EE 417 Introduction to Computer Vision / EE 569 3D vision

Line Detection and Hough Transform

Instructor: Associate Prof. Mehmet Keskinöz

Communication Theory &Technologies (CTT) Group,
Electronics Engineering Program,
Computer Science and Engineering,
Cyber-Security Program
Faculty of Engineering and Natural Sciences

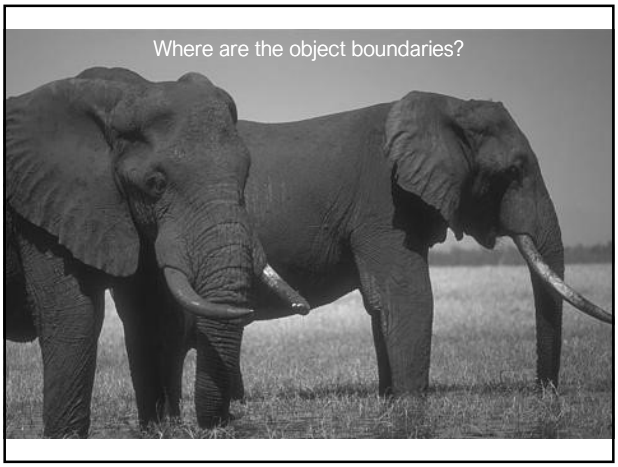
Email: keskinoz@sabanciuniv.edu

Sabancı Üniversitesi

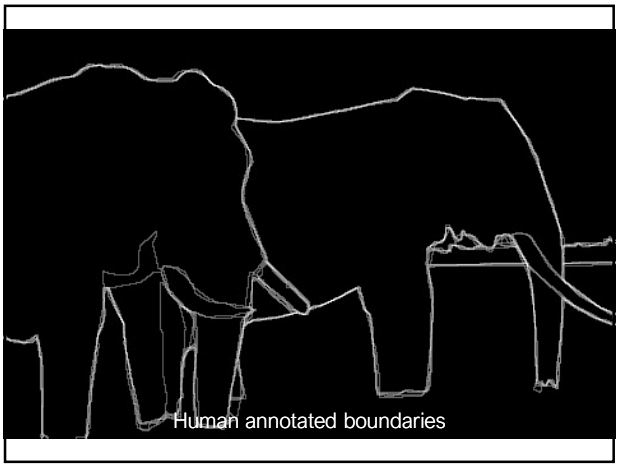
1

Finding boundaries

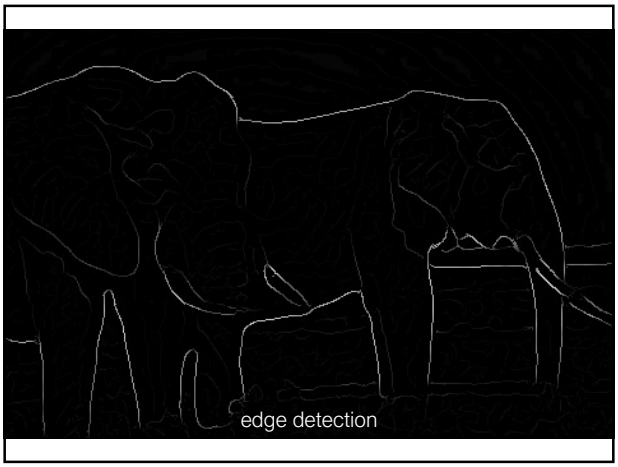
2



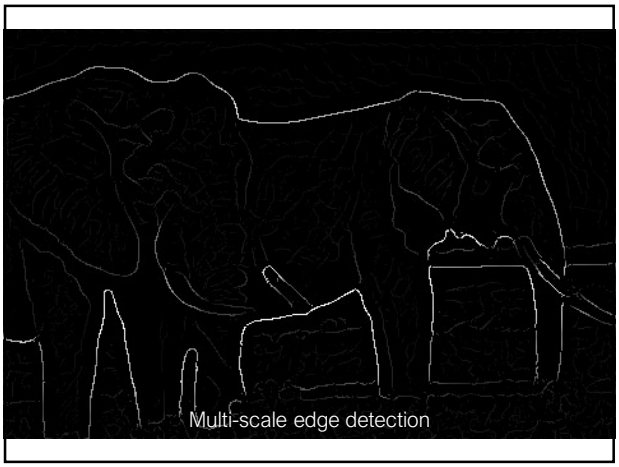
3



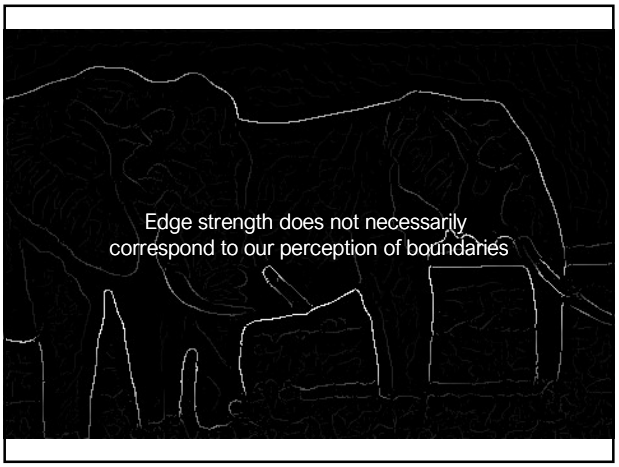
4



5

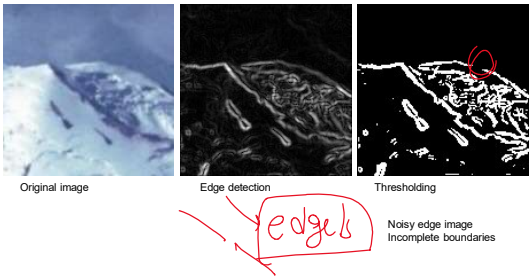


6



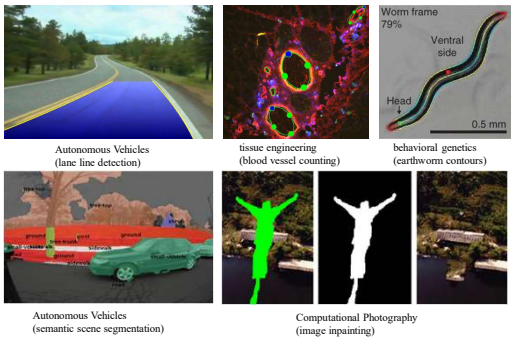
7

Lines are hard to find



8

Applications



9

Contours: Lines and Curves

- Edge detectors find "edgels" (pixel level)
- To perform *image analysis* :
  - edgels must be grouped into entities such as contours (higher level).
  - Canny does this to certain extent: the detector finds chains of edgels.

10

Line fitting

11

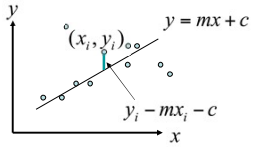
Line fitting

Given: Many (x_i, y_i) pairs

Find: Parameters (m, c)

Minimize: Average square distance:

$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$



12

Line fitting

Given: Many (x_i, y_i) pairs

Find: Parameters (m, c)

Minimize: Average square distance:

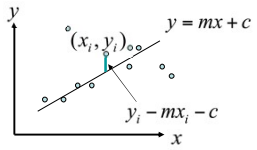
$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$

Using:

$$\frac{\partial E}{\partial m} = 0 \quad \& \quad \frac{\partial E}{\partial c} = 0$$

Note:

$$\bar{y} = \frac{\sum_i y_i}{N} \qquad \bar{x} = \frac{\sum_i x_i}{N}$$



$$c = \bar{y} - m\bar{x}$$

$$m = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

What are some problems with the approach?

