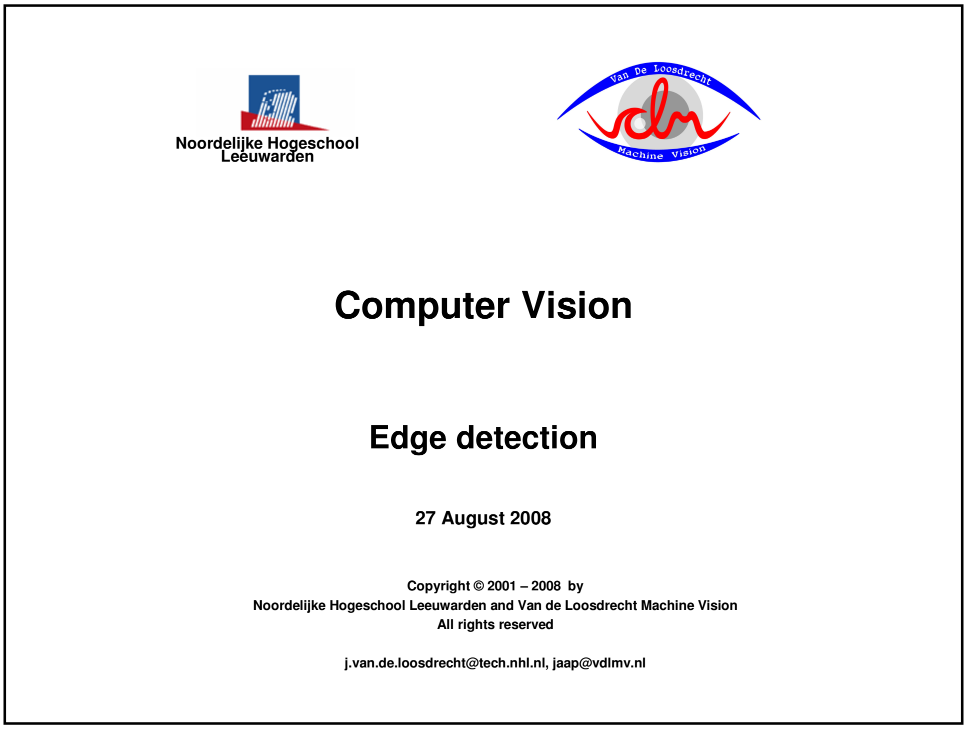


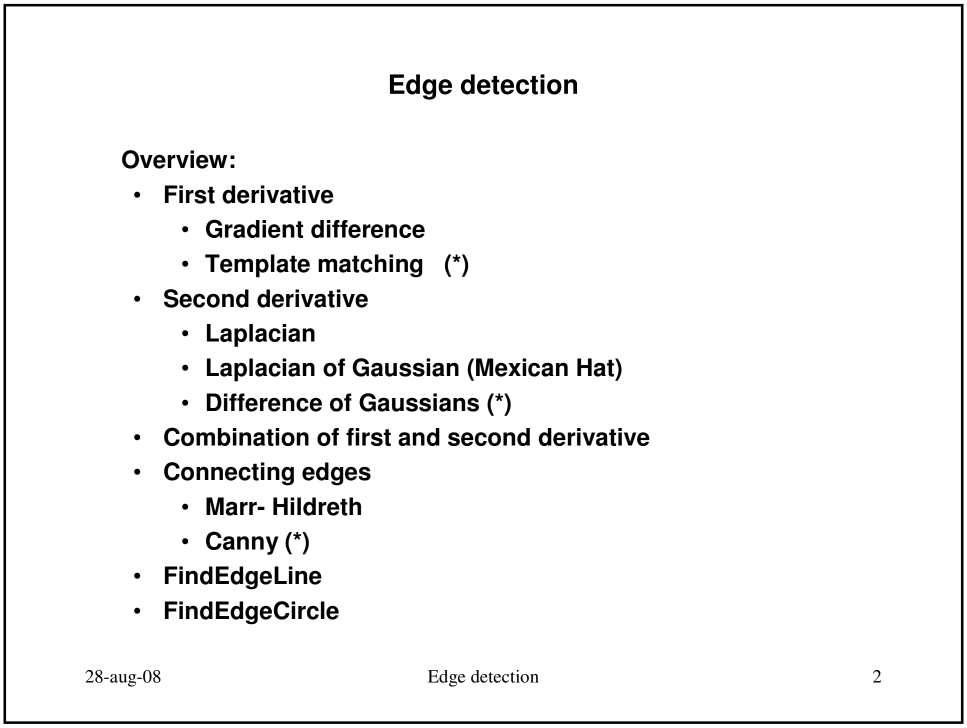
Noordelijke Hogeschool Leeuwarden

Computer Vision

Edge detection

27 August 2008

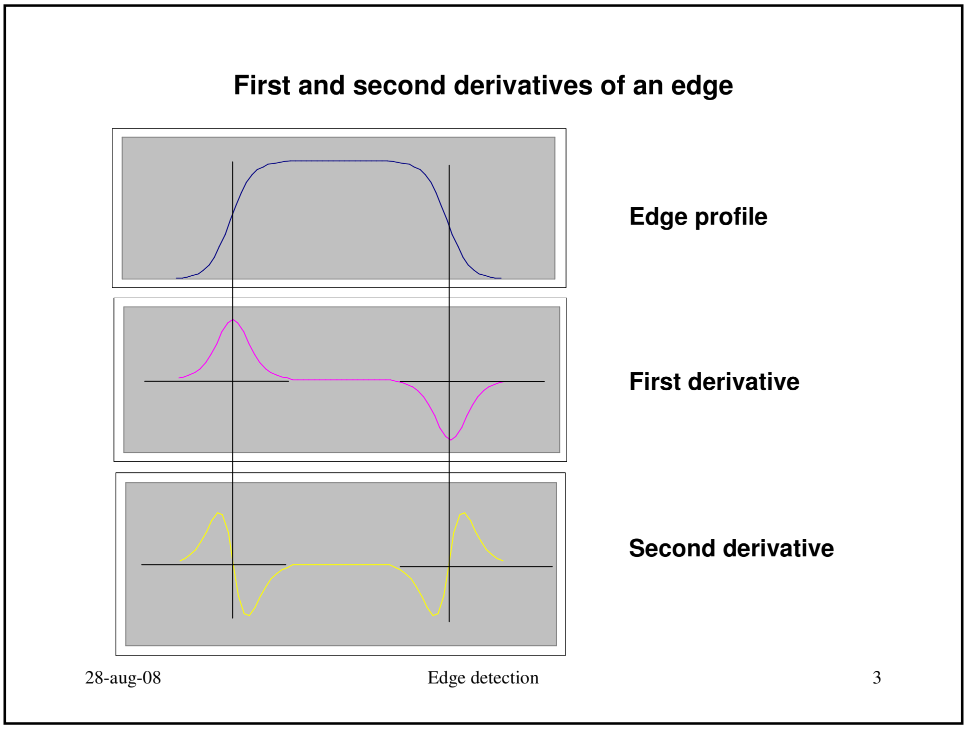
j.van.de.loosdrecht@tech.nhl.nl, jaap@vdlmv.nl



Edge detection

Overview:

- First derivative
  - Gradient difference
  - Template matching (*)
- Second derivative
  - Laplacian
  - Laplacian of Gaussian (Mexican Hat)
  - Difference of Gaussians (*)
- Combination of first and second derivative
- Connecting edges
  - Marr- Hildreth
  - Canny (*)
- FindEdgeLine
- FindEdgeCircle

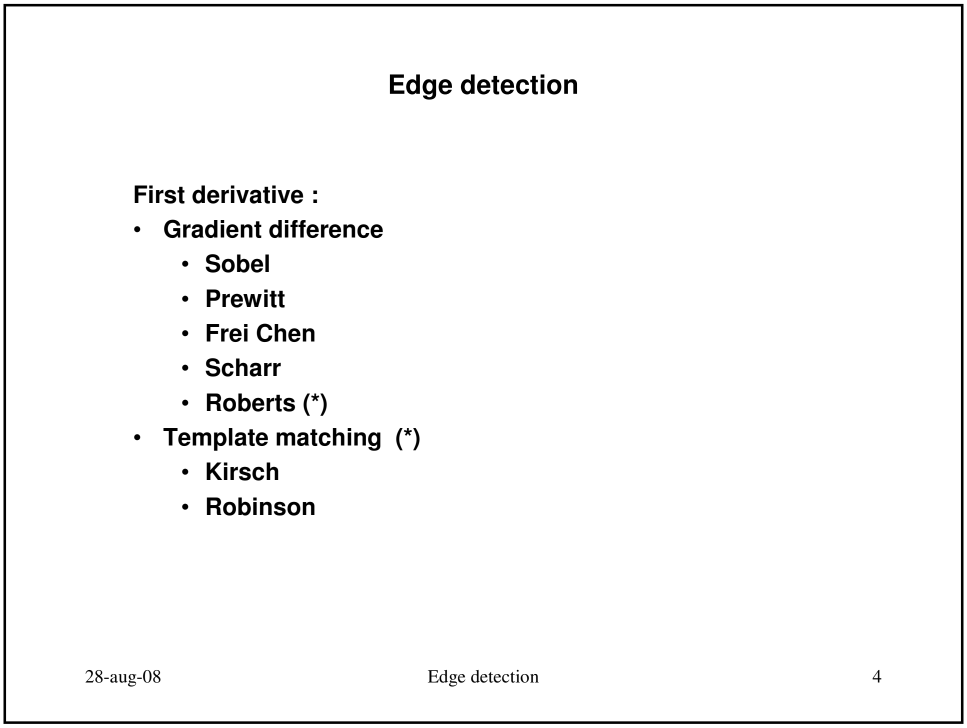


First and second derivatives of an edge

Edge profile

First derivative

Second derivative



Edge detection

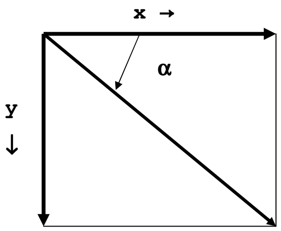
First derivative :

- **Gradient difference**
  - **Sobel**
  - **Prewitt**
  - **Frei Chen**
  - **Scharr**
  - **Roberts (*)**
- **Template matching (*)**
  - **Kirsch**
  - **Robinson**

Edge detection

Definitions:

- **magnitude: strength of edge**
- **direction: orientation of edge**
- **angles are measured in radians (- π .. π]**

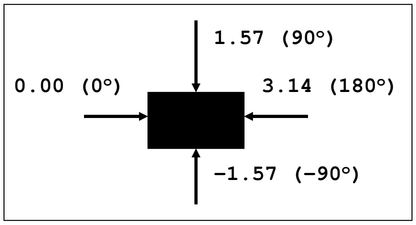


Edge detection

Definitions (continued):

- **edge directions are calculated from low towards high pixel values perpendicular at the edge contour**
  example:

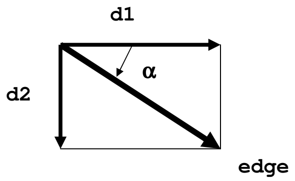
```
White = 0, Black = 10 !!!
```



Gradient difference

Idea:

- **Calculate the edge using convolution in two perpendicular directions d_1 and d_2**
- **Edge magnitude = sqrt ($d_1^2 + d_2^2$)**
- **Edge direction = arctan (d_1 / d_2)**



Gradient difference

Examples:

- **Sobel (src, magImage, dirImage, gradient, dirScale, minedge);**

  gradient: magnitude, direction or both
  dirScale: direction in radians * gradScale
  minEdge: if gradient is both, all directions with an edge magnitude lower then minEdge are not calculated and set to zero.

