

Camera Calibration

Using Matlab & OpenCV to Remove
Camera System Distortion



Links

- The Matlab Camera Calibration Toolbox (main site)
 - http://www.vision.caltech.edu/bouquetj/calib_doc/
- Toolbox Download
 - http://www.vision.caltech.edu/bouquetj/calib_doc/download/index.html
- Most Useful Tutorial
 - http://www.vision.caltech.edu/bouquetj/calib_doc/htmls/example.html
- Other Useful Toolbox Links
 - List of all Function in Toolbox
http://www.vision.caltech.edu/bouquetj/calib_doc/htmls/functions.html
 - Calibration Parameters
http://www.vision.caltech.edu/bouquetj/calib_doc/htmls/functions.html
- Open Source Computer Vision Toolbox
 - Open Source Computer Vision Library (OpenCV): C/C++; Win32, Linux
<http://sourceforge.net/projects/opencvlibrary/>
 - Using OpenCV with Visual Studio
<http://opencvlibrary.sourceforge.net/VisualC++>

Calibration Step 1

- Install the toolbox and add the path
- Start Matlab
- Type `calib_gui`
- Use unix like commands (`ls` and `cd`) to navigate to the dir with the calibration images
- Click 'Read images'

- NOTE: the images will be converted to a single channel using a formula found in:
 - `ima_read_calib.m`
 - line 74

- The formula they use to convert to one channel:

```
Ii =
0.299*Ii(:, :, 1) +
0.5870*Ii(:, :, 2) +
0.114*Ii(:, :, 3);
```

The screenshot displays the MATLAB environment. At the top, a window titled 'Camera Calibration Toolbox - Standard Version' is open, showing a menu with the following options:

Image names	Read images	Extract grid corners	Calibration
Show Extrinsic	Reproject on images	Analyse error	Recomp. corners
Add/Suppress images	Save	Load	Exit
Comp. Extrinsic	Undistort image	Export calib data	Show calib results

Below this, the MATLAB interface is visible. The Command Window shows the following text:

```
Using Toolbox Path Cache. Type "help toolbox_path_cache" for
Warning: Name is nonexistent or not a directory: d:\can_not_be_

To get started, select "MATLAB Help" from the Help menu.

>> calib_gui
>>
```

The Command History window shows the command `calib_gui` has been executed.

Select Calibration Images

- All calibration images need a common name prefix and a number scheme
 - Telephoto_Scene
 - 00 to 99
- Renaming all images is tedious, use of unix commands and scripting helpful, something
 (mv *.TIF prefix"\$i\$j".TIF)

The screenshot shows the MATLAB Camera Calibration Toolbox interface and a File Explorer window. The toolbox has a menu with options like 'Image names', 'Read images', 'Extract grid corners', etc. The File Explorer shows a directory 'variation 2--tmp' containing 16 TIF files named 'Telephoto_Scene01.TIF' through 'Telephoto_Scene16.TIF', each 6,080 KB.

The MATLAB Command Window shows the following commands and output:

```

cd 2007-WIP-Vision_Lab
ls
cd 2007_vislab-in_progres
ls
cd "09-01-07_Matlab Calib
cd '09-01-07_Matlab Calib
ls
clear
cd C:\Temp\variation 2--t
cd 'C:\Temp\variation 2--
ls
Telephoto_Scene