13

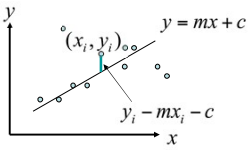
Line fitting

Given: Many (x_i, y_i) pairs

Find: Parameters (m, c)

Minimize: Average square distance:

$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$



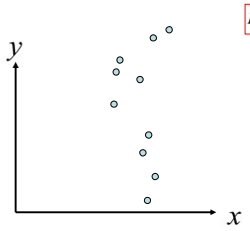
How can we solve this minimization?

14

Problems with parameterizations

Where is the line that minimizes E?

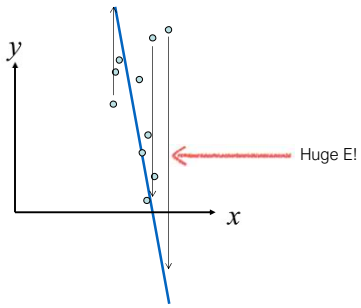
$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$



15

Problems with parameterizations

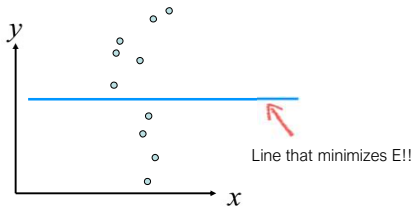
Where is the line that minimizes E? $E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$



16

Problems with parameterizations

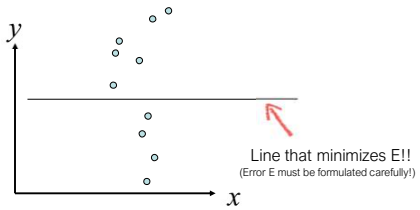
Where is the line that minimizes E? $E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$



17

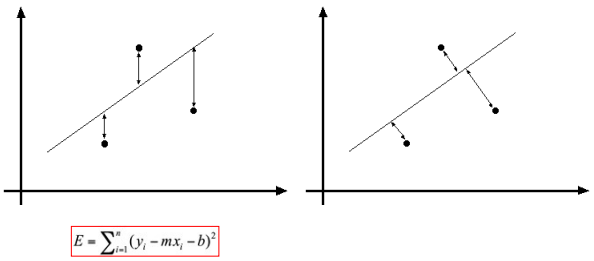
Problems with parameterizations

Where is the line that minimizes E?



How can we deal with this?

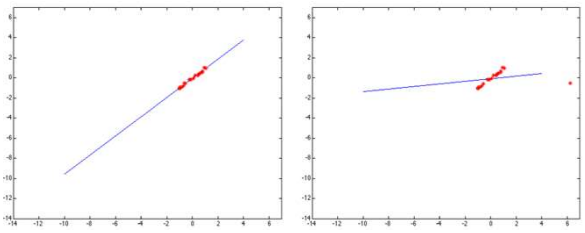
18



$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

19

Problems with noise



Least-squares error fit

Squared error heavily penalizes outliers

20

Model fitting is difficult because…

- **Extraneous data:** clutter or multiple models
  - We do not know what is part of the model?
  - Can we pull out models with a few parts from much larger amounts of background clutter?
- **Missing data:** only some parts of model are present
- **Noise**

- Cost:
  - It is not feasible to check all combinations of features by fitting a model to each possible subset

So what can we do?

21

Hough Transform

Line parameterizations

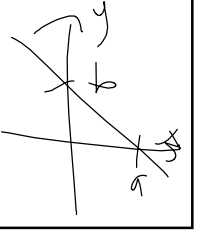
22

Equations of Line

1) $y = mx + b$ → y-intercept

↘ slope

2) $\frac{x}{a} + \frac{y}{b} = 1$



23

Slope intercept form

$$y = mx + b$$

slope y-intercept

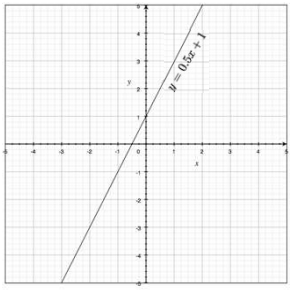
What are m and b?

24

Slope intercept form

$$y = mx + b$$

slope y-intercept



25

Double intercept form

$$\frac{x}{a} + \frac{y}{b} = 1$$

x-intercept y-intercept

What are x and y?

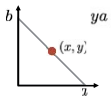
26

Double intercept form

$$\frac{x}{a} + \frac{y}{b} = 1$$

x-intercept y-intercept

Derivation:

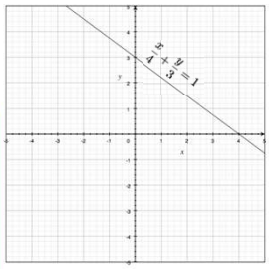


(Similar slope) $\frac{y-b}{x-0} = \frac{0-y}{a-x}$

$$ya + yx - ba + bx = -yx$$

$$ya + bx = ba$$

$$\frac{y}{b} + \frac{x}{a} = 1$$



27

Normal Form

$$x\cos\theta + y\sin\theta = \rho$$

What are rho and theta?

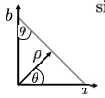
28

Normal Form

3-)

$$x\cos\theta + y\sin\theta = \rho$$

Derivation:

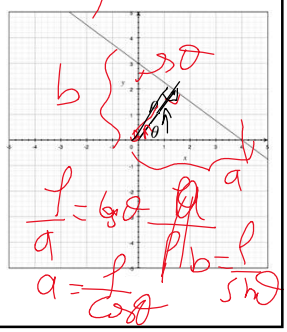


$$\cos\theta = \frac{\rho}{a} \rightarrow a = \frac{\rho}{\cos\theta}$$

$$\sin\theta = \frac{\rho}{b} \rightarrow b = \frac{\rho}{\sin\theta}$$

plug into: $\frac{x}{a} + \frac{y}{b} = 1$

$$x\cos\theta + y\sin\theta = \rho$$



29

Hough transform

30

Hough transform

- Generic framework for detecting a parametric model
- Edges don't have to be connected
- Lines can be occluded
- Key idea: edges **vote** for the possible models

31

Image and parameter space

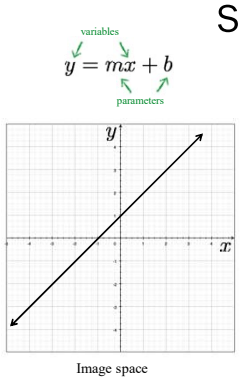


Image space

32

Image and parameter space

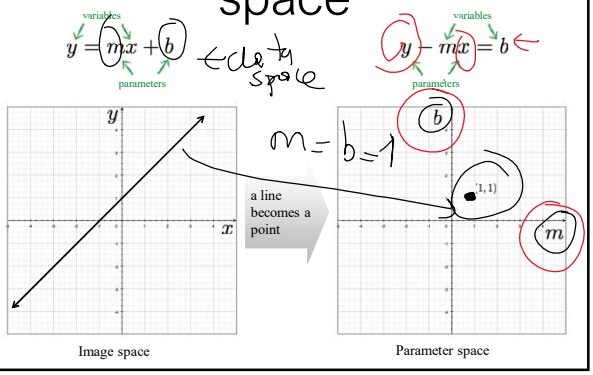
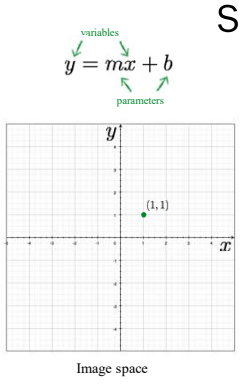


Image space Parameter space

33

Image and parameter space



What would a point in image space become in parameter space?