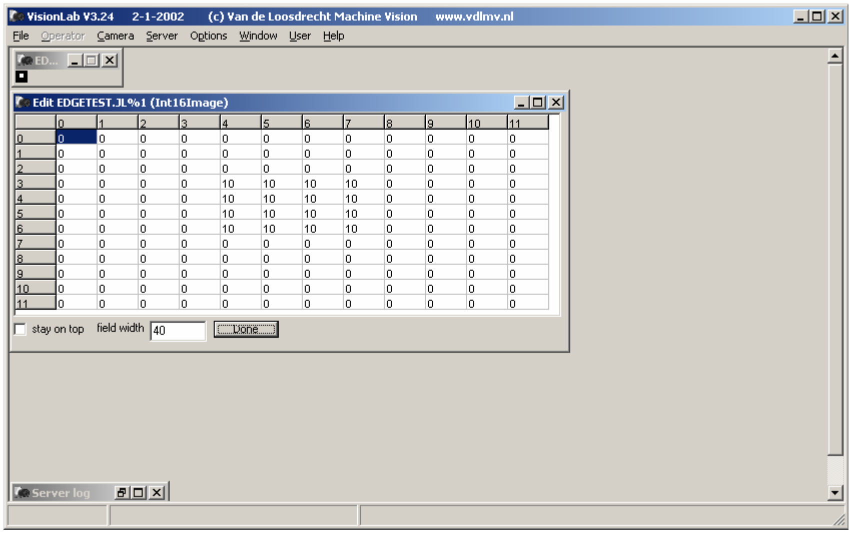
  Masks:

```
-1  -2  -1          -1   0   1
 0   0   0          -2   0   2
 1   2   1          -1   0   1
```

Demonstration Sobel

- Open image edgetest.jl *(do not use sq2.jl)*
- Apply Sobel magnitude on image
- Apply Sobel direction with scale = 10000 on image
- Edit images (3x) and explain

Image edgetest.jl

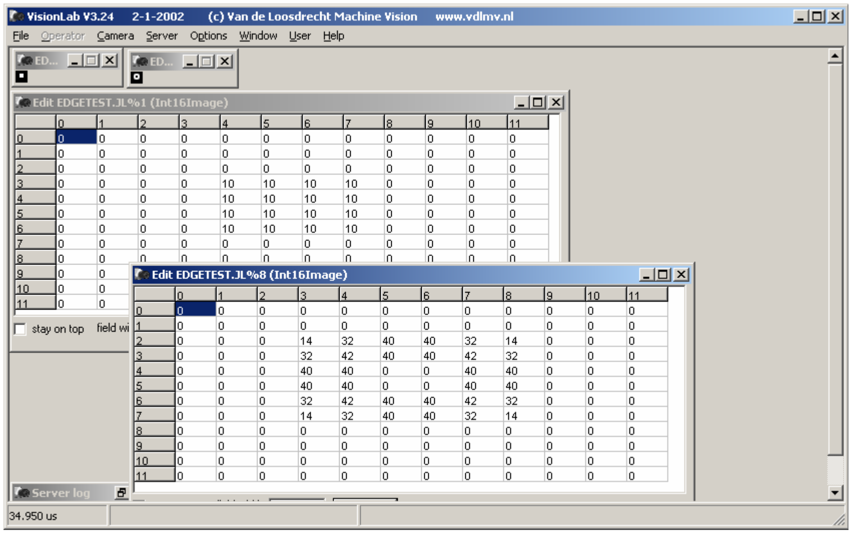

VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
ED...
Server log

Edit EDGETEST.JL%1 (Int16Image)

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

□ stay on top field width 40 Done

Apply Sobel magnitude


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
ED...
ED...
Server log

Edit EDGETEST.JL%1 (Int16Image)

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

□ stay on top field wi

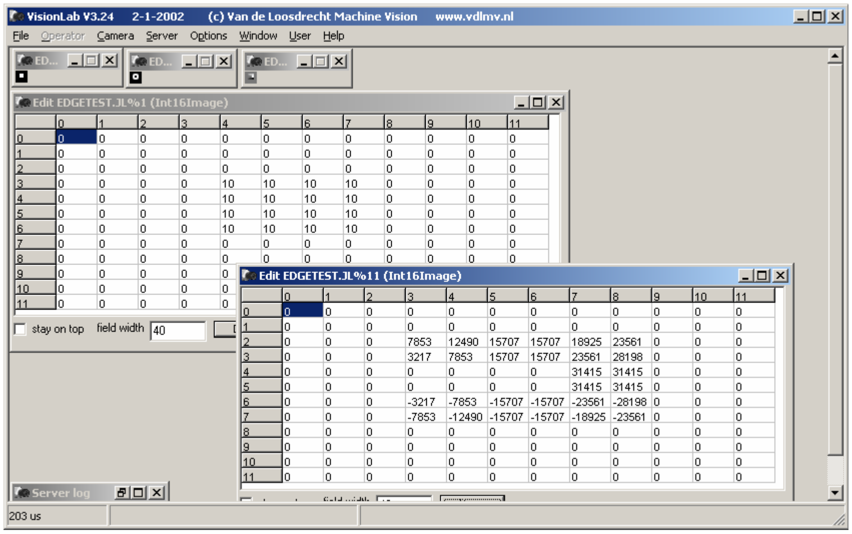
Edit EDGETEST.JL%8 (Int16Image)

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	14	32	40	40	32	14	0	0	0
3	0	0	0	32	42	40	40	42	32	0	0	0
4	0	0	0	40	40	0	0	40	40	0	0	0
5	0	0	0	40	40	0	0	40	40	0	0	0
6	0	0	0	32	42	40	40	42	32	0	0	0
7	0	0	0	14	32	40	40	32	14	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

34.950 us

28-aug-08 Edge detection 11

Sobel direction with scale = 10000


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
ED...
ED...
ED...
Server log

Edit EDGETEST.JL%1 (Int16Image)

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

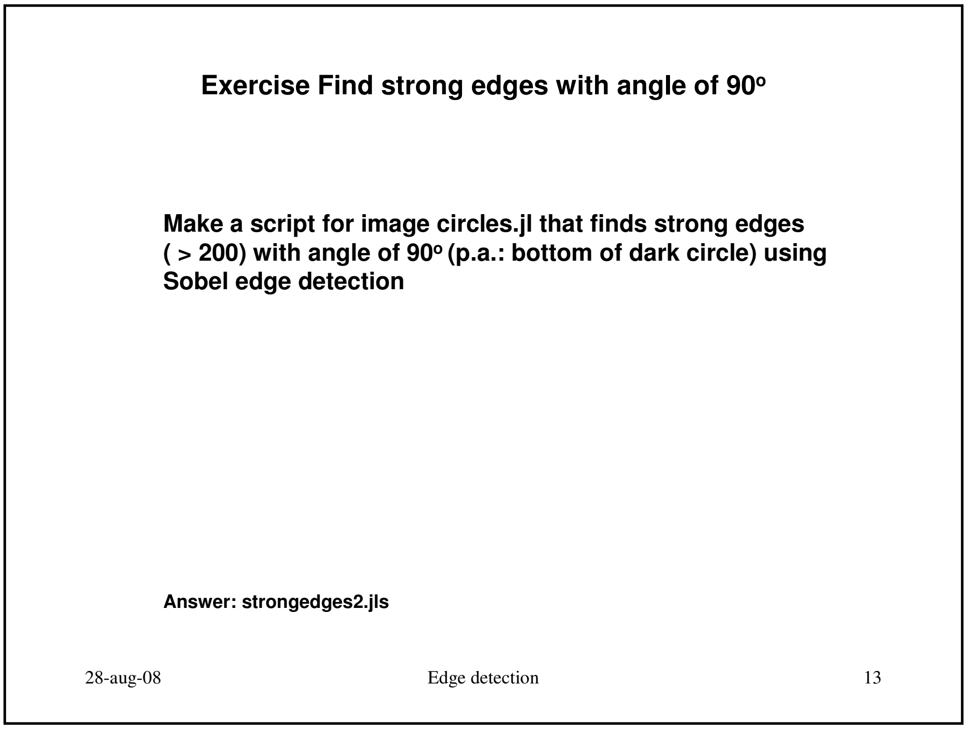
□ stay on top field width 40

Edit EDGETEST.JL%11 (Int16Image)

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	7853	12490	15707	15707	18925	23561	0	0	0
3	0	0	0	3217	7853	15707	15707	23561	28198	0	0	0
4	0	0	0	0	0	0	0	31415	31415	0	0	0
5	0	0	0	0	0	0	0	31415	31415	0	0	0
6	0	0	0	-3217	-7853	-15707	-15707	-23561	-28198	0	0	0
7	0	0	0	-7853	-12490	-15707	-15707	-18925	-23561	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

203 us

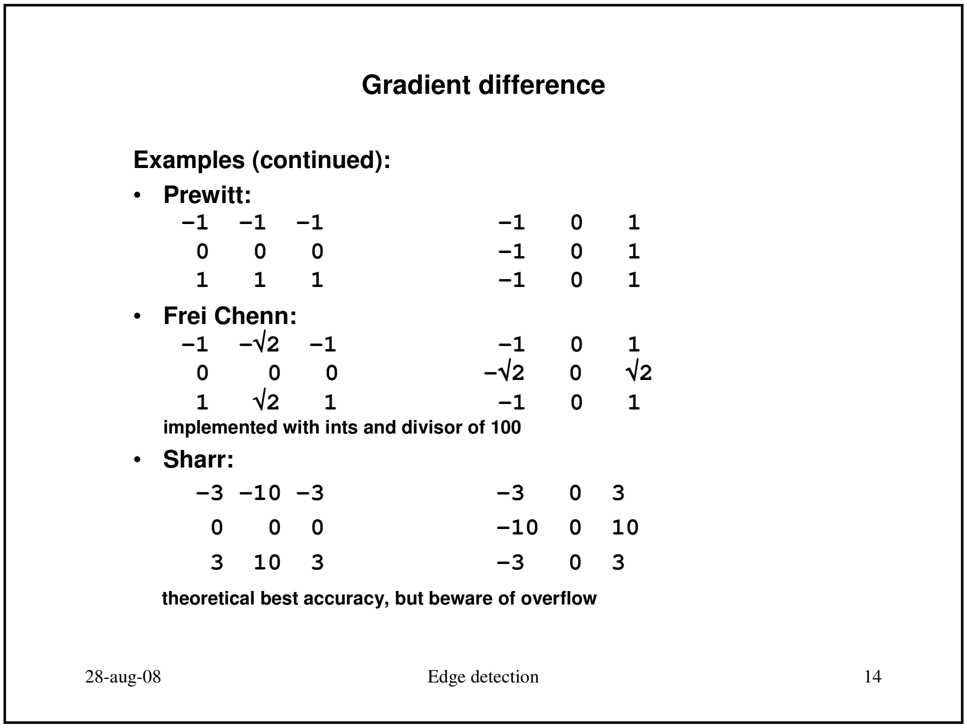
28-aug-08 Edge detection 12



Exercise Find strong edges with angle of 90^o

Make a script for image circles.jl that finds strong edges (> 200) with angle of 90^o (p.a.: bottom of dark circle) using Sobel edge detection

