



# Computer Vision

## Distance and Hough Transforms

27 August 2008

Copyright © 2001 – 2008 by  
Noordelijke Hogeschool Leeuwarden and Van de Loosdrecht Machine Vision  
All rights reserved

j.van.de.loosdrecht@tech.nhl.nl, jaap@vdlmv.nl

## Distance and Hough Transforms

### Overview:

- **Distance transform**
  - Pixel distance
  - Euclidean distance
- **Hough transform**
  - Circle
  - Line

28-aug-08

Distance and Hough Transforms

2

### Distance transform

**DistanceT srcImage destImage connected**

**This operator works on Binary Image (= source) and initialises a greyscale image (= destination).**

**Each pixel in an object is assigned a pixel value equal to its (eight or four connected) distance to the nearest background pixel.**

**Usage:**

- **Fast alternative for repetitive erosion**

28-aug-08

Distance and Hough Transforms

3

### Demonstration Distance Transform

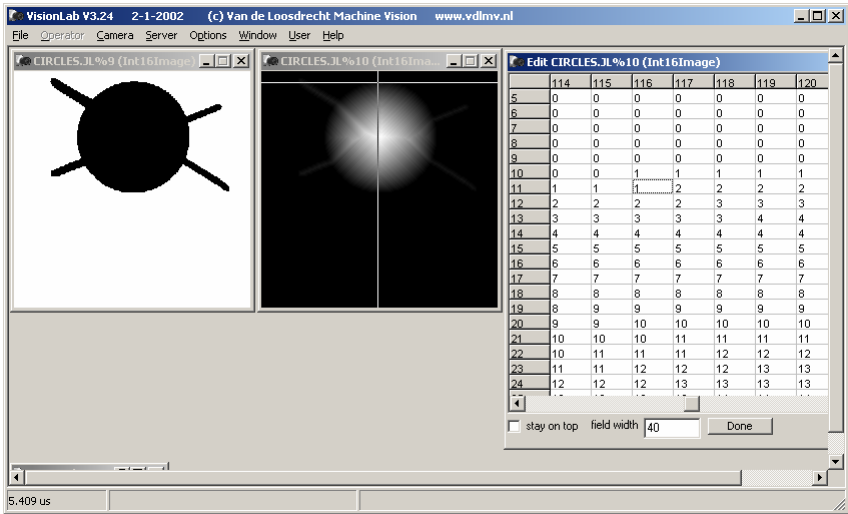
- Open image circles.jl
- Threshold 0 100
- Distance transform EightConnected
  - Analyse image with Edit (pixel 116,11 = 1)
- Distance transform FourConnected
  - Analyse image with Edit (pixel 116,11 = 2)
- Fast alternative for repetitive erosion:
- Use script dt\_speed.jls, use single step mode to show timer results
- Apply threshold 5 100 on EightConnected distance transform
- Apply 4 x Erosion with full 3x3 mask on thresholded image, note time needed for operations

28-aug-08

Distance and Hough Transforms

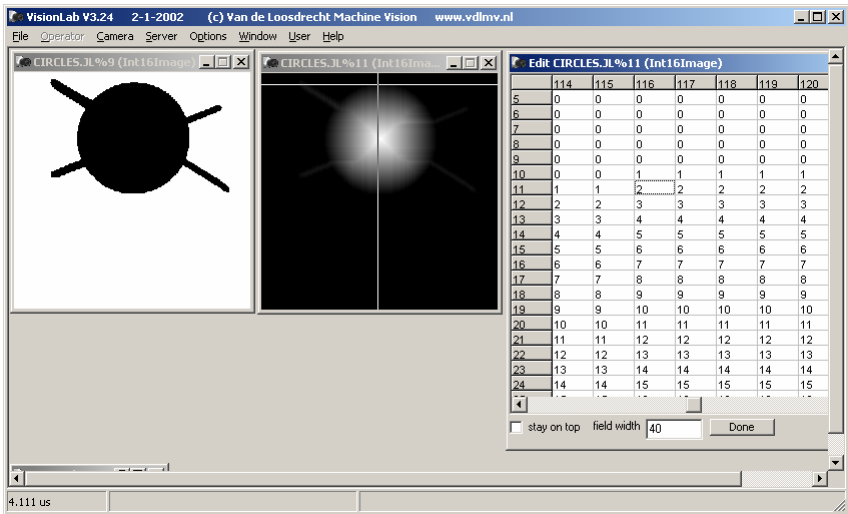
4

Distance transform EightConnected



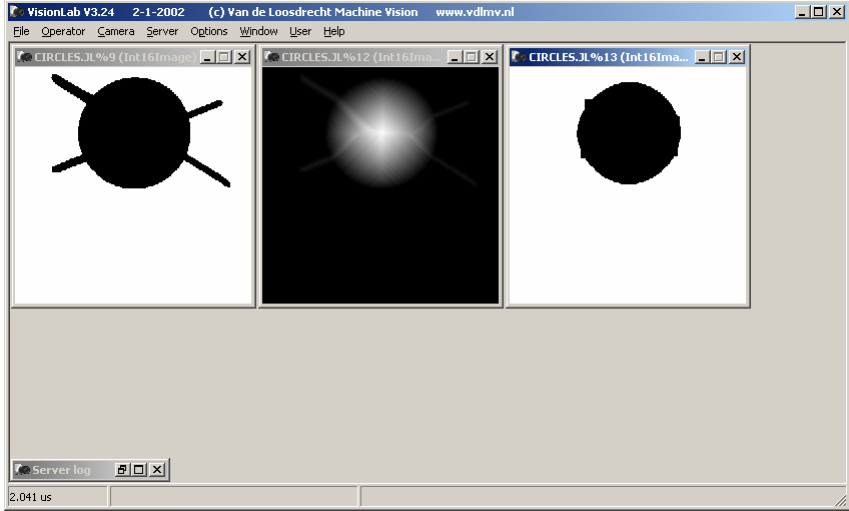
28-aug-08 Distance and Hough Transforms 5