Image space

34

Image and parameter space

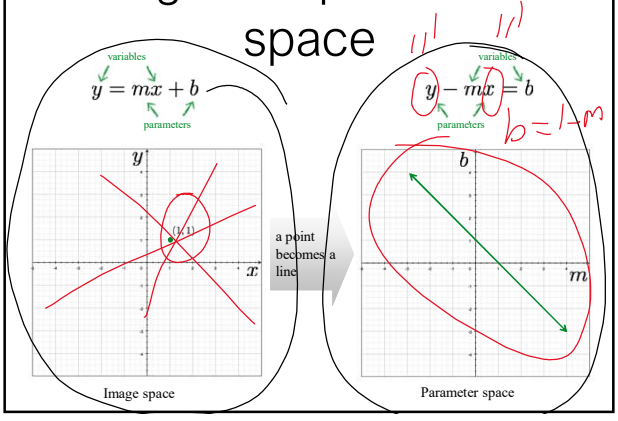


Image space Parameter space

35

Image and parameter space

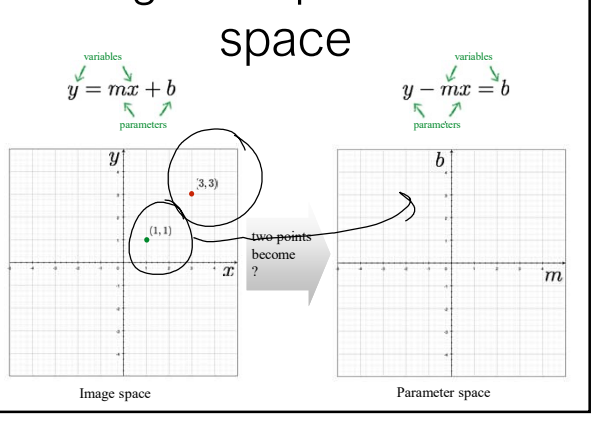
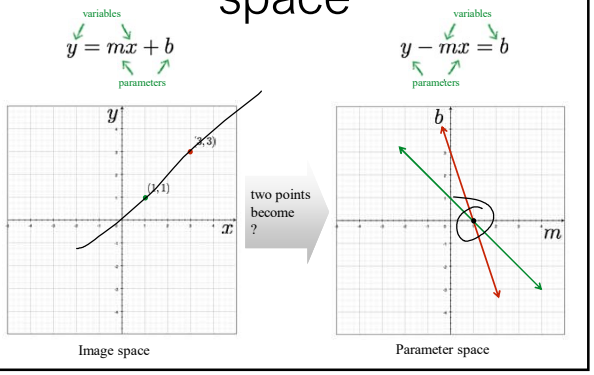


Image space Parameter space

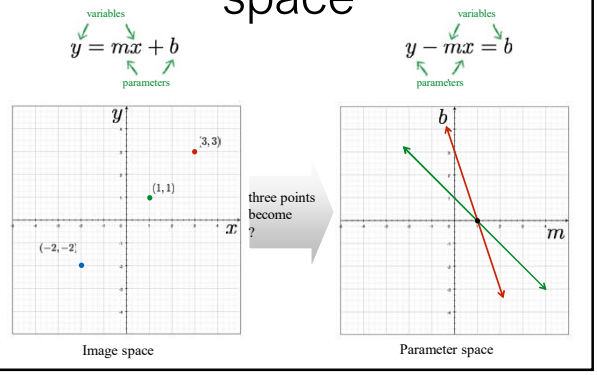
36

Image and parameter space



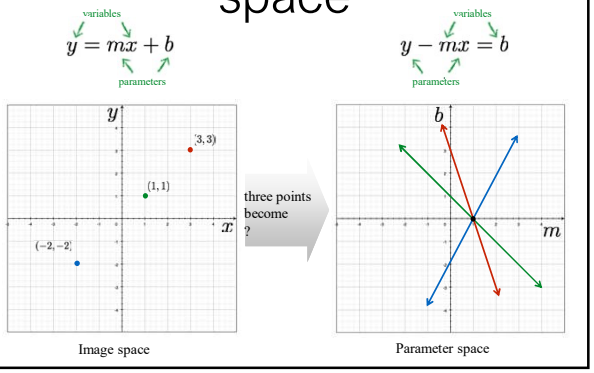
37

Image and parameter space



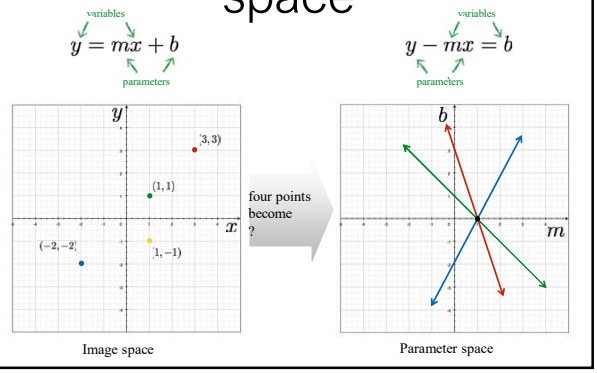
38

Image and parameter space



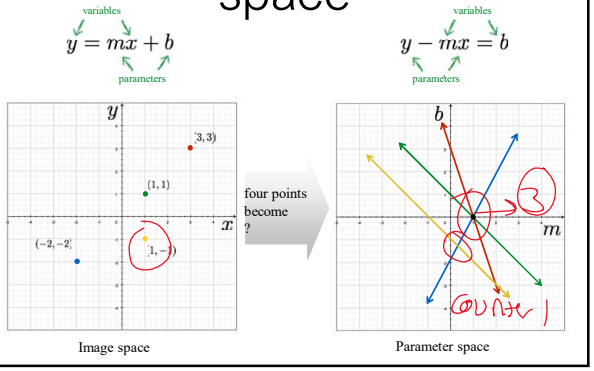
39

Image and parameter space



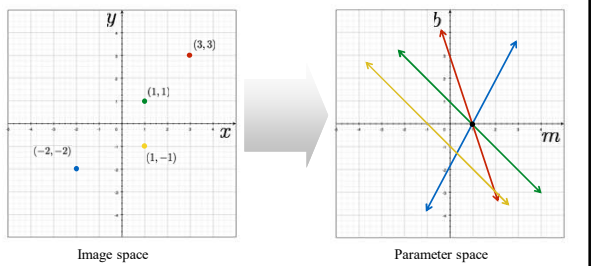
40

Image and parameter space



41

How would you find the best fitting line?



Is this method robust to measurement noise?

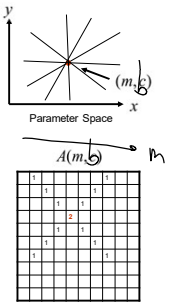
Is this method robust to outliers?

42

Line Detection by Hough Transform

Algorithm:

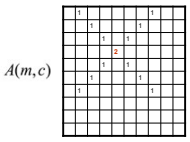
1. Quantize Parameter Space (m,c)
2. Create Accumulator Array $A(m,c)$
3. Set $A(m,c) = 0 \quad \forall m,c$
4. For each image edge (x_i, y_i)
   For each element in $A(m,c)$
   If (m,c) lies on the line: $c = -x_i m + y_i$
   Increment $A(m,c) = A(m,c) + 1$
5. Find local maxima in $A(m,c)$



43

Problems with parameterization

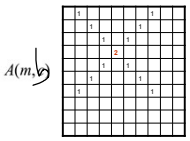
How big does the accumulator need to be for the parameterization (m,c)?