Answer: strongedges2.jls



Gradient difference

Examples (continued):

- **Prewitt:**

$$\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$$

- **Frei Chenn:**

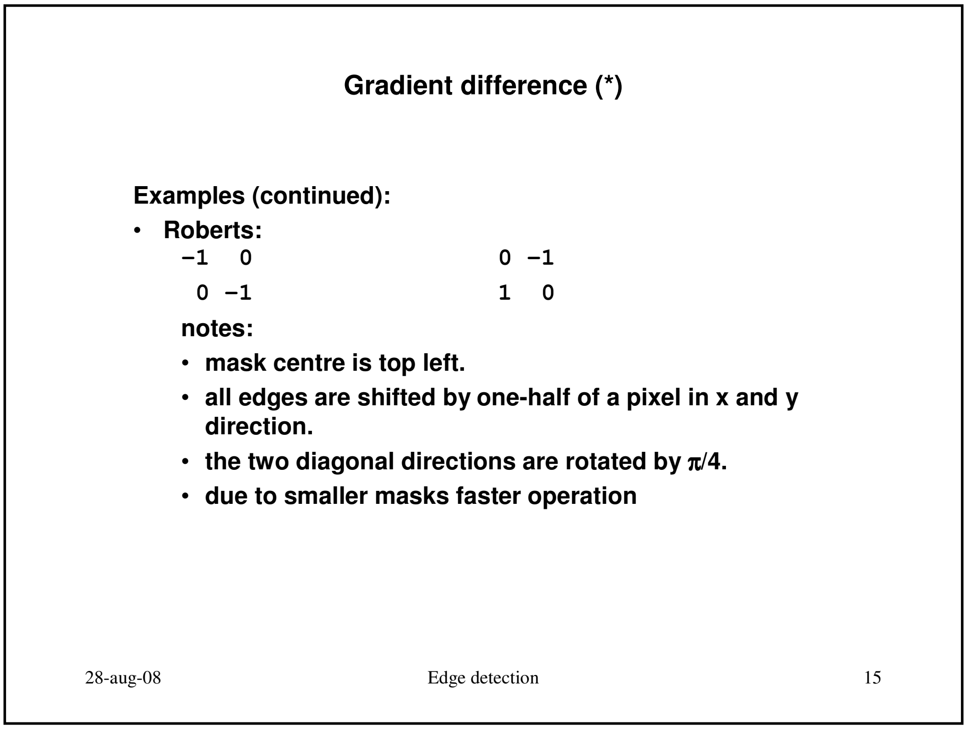
$$\begin{matrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{matrix}$$

implemented with ints and divisor of 100

- **Sharr:**

$$\begin{matrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{matrix} \qquad \begin{matrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{matrix}$$

theoretical best accuracy, but beware of overflow



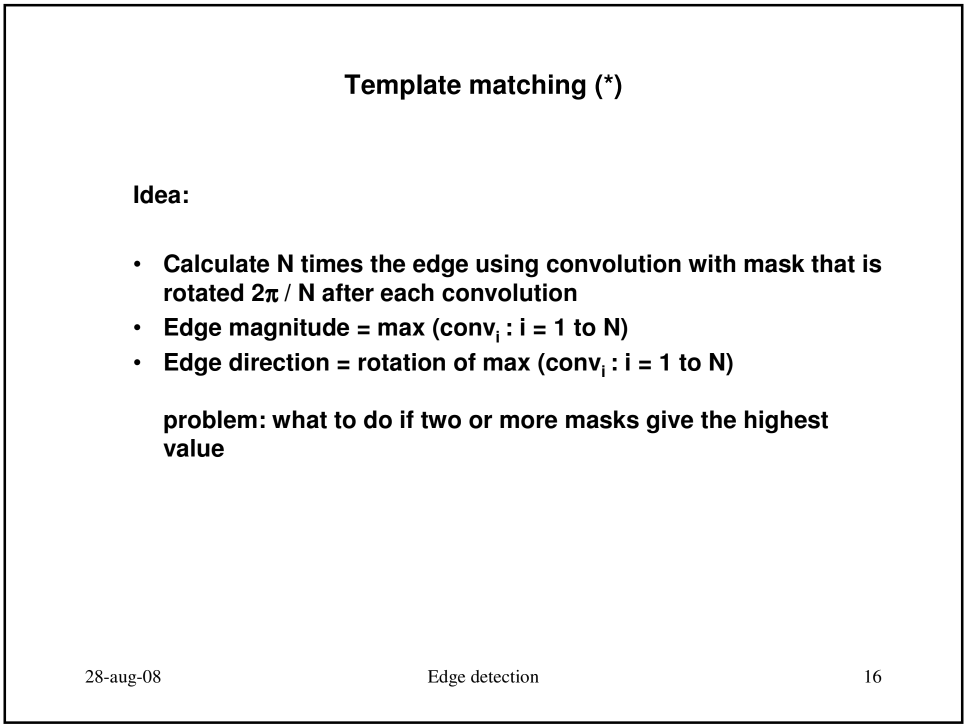
Gradient difference (*)

Examples (continued):

- Roberts:

```
-1  0                  0 -1
 0 -1                  1  0
```

  notes:
  - mask centre is top left.
  - all edges are shifted by one-half of a pixel in x and y direction.
  - the two diagonal directions are rotated by π/4.
  - due to smaller masks faster operation



Template matching (*)

Idea:

- Calculate N times the edge using convolution with mask that is rotated 2π / N after each convolution
- Edge magnitude = max ($conv_i$: i = 1 to N)
- Edge direction = rotation of max ($conv_i$: i = 1 to N)

  problem: what to do if two or more masks give the highest value

Template matching (*)

Examples:

- **Kirsch: (8 rotations)**

```
−3 −3  5
−3  0  5
−3 −3  5
```

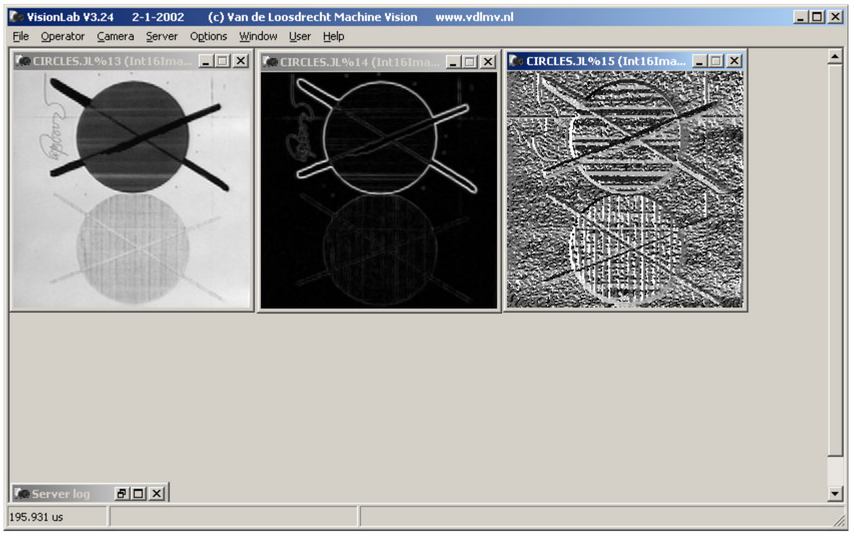
- **Robinson: (8 rotations)**

```
−1  0  1
−2  0  2
−1  0  1
```

Demonstration Kirsch (*)

- **Apply Kirsch on circles.jls for magnitude and direction, note increase of processor time compared with Sobel.**

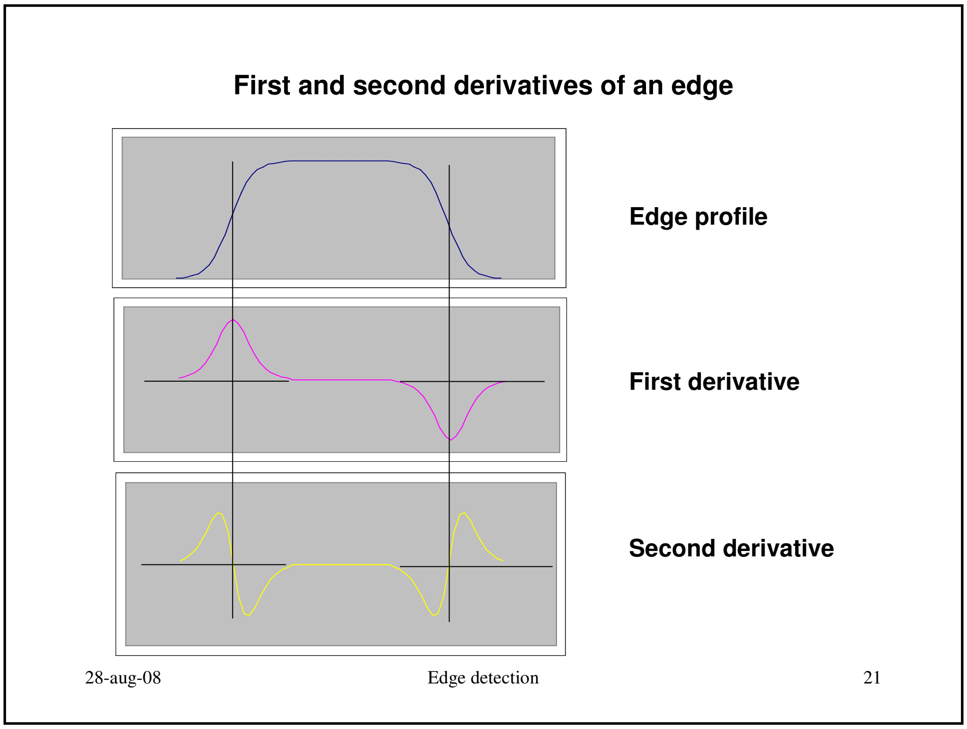
Kirsch on circles.jls for magnitude and direction (*)


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%13 (Int16Ima...
CIRCLES.JL%14 (Int16Ima...
CIRCLES.JL%15 (Int16Ima...
Server log
195.931 us

Edge detection

Second derivative

- **Laplacian**
- **Laplacian of Gaussian (Mexican Hat)**
- **Finding edges using zero crossings**