Distance transform FourConnected



28-aug-08 Distance and Hough Transforms 6

**Fast alternative for repetitive erosion:  
Threshold 5 100 on EightConnected distance transform**



28-aug-08 Distance and Hough Transforms 7

**Euclidean Distance Transform**

**EuclideanDistanceT srcImage destImage mask result precision ydivx**

**This operator works on Binary Image (= source) and initialises a greyscale image (= destination).**

**Each pixel in an object is assigned a pixel value equal to its Euclidean distance to the nearest background pixel.**

28-aug-08 Distance and Hough Transforms 8

### Euclidean Distance Transform

#### Implementation notes:

- calculations in “41 base number”,  $58/41 = \sqrt{2}$
- accuracy can be increased by using bigger mask
- ydivx gives the ratio of the pixelsize = 1 for square pixels

28-aug-08

Distance and Hough Transforms

9

### Euclidean Distance Transform (\*)

0	148	130	0	130	148	0
148	0	92	0	92	0	148
130	92	58	41	58	92	130
0	0	41	0	41	0	0
130	92	58	41	58	92	130
148	0	92	0	92	0	148
0	148	130	0	130	148	0

3x3 mask

5x5 mask

7x7 mask

- Masks for square pixels and base number 41
- Maximum blob diameter for Int16Image:  $(2^{15}/41) * 2 + 1 = 1599$  pixels

28-aug-08

Distance and Hough Transforms

10

### Euclidean Distance Transform (\*)

0	148	130	0	130	148	0
148	0	92	0	92	0	148
130	92	58	41	58	92	130
0	0	41	0	41	0	0
130	92	58	41	58	92	130
148	0	92	0	92	0	148
0	148	130	0	130	148	0

3x3 mask  
 5x5 mask  
 7x7 mask

- Sensitive to rotation of the object,  
maximum errors: mask3x3 = 7.9%, mask5x5 = 2.5% and  
mask7x7 = 1.2%
- Example: distance knight move with mask3x3:  $58 + 41 = 99$ .  
Error:  $(99/41) / \sqrt{5} = 1.079$

28-aug-08

Distance and Hough Transforms

11

### Demonstration Euclidean Distance Transform

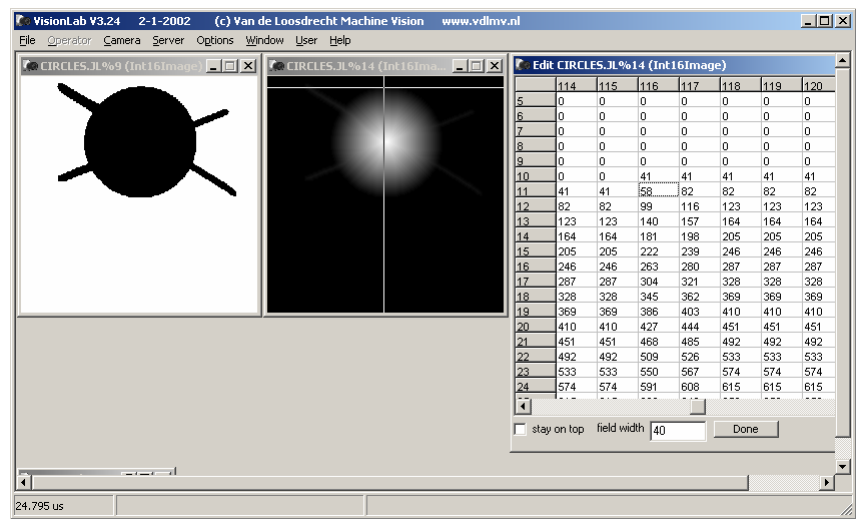
- Open image circles.jl
- Threshold 0 100
- (EuclideanDistanceT EDTMask3x3 NoScaleEDT 41 1 (\*))
- EuclideanDistanceT EDTMask3x3 ScaleEDT 41 1
- (Explain scaling (\*))

28-aug-08

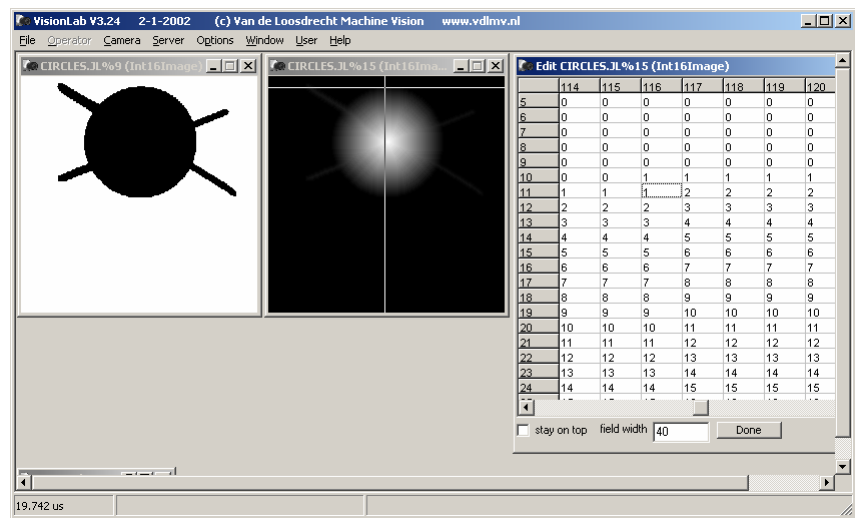
Distance and Hough Transforms

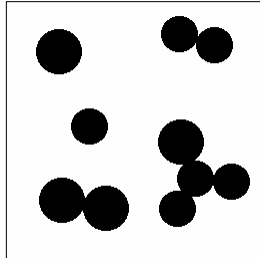
12

EuclideanDistanceT NoScaleEDT (\*)