```

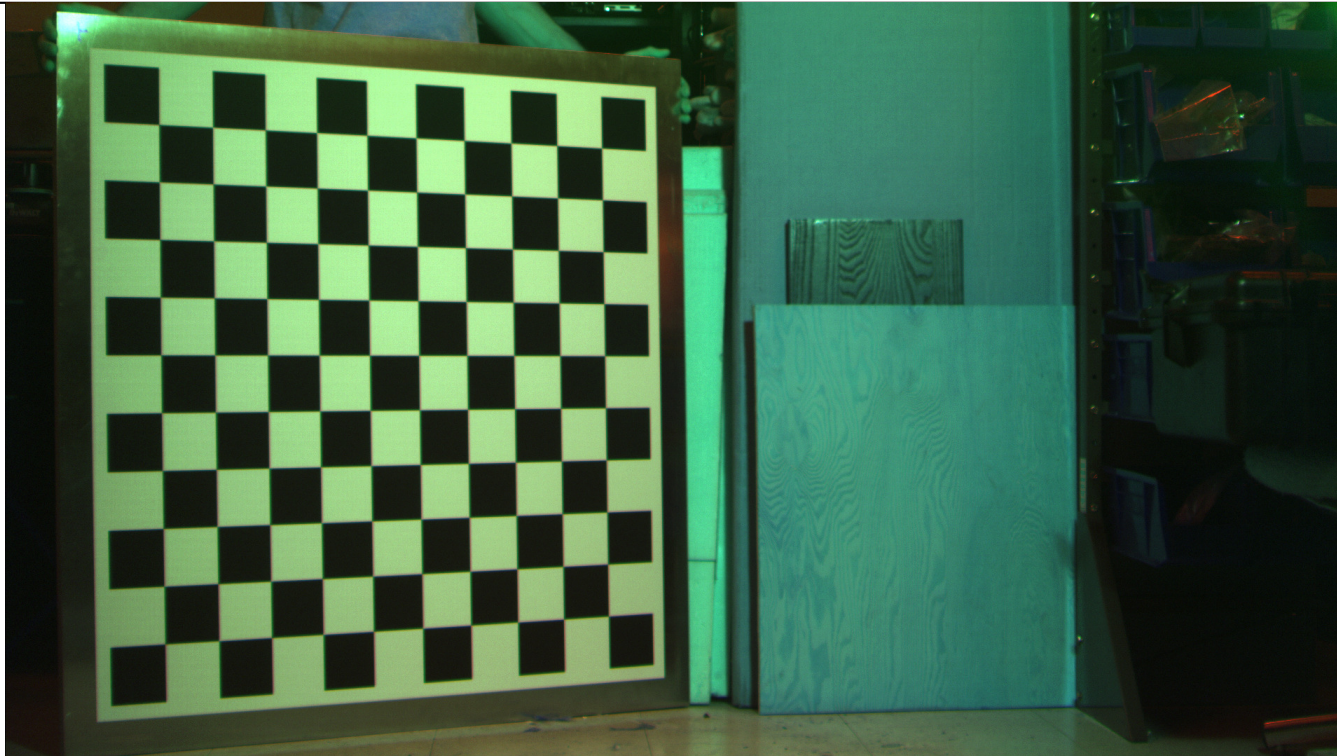
The Command Window output shows:

```

Basename camera calibration images (without number nor suffix): Telephoto_Scene
Image format: (['r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm'])

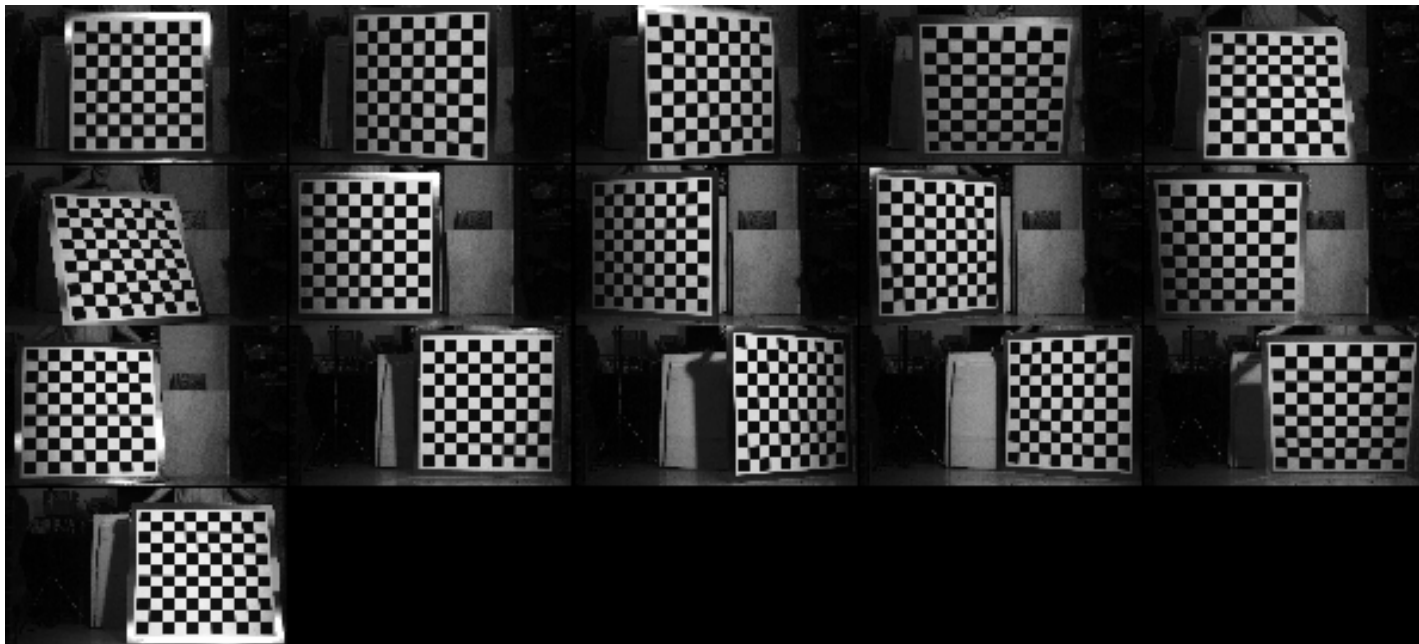
```

