Computer Vision

Non linear filters

27 August 2008

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

Non linear filters

Overview:

- **Image enhancement**
  - **Mean filter**
  - **Median filter**
  - **Mode filter**
  - **Sigma filter (*)**
- **Grey-scale morphology**
  - **Minimum filter**
  - **Maximum filter**
  - **Nth filter**

Non linear filters

Overview:

- **Local extremes**
  - **LocalMax filter**
  - **LocalMin filter**
- **Edge preserving smoothing (*)**
  - **Kuwahara filter**
  - **SNN_Mean filter**
  - **SNN_Median filter**
- **"Texture" (*)**
  - **Range filter**
  - **Variance filter**

28-aug-08 Non linear filters 3

Rank operators

The ranking operator initialises a destination image by sliding a mask across a source image.

The pixel values under the mask are used to calculate a new value. Each type of rank operator uses its own algorithm.

This new value is assigned to the destination image at the position of the centre (= origin) of the mask.

Rank operators are non-linear filters.

28-aug-08 Non linear filters 4

Image enhancement

- **Mean filter:**
  the new value is the mean value of the selected pixels under the mask.
- **Median filter:**
  the new value is the median value of the selected pixels under the mask.
  The median is the middle value in the sorted order of values.
- **Mode filter:**
  the new value is the mode value of the selected pixels under the mask.
  The mode is the value with the highest frequency of occurrence.

28-aug-08 Non linear filters 5

Image enhancement (*)

- **Sigma filter:**
  the new value is the mean value of the selected pixels under the mask if the absolute difference between the mean value and the original pixel is smaller then the specified deviation. Otherwise the new value is the value of the origin pixel.
- **Usage: noise reduction**

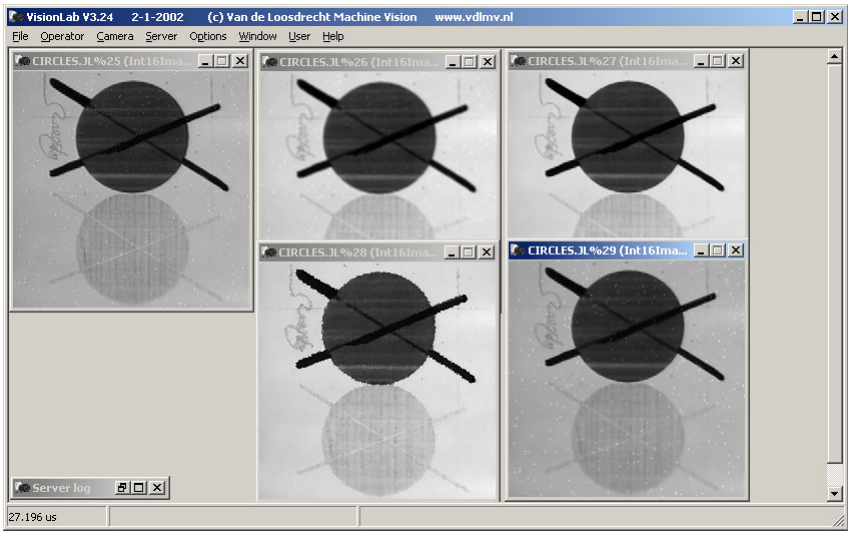
28-aug-08 Non linear filters 6

Demonstration image enhancement (*)

- Open image circles.jl
- Add noise 1 0 50
- Apply the the three filters with 3x3 mask to image with noise
- (* for Sigma use deviation = 10)
- Median filter gives best result
- Mode filter gives worse result
- Note there are different kinds of noise, in this case only impulse noise is investigated.
- Note Median with plus 3x3 mask gives even a better result.

28-aug-08 Non linear filters 7

Image with impulse noise **Mean** **Mode** **Median** **Sigma (*)**



28-aug-08 Non linear filters 8

Grey-scale morphology

- **Minimum filter:**
  the new value is the minimum value of the selected pixels under the mask.
  Used for grey-scale erosion and grey-scale dilation.
- **Maximum filter:**
  the new value is the maximum value of the selected pixels under the mask.
  Used for grey-scale dilation and grey-scale erosion.
- **Nth filter:**
  the new value is the nth value of the ascending sorted selected pixels under the mask.
  This is a generalisation of Maximum and Minimum filter.