EuclideanDistanceT ScaleEDT



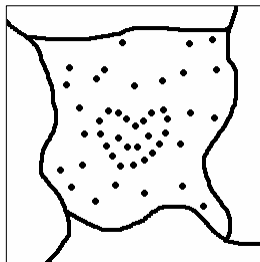
**Exercise 1 Distance Transform**

- Use image `connectedballs.jl`
- Calculate the number of balls in the image
- answer: `connectedballs.jls`

28-aug-08

Distance and Hough Transforms

15

**Exercise 2 Euclidean Distance Transform (\*)**

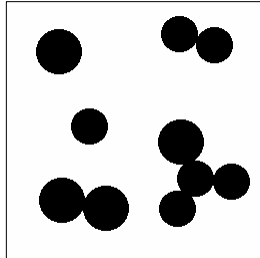
- Use image `city.jl`
- Calculate the average shortest distance from the centre of the cities to the highway
- answer: `city.jls`

28-aug-08

Distance and Hough Transforms

16



**Exercise 3 Euclidean Distance Transform (\*)**

- Use image `connectedballs.jl`
- Calculate “the watershed” between the balls
- answers: `watershed_conballs2.jls` `watershed_conballs.jls`

28-aug-08

Distance and Hough Transforms

17

**Hough Transform**

- Hough Circle Transform
- Hough Line Transform

28-aug-08

Distance and Hough Transforms

18

### Hough Circle Transform

**Task: find the best match for a circle**

<p><b>Image space</b></p> $(x-a)^2 + (y-b)^2 = R^2$ <p><math>a, b, R</math>: known  <math>x, y</math>: unknown</p>	<p><b>Hough space</b></p> $(a-x)^2 + (b-y)^2 = R^2$ <p><math>x, y</math>: known  <math>a, b</math>: unknown, <math>R</math>: fixed</p>
--	--

28-aug-08
Distance and Hough Transforms
19

### Hough Circle Transform

**Algorithm:**

- Initialise Hough space to 0
- For all object pixels 'draw' in Hough space a circle with radius  $R$  by incrementing all pixels in circle.
- Find maximum for position centre

This can be repeated for all radii  $R$  to be searched, maximum over all Hough spaces will give the answer

28-aug-08
Distance and Hough Transforms
20

### Demonstration Hough Circle Transform

- (note: log mode should be normal, not CSV)
- Open image circles.jl
- Threshold 0 100
- Erosion with full 3x3 mask
- Subtract eroded image from thresholded image in order to get a border image
- Hough circle transform on border image with radius 55
- (use gamma correction in point menu with factor 0.25 for better contrast in display with beamer)
- Analyse result with edit pixels
- Use Hough best circle with radius 55 to find position of circle
- An unknown radius can be found with  
FindBestCircle borderimage 50 58 0.1  
(result = centre of gravity radius nrofhits)

