NHL STENDEN
computer vision
& data science

Van De Loosdrecht
Machine Vision

Computer Vision

Image acquisition

10 April 2018

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

Image acquisition

Overview:

- Camera
- Lens
- Frame grabber
- Lighting
- Signal to Noise Ratio (*)

Cameras

- Web cams
- Medium tech
- High tech

- Price
- Quality
- Resolution
- Frames per second
- Specials:
  - Peltier cooling
  - Image amplifiers

Analogue camera, WAT-505EX

Camera

Sensor type:

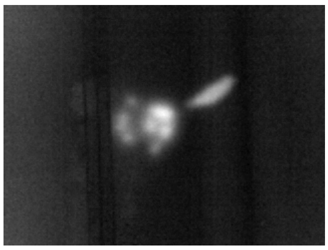
- **CCD**
- **CMOS**
- **Infra red**
- **X-rays**
- **Radar**
- **Sound**
- **MRI**
- **Radio telescope (astronomy)**

Infra red camera for thermal imaging from –20C to 250C

normal color image **infra red image**

27-Aug-18 Image acquisition 7

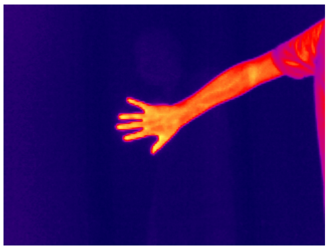
arm on cupboard (ir) **after arm removed (ir)**



