Computer vision

2D Camera Calibration

10 April 2018

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

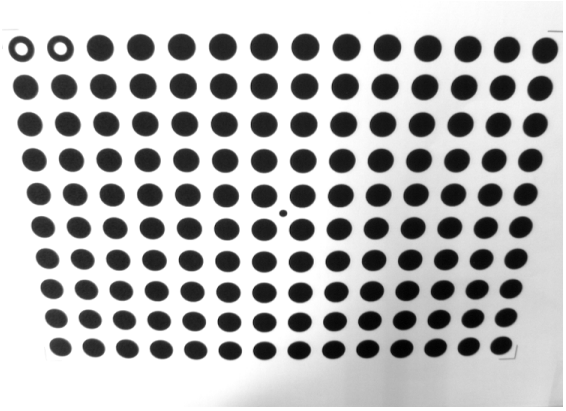
2D Camera Calibration

Overview:

- **Introduction**
- **Coordinate systems**
- **Camera parameters**
- **Calibration pattern**
- **Find calibration points**
- **Camera Calibration**
- **Pixel to world coordinate**
- **Correct for optical distortion**

Introduction

Problems in measuring the real size of 2D objects



Introduction

Some problems in measuring the real size of 2D objects:

- **Camera is tilted**
- **Dimensions of pixels horizontal and vertical are different**
- **Distance of object from camera**
- **Focal length of lens**
- **Lens distortion**
- **Mismatch frequency of frame grabber and analogue camera (*)**
- **Non-alignment of optical axis of camera ccd and lens**

Coordinate systems

Pixel coordinates:

- **Measured from the image**
- **Unit in pixels**
- **Origin: top left corner**
- **X values increase from left to right**
- **Y values increase from top to bottom**

World coordinates:

- **User defined**

In order to measure the real size of objects there must be a mapping from each pixel coordinate to a world coordinate.

Camera parameters

Camera parameters:

- **Interior parameters:**
  determined by the internal geometry of the camera and the properties of the frame grabber
- **Exterior parameters:**
  determined by the position and orientation of the camera relative to the world coordinate system

Interior parameters

- **Geometry of ccd chip:**
  - **dx: center-to-center distance of pixels in x direction**
  - **dy: center-to-center distance of pixels in y direction**
- **Principal point (alignment of optical axis of ccd and lens) :**
  - **xp: x-coordinate for principal point, relative to center of image**
  - **yp: y-coordinate for principal point, relative to center of image**
- **Camera constant:**
  - **f: <= focal length, at infinity = focal length**
- **Lens distortion coefficients:**
  - **kappa1: first order lens distortion coefficient**
  - **kappa2: second order lens distortion coefficient**
  - **kappa3: third order lens distortion coefficient**
- **Frame grabber property:**
  - **sx: scale factor for timing digitizer**

27-aug-18 2D Camera Calibration 7

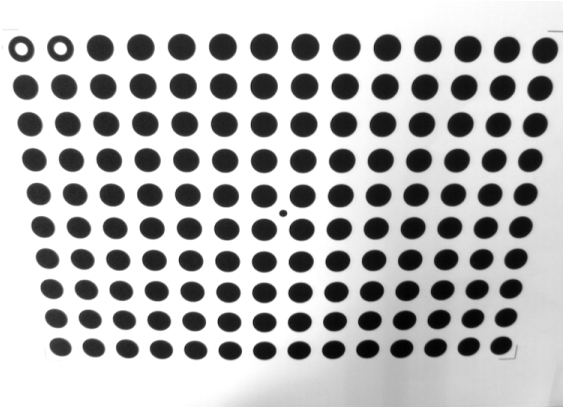
Exterior parameters

- **Rigid body transform:**
  - **Rx: rotation around x-axis**
  - **Ry: rotation around y-axis**
  - **Rz: rotation around z-axis**
  - **Tx: translation in x direction**
  - **Ty: translation in y direction**
  - **Tz: translation in z direction**

27-aug-18 2D Camera Calibration 8

Calibration Pattern

A fixed pattern of circles is used to calibrate the camera



Find calibration points

camera findcalpoints imageName nrRows nrCols minPixels maxPixels

This operator examines a binary image with imageName in order to find the calibration points.

A fixed pattern of circles is expected with nrCols circles vertical and nrRows circles horizontal.