28-aug-08

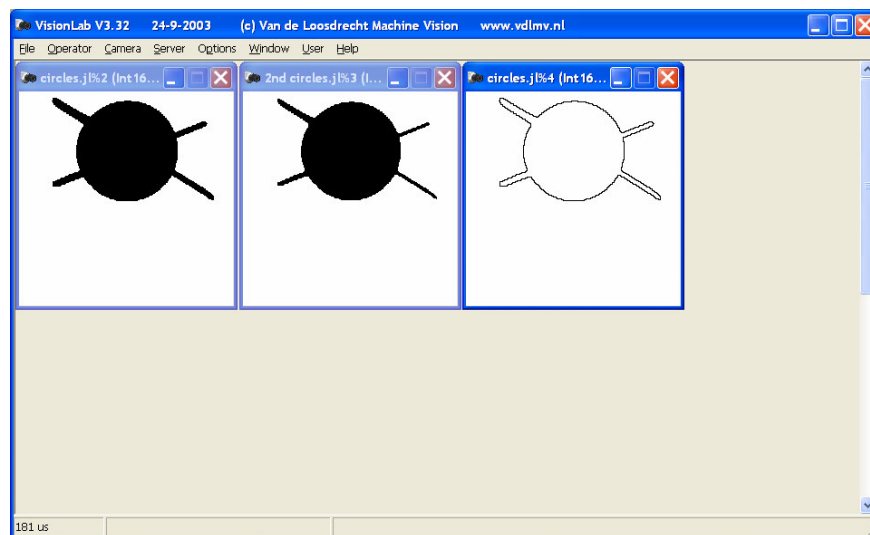
Distance and Hough Transforms

21

Threshold

Erosion

Border

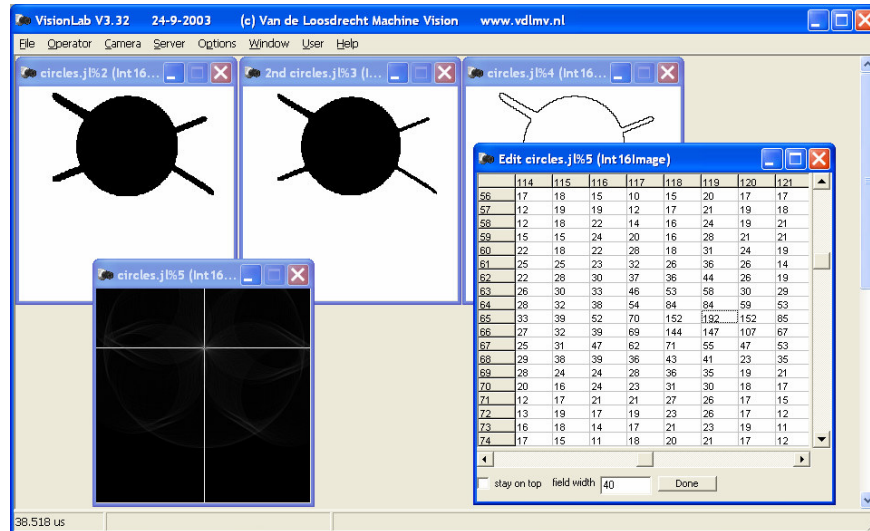


28-aug-08

Distance and Hough Transforms

22

## Hough Circle Transform with radius 55

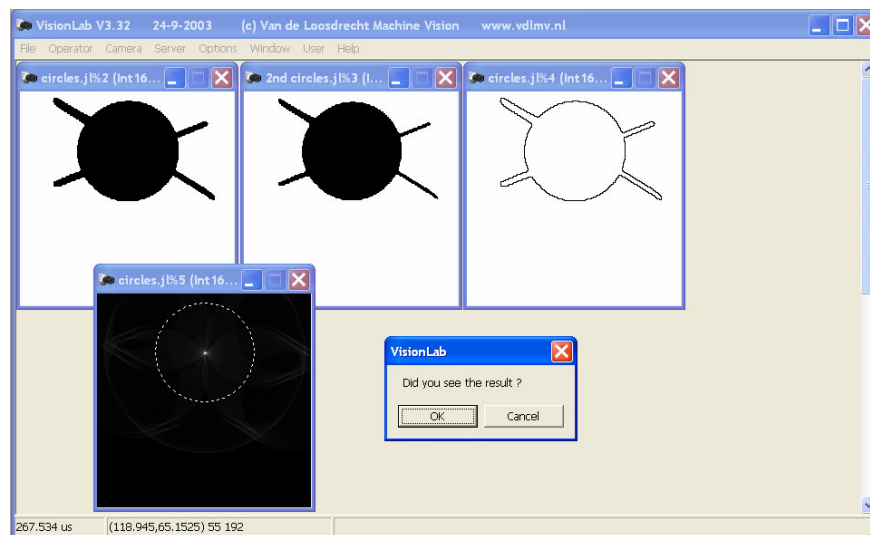


28-aug-08

Distance and Hough Transforms

23

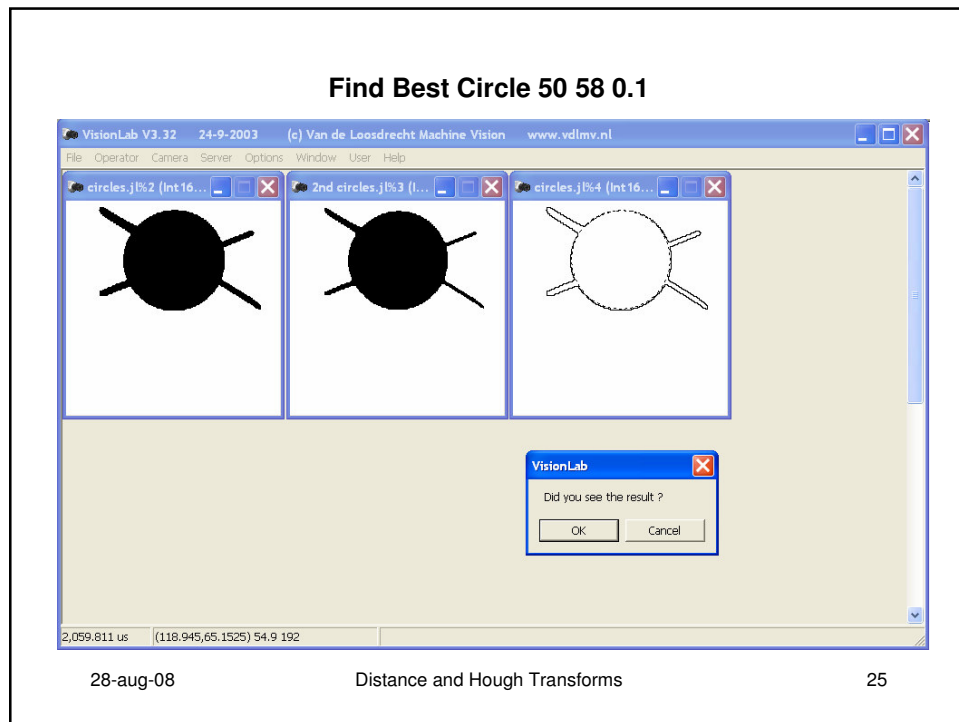
## Hough Best Circle with radius 55 to find position of circle



28-aug-08

Distance and Hough Transforms

24



### Find Best Circle

**findbestcircle imageName minR maxR deltaR**

**findbestcircle** is intended for a binary image

The operator searches in the image for the best match for a circle with minimum radius **minR** and maximum radius **maxR** and with a resolution of **deltaR** pixel.

The operator will give as result the following string:  
(centre co-ordinate) radius numberOfHits.

**Note sub-pixel precision.**

28-aug-08

Distance and Hough Transforms

26

### Find Fast Best Circle

**findfastbestcircle** imageName brightness edgeMin minR maxR  
deltaR

**findfastbestcircle** is intended for a grayscale image.

**findfastbestcircle** has two extra parameters:

- **brightness**: determines whether the circle is dark or light.
- **edgeMin**: the border of the circle is found using the Sharr edge detection method, all edge with a magnitude higher than **edgeMin** are considered as candidate for the circle.

**Note sub-pixel precision.**

28-aug-08

Distance and Hough Transforms

27

### Demonstration Find Fast Best Circle

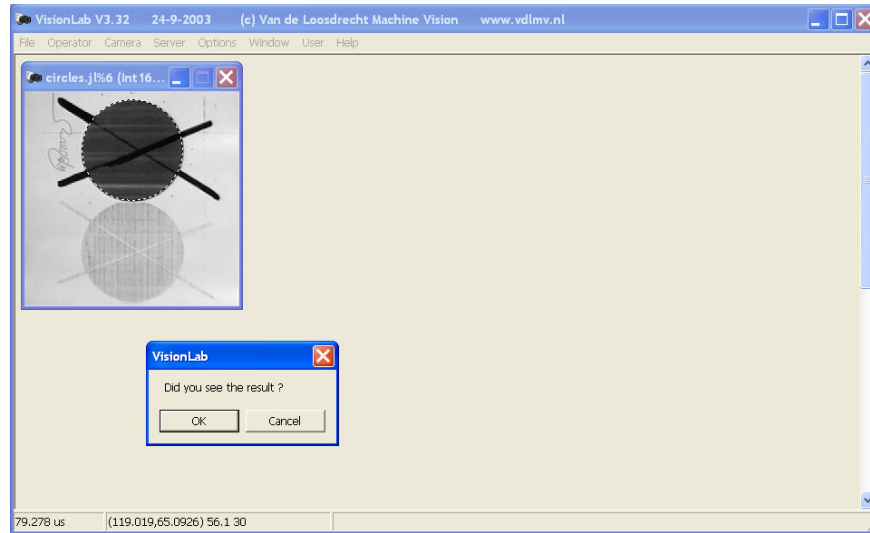
- Open image circles.jl
- **findfastbestcircle** DarkObject 500 50 58 0.1