Example Calibration Image



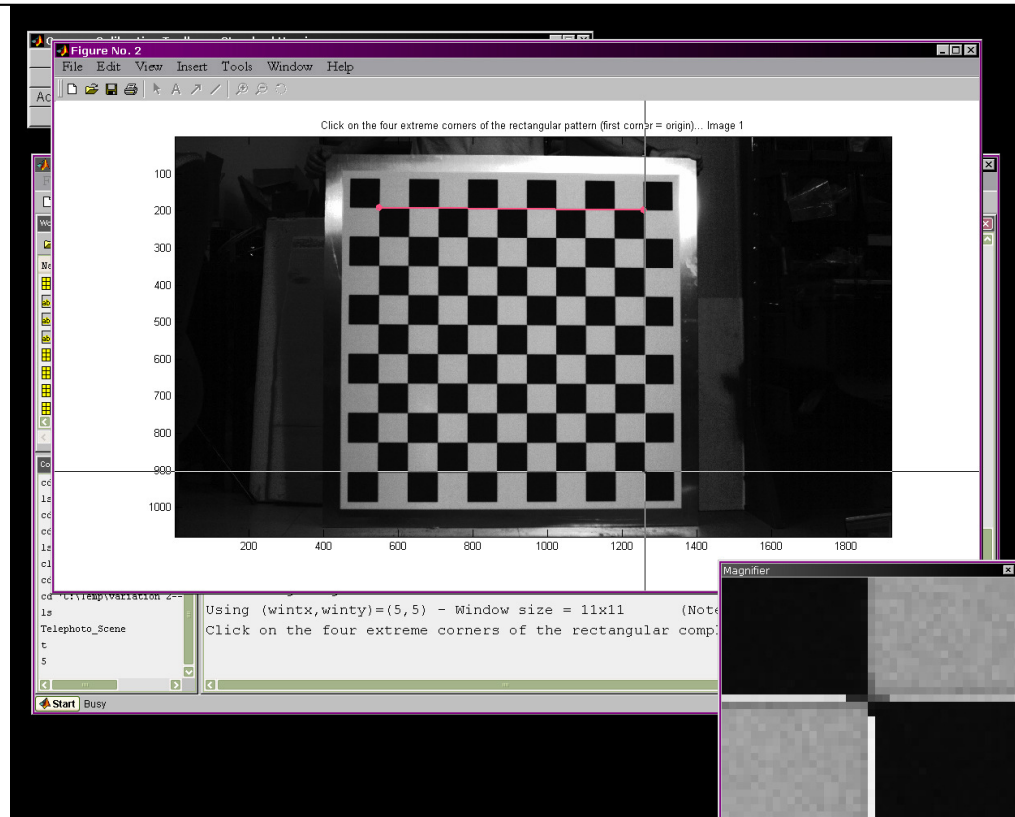
▪ Image taken in CIR, shown in RGB

Preview Read Images



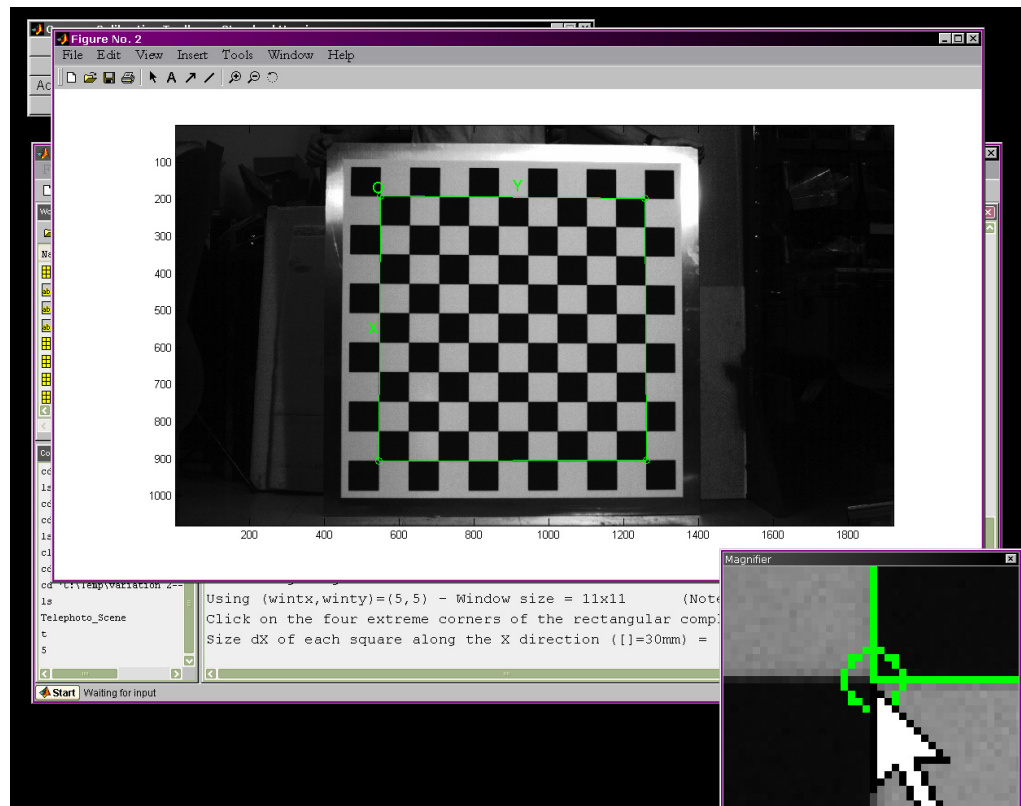
Extracting Grid Corners

- When starting this part of the process, the program will ask for
 - Window size
 - This is for corner detection
 - A smaller window is more accurate, but maybe require "better eyes"
- The zoomed in view is through windows magnifier.
- Most Linux GUI based distributions have a similar utility
- Notice the red line
 - Line is drawn connecting the last two points you selected