Each circle has minimal minPixels and maximal maxPixels.

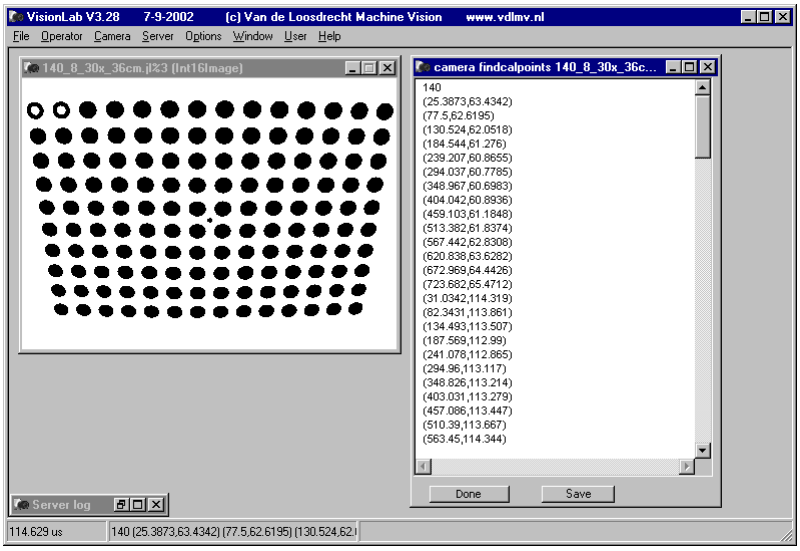
The positions of the calibration points are displayed on the screen.

Demonstration Find calibration points

Demonstration:

- **Open image 140_8_30x_36cm.jl**
- **ThresholdIsoData DarkObject**
- **camera findcalpoints 10 14 300 2000**
- **Examine results**

Demonstration Find calibration points



Camera Calibration

The 140 calibration points are used to find optimal values for the 15 camera parameters.

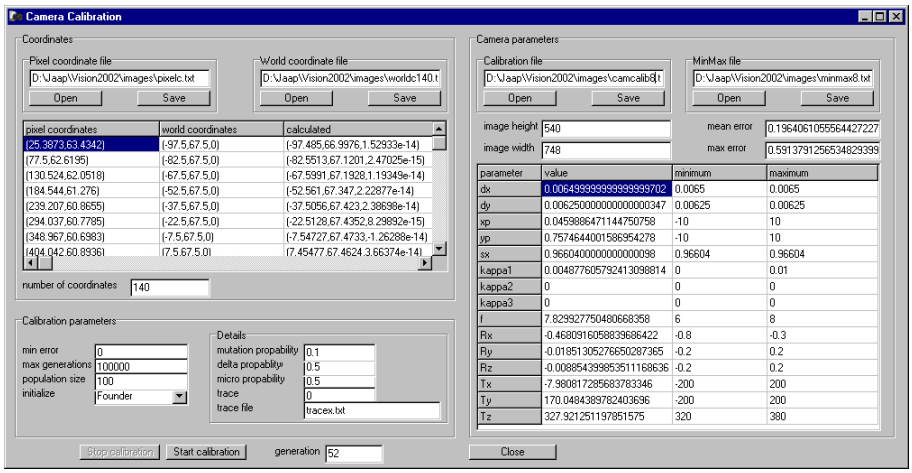
**So there are 140 equations with 15 unknown parameters.
The parameter space has 15 dimensions and consists of polynomial and goniometric equations.**

This non-linear optimalisation problem is solved in VisionLab with a combination of a Genetic Algorithm and Hill Climbing.

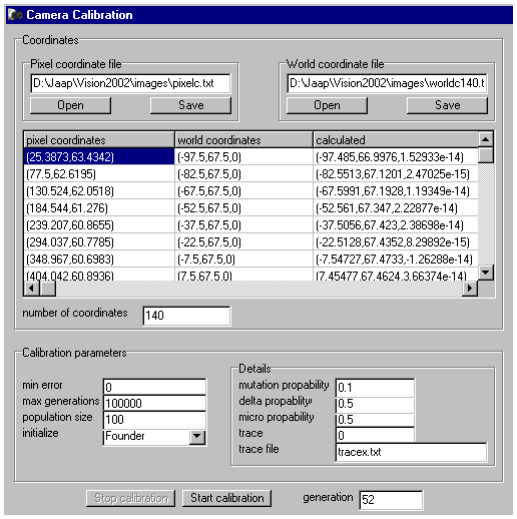
Demonstration Camera Calibration

- **Open dialog camera calibration**
  - **Pixel coordinate file: pixelc.txt**
  - **World coordinate file: worldc140.txt**
  - **Calibration file: camcalib8.txt**
  - **MinMax file: minmax8.txt**
- **Start calibration with:**
  - **Population size = 100**
  - **Mutation prop = 0.1**
  - **Delta prop = 0.5**
  - **Micro prop = 0.5**