44

Problems with parameterization

How big does the accumulator need to be for the parameterization (m,c)?



The space of m is huge!

$$-\infty \le m \le \infty$$

The space of c is huge!

$$-\infty \le c \le \infty$$

45

Better Parameterization

Use normal form:

$$x\cos\theta + y\sin\theta = \rho$$

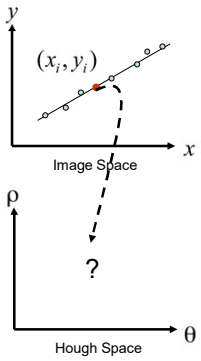
Given points (x_i, y_i) find (ρ, θ)

Hough Space Sinusoid

$0 \le \theta \le 2\pi$

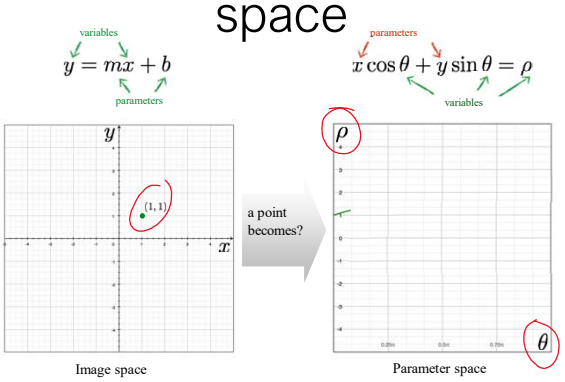
$0 \le \rho \le \rho_{max}$

(Finite Accumulator Array Size)



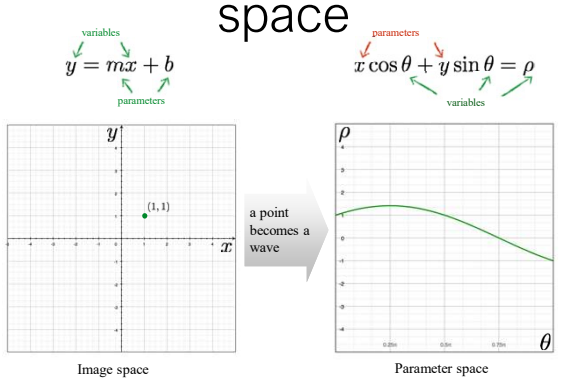
46

Image and parameter space



47

Image and parameter space



48

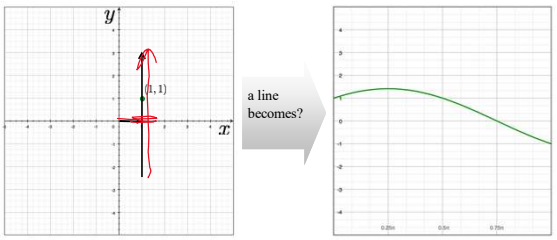
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes?

Image space | Parameter space

49

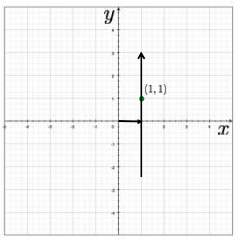
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes a point

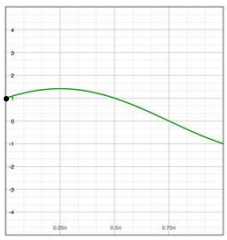


Image space | Parameter space

50

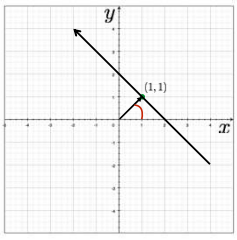
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes?

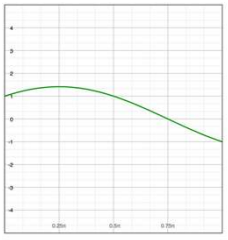


Image space | Parameter space

51

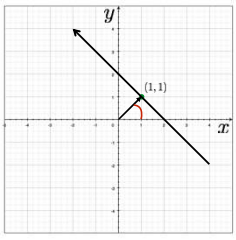
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes a point

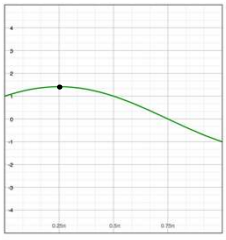


Image space | Parameter space

52

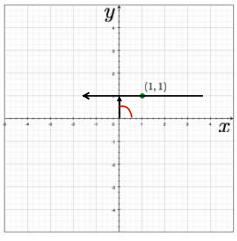
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes a point

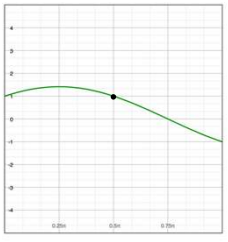


Image space | Parameter space

53

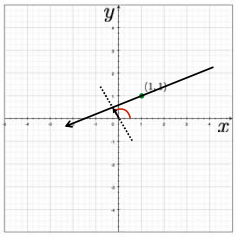
Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$



a line becomes a point

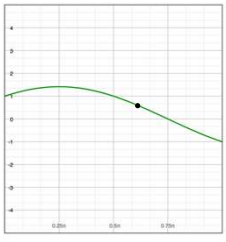


Image space | Parameter space

54

Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$

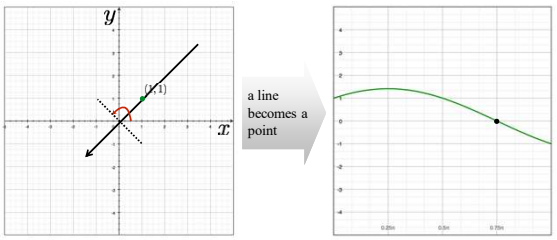


Image space | Parameter space

55

Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$

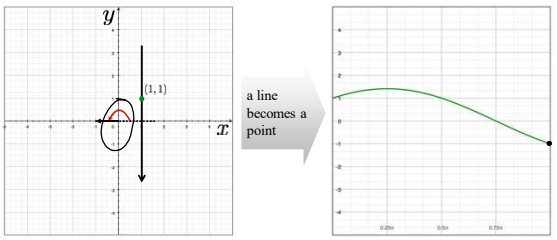


Image space | Parameter space

56

Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$

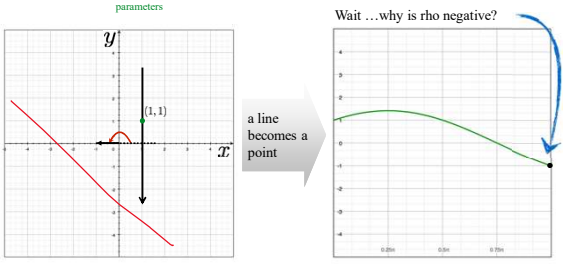


Image space | Parameter space

57

Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$

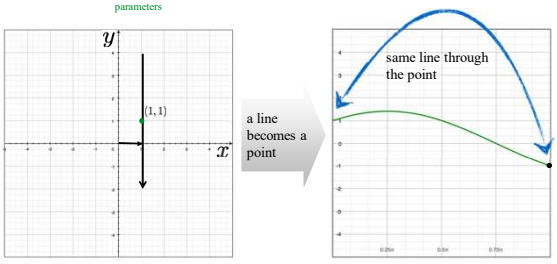


Image space | Parameter space

58

There are two ways to write the same line:

Positive rho version:

$x\cos\theta + y\sin\theta = \rho$

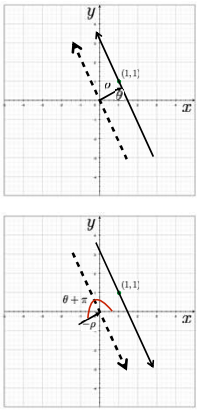
Negative rho version:

$x\cos(\theta + \pi) + y\sin(\theta + \pi) = -\rho$

Recall:

$\sin(\theta) = -\sin(\theta + \pi)$

$\cos(\theta) = -\cos(\theta + \pi)$



59

Image and parameter space

variables

$y = mx + b$

parameters

$x\cos\theta + y\sin\theta = \rho$

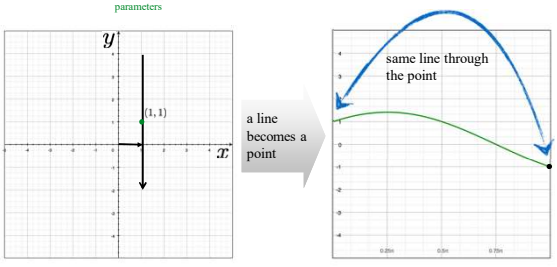


Image space | Parameter space

60

Image and parameter space

variables

$y = mx + b$

parameters

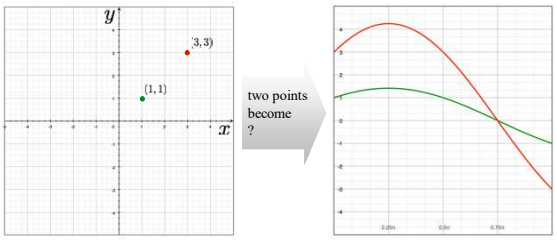


Image space Parameter space

61

Image and parameter space

variables

$y = mx + b$

parameters

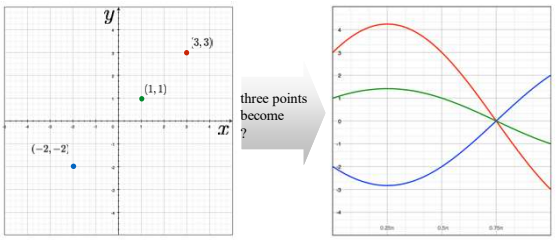


Image space Parameter space

62

Image and parameter space

variables

$y = mx + b$

parameters

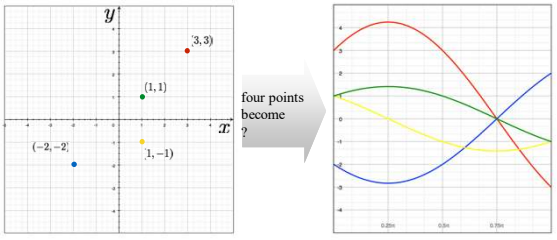


Image space Parameter space

63

Implementation

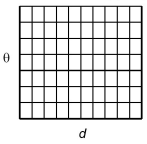
1. Initialize accumulator H to all zeros

2. For each edge point (x,y) in the image
   For θ = 0 to 180
   ρ = x cos θ + y sin θ
   H(θ, ρ) = H(θ, ρ) + 1
   end
   end

3. Find the value(s) of (θ, ρ) where H(θ, ρ) is a local maximum

4. The detected line in the image is given by
   ρ = x cos θ + y sin θ

H: accumulator array (votes)



NOTE: Watch your coordinates. Image origin is top left!

64

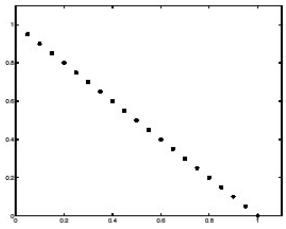
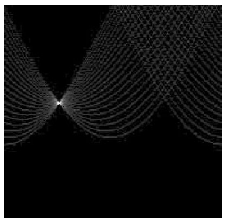


Image space

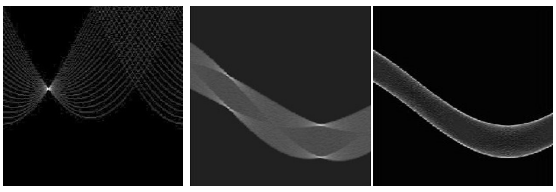


Votes

65

Basic shapes

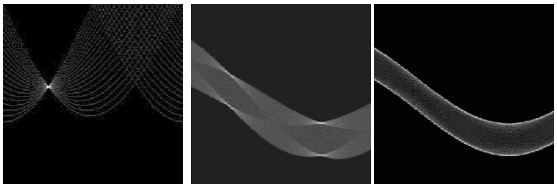
(in parameter space)



can you guess the shape?

66

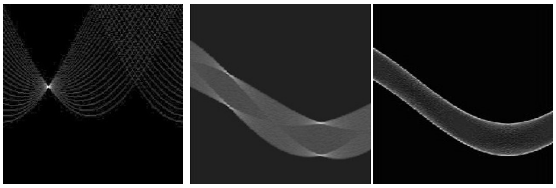
Basic shapes
(in parameter space)



line

67

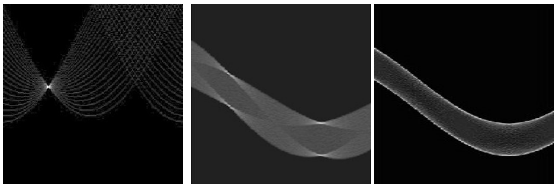
Basic shapes
(in parameter space)



line rectangle

68

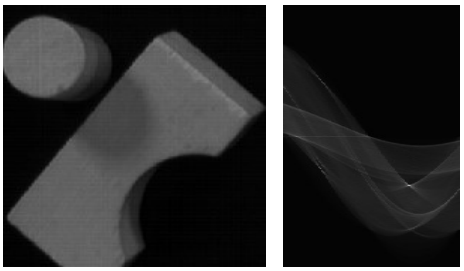
Basic shapes
(in parameter space)



line rectangle circle

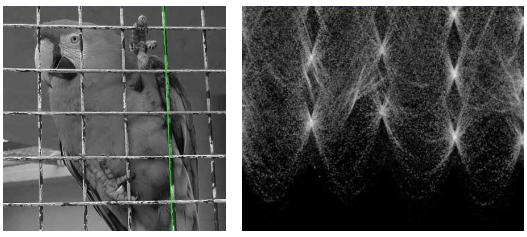
69

Basic Shapes



70

More complex image



71

In practice, measurements are noisy…

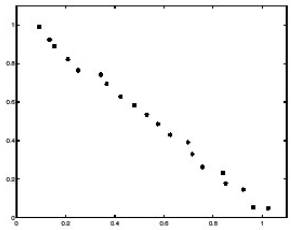
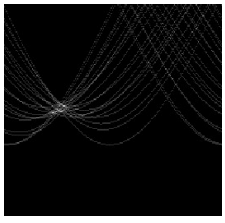
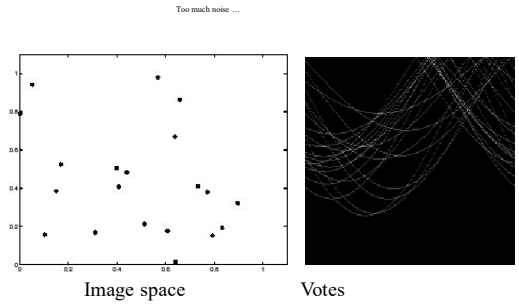