28-aug-08 Non linear filters 9

Grey-scale morphology

- **Usage:**
  - **Grey-scale opening**
  - **Grey-scale closing**
  - **Noise reduction**

28-aug-08 Non linear filters 10

Demonstration grey-scale morphology

- Open image circles.jl
- Apply 3 x Maximum filter with full 3x3 mask
- Result: object border is replaced by background, little 'noise' dots are disappeared

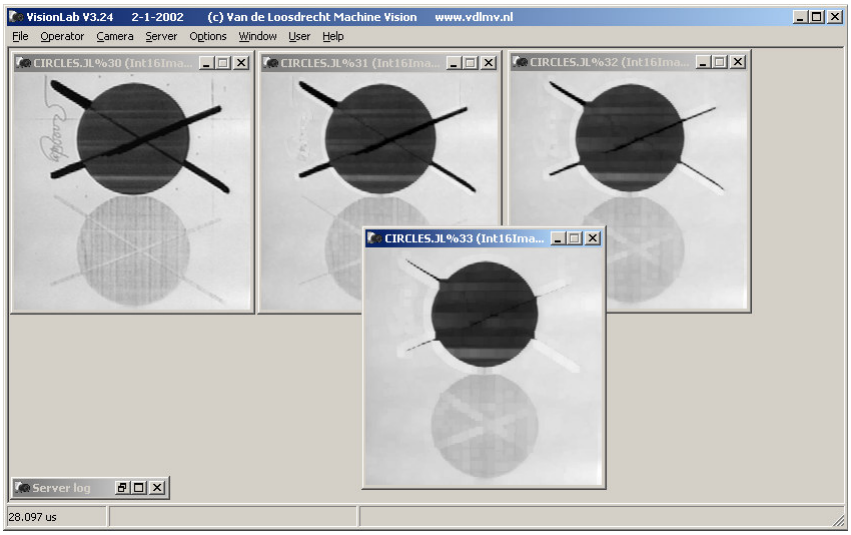
- Subtract 1x Maximised image from original (= 2nd image)
- Threshold 40 1000 on result gives edge of dark circle

- Apply 2x Minimum filter with full 3x3 mask on circle.jl
- Result: objects grow

- Apply 3 x Maximum filter with 7x7 mask (no slide) result object shrink faster in one operation