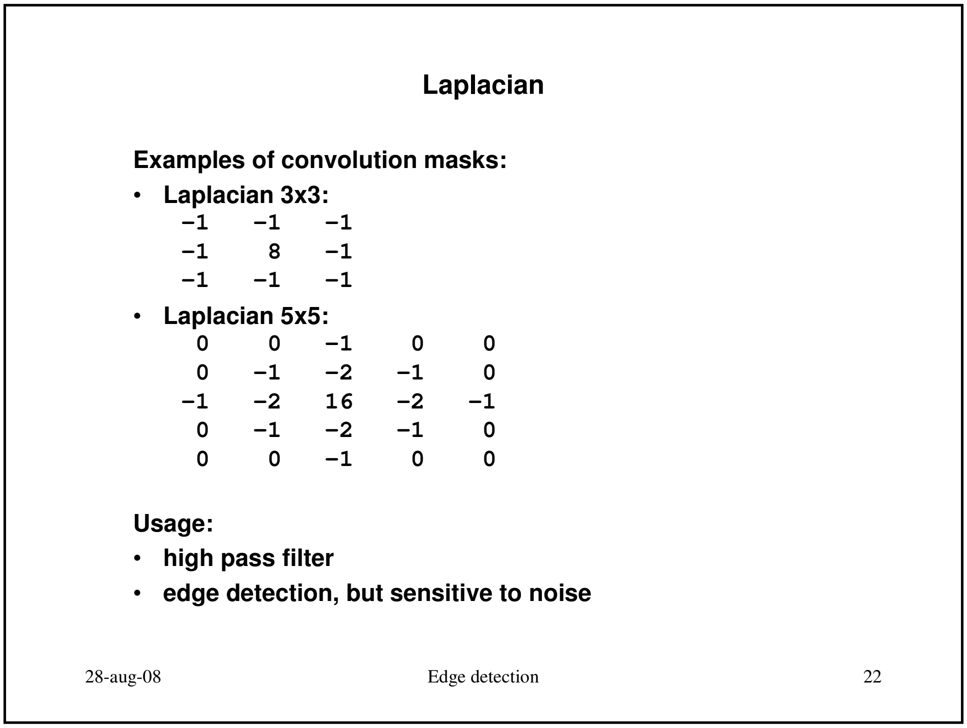


First and second derivatives of an edge

Edge profile

First derivative

Second derivative



Laplacian

Examples of convolution masks:

- **Laplacian 3x3:**

-1	-1	-1
-1	8	-1
-1	-1	-1

- **Laplacian 5x5:**

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

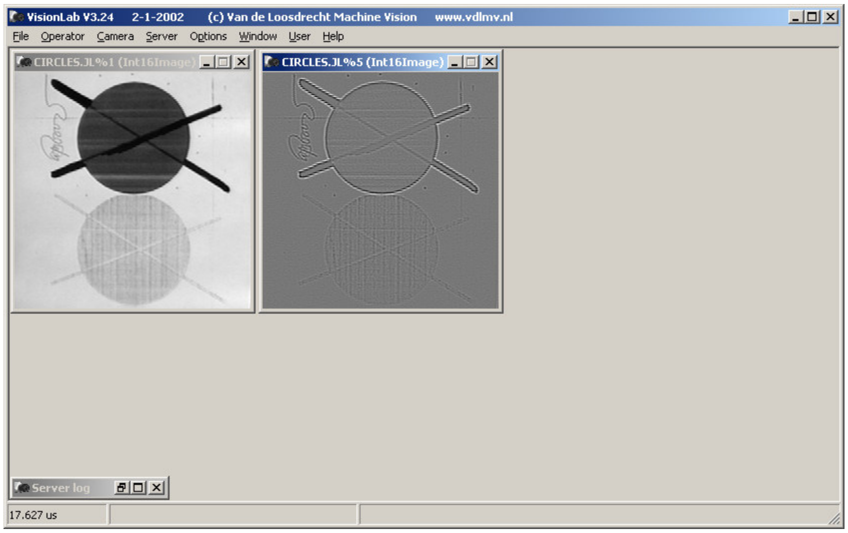
Usage:

- **high pass filter**
- **edge detection, but sensitive to noise**

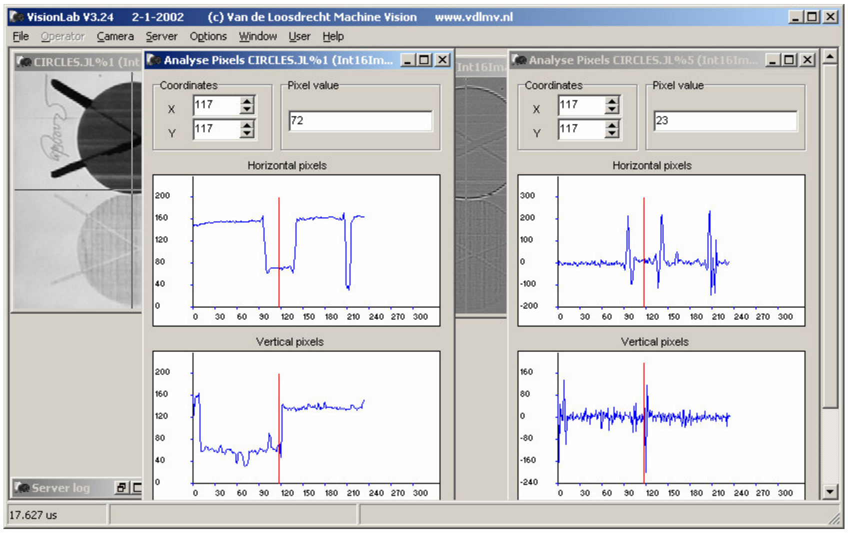
Demonstration high pass filter

- Open image circles.jl
- Convolution with Laplacian 3x3
- 2x analyse pixels:
    - low frequencies -> 0
    - high frequencies -> |pixel value| >> 1

Convolution with Laplacian 3x3


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%1 (Int16Image)
CIRCLES.JL%5 (Int16Image)
Server log
17.627 us

original after convolution with Laplacian


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%1 (Int
Analyse Pixels CIRCLES.JL%1 (Int16Im...
Int16Im
Analyse Pixels CIRCLES.JL%5 (Int16Im...
Horizontal pixels
Vertical pixels
200 160 120 80 40 0
0 30 60 90 120 150 180 210 240 270 300
300 200 100 0 -100 -200
160 80 0 -80 -160 -240
Server log

Coordinates X 117 Y 117 — Pixel value 72

Coordinates X 117 Y 117 — Pixel value 23

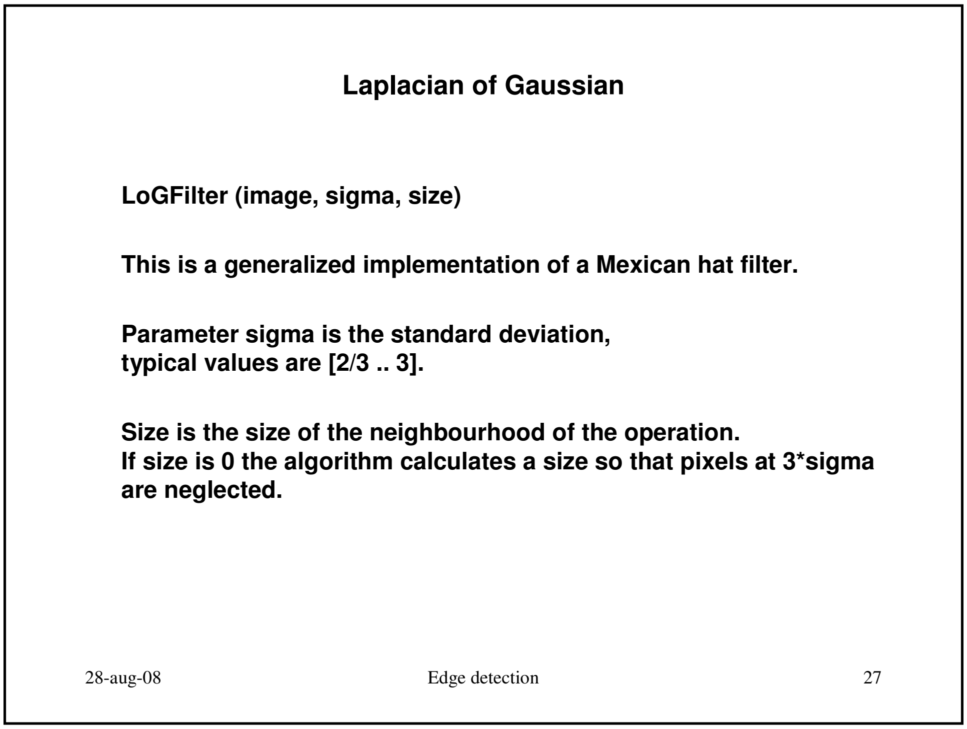
17.627 us

Mexican hat

- Variant of low and high pass filters
- Combination of low and high pass filter
- Mask (7x7):

$$\begin{matrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & -3 & -3 & -3 & -1 & 0 \\ -1 & -3 & 0 & 7 & 0 & -3 & -1 \\ -1 & -3 & 7 & 24 & 7 & -3 & -1 \\ -1 & -3 & 0 & 7 & 0 & -3 & -1 \\ 0 & -1 & -3 & -3 & -3 & -1 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \end{matrix}$$

- Local noise is smoothed out by low pass filter in centre



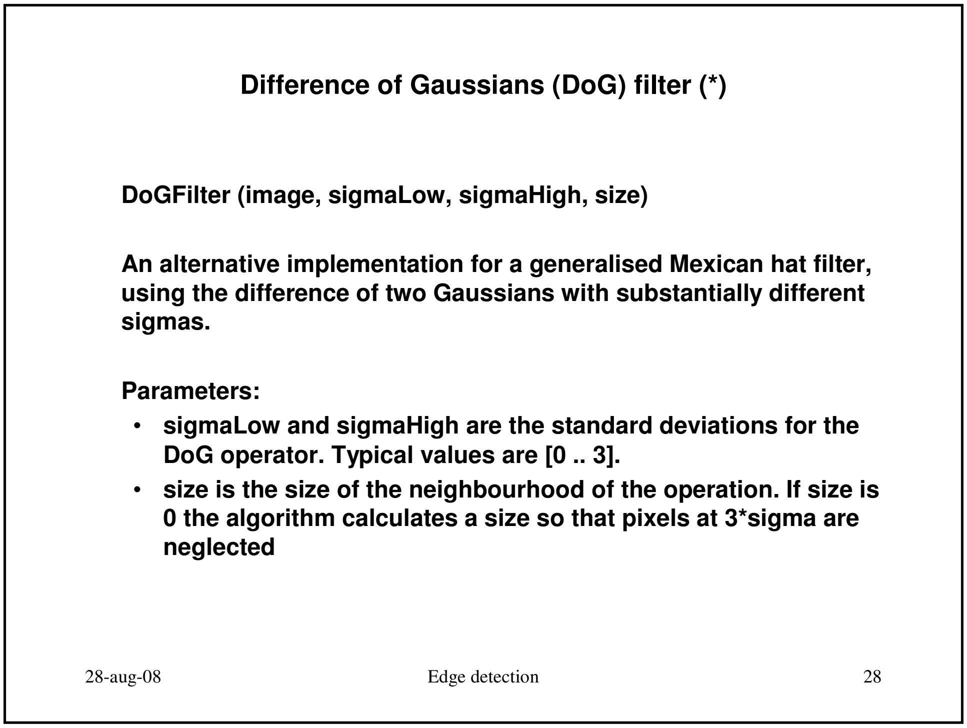
Laplacian of Gaussian

LoGFilter (image, sigma, size)

This is a generalized implementation of a Mexican hat filter.

**Parameter sigma is the standard deviation,
typical values are [2/3 .. 3].**

**Size is the size of the neighbourhood of the operation.
If size is 0 the algorithm calculates a size so that pixels at 3*sigma are neglected.**



Difference of Gaussians (DoG) filter (*)

DoGFilter (image, sigmaLow, sigmaHigh, size)

An alternative implementation for a generalised Mexican hat filter, using the difference of two Gaussians with substantially different sigmas.