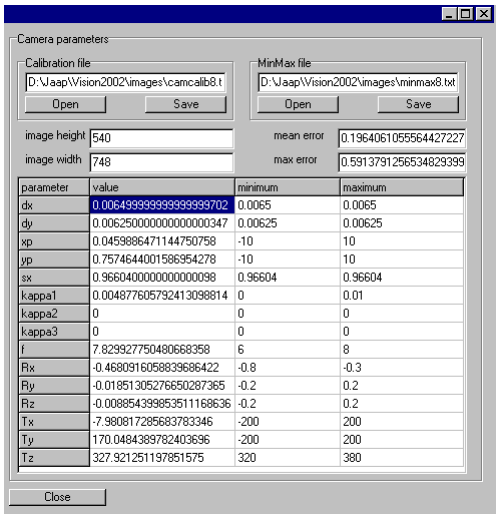
Demonstration Camera Calibration



Left part Camera Calibration screen



Right part Camera Calibration screen



27-aug-18 2D Camera Calibration 17

Pixel to world coordinate

camera pixelctoworldc pixelCoord camcalibfile

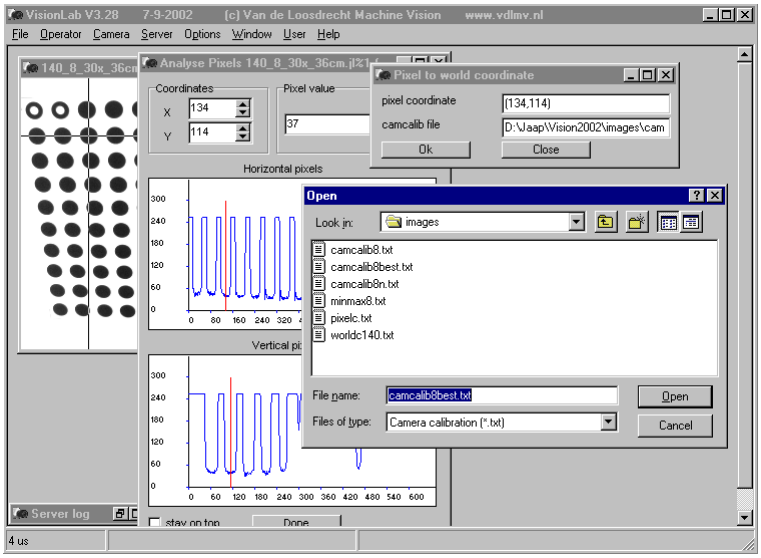
This operator translates a pixelCoordinate to a worldCoordinate using a calibration file camcalibfile.