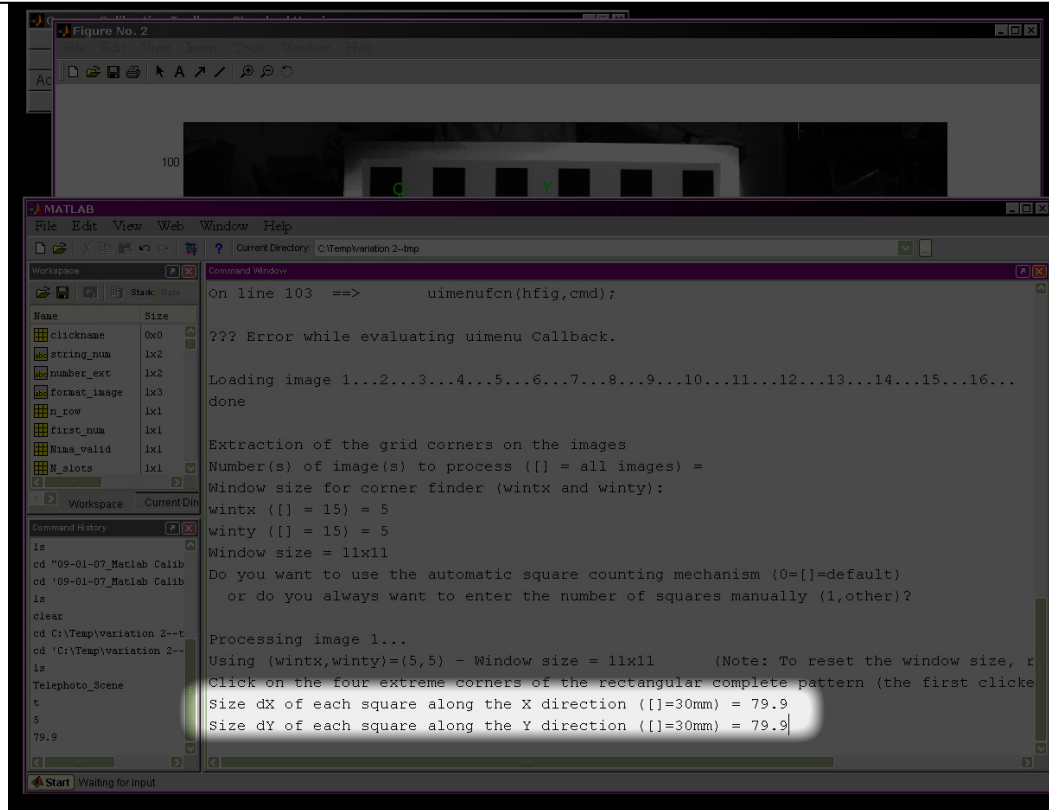
Picking Last Corner

- After picking last corner, green square is drawn
- Extracting grid corners from the first image is a special case
 - After picking the last corner, the program will ask you to supply measurements of the grid squares.