28-aug-08

Distance and Hough Transforms

28

### Demonstration Find Fast Best Circle



28-aug-08

Distance and Hough Transforms

29

### Implementation FindFastCircle

- The border of the circle is searched with an edge detection operator
- For each border pixel the magnitude and direction of the edge is calculated
- With the direction the tangent of the circle is calculated
- The position of the center of the circle can be calculated
- For each candidate pixel only one point (the center) is 'drawn' in the Hough space

28-aug-08

Distance and Hough Transforms

30

### Hough Circle Transform

Explain difference in found circle radius:

- **FindBestCircle:**  $r = 54.9$
- **FindFastBestCircle:**  $r = 56.1$

28-aug-08

Distance and Hough Transforms

31

### Demonstration Find Fast Best Circles

- `Open image circles.jl`
- `findfastbestcircles DarkObject 130 30 90 1 2 40 8`

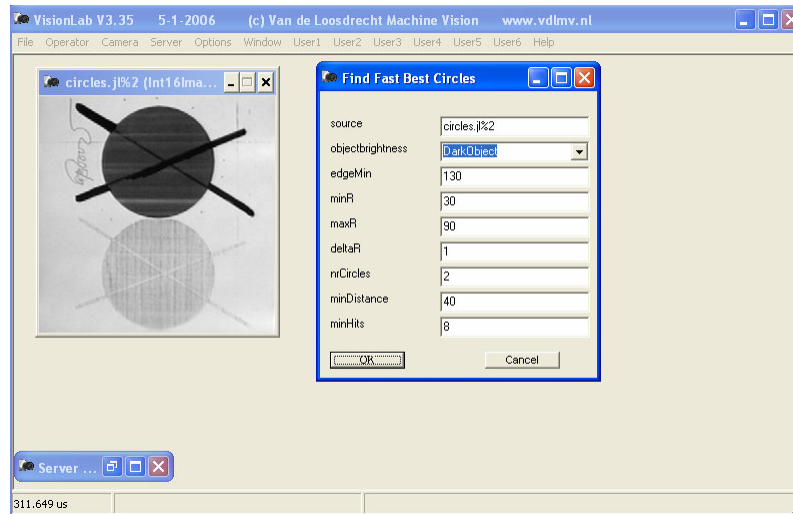
28-aug-08

Distance and Hough Transforms

32



### Find Fast Best Circles

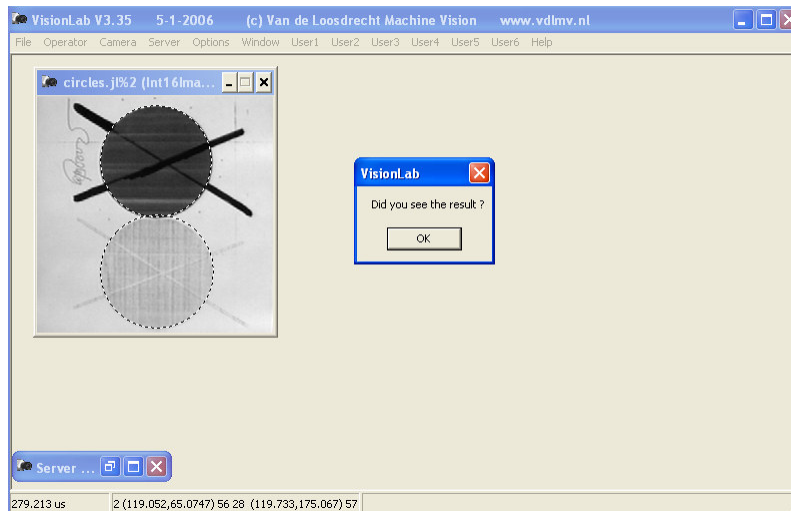


28-aug-08

Distance and Hough Transforms

33

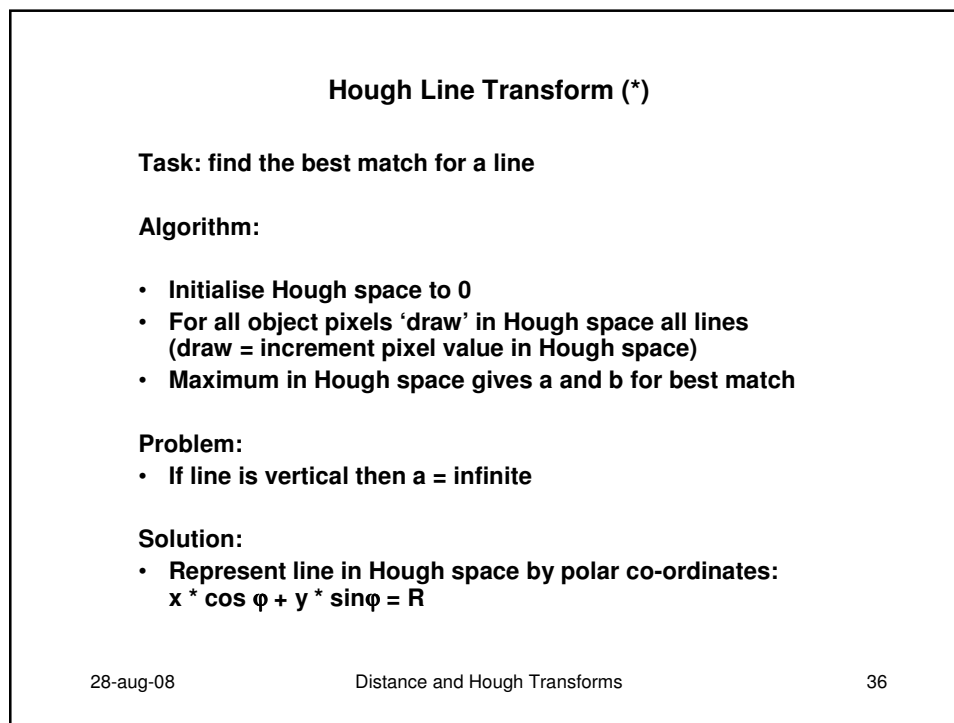