27-aug-18 2D Camera Calibration 18

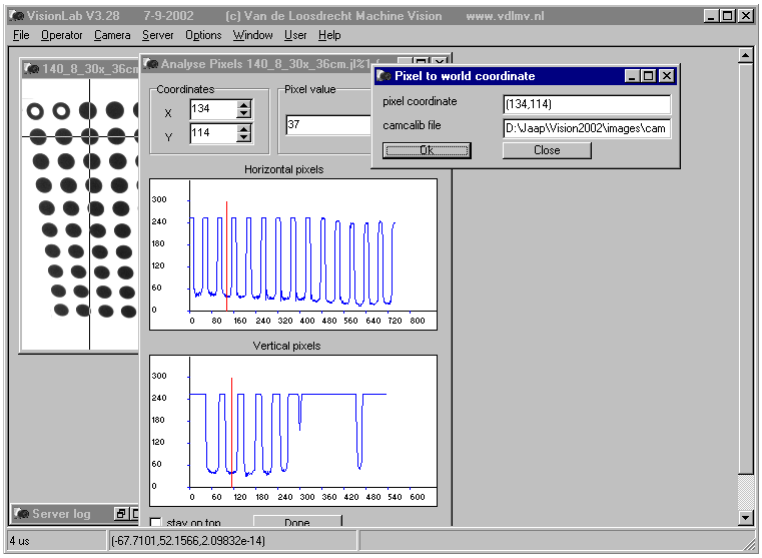
Demonstration Pixel to world coordinate

- **Open image 140_8_30x_36cm.jl**
- **Find coordinate of pixel to convert to world coordinate**
- **Use in camera menu function Pixel to world coordinate**
  - **Fill in chosen pixel coordinate**
  - **Click on camcalib file field and select camera calibration file camcalib8best.txt**
  - **Execute function and see result in result bar**

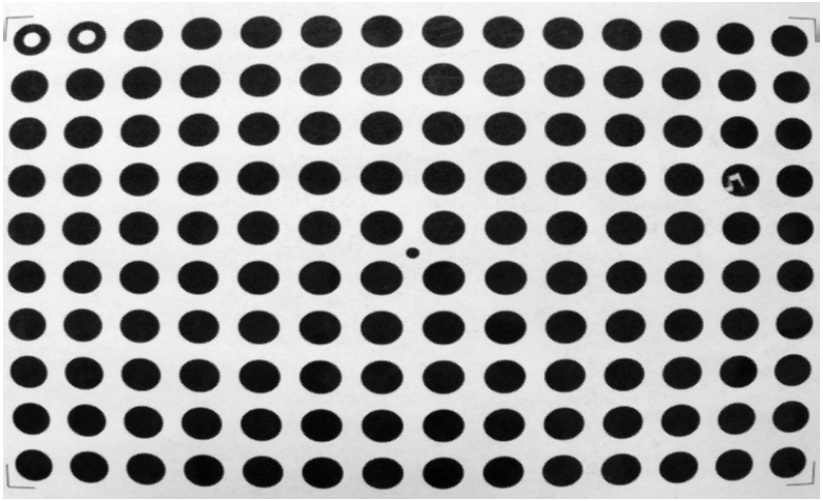
Demonstration Pixel to world coordinate



Demonstration Pixel to world coordinate



Correct for optical distortion



Correct for optical distortion (*)

CorrectXYCoord (xyCoord, heightImage, widthImage, principal_point, sx, dx, dy, kappa1, kappa2, kappa3)
Camera CorrectCoord2D (Coord2D, heightImage, widthImage, principal_point, sx, dx, dy, kappa1, kappa2, kappa3)

This operators corrects the "barrel shape" lens distortion for one pixel coordinate

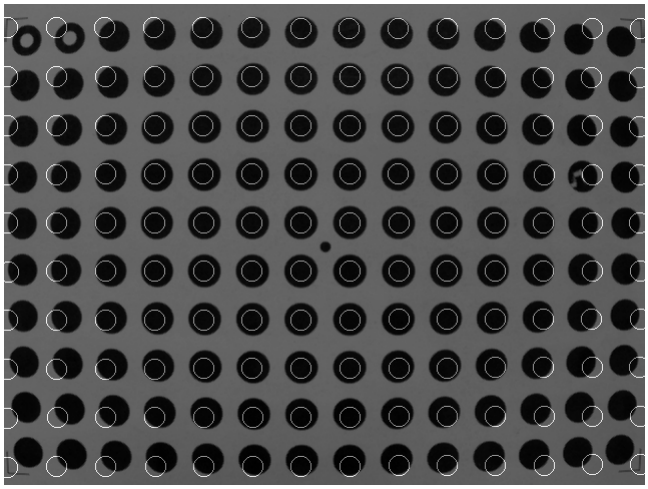
Camera CamDistortion (srcImage, destImage, principal_point, sx, dx, dy, kappa1, kappa2, kappa3)