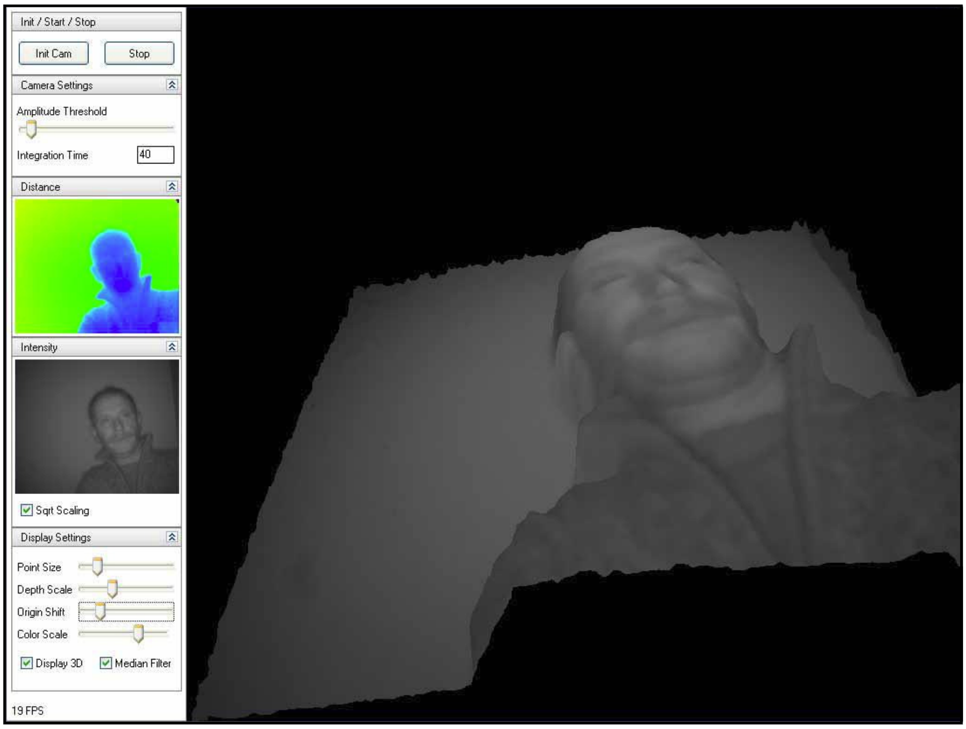
27-Aug-18 Image acquisition 8

infra red raw image

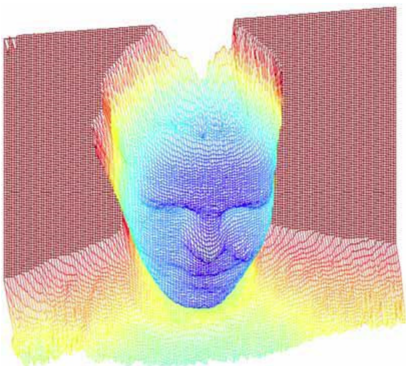
in false colors



Range camera, Time Of Flight

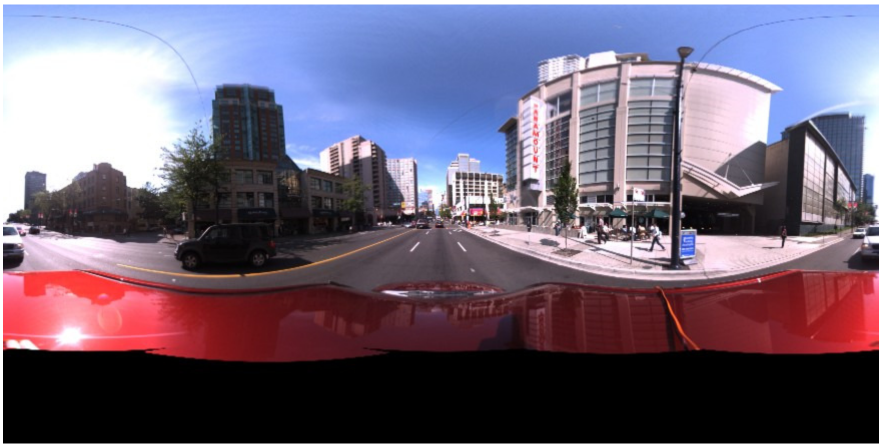


Range camera, Time Of Flight



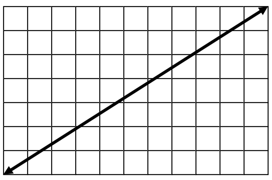
Lady bug, 360 graden beeld

Lady bug, 360 graden beeld



Stereo vision

Sensor: array of pixels



- **Resolution:**
  - **Number pixels width x height**
  - **Typical values: 640 x 480, 800 x 600, 1280 x 1024**
- **Frames per second:**
  - **Typical values: 5 to 200**
- **Diagonal chip is the size:**
  - **Typical values: 1/3", 1/2", 2/3" en 1"**
  - **Important for choice of lens (vignetting)**
- **Fill factor:**
  - **Light sensitivity**
  - **Micro lenses**

Vignetting

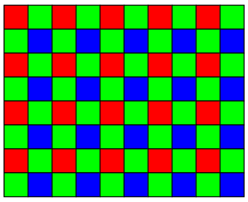
27-Aug-18 Image acquisition 17

Sensor: array of pixels

- **Pixel size**
  - **Important for the light sensitivity**
  - **Typical values: 3 - 12 µm**
  - **Square: important by size measurements**
- **Pixel "depth" (dynamic range)**
  - **Number of gray values**
  - **8 bits = 256 (usually enough)**
  - **10 bits = 1024**
  - **12 bits = 4096**
- **Gain and offset**
  - **Pixel value = offset + amount of light * gain**
  - **High gain -> more noise**
- **Spectral sensitivity (quantum efficiency)**

27-Aug-18 Image acquisition 18

Color sensor



- 1 chip
  - Pattern with filters and interpolation
  - Less sharpness than comparable grayscale camera
- 3 chips
  - More expensive
  - Less shockproof

Raw format

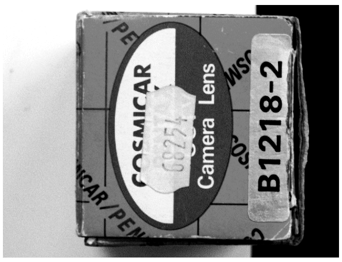


Image is grayscale image with raw color sensor information

Demonstration:

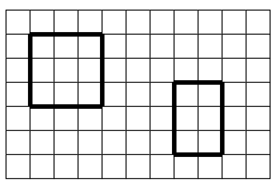
- **Open image: raw_image.jl**
- **ConvertCFAtoRGB888Image image BayerGB**

Sensor: CCD versus CMOS

- CCD
  - Better signal / noise ratio
    better suitable in low light situations
- CMOS
  - Cheaper
  - Easy integration at chip level
  - Pixel addressable
    - Windowing
    - Sub sampling and binning
  - High Dynamical Range

Pixel addressable

- Windowing
  - Part of sensor surface -> higher frame rate
  - Multiple windows in one snapshot

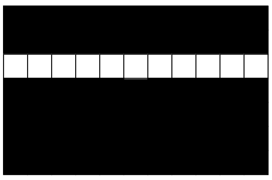


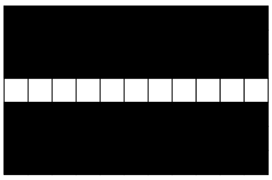
- Sub sampling and binning
  - n by n neighbour pixels are combined to 1 pixel
    image factor n smaller-> higher frame rate

High Dynamical Range

Shutter

- **Typical values : 1/15 .. 1/10.000 second**
- **Mechanic**
- **Electronic**
  - **Global shutter**
  - **Rolling shutter**
    problems with fast moving objects





Rolling shutter

Fokker Dr I, Stichting Vroege Vogels Lelystad

Digitalization

From "continuous analogue light" to sampled digital image:

- **Spatial co-ordinates**
- **Intensity value**
- **Time**

Sensitive to a specific part of the spectrum

Analogue CCD Camera

Analogue output

- **Monochrome**
  - **(EIA) RS-170 Video, 30 fps, 640 x 480 lines**
  - **CCIR, 25 fps, 768 x 576 lines**
  - **Frame grabber converts typically to 8-bit grey scale**
- **Colour composite**
  - **NTSC, 30 fps**
  - **PAL, 25 fps**
- **Non standard video**

Analogue CCD Camera

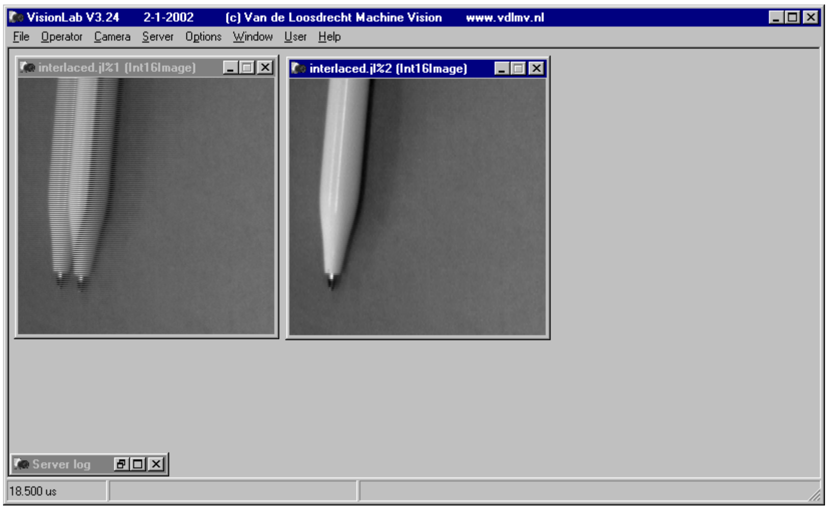
Scanning

- **Interlaced**
  - **Cheap**
  - **First, even, odd, both**
- **Progressive scan**
  - **Expensive**
  - **Motion applications**
- **External triggering**
- **Pixel dimensions / dimensions of ccd chip**
  square pixels are important for measurements
- **Shutter speed (typical: 1 - 1/10.000 second)**

Demonstration Interlaced versus Progressive Scan

- Open image interlaced.jl
- DeInterlace image (from point menu)