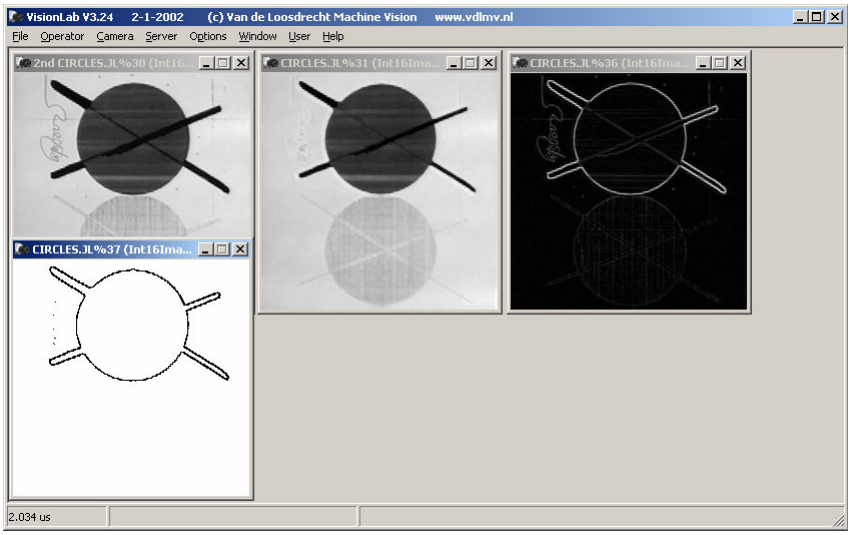
28-aug-08 Non linear filters 11

Apply 3 x Maximum filter with full 3x3 mask



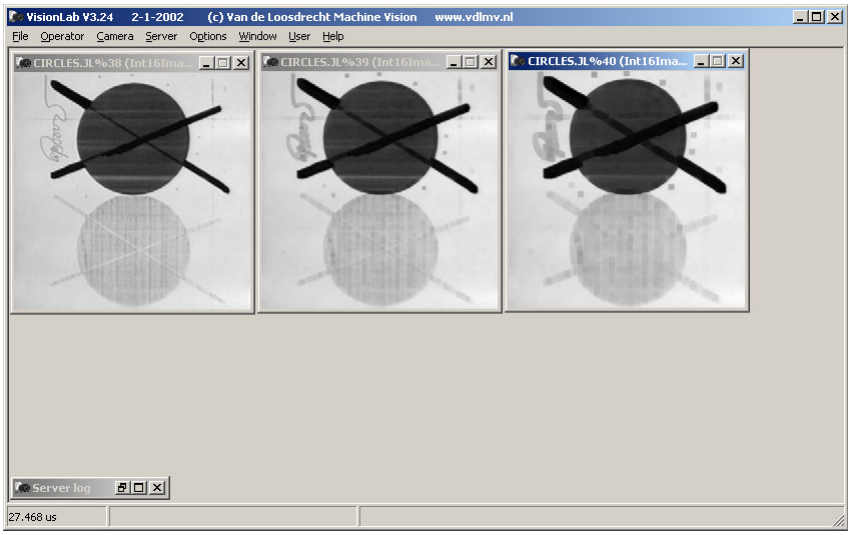
28-aug-08 Non linear filters 12

Subtract 1x Maximised image from original (= 2nd image) followed by Threshold 40 1000



28-aug-08 Non linear filters 13

Apply 2x Minimum filter with full 3x3 mask



28-aug-08 Non linear filters 14

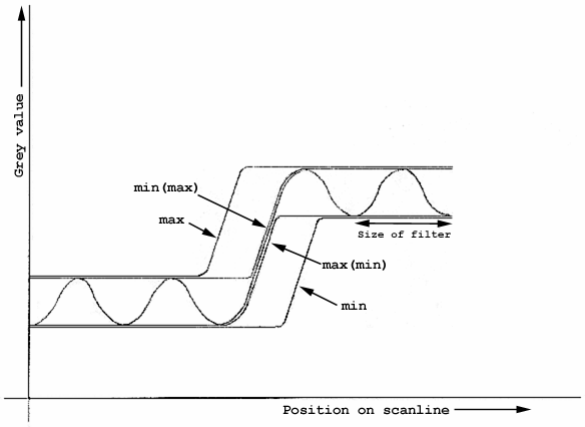
Grey-scale morphology

- **Finding back-ground behind small objects**

  Max and Min used in combination,

  Max(Min(image)) or Min(Max(image))

Max(Min(image)) versus Min(Max(image))

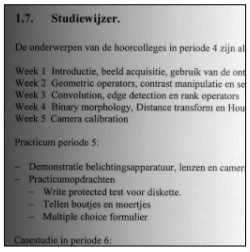


Max(Min(image)) versus Min(Max(image))

- For large edges same result
- Max(Min(image))
  - On bottom side of the small edges
  - Used for generation of dark backgrounds, "removes the bright spots"
- Min(Max(image))
  - On top side of the small edges
  - Used for generation of bright backgrounds, "removes the dark spots"

28-aug-08 Non linear filters 17

Exercise background generation 1


1.7. Studiewijzer.

De onderwerpen van de hoorcolleges in periode 4 zijn al

Week 1 Introductie, beeld acquisitie, gebruik van de ont
Week 2 Geometrie operators, contrast manipulatie en se
Week 3 Convolution, edge detection en rank operators
Week 4 Binary morphology, Distance transform en Hou
Week 5 Camera calibration

Practicum periode 5:

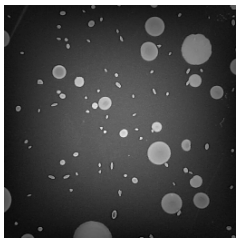
- Demonstratie belichtingsapparatuur, lenzen en camer
- Practicumopdrachten
  - Write protected test voor diskette.
  - Tellen boutjes en moertjes
  - Multiple choice formulier

Casestudie in periode 6:

- Use image stdhand_r.jl.
- Try to find good threshold value in order to separate characters from the background.
  This will be unsuccessful due to uneven lightning conditions
- Remove background from image
- Now try to find good threshold value in order to separate characters from the background.
- answer: stdhand_r.jls

28-aug-08 Non linear filters 18

Exercise background generation 2



- **Use image shading_c.jl**
- **Try to find good threshold value in order to separate the cells from the background.**
  This will be unsuccessful due to uneven lightning conditions
- **Remove background from image**
- **Now try to find good threshold value in order to separate the cells from the background.**

- **answer: shading_c_back.jls**

28-aug-08 Non linear filters 19

Local extremes filters (* for exercise city only)

- **LocalMax filter**
  - **The local maximum filter operator initialises a destination image by sliding a mask across the source image. A new value is calculated for the destination image at the position of the centre (= origin) of the mask.**
    This new value is the value of the origin if the origin value is the local maximum of the pixels under the mask otherwise the new value is set to the background value.
  - **If the source pixel value at the origin equals to the background value (=0) the corresponding destination pixel is assigned the background value.**
- **LocalMin filter**

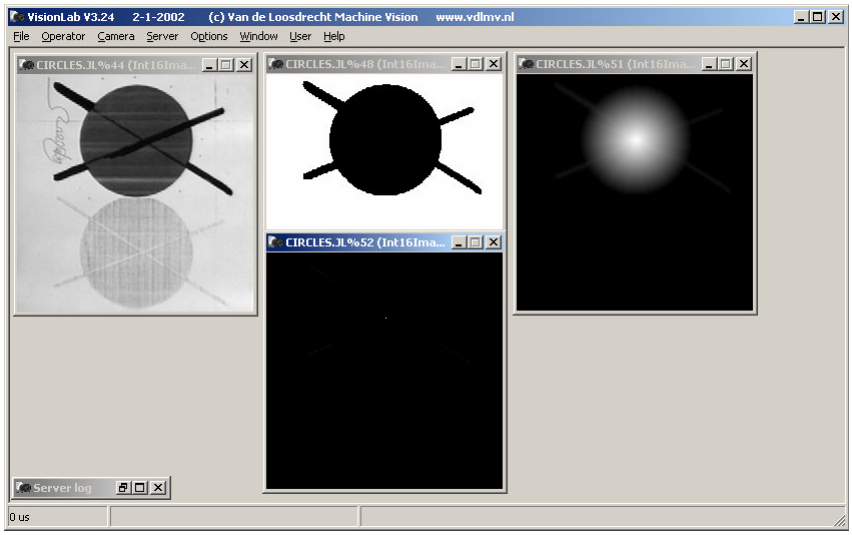
28-aug-08 Non linear filters 20

Demonstration LocalMax filter (*)

Find middle of circle in circle.jl:

- **Open image circle.jl**
- **Threshold 0 100**
- **EuclideanDistanceTransform EDTMask7x7 NoScaleEDT 100 1**
  note: there will be a special lecture about distance transforms
- **LocalMaxFilter 0 EdgeExtend 3 3 1 1 1 1 1 1 1 1 1 1 1**
- **Gamma 0.25 // for better displaying**

Demonstration LocalMax filter (*)



Edge preserving smoothing Filters (*)

Image is smoothed but edges are preserved
- **Kuwahara filter**
- **Symmetric Nearest Neighbour (SNN) filter**

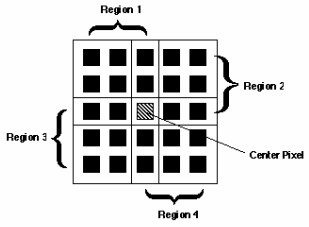
Usage:
- **Sharpen up vague edges before using edge detection**
- **Artistic effects**

Kuwahara filter (*)

KuwaharaFilter (srcImage, destImage, radius, edge)

The square window with the defined radius around the center pixel is divided in four overlapping regions.