This operator corrects the "barrel shape" lens distortion in the whole image

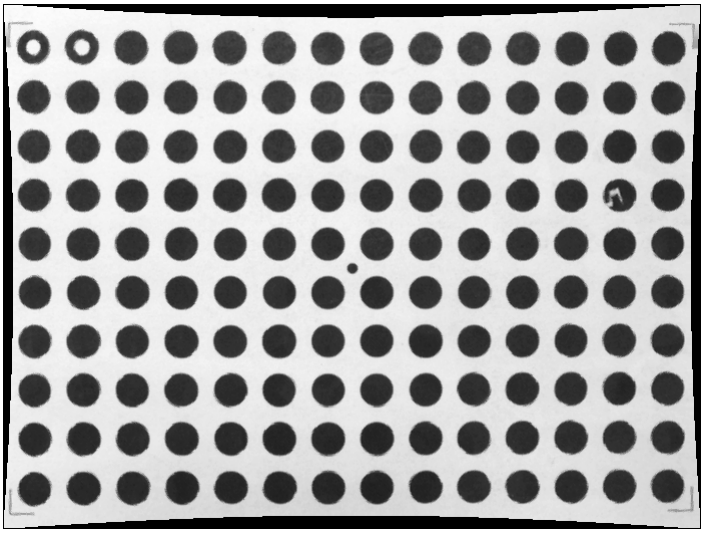
Demonstration Correct for optical distortion

- **Close open images**
- **(note image taken with different camera as previous example, we are now interested only in the optical distortion)**
- **Show position of centre circles:**
  - **run script correct_distortion.jls**
- **Open file distortion.jl**
- **(Show corrected image:**
  - **CamDistortion (0.002,0.09) 1 0.0099 0.0099 0.00532 0 0)**

Positions of centre circles

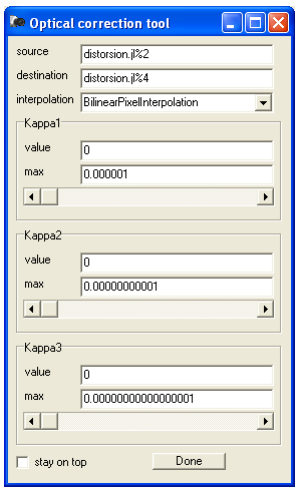


Corrected image



Optical correction tool

Used for interactively correcting optical distortions.

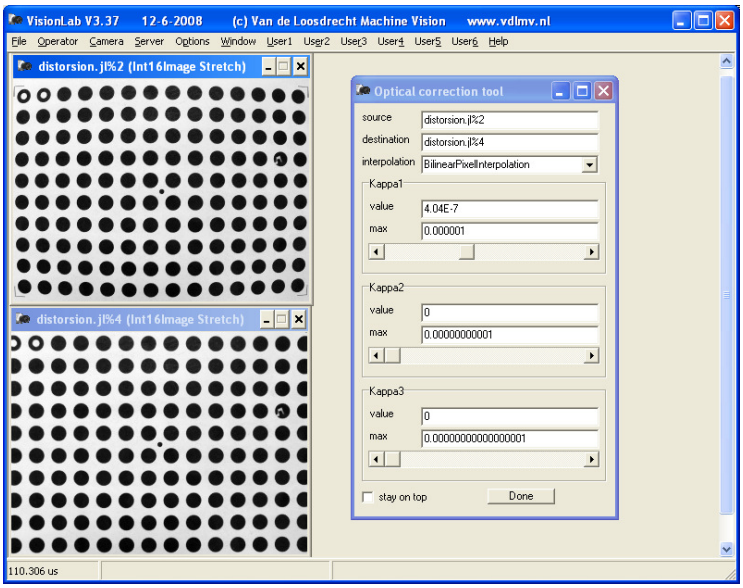


Note: Note the kappa's for this operator are different from the kappa's used by CamDistortion.

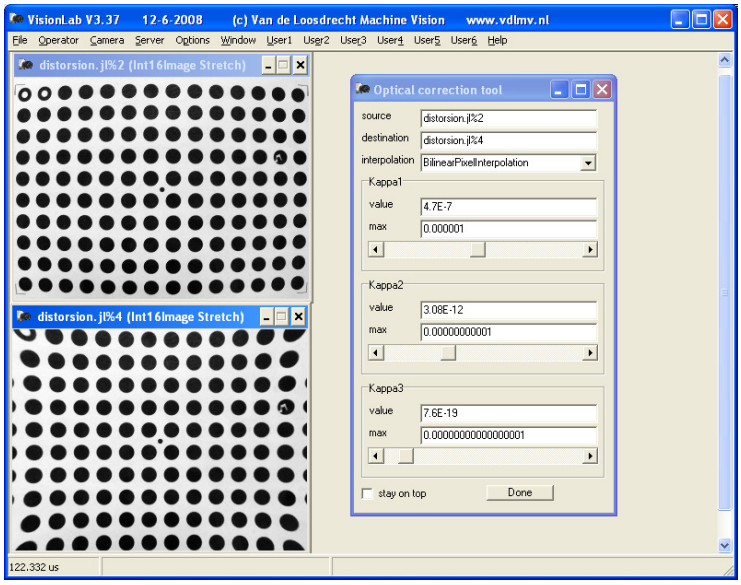
This operator performs "the inverse operation" and is much faster then CamDistortion.