Parameters:

- **sigmaLow and sigmaHigh are the standard deviations for the DoG operator. Typical values are [0 .. 3].**
- **size is the size of the neighbourhood of the operation. If size is 0 the algorithm calculates a size so that pixels at 3*sigma are neglected**

Demonstration Mexican hat (*)

- Open image circles.jl (use LUT stretch)
- Add noise 1 0 50
- Convolution Laplacian 5x5 on noise image
- Convolution Mexican hat on noise image
  (smooth noise and enhance high frequencies)

- Open image circles.jl
- Convolution Mexican hat on image
- LoGFilter 1.4 7 on Image (same result)

With noise Mexican hat Laplacian (*)

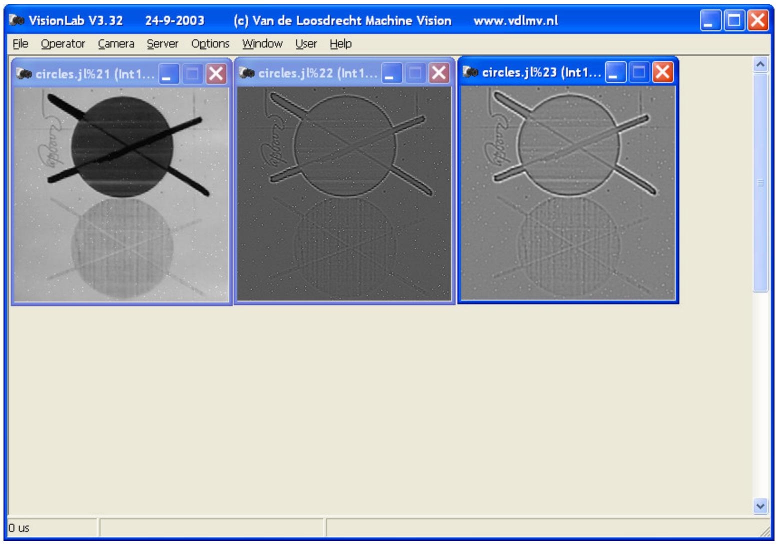
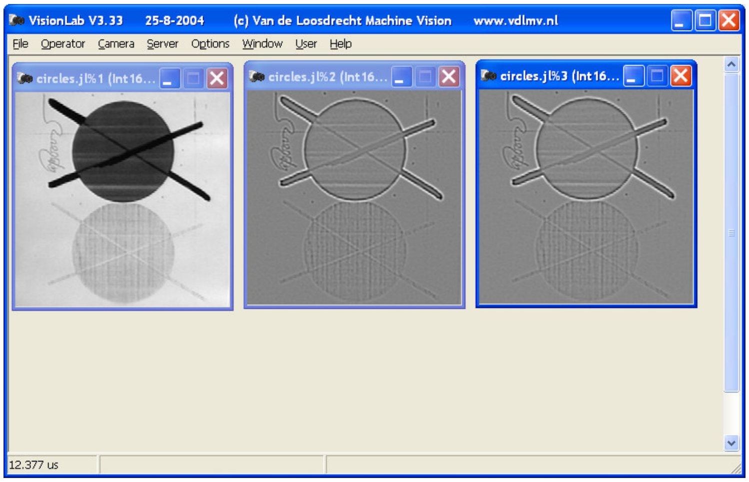

VisionLab V3.32 24-9-2003 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%21 (Int1...
circles.jl%22 (Int1...
circles.jl%23 (Int1...
0 us

image **MexicanHat7x7** **LoGFilter 1.4 7 (*)**


VisionLab V3.33 25-8-2004 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%1 (Int16...
circles.jl%2 (Int16...
circles.jl%3 (Int16...
12.377 us

Combination of first and second derivative

The positions of the edges can not be found exactly using first derivative edge detectors because the maximum value are are position dependant.

The zero crossings in the second derivative ‘ANDed’ with the strong edges in the first derivative give the exact position of the edges.

Zero crossings

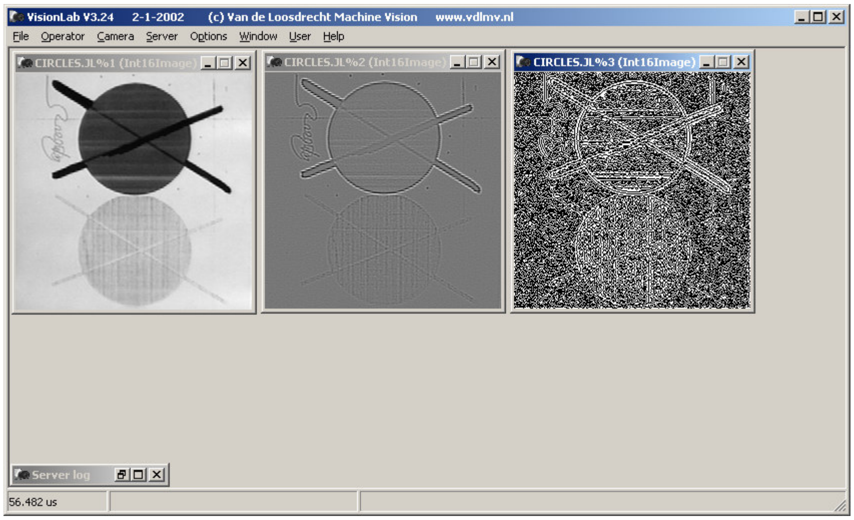
- Open image circles.jl
- Convolution Laplacian 3x3
  - Show zero crossings with pixel analysis
- Zero crossings FourConnected on laplacian
- Sobel Magnitude on circles
- Threshold 200 1000 on sobel
- Multiply zero crossing with threshold gives the exact position of the edges

Note:

- With zero crossings the “middle” of the edge will be found and the edge will be approximately one pixel thick
- With Sobel magnitude followed by threshold an edge will be found which is in general more then one pixel thick, its thickness will vary with the intensity of the lighting.

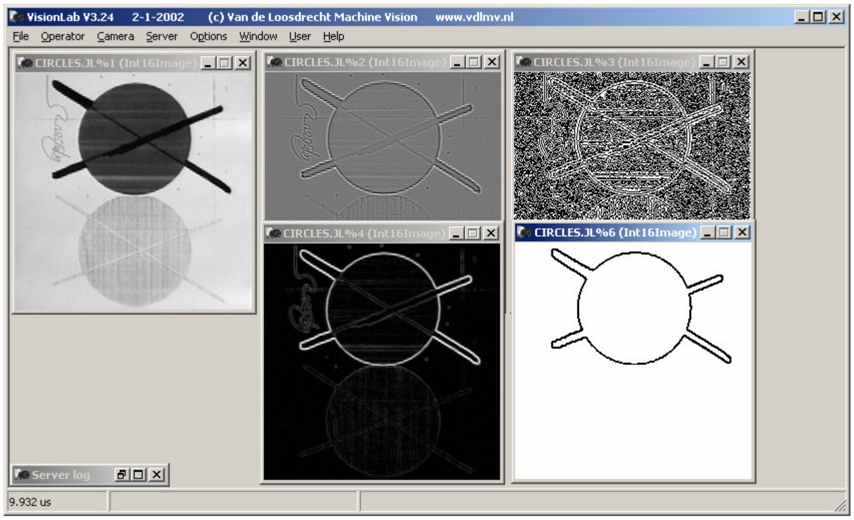
28-aug-08 Edge detection 33

original Laplacian Zero crossings

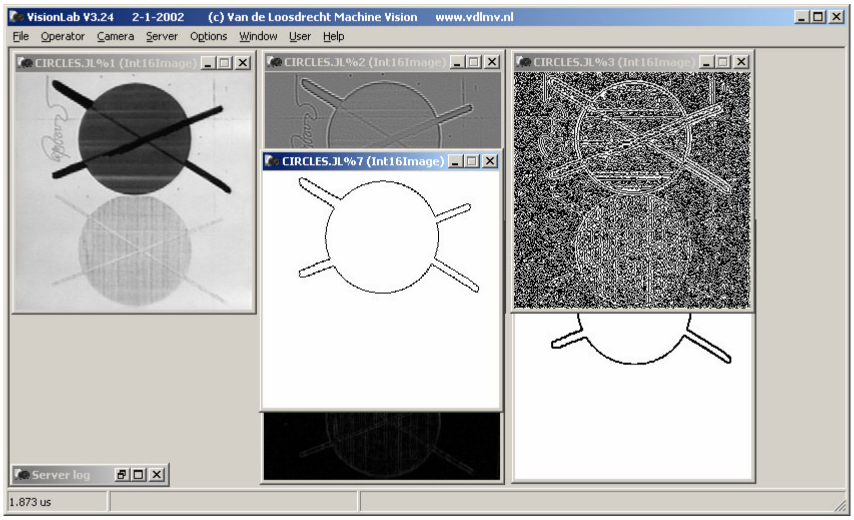

VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%1 (Int16Image)
CIRCLES.JL%2 (Int16Image)
CIRCLES.JL%3 (Int16Image)
Server log
56.482 us

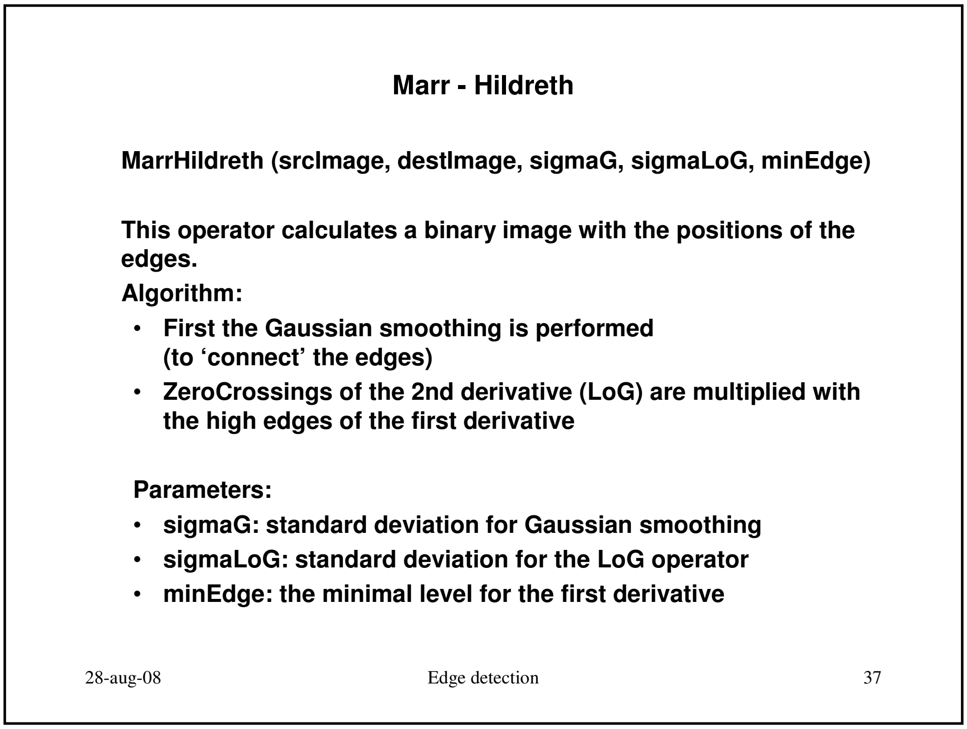
28-aug-08 Edge detection 34

Sobel Threshold 200 10000


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%1 (Int16Image)
CIRCLES.JL%2 (Int16Image)
CIRCLES.JL%3 (Int16Image)
CIRCLES.JL%4 (Int16Image)
CIRCLES.JL%6 (Int16Image)
Server log
9.932 us