Image space



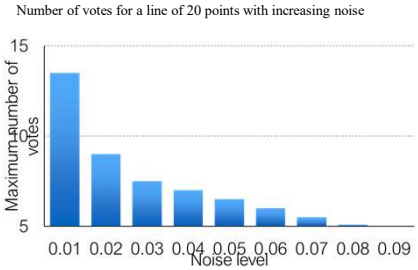
Votes

72



73

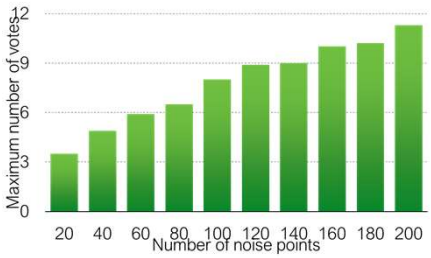
Effects of noise level



More noise, fewer votes (in the right bin)

74

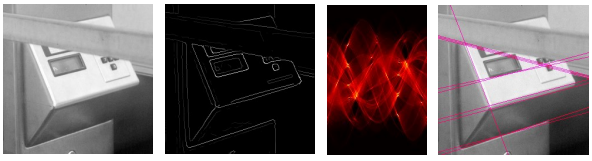
Effect of noise points



More noise, more votes (in the wrong bin)

75

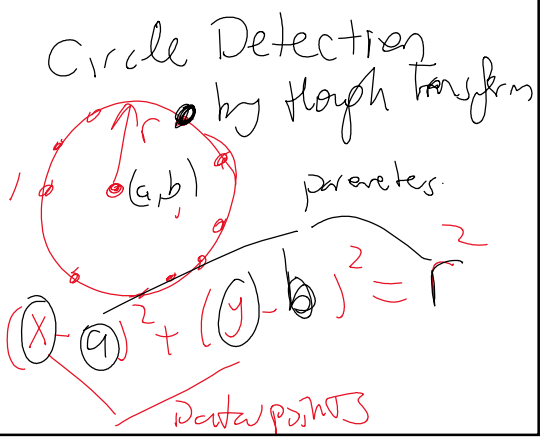
Real-world example



76

Hough Circles

77



78

Let's assume radius known

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

What is the dimension of the parameter space?

79

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

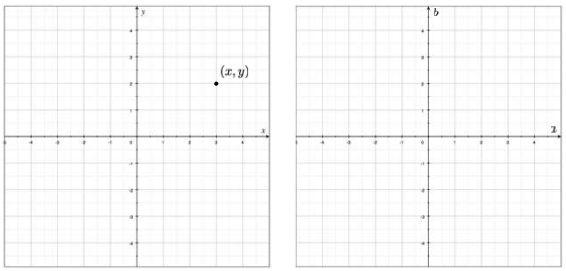


Image space

Parameter space

What does a point in image space correspond to in parameter space?

80

parameters

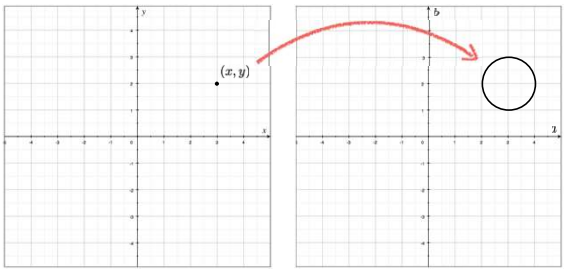
$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables



81

parameters

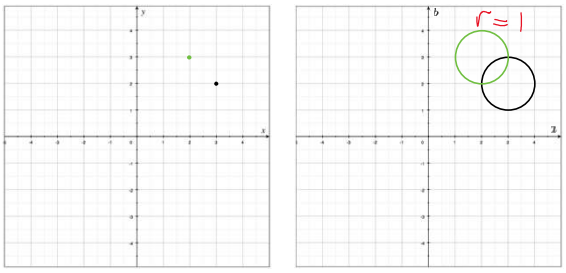
$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables



82

parameters

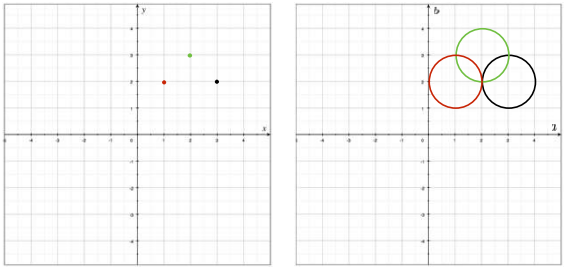
$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables



83

parameters

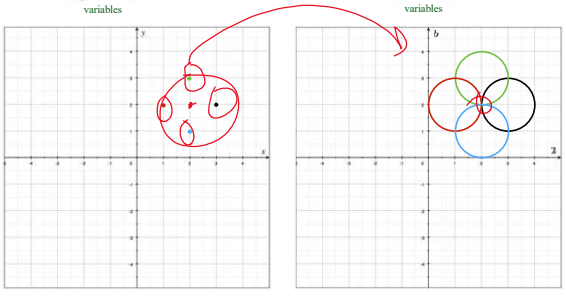
$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables



84

What if radius is unknown?

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

85

What if radius is unknown?

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

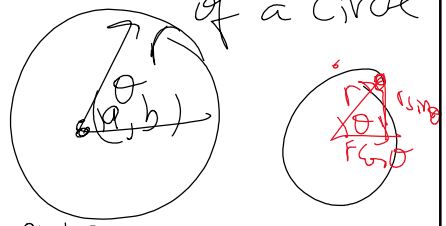
If radius is not known: 3D Hough Space!

Use Accumulator array $A(a,b,r)$

Surface shape in Hough space is complicated

86

Parametric representation of a circle



$x = a + r\cos\theta$

$y = b + r\sin\theta$

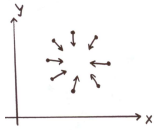
87

Using Gradient Information

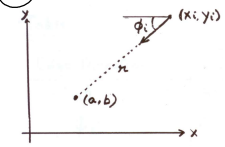
Gradient information can save lot of computation:

Edge Location (x_i, y_i)

Edge Direction ϕ_i



Assume radius is known:



$a = x - r\cos\phi$

$b = y - r\sin\phi$

Need to increment only one point in accumulator!

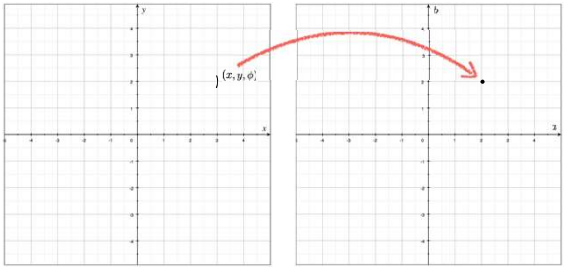
88

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)



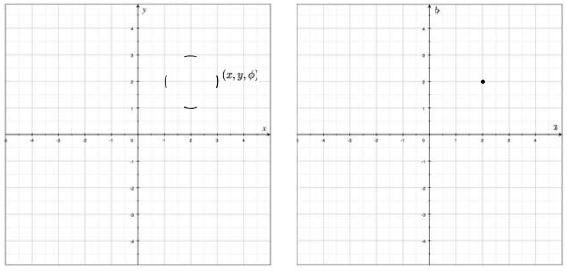
89

$$(x-a)^2+(y-b)^2=r^2$$

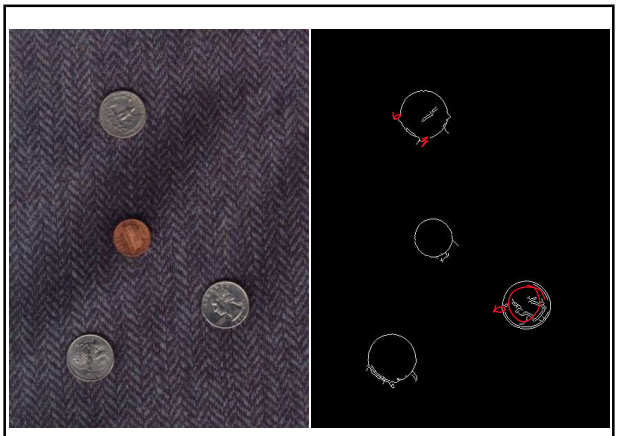
(parameters: a, b; variables: x, y)

$$(x-a)^2+(y-b)^2=r^2$$

(parameters: a, b; variables: x, y)

