
5) Access and set individual pixel values

Example Color Detection in ROS using OpenCV

Resources

- OpenCV in ROS:
  - http://wiki.ros.org/vision_opencv
  - http://wiki.ros.org/cv_bridge/Tutorials
  - http://docs.opencv.org/2.4/doc/tutorials/tutorials.html

THE END