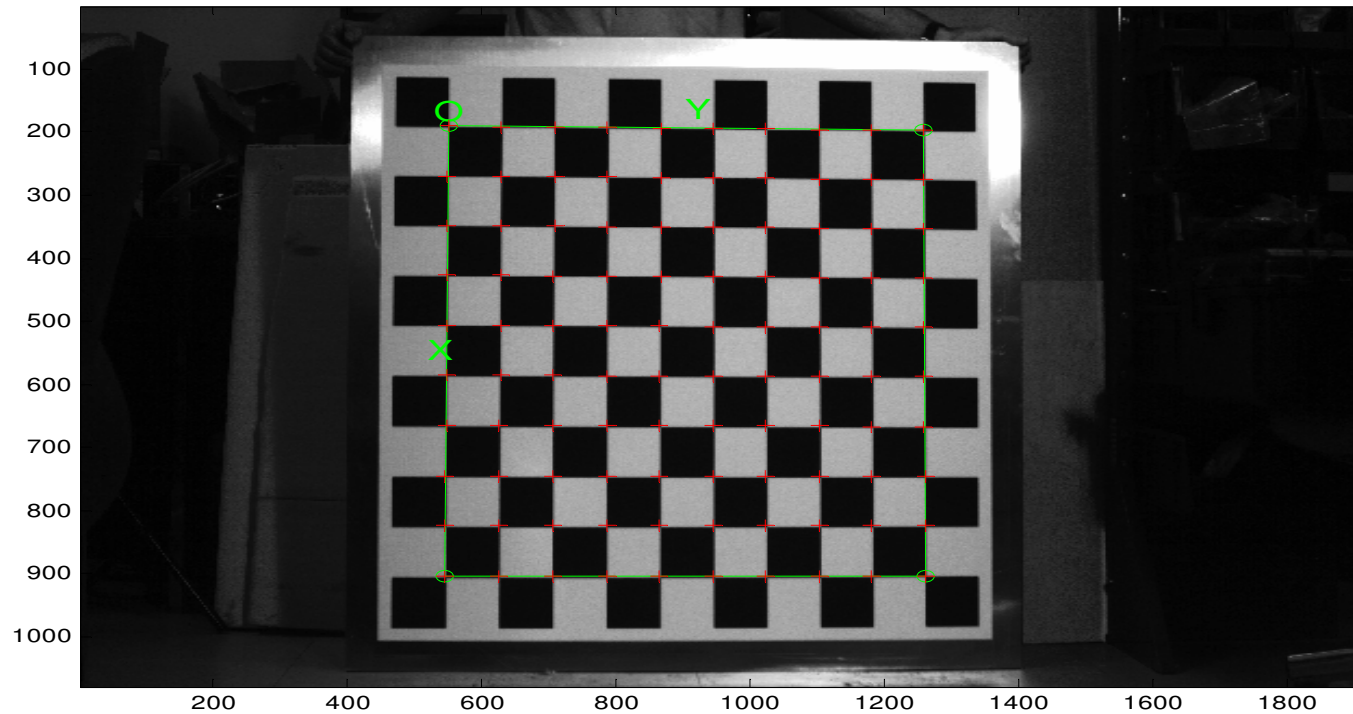
First Corner Extraction

- Oddly, the program does not ask for measurements until after the first corner extraction
- Our most accurate measurement found the grid squares to be 79.9mm^2

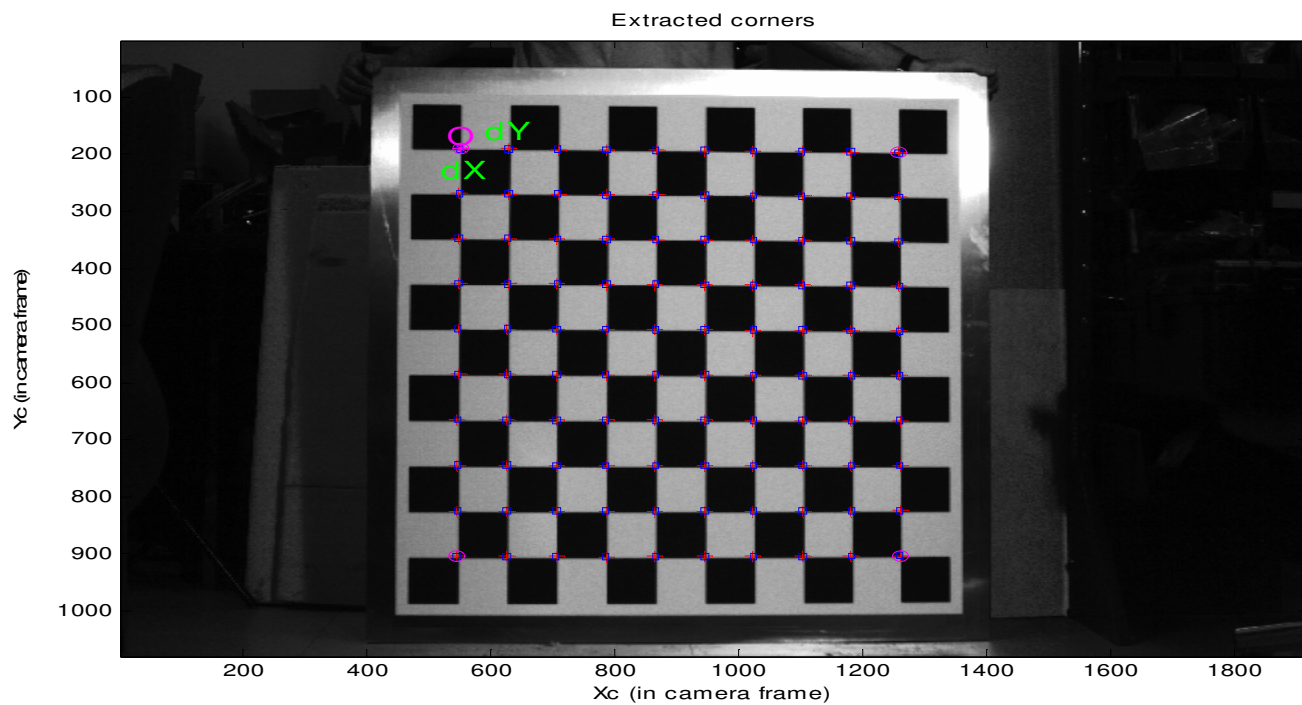


Verify Corner Extraction

The red crosses should be close to the image corners



Grid Points Extracted



Run Calibration

- After extracting the corners from every image, click 'Calibration'
- The system will carry out a number of iterations of the gradient descent method and will converge if the corners are found accurately

Calibration results after optimization (with uncertainties):