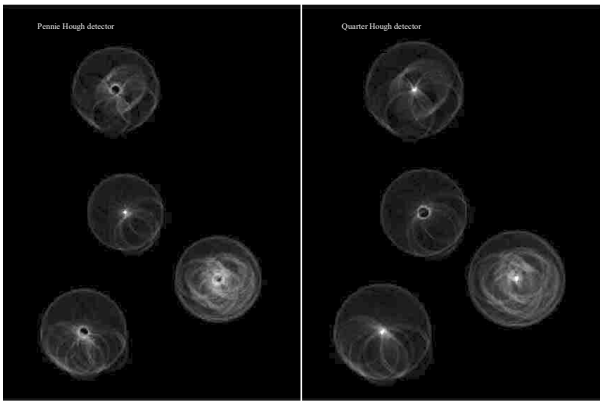


90



91

92



93

The Hough transform …

Deals with occlusion well?
Detects multiple instances?
Robust to noise?
Good computational complexity?
Easy to set parameters?

94

Can you use Hough Transforms for other objects, beyond lines and circles?

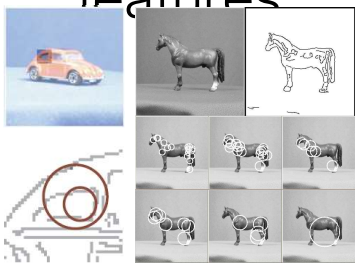
Do you have to use edge detectors to vote in Hough Space?

95

Application of Hough transforms

96

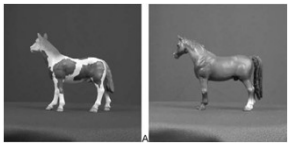
Detecting shape features

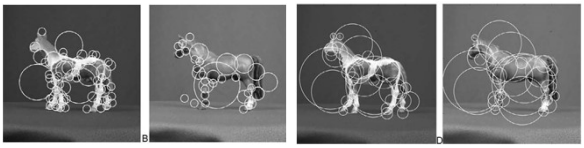


F. Jurie and C. Schmid, Scale-invariant shape features for recognition of object categories, CVPR 2004

97

Original images



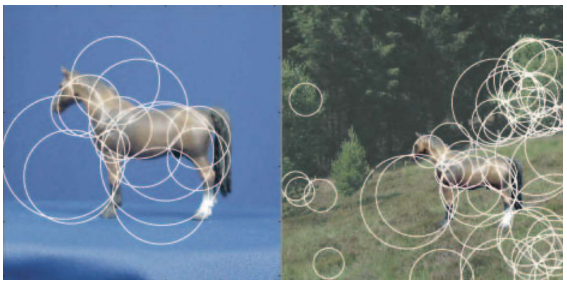


Laplacian circles

Hough-like circles

Which feature detector is more consistent?

98

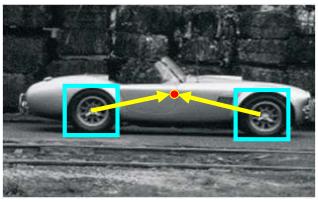


Robustness to scale and clutter

99

Object detection

Index displacements by "visual codeword"



training image

visual codeword with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model,
ECCV Workshop on Statistical Learning in Computer Vision 2004

100

101

References

Basic reading:
- Szeliski textbook, Sections 4.2, 4.3.

102

More Image Features

(Grouping edges)

103

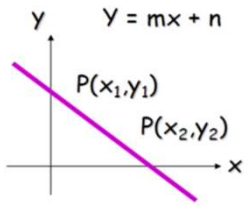
Contours: Lines and Curves

- Edge detectors find "edgels" (pixel level)
- To perform *image analysis* :
  - edgels must be grouped into entities such as contours (higher level).
  - Canny does this to certain extent: the detector finds chains of edgels.

104

Line detection

- Mathematical model of a line:



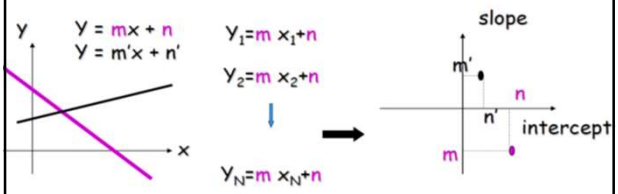
$Y_1 = m\,x_1 + n$

$Y_2 = m\,x_2 + n$

$Y_N = m\,x_N + n$

105

Image and Parameter Spaces



Line in Img. Space ~ Point in Param. Space

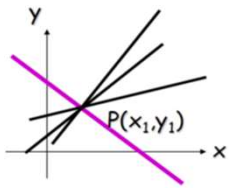
106

Looking at it backwards …

Image space

Fix (m,n), Vary (x,y) - Line $Y = mx + n$

Fix (x_1,y_1), Vary (m,n) – Lines thru a Point $Y_1 = m\,x_1 + n$



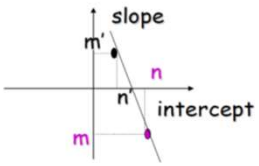
107

Looking at it backwards …

Parameter space

$Y_1 = m\,x_1 + n$ Can be re-written as: $n = -x_1\,m + Y_1$

Fix $(-x_1,y_1)$, Vary (m,n) - Line $n = -x_1\,m + Y_1$



108

Img-Param Spaces

- Image Space
  - Lines
  - Points
  - Collinear points
- Parameter Space
  - Points
  - Lines
  - Intersecting lines

109

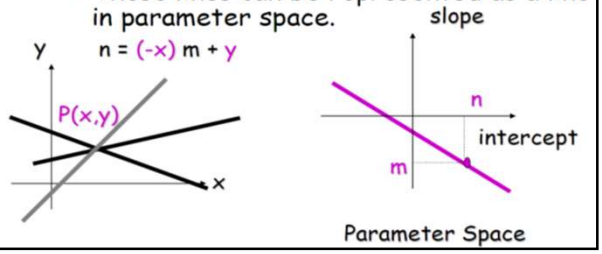
Hough Transform Technique

- H.T. is a method for detecting straight lines (and curves) in images.
- Main idea:
  - Map a difficult pattern problem into a simple peak detection problem

110

Hough Transform Technique

- Given an edge point, there is an infinite number of lines passing through it (Vary m and n).
  - These lines can be represented as a line in parameter space.

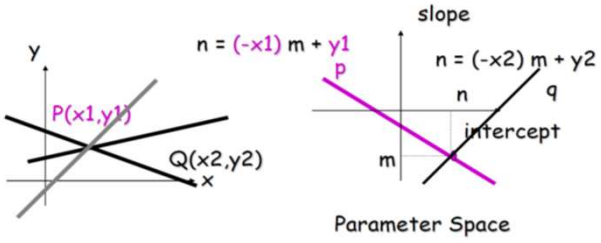


111

Hough Transform Technique

- Given a set of collinear edge points, each of them have associated a line in parameter space.
  - These lines intersect at the point (m,n) corresponding to the parameters of the line in the image space.