Multiply zero crossing with threshold


VisionLab V3.24 2-1-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
CIRCLES.JL%1 (Int16Image)
CIRCLES.JL%2 (Int16Image)
CIRCLES.JL%3 (Int16Image)
CIRCLES.JL%7 (Int16Image)
Server log
1.873 us



Marr - Hildreth

MarrHildreth (srcImage, destImage, sigmaG, sigmaLoG, minEdge)

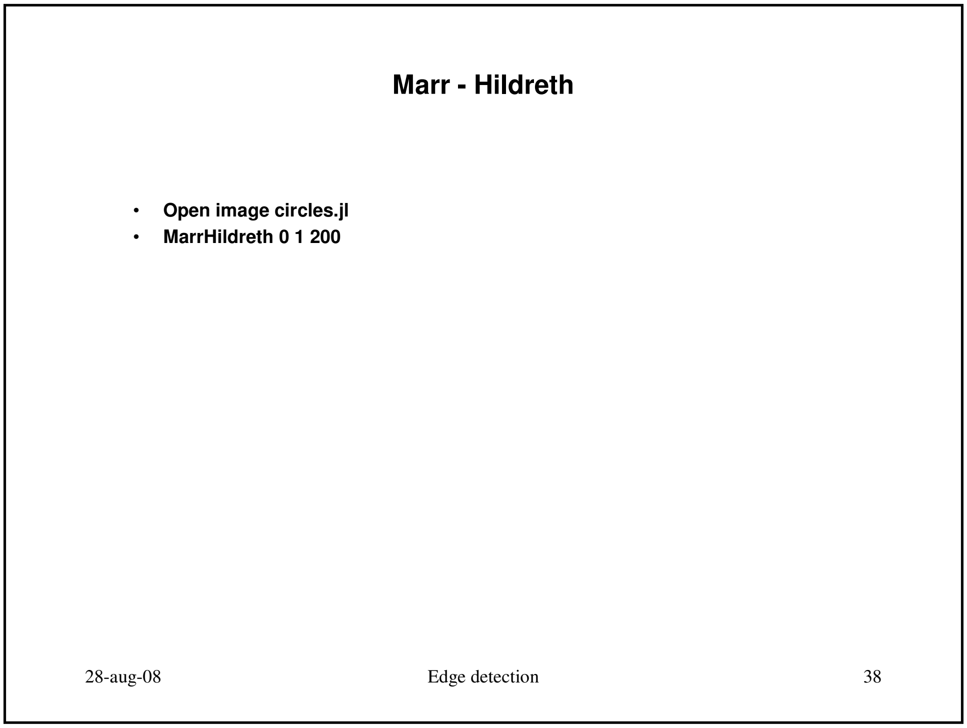
This operator calculates a binary image with the positions of the edges.

Algorithm:

- **First the Gaussian smoothing is performed (to ‘connect’ the edges)**
- **ZeroCrossings of the 2nd derivative (LoG) are multiplied with the high edges of the first derivative**

Parameters:

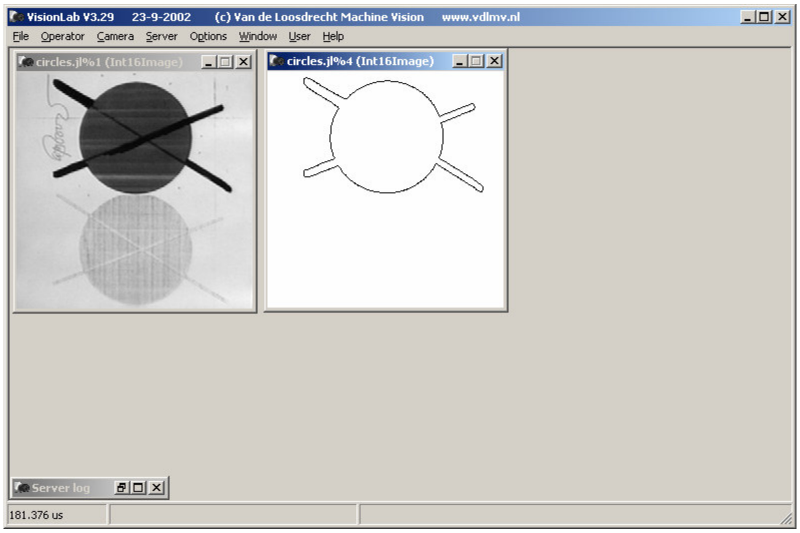
- **sigmaG: standard deviation for Gaussian smoothing**
- **sigmaLoG: standard deviation for the LoG operator**
- **minEdge: the minimal level for the first derivative**



Marr - Hildreth

- **Open image circles.jl**
- **MarrHildreth 0 1 200**

MarrHildreth 0 1 200


VisionLab V3.29 23-9-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%1 (Int16Image)
circles.jl%4 (Int16Image)
Server log

181.376 us

Canny (*)

Canny (srcImage, destImage, sigma, low, high, connected)

This operator calculates a binary image with the positions of the edges.

Algorithm and parameters:

- **Gaussian smoothing with *sigma***
- **Sobel edge detection**
- **Maxima of the edges magnitudes are searched for and linked. All pixels with a edge greater than *high* are selected as object pixels. These object pixels are used as seeds. All connected neighbours of the seeds with a edge greater than *low* are added to the object pixels. This growing process is repeated until no pixels are added.**

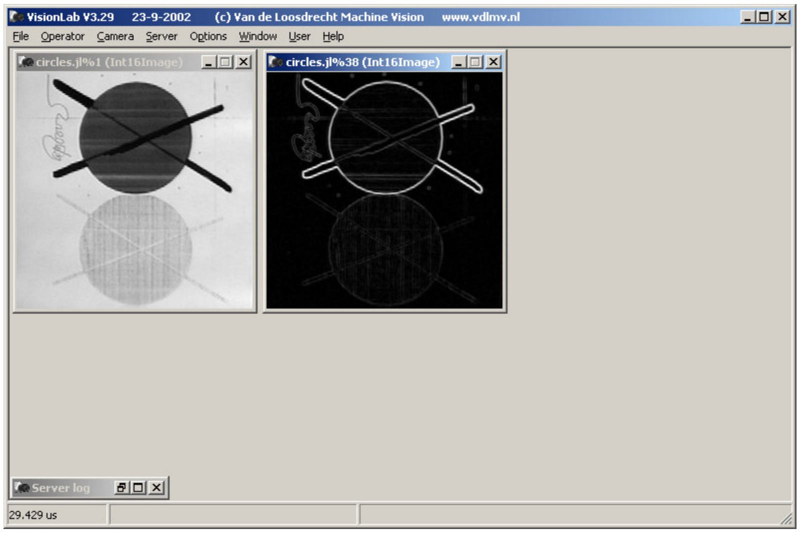
Demonstration Canny (*)

Try to find the signature and 'big circle'

- **Open image circels.jl**
- **Sobel GradientMagnitude 10000 0**
- **Threshold edge 60 1000, -> to much**
- **Threshold edge 150 1000, -> only 2 disconnected pixels of signature**
- **Canny image 0 60 150 EightConnected -> position of signature**

28-aug-08 Edge detection 41

Sobel GradientMagnitude 10000 0 (*)

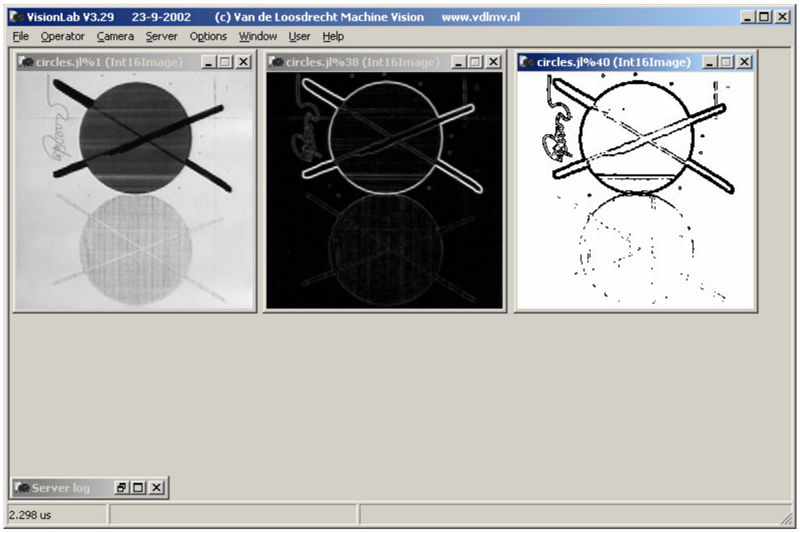


VisionLab V3.29 23-9-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl

29.429 us

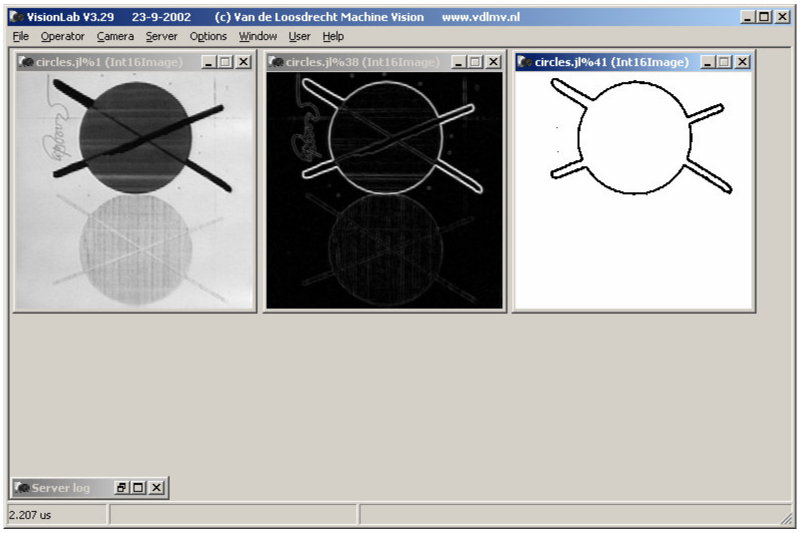
28-aug-08 Edge detection 42

Threshold edge 60 1000 (*)


VisionLab V3.29 23-9-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%1 (Int16Image)
circles.jl%38 (Int16Image)
circles.jl%40 (Int16Image)
Server log
2.298 us

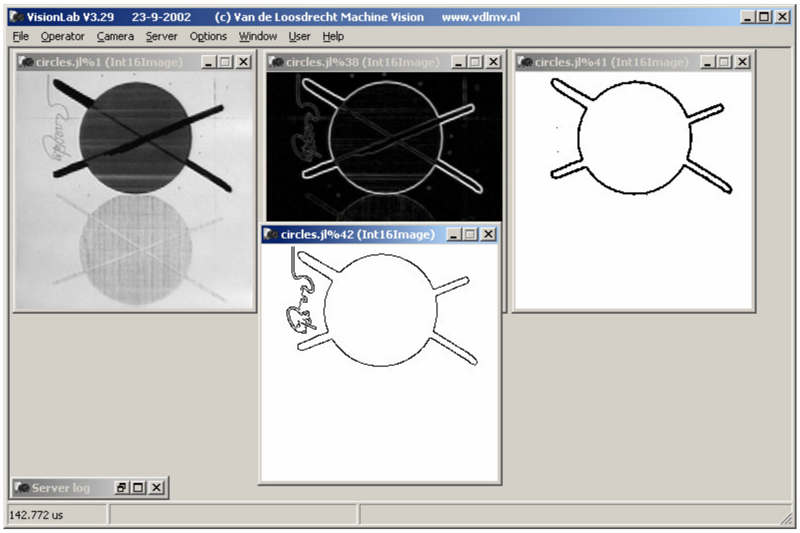
28-aug-08 Edge detection 43

Threshold edge 150 1000 (*)