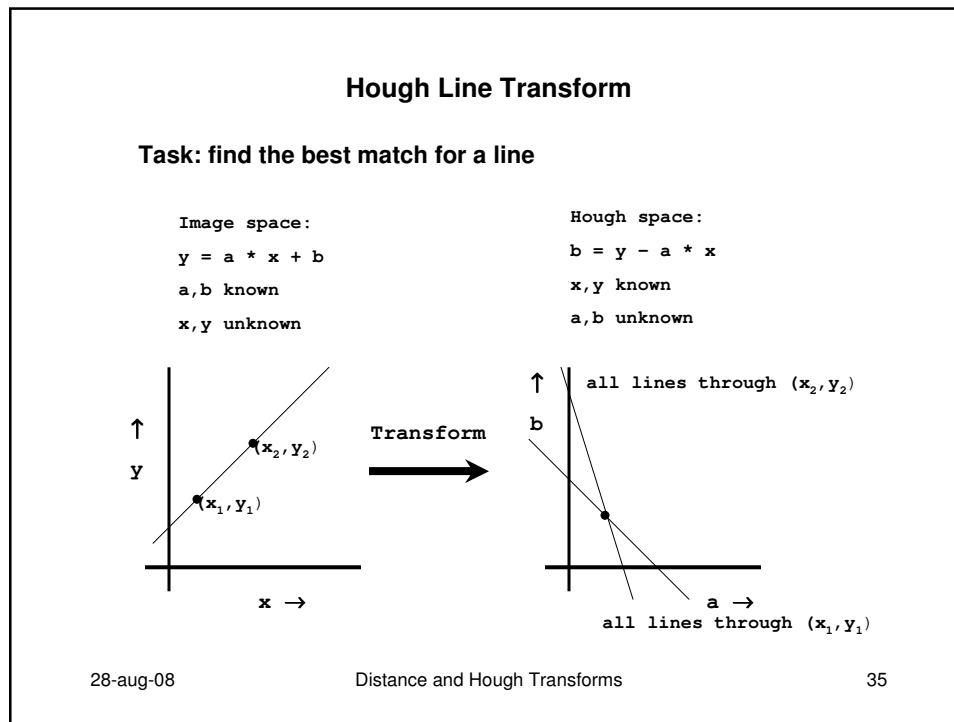
### Find Fast Best Circles



28-aug-08

Distance and Hough Transforms

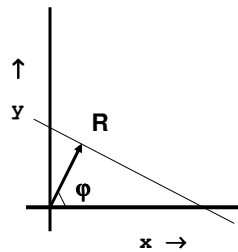
34



**Polar co-ordinates (\*)**

A line is represented by a normal vector with:

- length  $R$
- angle  $\phi$



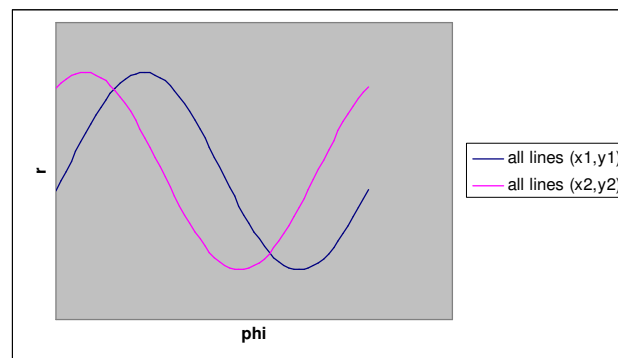
28-aug-08

Distance and Hough Transforms

37

**Polar co-ordinates (\*)**

Hough space of all lines through two points



28-aug-08

Distance and Hough Transforms

38

### Find Best Line

**Task:** find the best match for a line

**findbestline** srcName minR maxR deltaR minPhi maxPhi deltaPhi

**findbestline** is intended for a binary image

This operator searches in the image for the best match for a line. The normal vector of this line (r,phi) and the number of hits are returned as result.

This line is searched for in the area of the image with the following limitations (in polar co-ordinates): r in [minR..maxR] and phi in [minPhi..maxPhi] in degrees. Limitation of phi:  $-\pi/2 \leq \phi \leq \pi$ , due to the fact that a normal vector can not be in the fourth quadrant.

The resolution of the search is determined by deltaR and deltaPhi.