See Geometry menu.

Optical correction tool

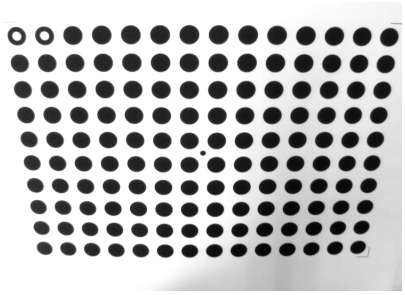


Optical correction tool



27-aug-18 2D Camera Calibration 29

Exercise correct distortion

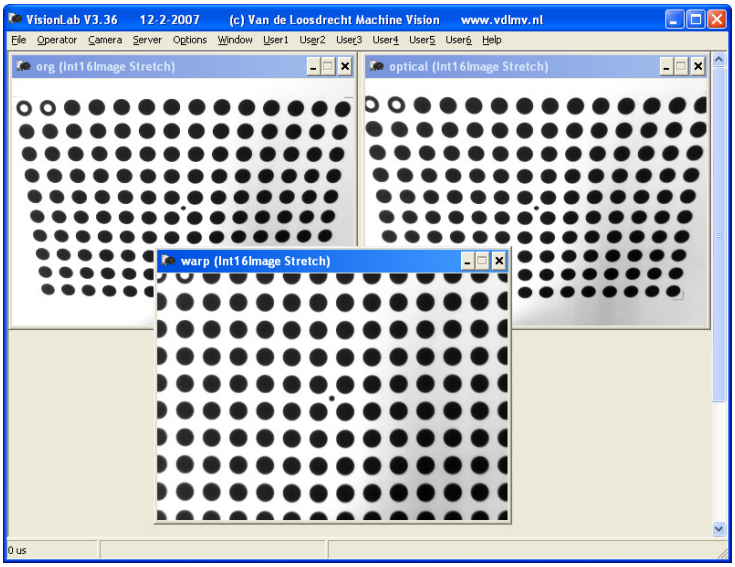


- **Use image 140_8_30x_36cm.jl**
- **Correct both optical distortion and distortion by tilting**

Answer: correct_opt_warp.jls

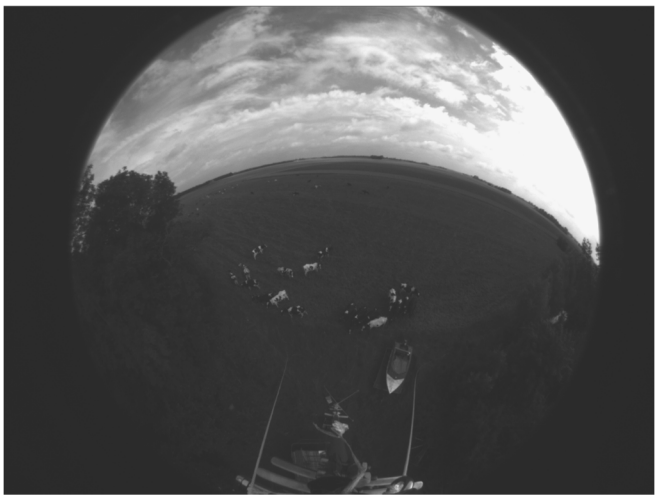
27-aug-18 2D Camera Calibration 30

Orginal -> Optical correction -> Warp



27-aug-18 2D Camera Calibration 31

Fish Eye lens



27-aug-18 2D Camera Calibration 32

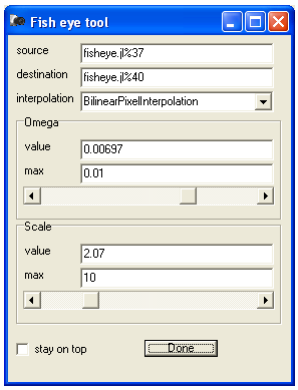
Fish Eye Tool

Used for interactively correcting Fish Eye distortions.

