



2015.06.19

Basic of Camera Calibration : Tool Manual

강한솔



INDEX



01 MATLAB Camera Calibration Toolbox

02 Camera Calibration Using the Open CV

03 GML C++ Camera Calibration Toolbox

04 Comparison of Results

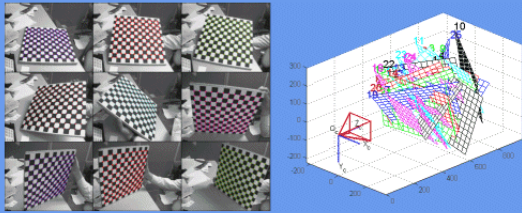
01 MATLAB Camera Calibration Toolbox

Set up

Browser

← → ↻  http://www.vision.caltech.edu/bouguetj/calib_doc/

Camera Calibration Toolbox for Matlab



This is a release of a Camera Calibration Toolbox for Matlab[®] with a complete documentation. This document may also be used as a tutorial on camera calibration since it includes general information about calibration, references and related links. Please report bugs/questions/suggestions to Jean-Yves Bouguet at "jbouguet at gmail dot com".

The C implementation of this toolbox is included in [OpenCV](#), an open source computer vision library freely available [online](#).

Content:

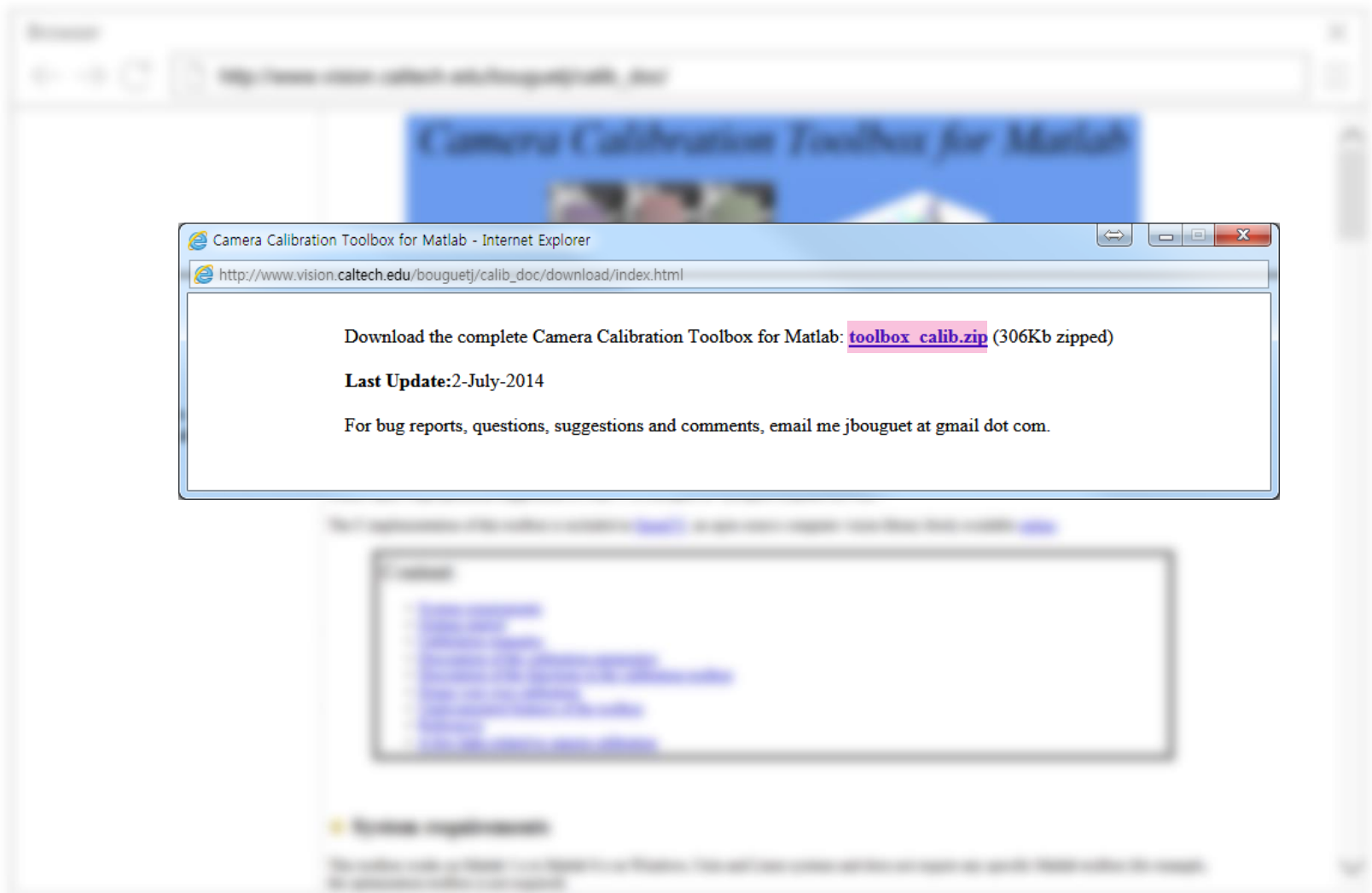
- [System requirements](#)
- [Getting started](#)
- [Calibration examples](#)
- [Description of the calibration parameters](#)
- [Description of the functions in the calibration toolbox](#)
- [Doing your own calibration](#)
- [Undocumented features of the toolbox](#)
- [References](#)
- [A few links related to camera calibration](#)