Example for radius = 3:
- **window is 5x5 pixels**
- **4 regions of 3x3 pixels**
- **center pixel is in all regions**



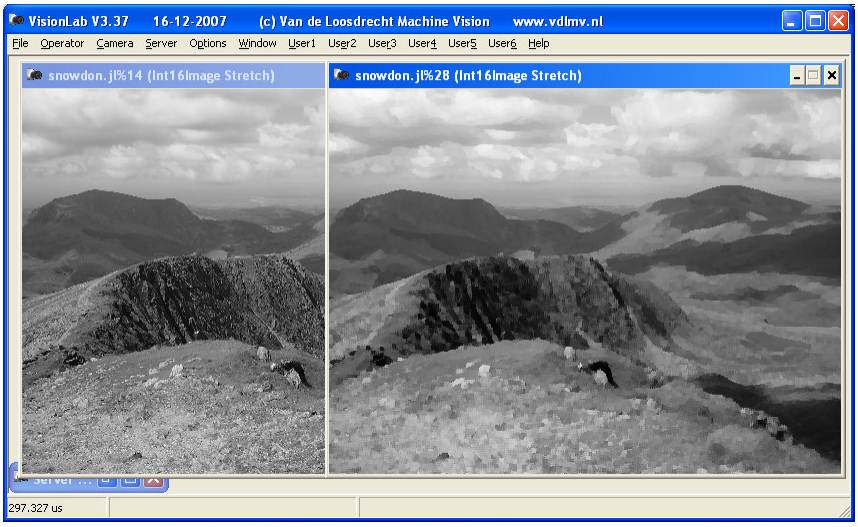
The output value for the central pixel in the window is the mean value of that region that has the smallest variance.

Demonstration Kuwahara filter (*)

- Open image snowdon.jl
- Apply on image KuwaharaFilter 3 EdgeExtend
- Apply on image KuwaharaFilter 8 EdgeExtend
- Apply Sobel on orginal image
- Apply Sobel on KuwaharaFilter with radius 8

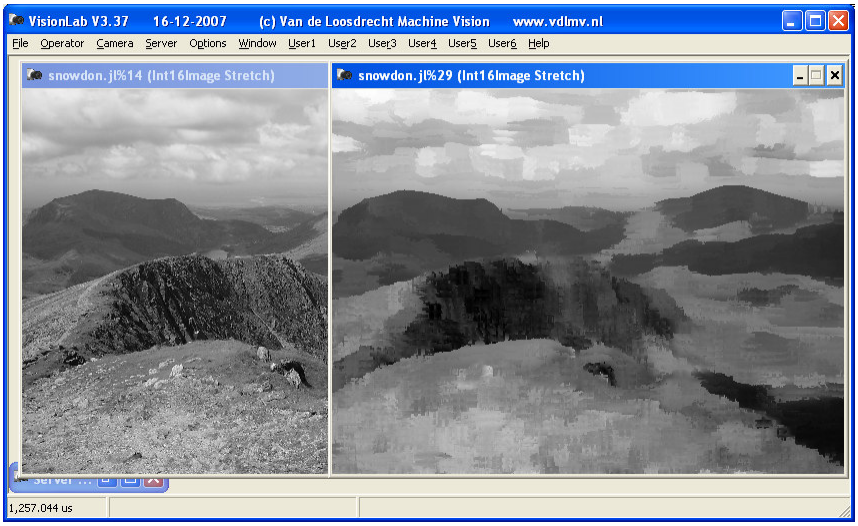
Original image (*)

Kuwahara with radius 3 (*)