If omega is positive an image taken with a fish eye objective will be stretched flat. With a negative omega a normal image can be transform to a fish eye image.

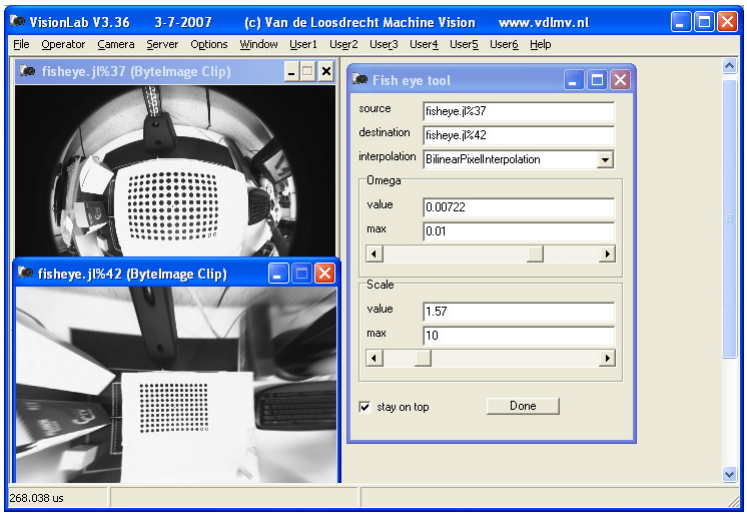
A good starting point for experimenting with a value for omega = PI / (diameter fish eye circle in pixels) .

The scale factor specifies a the zoom factor.



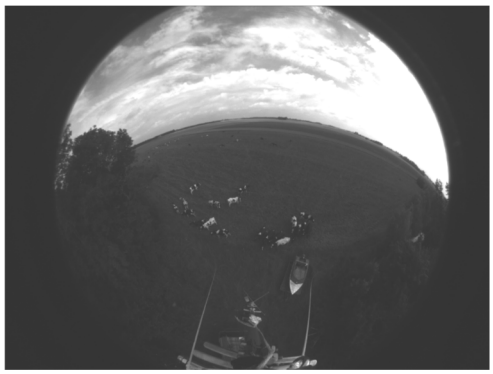
27-aug-18 2D Camera Calibration 33

Fish Eye Tool (on image fisheye.jl)



27-aug-18 2D Camera Calibration 34

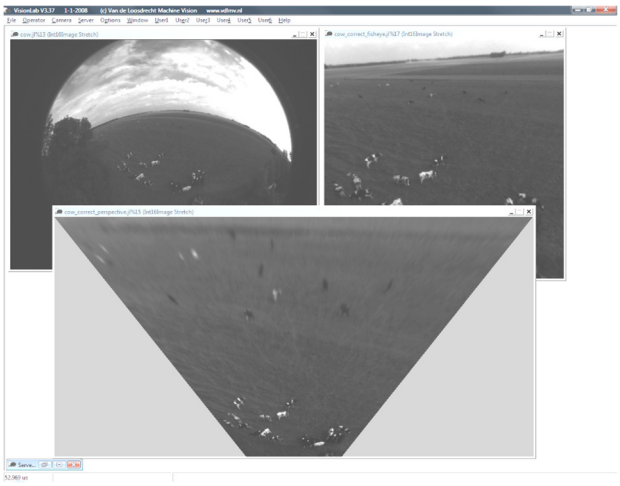
Exercise correct distortion



- **Use image cow.jl**
- **Correct both optical distortion and distortion by tilting**

Answer: cow_correct.jls

Orginal -> Optical correction -> Warp



- **All objects are of the same size, regardless of the distance to the camera**
- **Advantage: Binary or Greyscale filters can have fixed sizes**

Fish Eye correction operator (*)

FishEye srcImage destImage height width omega scale border interpolation

Menu: Operator | Geometry

This operator performs a FishEye transform. If omega is positive an image taken with a fish eye objective will be stretched flat. With a negative omega a normal image can be transform to a fish eye image.

A good starting point for experimenting with a value for omege = PI / (diameter fish eye circle in pixels) .

The scale factor specifies a the zoom factor.
The parameters height and width determine the size of the destination image.
The specified border value will be used as result pixel if information outside the source image is necessary for the calculation.