DeInterlace



Calculation ‘real’ pixel ratio for analogue camera (*)

Example

- **WAT-505EX (old type):**
  - **chip: 596 (V) x 795 (H) pixels**
  - **pixel: 6.5 ųm (V) x 6.25 ųm (H)**
- **Frame grabber (CCIR):**
  - **576 lines of 768 pixels**
- **‘real’ pixel width: (795 / 768) * 6.25 = 6.47 ųm**
  squareness of pixel: (6.5 / 6.47) = 1.005
- **596 - 576 = 20 lines are not used**

Analogue frame grabber, RIO full version

Analogue frame grabber (*)

- **Host bus (PCI, AT)**
- **Spatial resolution (768 x 576 25 Hz)**
- **Intensity resolution (8 bit)**
- **Video input**
  - **Number of input channels**
  - **Type (RS-170, PAL, etc)**
- **Accuracy**
- **On board processing**
  Input LUT's, ROI, scaling, etc
- **Digital I/O**
- **Video display**
- **Software driver**

Analogue camera triggering (*)

- **Analogue camera in free running mode**
  example 25 frames (images) per second (fps)
  - **Triggering in software**
  - **Triggering by frame grabber**
- **Triggering of camera**
  - **Asynchrone reset**

Digital CCD Camera

Digital output

- **Digital area scan**
  - **Larger image size possible (4k x 4k)**
  - **Higher resolution (8 .. 12 bit)**
  - **Fast acquisition (>100 Mbytes/s)**
    example: 1k x 1k, 100 fps
  - **Windowing**
  - **Binning**
  - **Applications: machine vision, scientific**
- **Digital line scan camera**
  - **10k - 100k lps**
  - **Processing**
    - **Line by line**
    - **Stitched together into 2D image**
  - **Application: high speed motion**

Digital line scan camera

Standards for digital camera interfaces

- Camera link
  - Highest performance
  - Expensive frame grabber
- FireWire (IEEE 1394a and 1394b)
  - Available on common main boards
  - IIDC standard DCAM
  - DV is for "handy cams"
- USB (1,2 and 3)
  - Available on common main boards (1 and 2)
- GigE Vision
  - Gigalink ethernet
- CoaXPress

Standards for digital camera interfaces: comparision

Standard	Max Length	Speed
USB 2.0	5 m	480 Mb/s
USB 3.0 (USB3 Vision)	5 m	4800 Mb/s
1394a (Firewire A)	4.5 m	400 Mb/s
1394b (Firewire B)	4.5 m	800 Mb/s
Future: 1394c (FireWire C)		3200 Mb/s
CameraLink	10 m	6120 Mb/s
Gigabit Ethernet(GigE)	100 m	1000 Mb/s
Gigabit Ethernet(10GigE)	100 m	10000 Mb/s
CoaxPress	100 m	10000 Mb/s

- The max length can be enlarged using repeaters
- The real speed (fps) is also depended on the overhead of the protocol used

Standards for digital camera interfaces

- Gen<I>Cam
  - One SDK for
    - GigE Vision
    - IEEE 1394
    - Camera Link
  - GenApi: configuring the camera
  - SFNC: Standard Feature Naming Convention
  - GenTL: Transport Layer convention