112



113

Hough Transform Technique

- At each point of the (discrete) parameter space, count how many lines pass through it.
  - Use an array of counters
  - Can be thought as a " parameter **image**"
- The higher the count, the more edges are collinear in the image space.
  - Find a peak in the counter array
  - This is a "bright" point in the parameter image
  - It can be found by thresholding

114

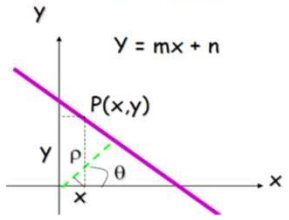
Practical Issues

- The slope of the line is $-\infty<m<\infty$
  - The parameter space is INFINITE
- The representation y = mx + n does not express lines of the form x = k

115

Solution:

- Use the "Normal" equation of a line:



$\rho = x\cos\theta + y\sin\theta$

θ Is the line orientation

ρ Is the distance between the origin and the line

116

New Parameter Space

- Use the parameter space (ρ, θ)
- The new space is FINITE
  - $0 < \rho < D$, where D is the image diagonal.
  - $0 < \theta < \pi$
- The new space can represent all lines
  - Y = k is represented with $\rho = k, \theta=90$
  - X = k is represented with $\rho = k, \theta=0$

117

Consequence:

- A Point in Image Space is now represented as a SINUSOID
  - $\rho = x\cos\theta + y\sin\theta$

118

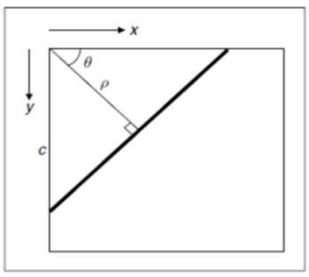
Hough Transform Algorithm

Input is an edge image (E(i,j)=1 for edgels)

1. Discretize θ and ρ in increments of dθ and dρ. Let A(R,T) be an array of integer accumulators, initialized to 0.
2. For each pixel E(i,j)=1 and h=1,2,...T do
   1. ρ = i cos(h * dθ) + j sin(h * dθ)
   2. Find closest integer k corresponding to ρ
   3. Increment counter A(k,h) by one
3. Find local maxima in A(R,T)

119

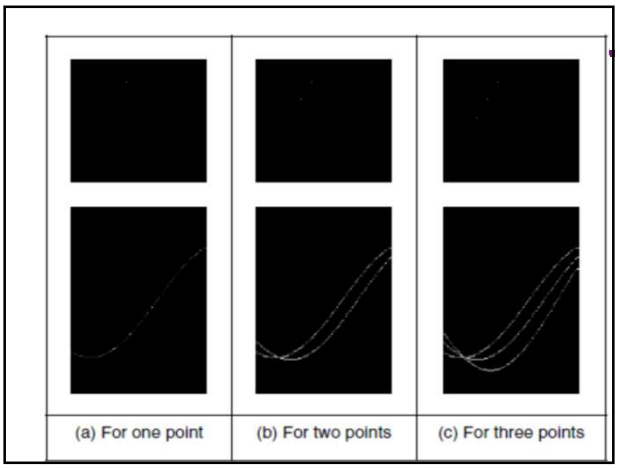
Summary: Hough Transform for Lines



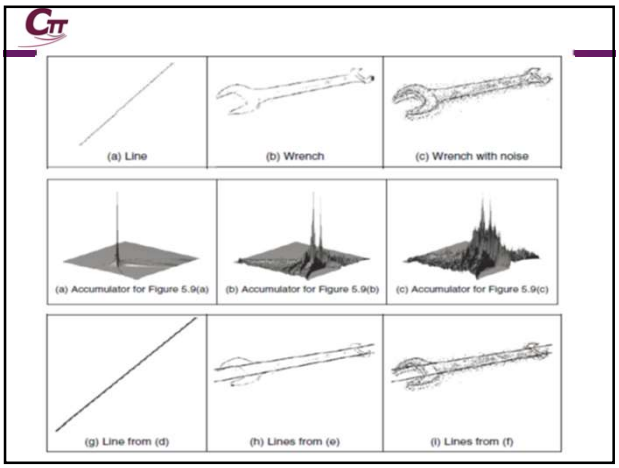
$\rho = x\cos(\theta) + y\sin(\theta)$

Figure 5.10 Polar consideration of a line

120



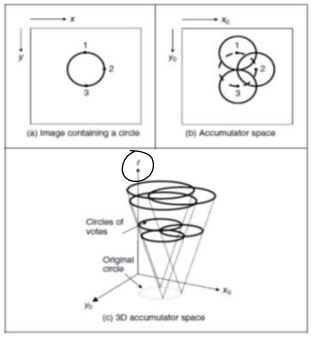
121



122

Hough Transform for Circles

$$(x-x_0)^2+(y-y_0)^2=r^2$$



123

Hough Transform for Circles

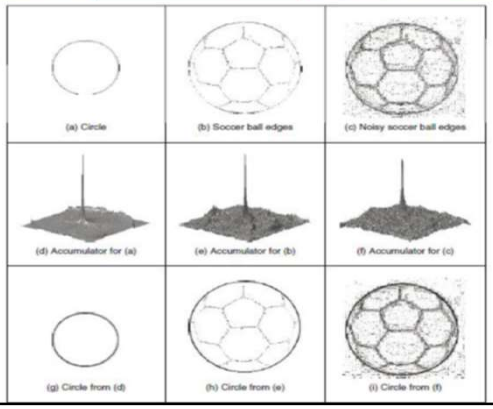
Implicit Form: $(x-x_0)^2+(y-y_0)^2=r^2$

Parametric Form: $x=x_0+r\cos(\theta) \quad y=y_0+r\sin(\theta)$

Parameters: $x_0=x-r\cos(\theta) \quad y_0=y-r\sin(\theta)$

124

HT Circles



125

Hough Transform Speed Up

- If we know the orientation of the edge – usually available from the edge detection step
  - We fix theta in the parameter space and increment **only one** counter!

126

Hough Transform for Curves

- The H.T. can be generalized to detect any curve that can be expressed in parametric form:
  - Y = f(x, a1,a2,…ap)
  - a1, a2, … ap are the parameters
  - The parameter space is p-dimensional
  - The accumulating array is LARGE!

127

H.T. Summary

- H.T. is a "voting" scheme
  - points vote for a set of parameters describing a line or curve.
- The more votes for a particular set
  - the more evidence that the corresponding curve is present in the image.
- Can detect MULTIPLE curves in one shot.
- Computational cost increases with the number of parameters describing the curve.

128

```
%Polar Hough Transform for Lines

function HTPLine(inputimage)

%image size
[rows,columns]=size(inputimage);

%accumulator
rmax=round(sqrt(rows^2+columns^2));
acc=zeros(rmax,180);

%image
for x=1:columns
  for y=1:rows
    if(inputimage(y,x)==0)
      for m=1:180
        r=round(x*cos((m*pi)/180)
          +y*sin((m*pi)/180));
        if(r<rmax & r>0)
          acc(r,m)= acc(r,m)+1; end
      end
    end
  end
end
```

129

HT Algorithm for Circles

```
%Hough Transform for Circles

function HTCircle(inputimage,r)

%image size
[rows,columns]=size(inputimage);

%accumulator
acc=zeros(rows,columns);

%image
for x=1:columns
  for y=1:rows
    if(inputimage(y,x)==0)
      for ang=0:360
        t=(ang*pi)/180;
        x0=round(x-r*cos(t));
        y0=round(y-r*sin(t));
        if(x0<columns & x0>0 & y0<rows & y0>0)
          acc(y0,x0)=acc(y0,x0)+1;
        end
      end
    end
  end
end
```

130