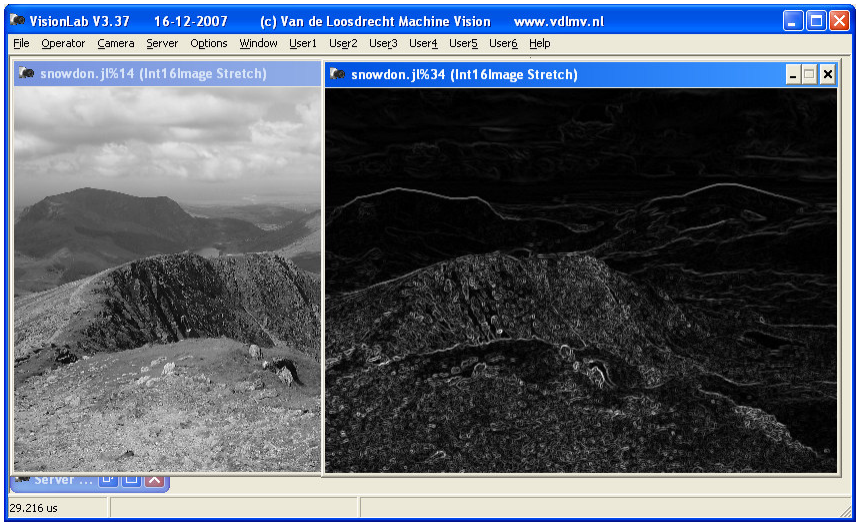
28-aug-08 Non linear filters 27

Kuwahara with radius 8 (*)



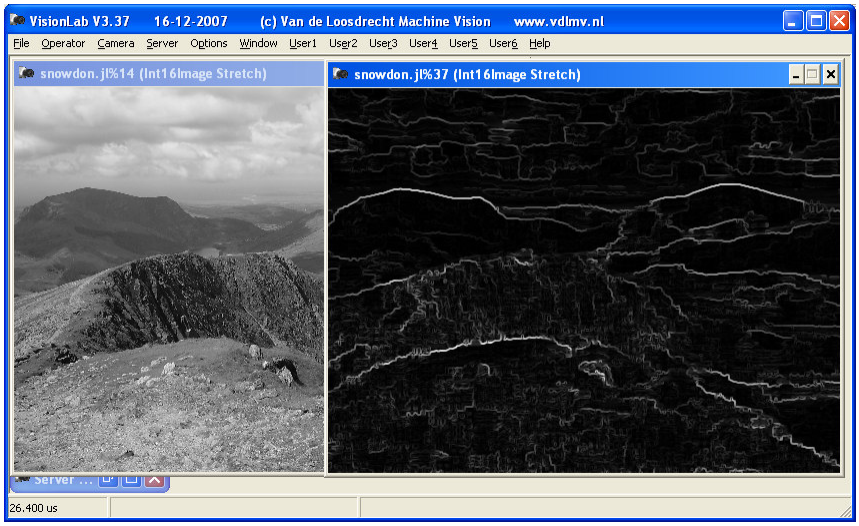
28-aug-08 Non linear filters 28

Sobel on original image (*)



28-aug-08 Non linear filters 29

Sobel on Kuwahara with radius 8 (*)



28-aug-08 Non linear filters 30

Symmetric Nearest Neighbour (SNN) filter (*)

SNN_MeanFilter (srcImage, destImage, radius, edge)
SNN_MedianFilter (srcImage, destImage, radius, edge)

SNN compares symmetric pairs of pixels within a defined radius with the center pixel.
For each pair of pixels the one which is closest in value to the center pixel is calculated.

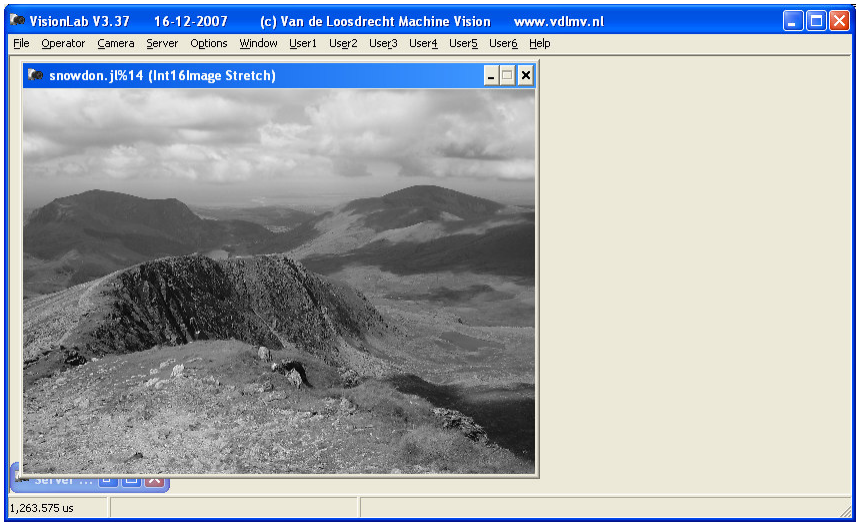
For the SNN_MeanFilter the new pixel value assigned to the center pixel is the mean of the closest pixels.