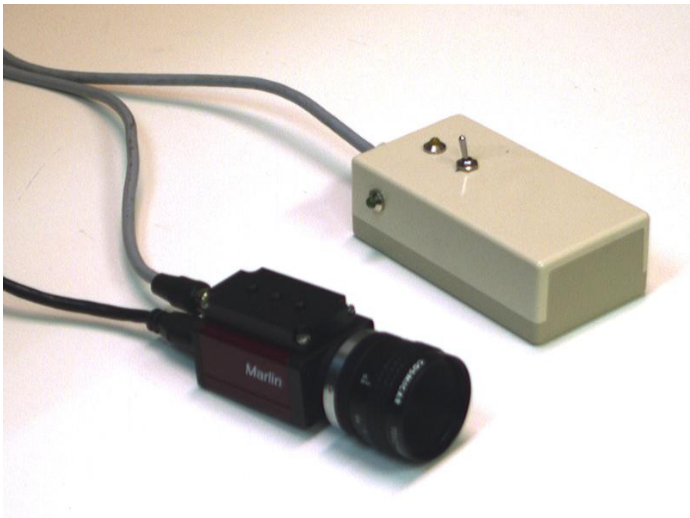
27-Aug-18 Image acquisition 39

Real time behaviour

- Real time loop:
  - Acquisition image
  - Process image, do measurement
  - Activate outputs / log result
- Process time will fluctuate,
  buffering of images needed by:
  - operating system
  - frame grabber
  - camera

27-Aug-18 Image acquisition 40

Camera triggering



Camera trends

- Analogue cameras will be used less
- CMOS sensor will increase its market share, CCD will lose
- Digital FireWire and USB cameras have become cheaper and better
- Now: for best performance Camera Link
- Since 2005: FireWire IEEE 1394b camera's on the market
- Since 2005: GigaE Vision
- Gen<I>Cam

Lenses

Lens

- Lens mount: C, CS, F, ….
- Focal length (mm), fixed or zoom
- Aperture or diaphragm (F)
- Depth of field
- Minimum focal distance
- Geometric distortion -> telecentric lenses
- Field of view
  - Size of CCD chip
  - Focal length
  - Distance to object

Lighting

Requirements:

- **Homogeneous light over field of view**
- **Maximum contrast for features of interest**
- **Minimum contrast for features of non interest**
- **Minimum sensitivity to:**
  - **Environmental variations (ambient light)**
  - **Feature variations**

Lighting techniques

- **Natural ambient light**
- **Direct light, creates shadows and reflection**
- **Diffuse light, minimises shadows and reflection**
- **Back lighting, high contrast**
- **Dark field lighting**
- **Strobed light, freeze motion**
- **Structured light, measurement**
- **Polarised light, reduction of reflection**

- **<u>Warning: using laser light can be dangerous !!!</u>**

Signal to Noise Ratio (*)

The amount of noise in an image is measured in the Signal to Noise Ratio (SNR)

Is measured by computing its value in a homogeneously illuminated background section of the image

First the standard deviation is measured:

$$\sigma^2 = \frac{1}{N-1}\left[\sum_{i=1}^{N} b_i^{\,2} - \frac{1}{N}\left(\sum_{i=1}^{N} b_i\right)^2\right]$$

where b_i is the brightness of the image at position i

27-Aug-18 Image acquisition 47

Signal to Noise Ratio (*)

This can be rearranged to:

$$\sigma^2 = \frac{1}{N-1}\left[\sum_{i=1}^{N}\left(b_i - \frac{1}{N}\left(\sum_{i=1}^{N} b_i\right)\right)^2\right]$$

The SNR is calculated as:

$$SNR = \frac{\max(b) - \min(b)}{\sigma} : 1$$

where max(b) and min(b) are the maximum and minimum possible brightness value in the image b.

27-Aug-18 Image acquisition 48

Signal to Noise Ratio (*)

Example:

If max(b) = 255, min(b) = 0, and stddev = 5.0, then SNR = 51:1.

The SNR is sometimes expressed in decibels as

SNR(dB) = 20 $\log_{10}$(SNR).