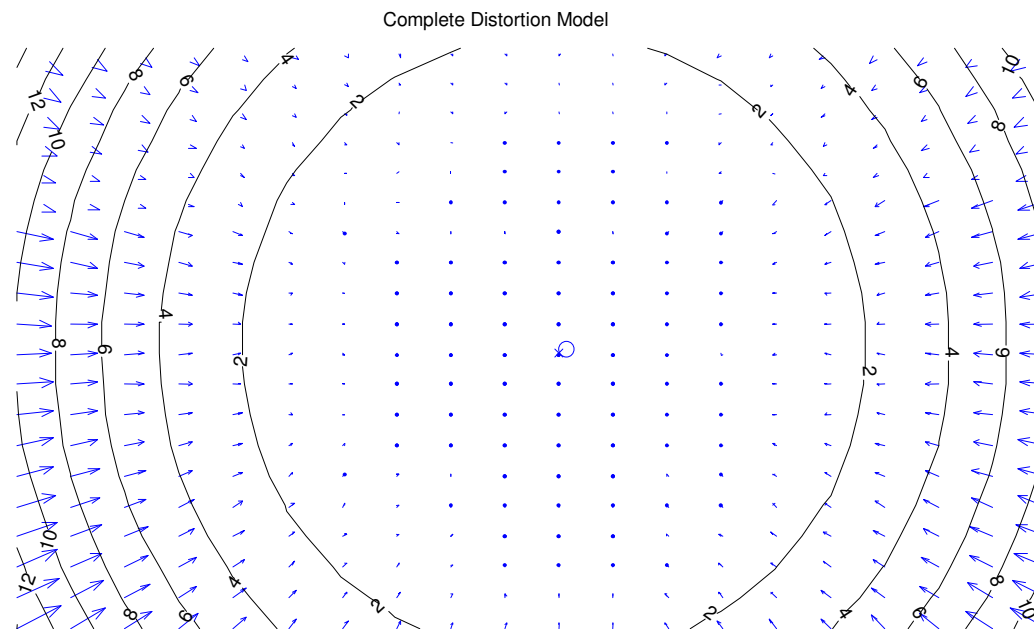
```
Focal Length:      fc = [ 3513.44159  3510.48363 ] ± [ 4.16406  4.17913 ]
Principal point:   cc = [ 974.00660  532.29039 ] ± [ 4.53338  4.50995 ]
Skew:              alpha_c = [ 0.00000 ] ± [ 0.00000 ]
                   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:        kc = [ -0.14662  0.06462  -0.00020  -0.00088  0.00000 ]
                   ± [ 0.00606  0.08436  0.00028  0.00028  0.00000 ]
Pixel error:       err = [ 0.10706  0.11851 ]
```

Note: The numerical errors are approximately three times the standard deviations (for reference).

- Values in units of pixels

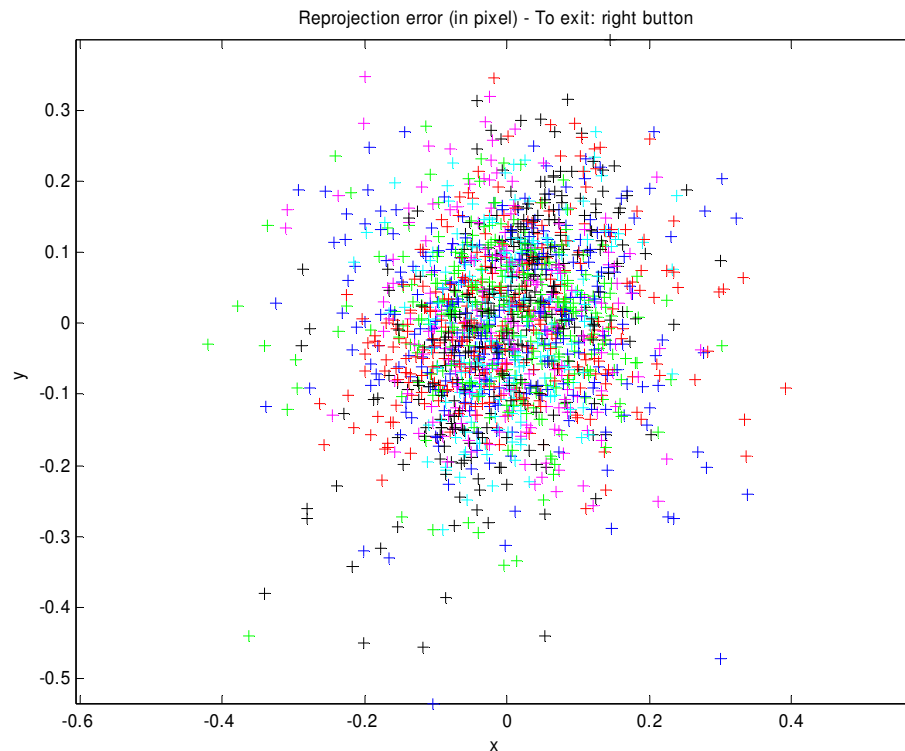
See Distortion model

- This command is not in the GUI menu
- `visualize_distortions`



Find Errors

- From the GUI menu select 'Analyse Errors'
 - They spelled it wrong?
- Click on outliers to see which image and corner they are coming from
- To fix
 - remove that image from the calibration
 - pick the corners for that image manually by pixel value
 - consider different calibration images
- Also, for more accurate results in general
 - Optimal calibration images
 - Number of images
 - Where the grid resides in the image (WRT the distortion)