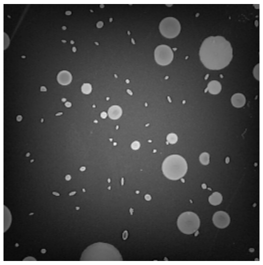

VisionLab V3.29 23-9-2002 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%1 (Int16Image)
circles.jl%38 (Int16Image)
circles.jl%41 (Int16Image)
Server log
2.207 us

28-aug-08 Edge detection 44

Canny image 0 60 150 EightConnected   (*)


VisionLab V3.29   23-9-2002   (c) Van de Loosdrecht Machine Vision   www.vdlmv.nl
File Operator Camera Server Options Window User Help
circles.jl%1 (Int16Image)
circles.jl%38 (Int16Image)
circles.jl%41 (Int16Image)
circles.jl%42 (Int16Image)
Server log
142.772 us

Exercise Segmentation using edge detection



- **Use image shading_c.jl in the excerise directory**
- **Try to find good threshold values in order to separate the cells from the background**
  This will be unsuccessful due to uneven lightning conditions
- **Use Sobel edge detection to find the borders of the cells and then to segment the image**
- **See shading_c_sobel.jls for answer**

Exercise (*)

- **Experiment with the the other edge detection operators on image shading_c.jl**

FindEdgeLine

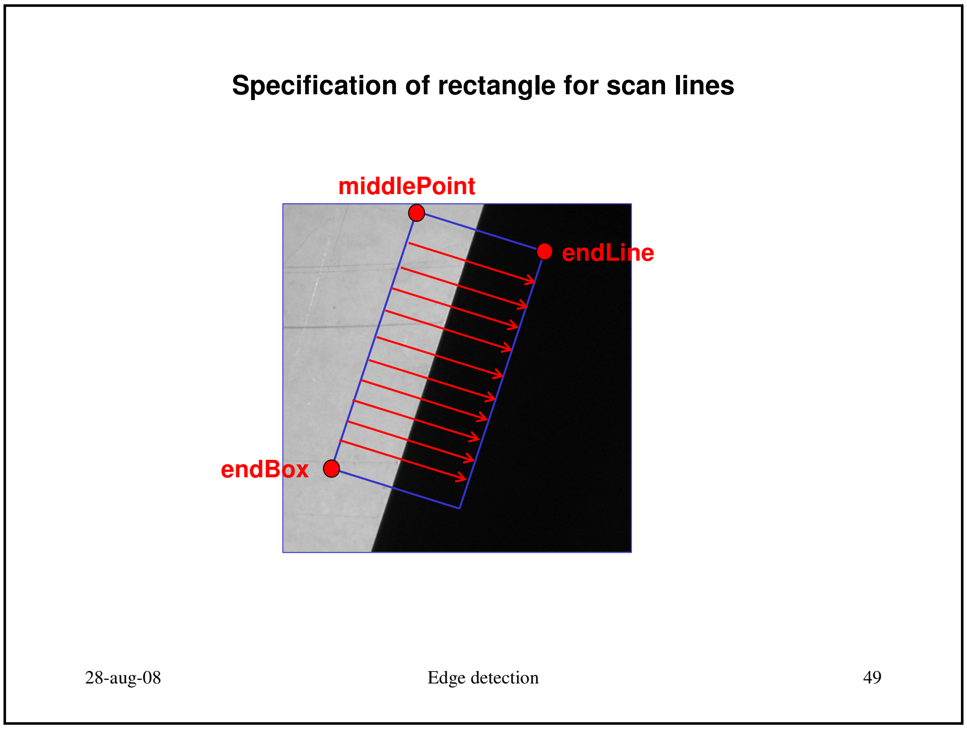
FindEdgeLine (image, middlePoint, endLine, endBox, lineDistance, outlierDistance, nrIterations)

This operator finds with subpixel precision a line with the largest edges within the specified rectangle middlePoint, endLine and endBox.

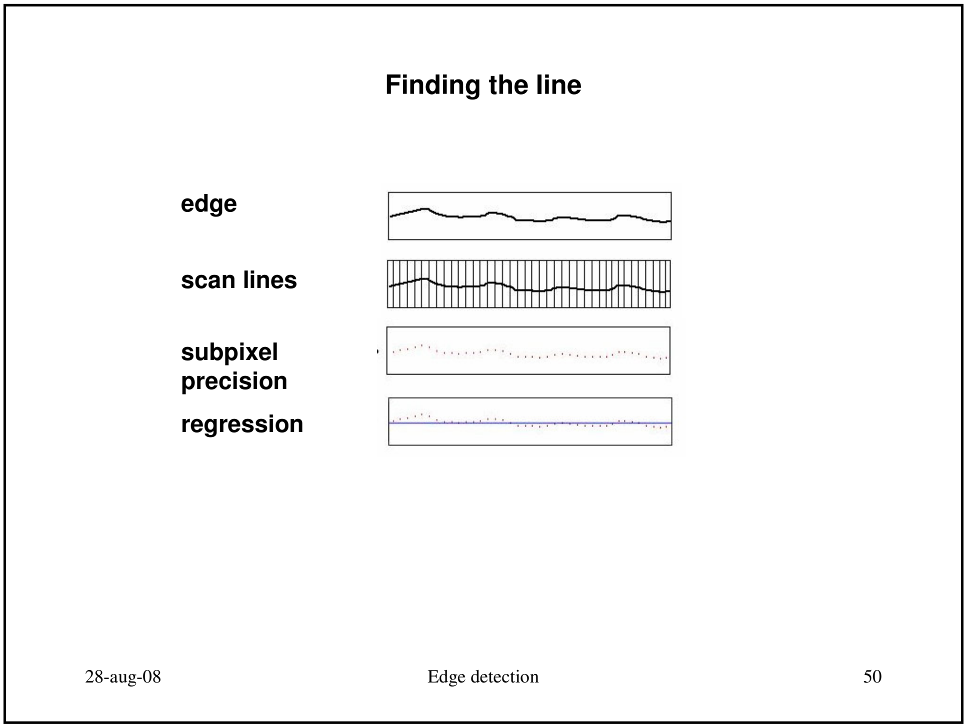
In the specified rectangle scan lines will be tested at the specified lineDistance. The rectangle should have a width of at least 5 pixels.

If outLayerDistance is greater then zero then the regression algorithm is repeated for nrIterations. In each next iterations only pixel with a distance smaller then outlierDistance to the previous found line are used in the calculation of the next line.

The function result is the start and the end coordinate of the line found and the number of pixels found on the line.



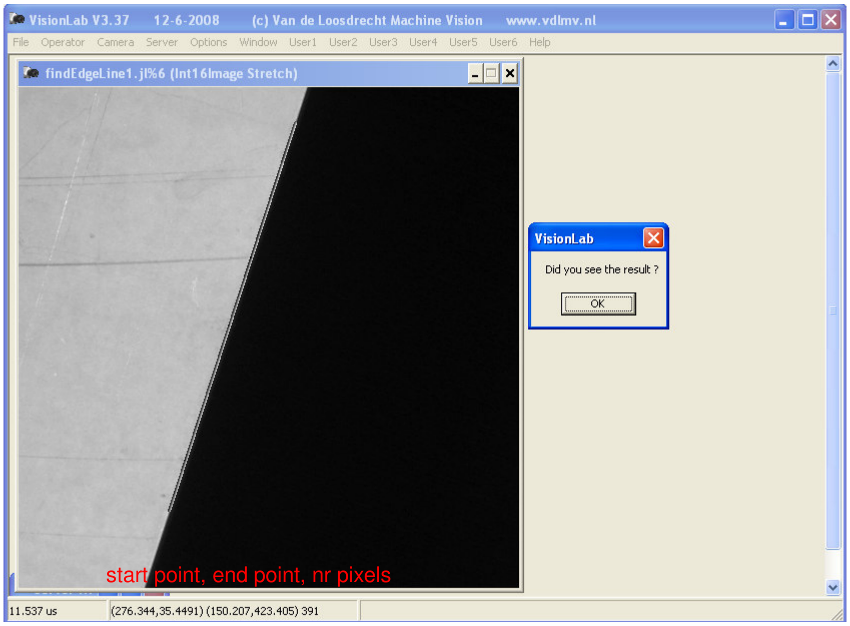
Specification of rectangle for scan lines



Finding the line

Demonstration FindEdgeLine

- Open image findEdgeLine1.jl
- FindEdgeLine (200,10) (350,60) (80,400) 1 10 1


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
findEdgeLine1.jl%6 (Int16Image Stretch)
VisionLab
OK

Did you see the result ?

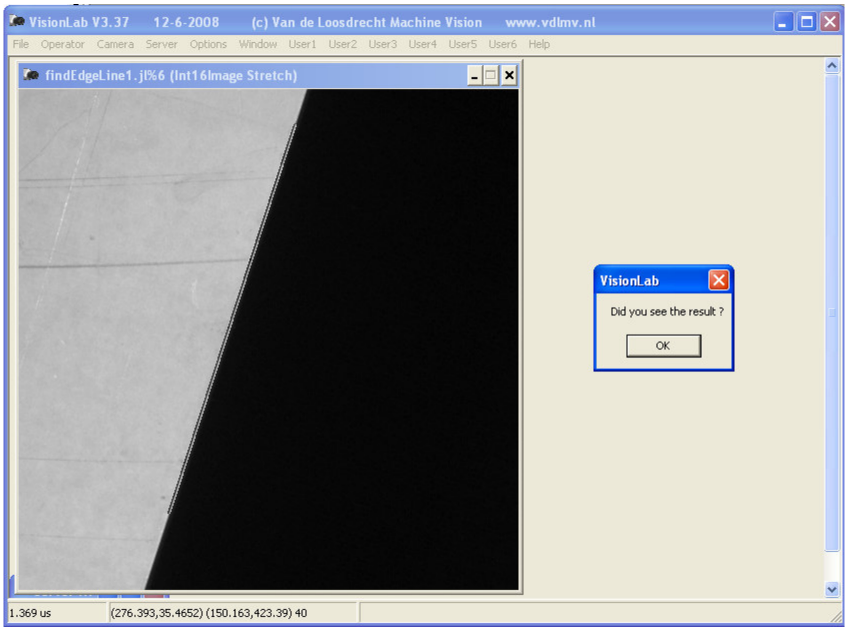
start point, end point, nr pixels

```
11.537 us    (276.344,35.4491) (150.207,423.405) 391
```

Demonstration FindEdgeLine

Increase speed by setting lineDistance to 10, decrease of accuracy (see number of points found)

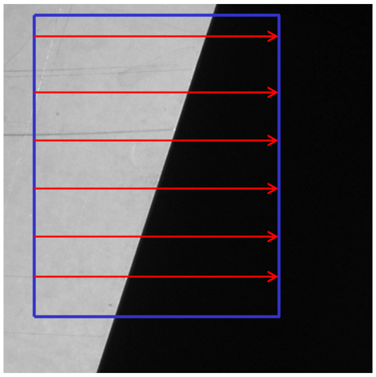
- FindEdgeLine (200,10) (350,60) (80,400) 10 10 1


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
findEdgeLine1.jl%6 (Int16Image Stretch)
VisionLab
OK

Did you see the result ?