For the SNN_MedianFilter the new pixel value assigned to the center pixel is the median of the closest pixels.

Demonstration SNN filter (*)

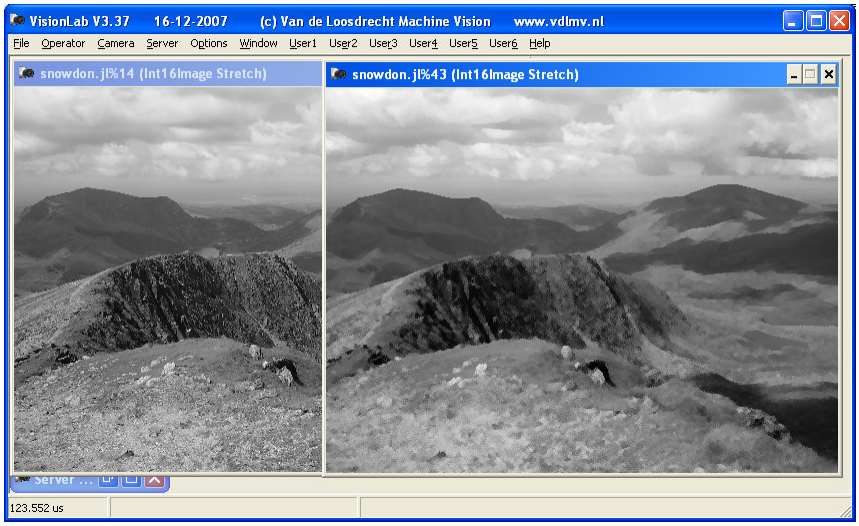
- **Open image snowdon.jl**
- **Apply on image SNN_MeanFilter 3 EdgeExtend**
- **Apply on image SNN_MeanFilter 8 EdgeExtend**
- **Apply on image SNN_MedianFilter 8 EdgeExtend**
- **Note: not much difference between Mean and Median**

Original image (*)



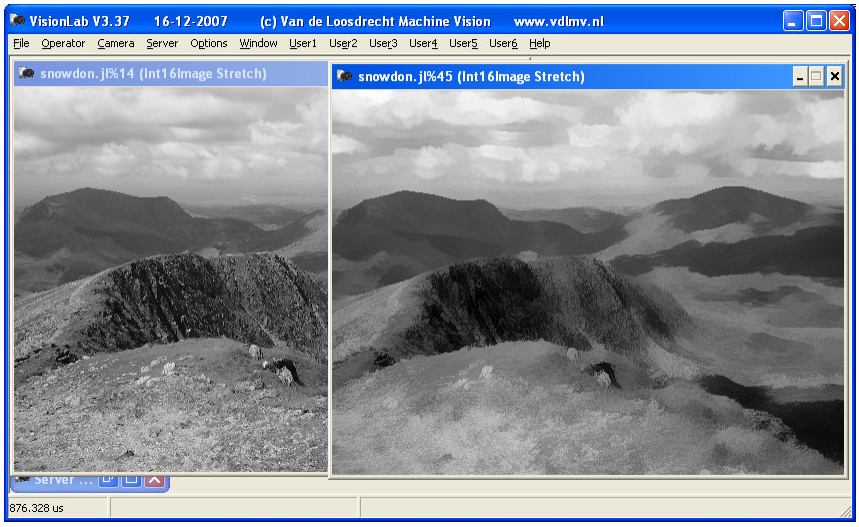
28-aug-08 Non linear filters 33

SNN_Mean with radius 3 (*)