★ **System requirements**

This toolbox works on Matlab 5.x to Matlab 8.x on Windows, Unix and Linux systems and does not require any specific Matlab toolbox (for example, the optimization toolbox is not required).

01 MATLAB Camera Calibration Toolbox

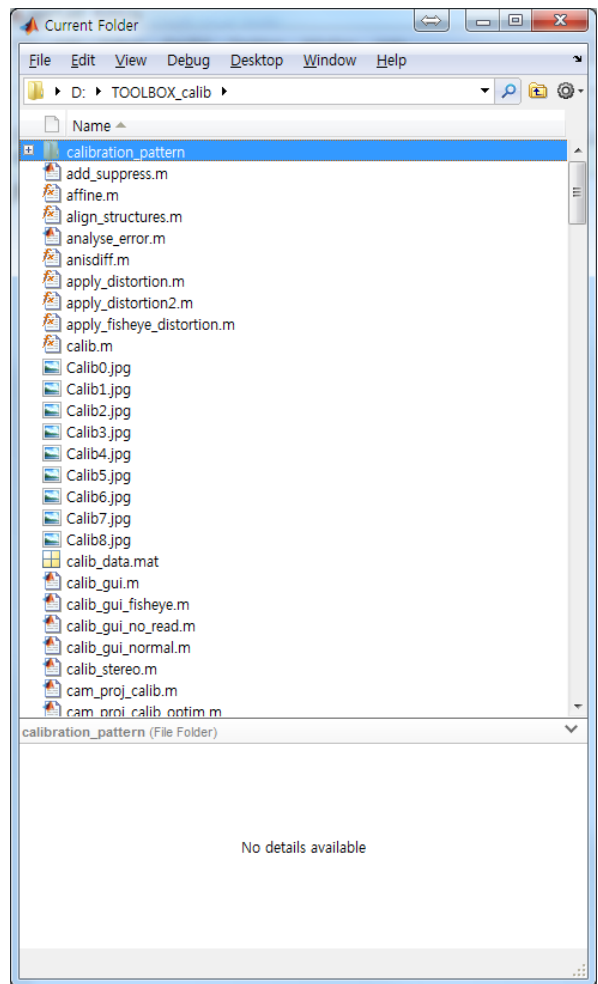
- Set up



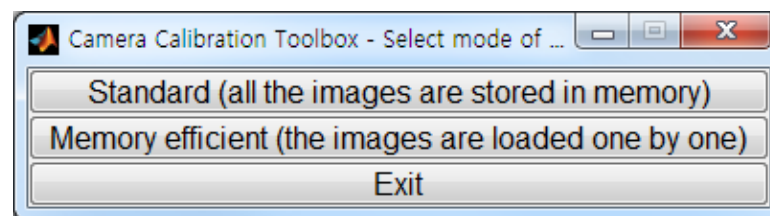
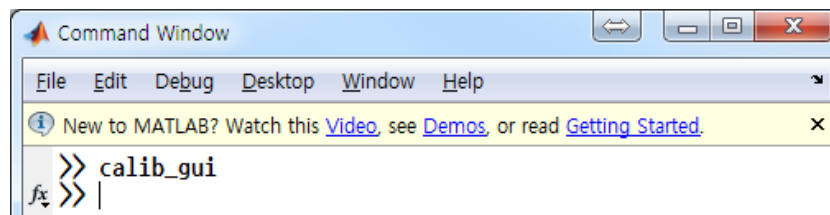
01 MATLAB Camera Calibration Toolbox

Manual

1. Set the Current Folder



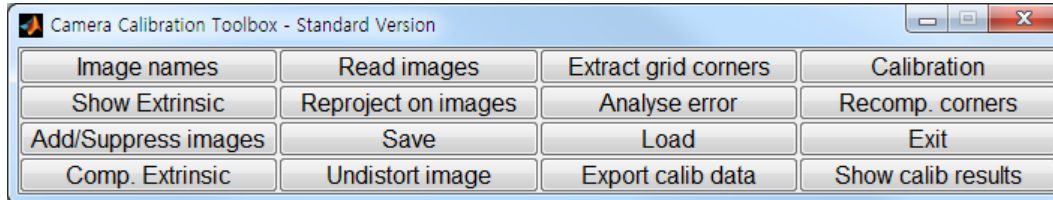
2. Enter as "calib_gui" on your MATLAB Command Window



01 MATLAB Camera Calibration Toolbox

Manual

3. Select the Standard button.



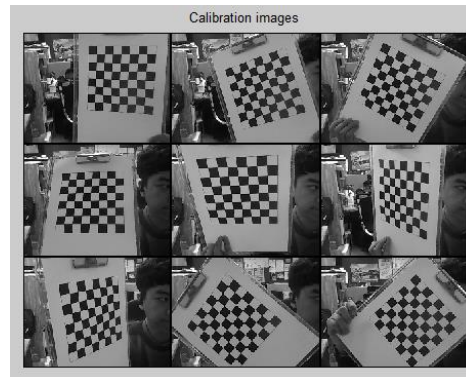
4. Select the Image names.

Baseline camera calibration images (without number nor suffix): **Calib**

Image format: ([]='r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm') **j**



Loading image 1...2...3...4...5...6...7...8...9...
done



cf. Basename



Calib0.jpg



Calib1.jpg

01 MATLAB Camera Calibration Toolbox

Manual

5. Select the **Extract grid corners** button.

Extraction of the grid corners on the images

Number(s) of image(s) to process (0 = all images) =

Window size for corner finder (wintx and winty):

wintx (0 = 5) =

winty (0 = 5) =

Window size = 11x11

Do you want to use the automatic square counting mechanism (0=0=default)
or do you always want to enter the number of squares manually (1,other)?

Processing image 1...

Using (wintx,winty)=(5,5) - Window size = 11x11 (Note: To reset the window size, run script clearwin)

Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...