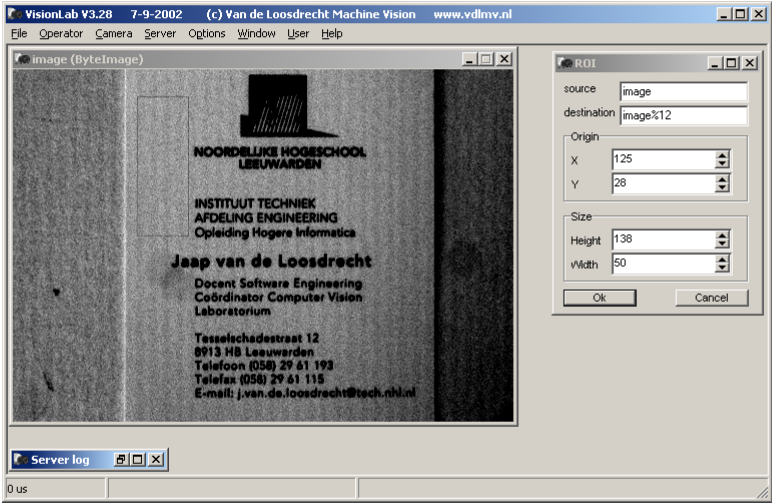
For the example, this would mean a SNR(dB) = 34.2 dB.

Demonstration Calculation of SNR (*)

- **open image card1.jl (low light, low contrast, extreme gain)**
- **roi 125 28 138 50 to get homogeneously illuminated background section**
- **Minmaxpixel is 8 21 (should normally be 0 .. 255 for 8 bit camera)**
- **standarddeviation on roi gives 1.75 , so SNR = 13 / 1.75 = 7.4 : 1**

- **Calculation stddev "by hand": (no slides)**
  - ***Convert roi to Int16Image (beware of overflow !!)***
  - **Calculate AveragePixel of roi (= 14)**
  - **Create new image with SetAllPixels 14 on roi (synthetic menu)**
  - **Subtract "image14" from roi**
  - **Multiply this result with itself**
  - **SumIntPixels on result of multiply (=23858)**
  - **stdev = sqrt(23858/(138*50-1)) = 1.86**
  - **(difference due to rounding error in AveragePixel)**

Calculate Standard Deviation (*)



Reduce noise by averaging images

Noise can be reduced by averaging images

avgImage = for each pixel: (Sum pixel of all images) / nr of images

The SNR improves theoretically linear to the square root of the number of images

Exercise noise reduction by averaging images (*)

Exercise:

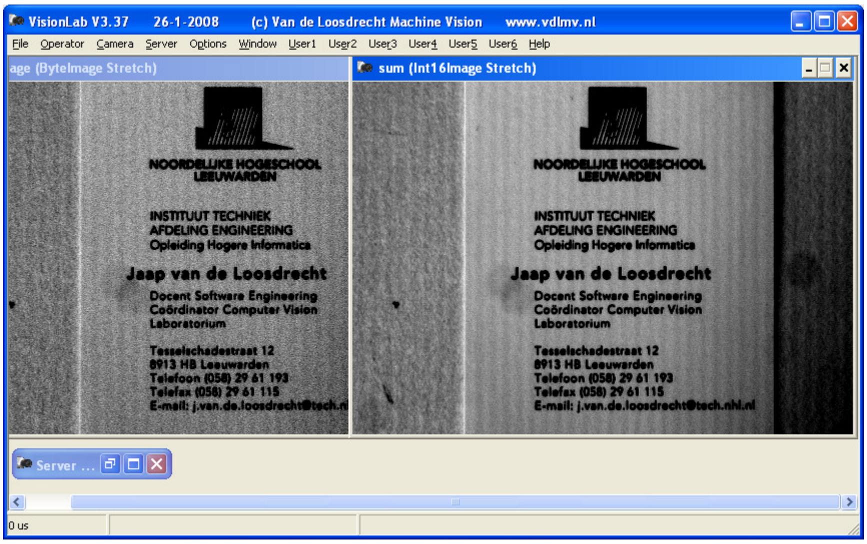
- **average ten images card1.jl, …, card10.jl and calculate SNR of result, use min/max pixel of ROI original image**
- **Explain why improved is not as good as could be expected in theory**

- **see for answer script card_noise.jls, examine variables**
- **answer exercise: stdev = 0.856, SNR = 15.2, so improvement is by factor 2**

Demonstration noise reduction by averaging images

- **use script card_noisedemo.jls**
- **average ten under exposed images card1.jl, …, card10.jl**

Demonstration noise reduction by averaging images



27-Aug-18 Image acquisition 55