28-aug-08

Distance and Hough Transforms

39

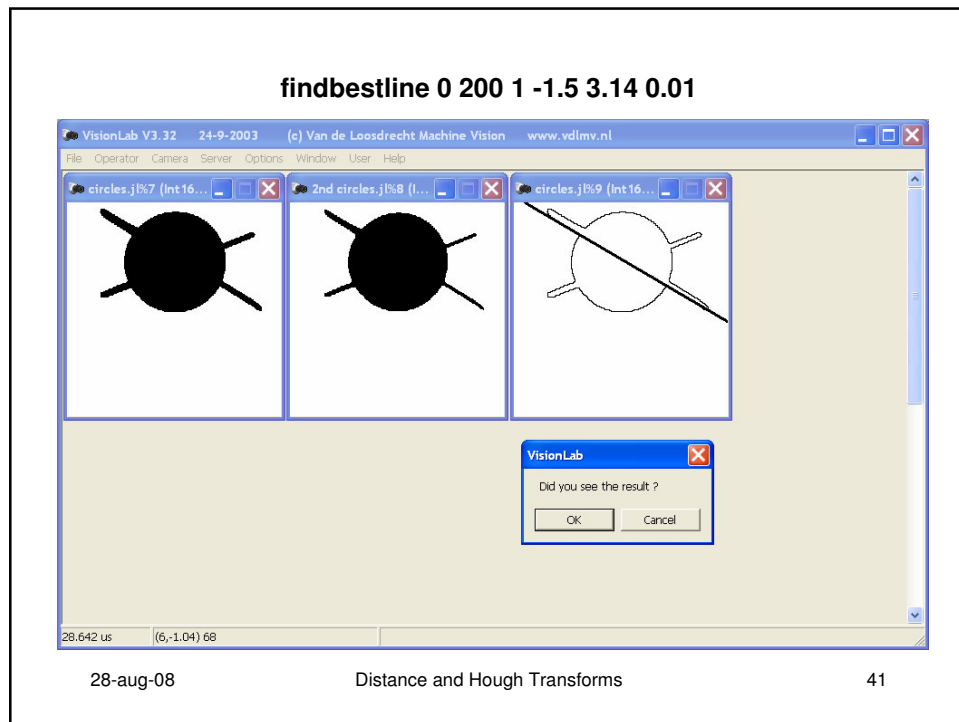
### Demonstration Hough Line Transform

- Open image circles.jl
- Threshold 0 100
- Erosion with full 3x3 mask
- Subtract eroded image from thresholded image in order to get a border image
- **findbestline** borderimage 0 200 1 -1.5 3.14 0.01

28-aug-08

Distance and Hough Transforms

40



### Find Fast Best Line

**findfastbestline srcName minR maxR deltaR minPhi maxPhi  
deltaPhi edgeMin**

**findfastbestline** is intended for a grayscale image.

**findfastbestline** has an extra parameter:

- **edgeMin**: the border of the line is found using the Sharr edge detection method, all edges with a magnitude higher than **edgeMin** are considered as candidates for the line.

Implementation with an edge detection operator which finds the orientation of the line segments. For each pixel in the image only one pixel in the Hough space is incremented

### Demonstration Find Fast Best Line

- Open image circles.jl
- `findfastbestline 400 0 200 1 -1.5 3.14 0.1`

28-aug-08

Distance and Hough Transforms

43

### Demonstration Find Fast Best Line



28-aug-08

Distance and Hough Transforms

44

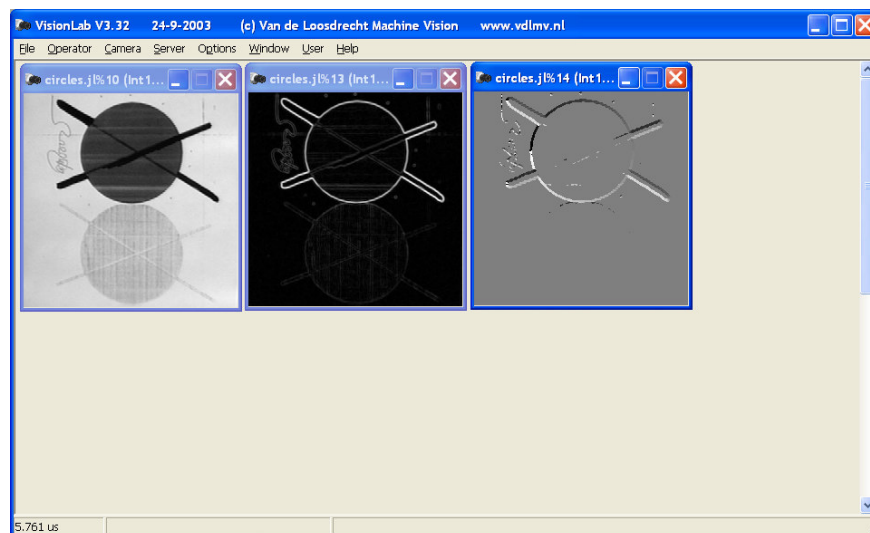
**Find (Fast) Best Line (\*)****Question:**

**Why is the result from Find Fast Best Line different from Find Best Line ?**

28-aug-08

Distance and Hough Transforms

45

**Scharr with minEdge = 400 (\*)**

28-aug-08

Distance and Hough Transforms

46

### Demonstration Find Best Lines (\*)

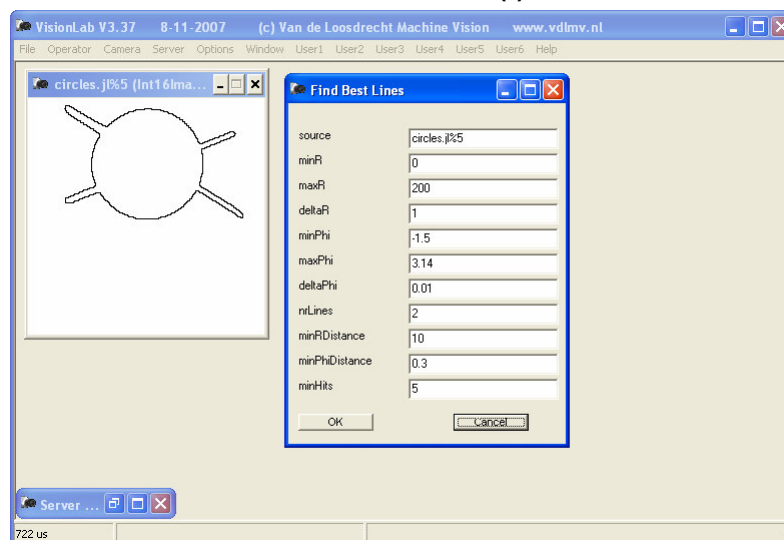
- Open image circles.jl
- Threshold 0 100
- Erosion with full 3x3 mask
- Subtract eroded image from thresholded image in order to get a border image
- `findbestlines borderimage 0 200 1 -1.5 3.14 0.01 2 10 0.3 5`