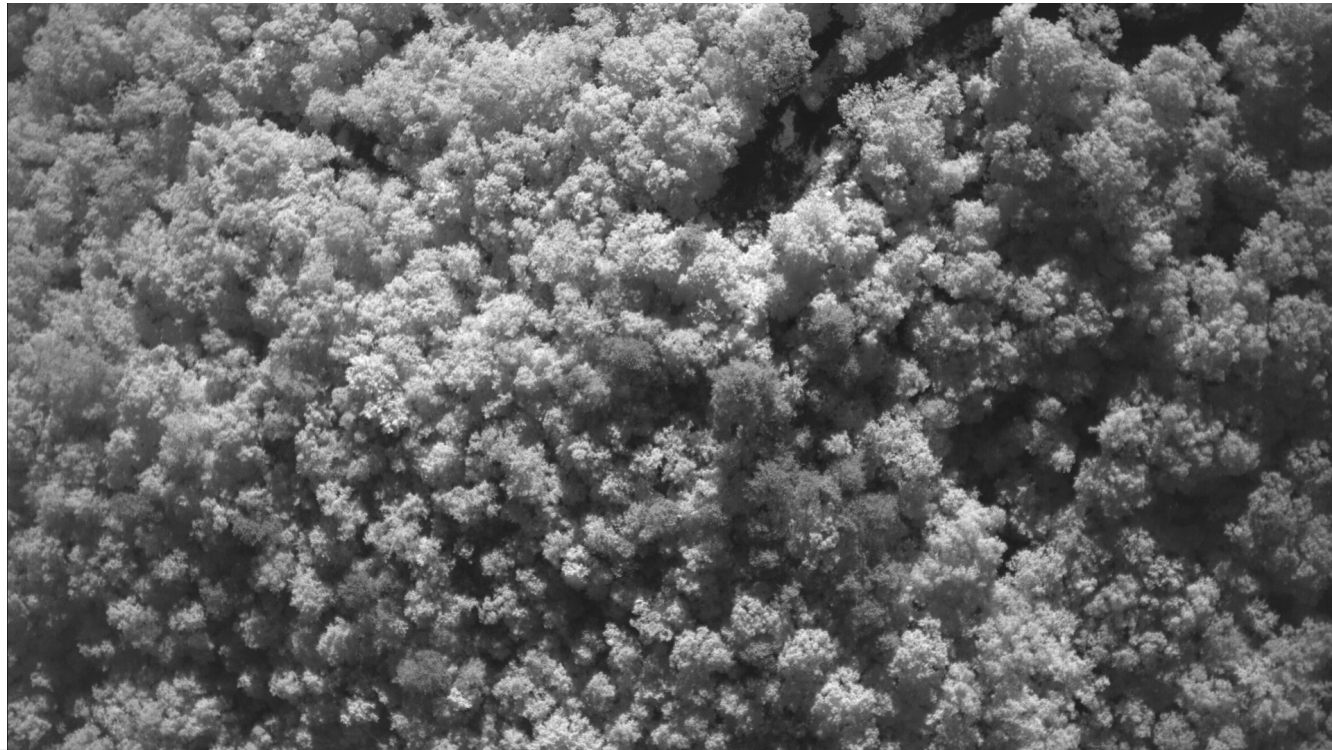
Undistort

- Once the calibration is found to be satisfactory
 - Click 'Undistort Image'
 - Can undistort all calibration images or any image of choice

- OpenCV has the same undistort function
 - Uses the same parameters and units
 - Allows different interpolation models to be used