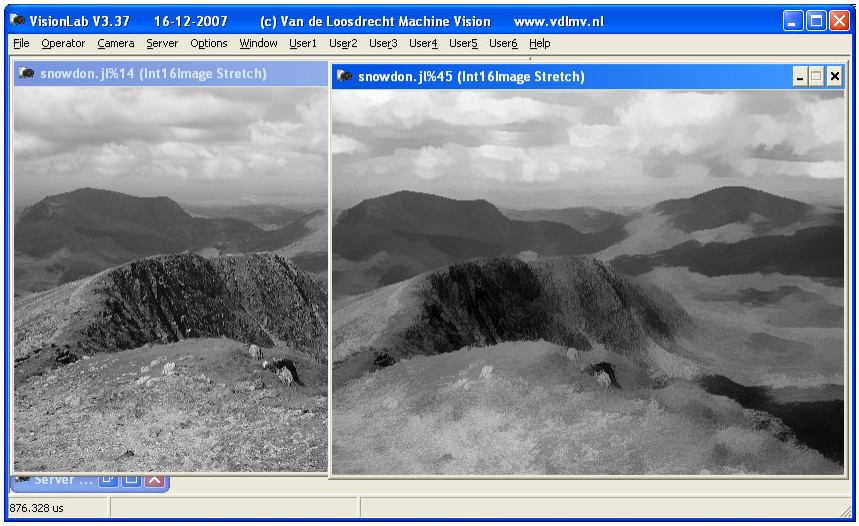
28-aug-08 Non linear filters 34

SNN_Mean with radius 8 (*)



28-aug-08 Non linear filters 35

SNN_Median with radius 8 (*)



28-aug-08 Non linear filters 36

"Texture" (*)

If brightness is interpreted as elevation then a variation in brightness is called a texture.
A texture is a measure of surface roughness.

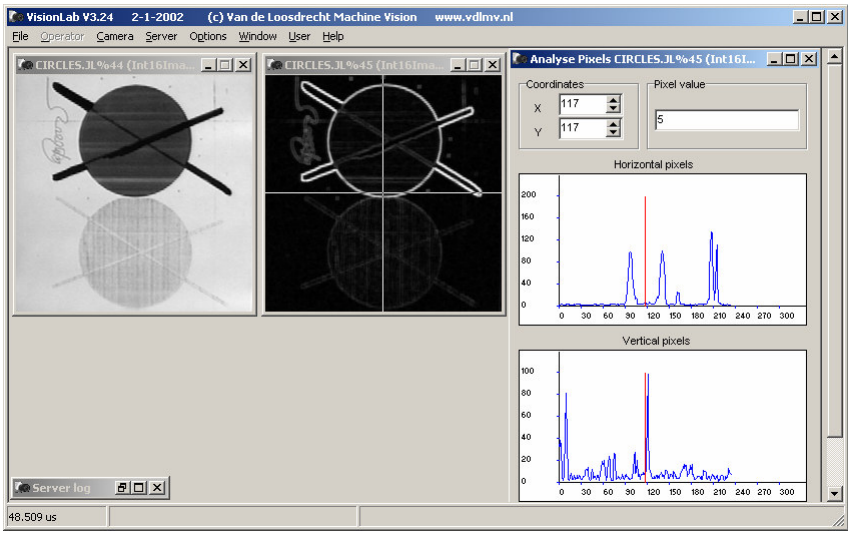
- **Range filter:**
  the new value is the difference between the maximum value and the minimum value of the selected pixels under the mask.
- **Variance filter:**
  the new value is the square root of the sum of the squares of the difference between values of the central pixel and its neighbours.

Both filters give a primitive measurement for texture. Looks similar to edge detection, but gives a lower response on the edges.

Demonstration "texture" (*)

- **Open image circles.jl**
- **Apply both operations with EdgeExtend to both images**

Range filter (*)



Variance filter (*)