28-aug-08

Distance and Hough Transforms

47

### Find Best Lines (\*)



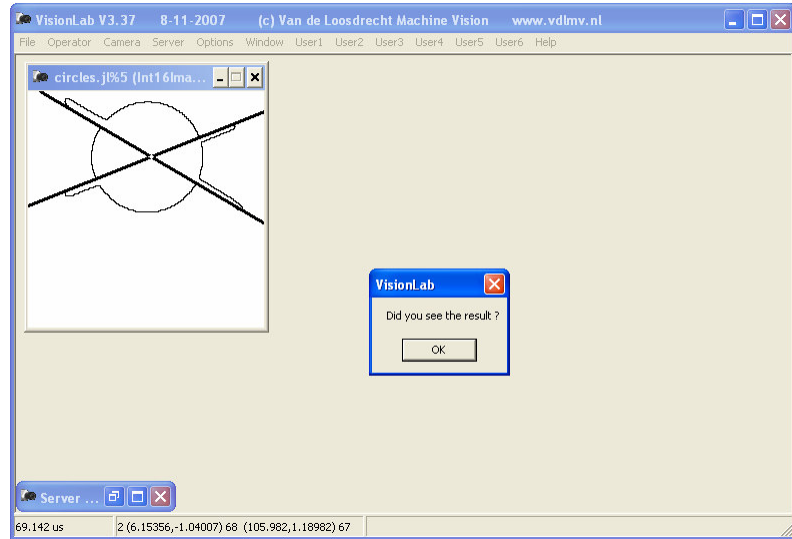
28-aug-08

Distance and Hough Transforms

48



## Find Best Lines (\*)



28-aug-08

Distance and Hough Transforms

49

## Demonstration Find Fast Best Lines

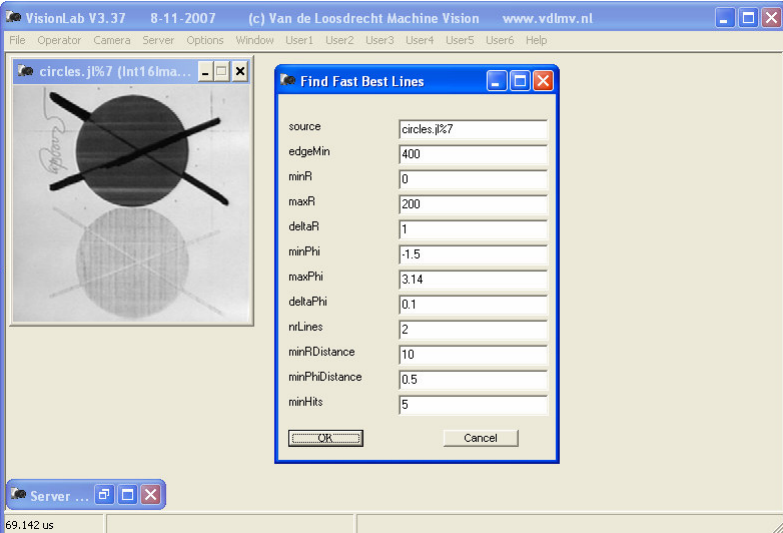
- Open image circles.jl
- FindFastBestLines circles 7 400 0 200 1 -1.5 3.14 0.1 2 10 0.5 5

28-aug-08

Distance and Hough Transforms

50

### Find Fast Best Lines

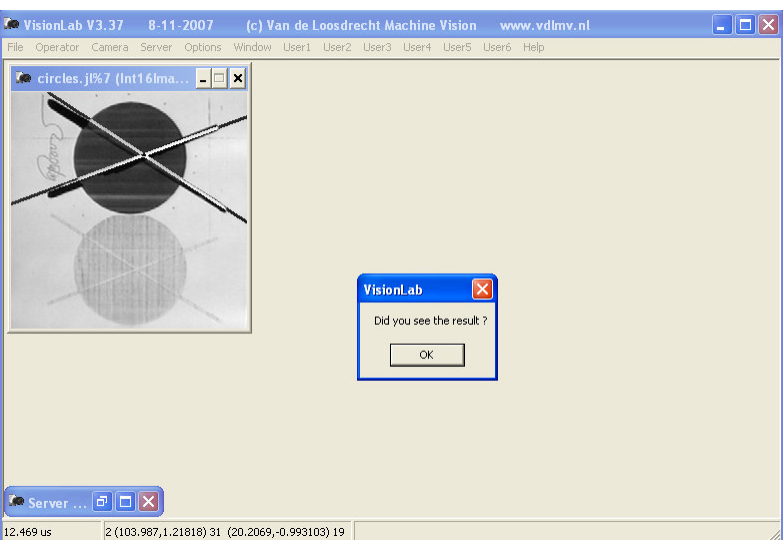


28-aug-08

Distance and Hough Transforms

51

### Find Fast Best Lines



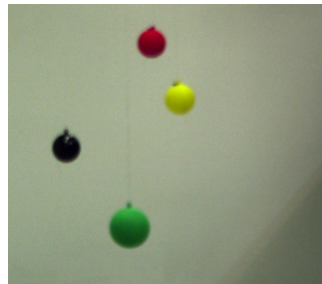
28-aug-08

Distance and Hough Transforms

52

### Exercise find balls

- Use image robot\_balls.jl



- Answer: ht\_robot\_balls.jls

28-aug-08

Distance and Hough Transforms

53

### Alternative for finding lines and circles

Alternative operators to find lines and circles are based on the Edge detection, see the chapter about Edge detection

#### Edge based:

- Fast
- Search area must contain only edges to find
- Can find only 1 line or circle
- Outliers cause problems

#### Hough based:

- Slower
- Search area can be whole image
- Can find more than 1 lines or circles
- Less problems with outliers

28-aug-08

Distance and Hough Transforms

54