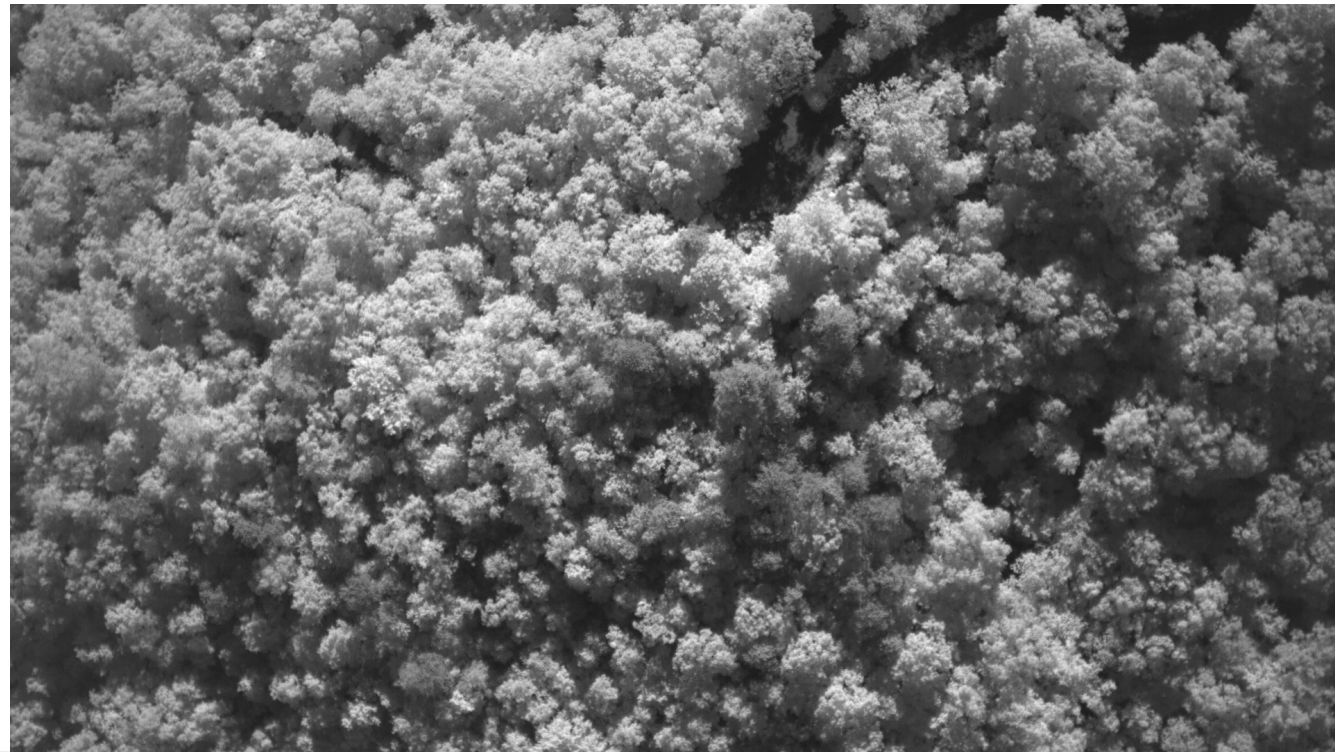
Undistort Example - Before

- This is only the red channel



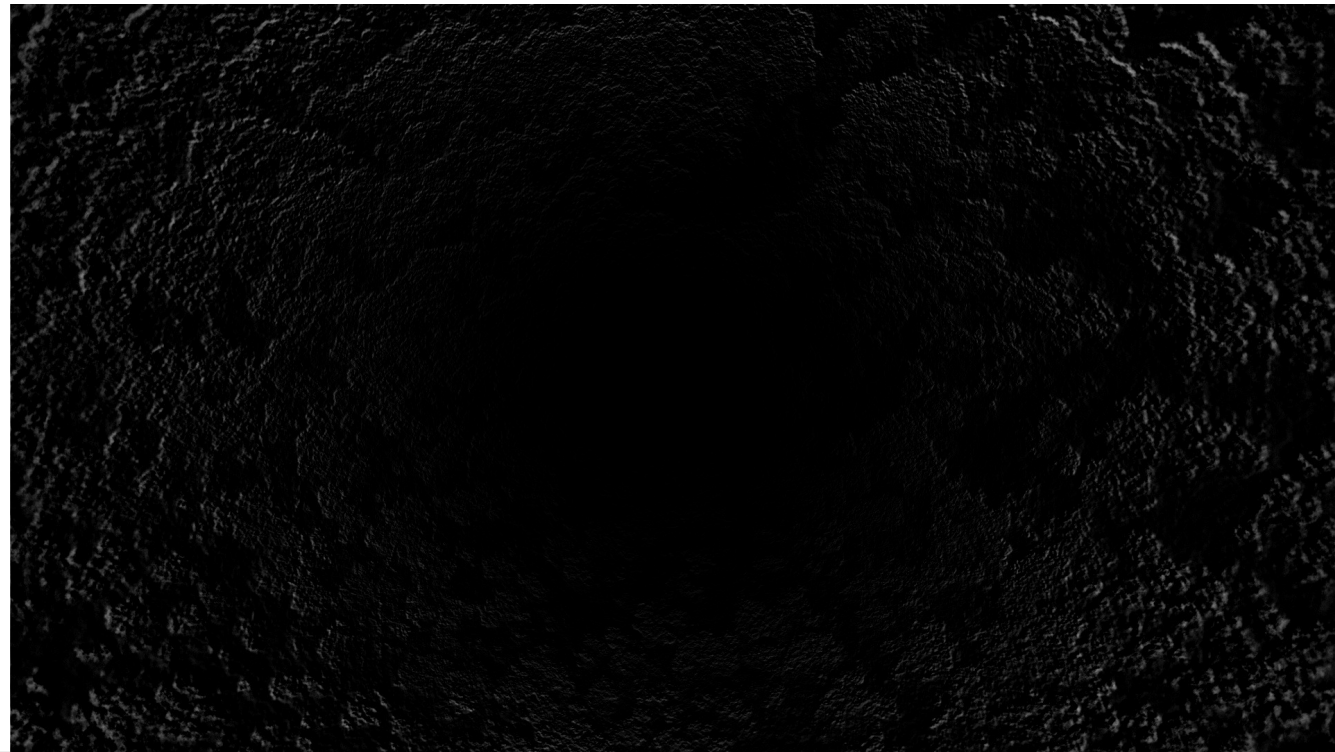
Undistort Example - After

- This is only the red channel



Undistort – See What it Did

- Subtract one image from the other (matrix subtraction)



Comparing

- Undistorted minus distorted with distortion model overlaid