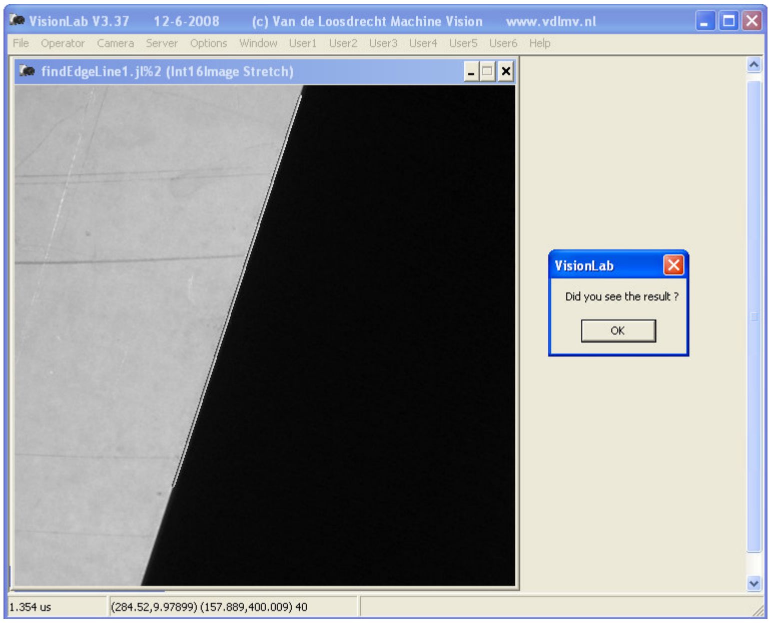
```
1.369 us    (276.393,35.4652) (150.163,423.39) 40
```

Example scan lines not perpendicular at edge



Demonstration scan lines not perpendicular to edge

- **Open image findEdgeLine1.jl**
- **FindEdgeLine (20,10) (350,10) (20,400) 1 10 1**


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
findEdgeLine1.jl%2 (Int16Image Stretch)
VisionLab
OK

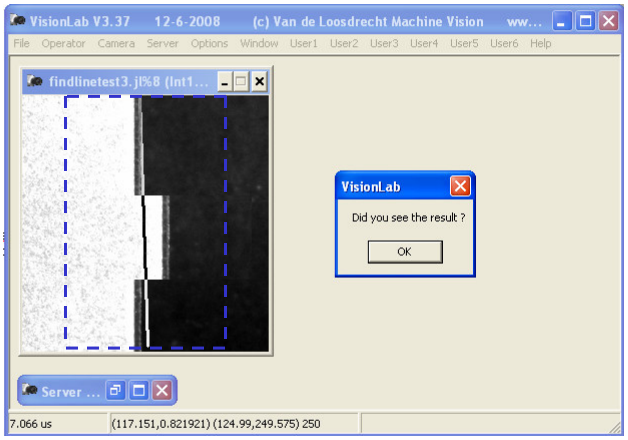
Did you see the result ?

```
1.354 us	(284.52,9.97899) (157.889,400.009) 40
```

Demonstration FindEdgeLine with outliers

- **Open image findedgeline2.jl**
- **FindEdgeLine (50,1) (200,1) (50,250) 1 0 1**

Demonstration FindEdgeLine with outliers no iterations


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision ww...
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
findlinetest3.jl%8 (Int1...
VisionLab
OK
Server ...

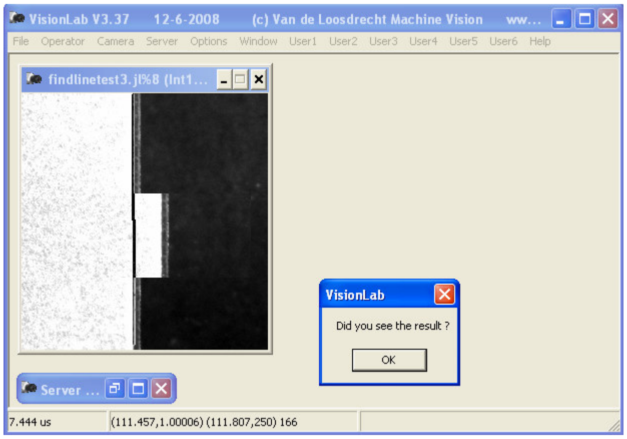
Did you see the result ?

```
7.066 us    (117.151,0.821921) (124.99,249.575) 250
```

Demonstration FindEdgeLine with outliers

- **FindEdgeLine (50,1) (200,1) (50,250) 1 10 4**

Demonstration FindEdgeLine with outliers 4 iterations


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision ww...
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
findlinetest3.jl%8 (Int1...
Server ...
VisionLab
OK

Did you see the result ?

7.444 us (111.457,1.00006) (111.807,250) 166

FindEdgeCircle

FindEdgeCircle (image, middlePoint, nrSamples, minR, maxR, outlierDistance, nrIterations)

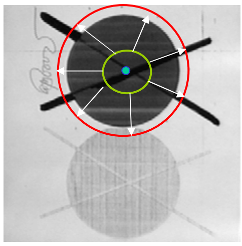
This operator finds with subpixel precision a circle within the specified disk shape specified by middlePoint, minR and maxR. In the specified disk shape nrSamples scan lines tested starting from a distance minR from the middlepoint and ending at a distance maxR from the middlepoint. The probe lines will be equally divided in the space bounded by middlePoint, minR and maxR.

Note: the number of probe lines will be nrSamples rounded up to the next multiple of 4. maxR must be > minR + 5.

If outLayerDistance is greater then zero then the regression algorithm is repeated for nrIterations. In each next iterations only pixel with a distance smaller then outlierDistance to the previous found line are used in the calculation of the next line.

The function result is the center coordinate, the radius and the number of pixels found on the circle.

Specification of disk shape for scanlines



middlePoint

minR

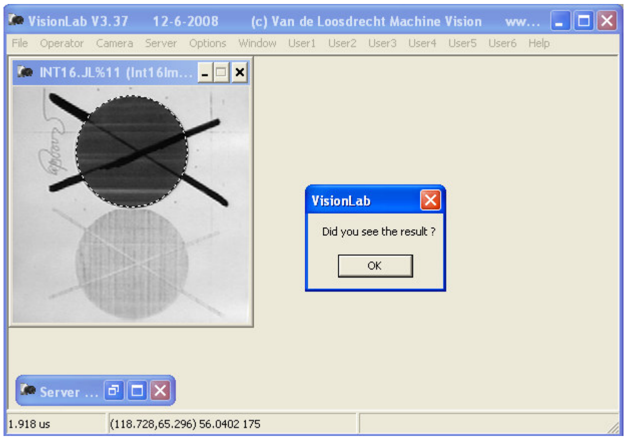
maxR

Scan lines

Demonstration FindEdgeCircle

- Open image circles.jl
- FindEdgeCircle (118,65) 200 20 65 1 1
- FindEdgeCircle (118,65) 50 20 65 1 1 (faster)

FindEdgeCircle (118,65) 200 20 65 1 1


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision ww...
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
INT16.JL%11 (Int16Im...
VisionLab
OK
Server ...

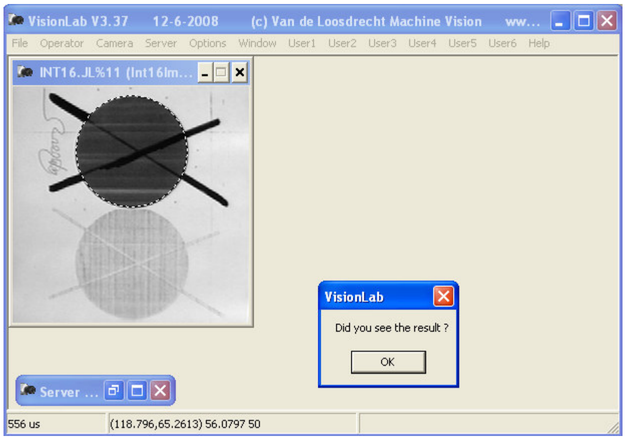
Did you see the result ?

```
1.918 us    (118.728,65.296) 56.0402 175
```

center, radius, nr pixels

28-aug-08 Edge detection 65

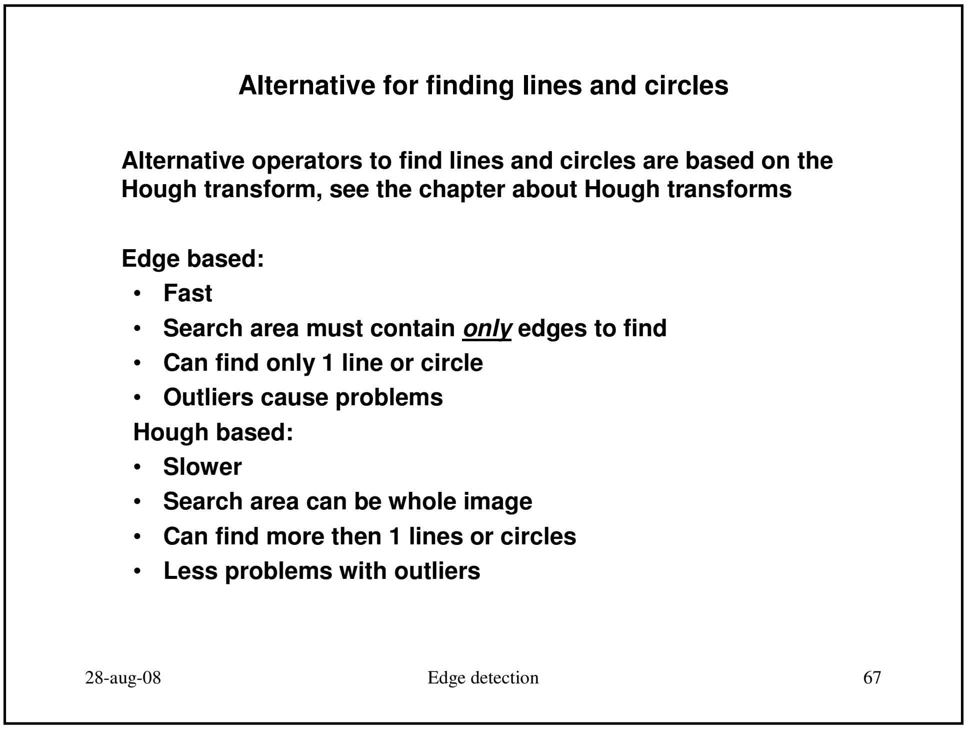
FindEdgeCircle (118,65) 50 20 65 1 1


VisionLab V3.37 12-6-2008 (c) Van de Loosdrecht Machine Vision ww...
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
INT16.JL%11 (Int16Im...
VisionLab
OK
Server ...

Did you see the result ?

```
556 us    (118.796,65.2613) 56.0797 50
```

28-aug-08 Edge detection 66



Alternative for finding lines and circles

Alternative operators to find lines and circles are based on the Hough transform, see the chapter about Hough transforms

Edge based:

- **Fast**
- **Search area must contain *<u>only</u>* 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 then 1 lines or circles**
- **Less problems with outliers**