~~LEC 1~~

## INTRODUCTION

Goal: provided an image, make inferences about the scene

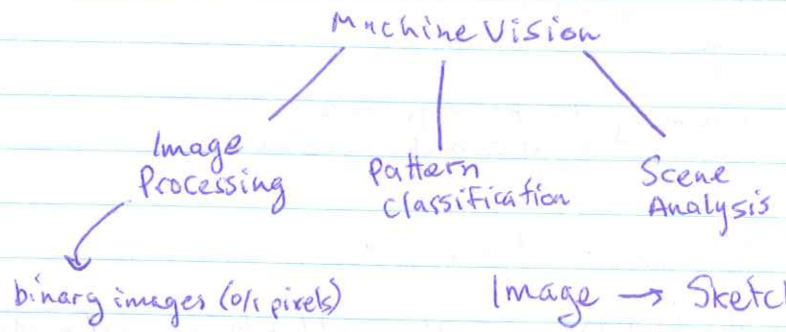



Image → Sketch

## CHAPTER 2: Image Formation & Image Sensing

2 key questions ⟶ 1) what determines 3D→2D?
2) Brightness?

Lens equation:

$$\frac{1}{z'} + \frac{1}{-z} = \frac{1}{f}$$

vignetting → brightness drop-off away from optical axis

↳ not a problem for people, but a big one for machines!

A key simplification is that, typically, we are looking at opaque objects in transparent media (air, water, vacuum)

quantum efficiency - frequency-dependent electron-flux-to-photon-flux

Because humans have only 3 cones many different "true" colors are metameric.

# CHAPTER 3 : Binary Images: Geometric Properties

Think of image components as sets → size, shape, position

Simple example: <u>area</u>

$$A = \iint b(x,y)\,dx\,dy$$

↑ assuming dark = object = 1, light = background = 0

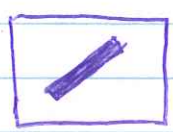
Position of center of mass:

$$\bar{x} \iint b(x,y)\,dx\,dy = \iint x\,b(x,y)\,dx\,dy$$

↑ center of mass is where all mass could be concentrated and not change the first moment of the object.

Orientation:

Suppose we have a ruler



axis of elongation = least second moment

$$E_{min} = \iint r^2 b(x,y)\,dx\,dy$$

↑ $r$ = ⊥ distance to axis line

$$= \iint (x\sin\theta - y\cos\theta + \rho)^2 b(x,y)\,dx\,dy$$

$$\tan 2\theta = \frac{b}{a-c}$$

- $b$ ← $2^{nd}$ xy-moment
- $a$ ← $2^{nd}$ x-moment
- $c$ ← $2^{nd}$ y-moment
- $\theta$ ← angle of orientation

We can speed up this analysis by computing the vertical, horizontal and diagonal <u>projections</u> of the object to get at the 1st & 2nd moments which are all that is necessary for position & orientation.

Run-length coding: store sequences of 1's and 0's

1 1 1 0 0 0 0 1 1 0 1 1 1 1 — 14 bits
3 4 2 1 4 — 11 bits

→ really fast computations:

$\text{Area} = \sum_{\text{all ones}}$ $\bar{y}: A\bar{y} = \sum y h_i$ ($h_i$ ← horizontal projection)

## CHAPTER FOUR: Binary Images: Topological Properties

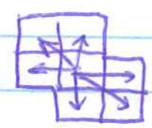
How do we look at area/pos/orientation when we have multiple objects?

→ 1) Labeling components (bucket-fill approach)

→ Q: Use 4-way or 8-way connectedness?

Problem — neither is perfect in C→D

→ 6-connectedness:



→ works better!

→ it is essentially a conversion to unambiguous hexagonal tessellation.

2) Sequential Labelling ~~[illegible]~~

→ check  for labels, ~~[illegible]~~

## Using Parallel Speedups: Local Counting

→ Σ counters = Area
Σ XOR = Perimeter
Euler #

} all have so-called "additive set property" allowing local calculation

 $f(X) + f(Y) = f(X \cup Y) + f(X \cap Y)$

Iterative Modification

↳ instead of adding local results, create a new image

↓

create a skeleton, which retains connectivity w/ min area

## CHAPTER FIVE : Regions & Image Segmentation

How can we automatically pick a threshold for converting grayscale to binary?

↓

brightness distribution function : $p(x)$

% of image brighter than $x$ but < $\delta x$

Salt and pepper noise : static on binary images

Color can be used to increase separation distance (think separability in high-D space!)

## CHAPTER SIX : Image Processing : Continuous Images

LSI Systems → derivative filters are good for preprocessing edge detection

↑ shift = time

2D convolution $f * h = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-\varepsilon, y-n)\,h(\varepsilon, n)\,d\varepsilon\,dn$

All LSI systems can be characterized by a system functional $H$