01 MATLAB Camera Calibration Toolbox

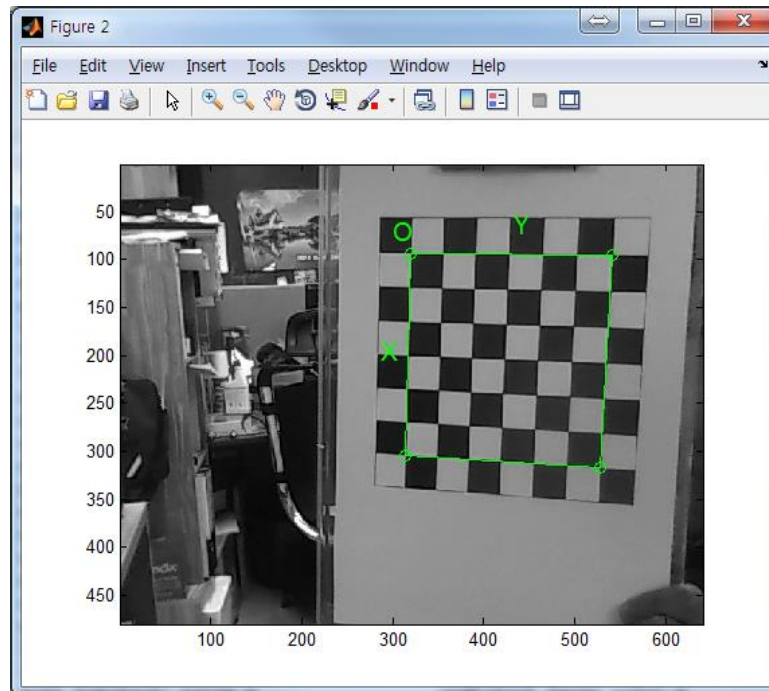
Manual

5. Select the **Extract grid corners** button.

Processing image 1...

Using $(wintx, winty) = (5, 5)$ - Window size = 11x11 (Note: To reset the window size, run script clearwin)

Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...



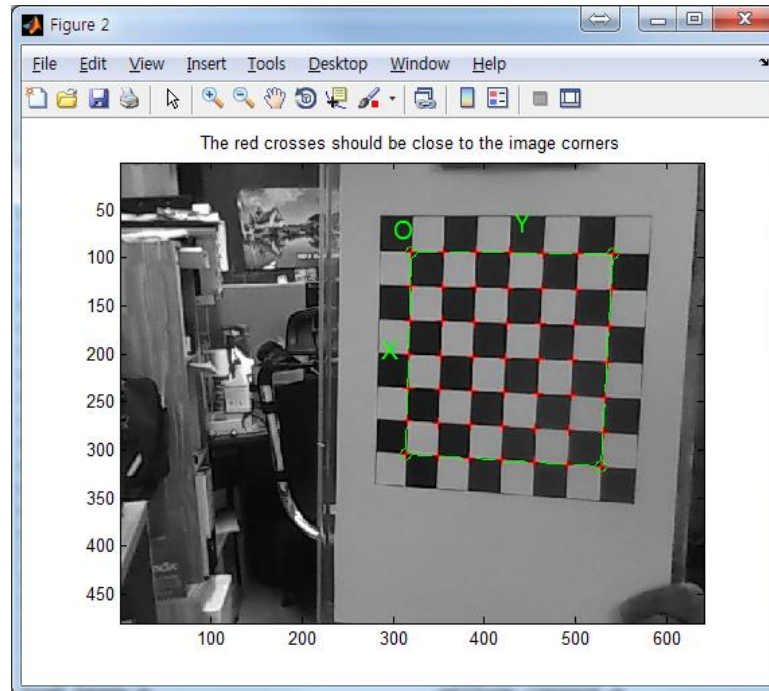
01 MATLAB Camera Calibration Toolbox

Manual

5. Select the **Extract grid corners** button.

Size dX of each square along the X direction ($\square=100\text{mm}$) = 30

Size dY of each square along the Y direction ($\square=100\text{mm}$) = 30



01 MATLAB Camera Calibration Toolbox

Manual

5. Select the **Extract grid corners** button.

If the guessed grid corners (red crosses on the image) are not close to the actual corners, it is necessary to enter an initial guess for the radial distortion factor k_c (useful for subpixel detection)
Need of an initial guess for distortion? (Y=no, other=yes)

Corner extraction...

Processing image 2...

Using (wintx,winty)=(5,5) - Window size = 11x11 (Note: To reset the window size, run script clearwin)

Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...



01 MATLAB Camera Calibration Toolbox

Manual

5. Select the **Extract grid corners** button.

-
-
-

Processing image 9...

Using (wintx,winty)=(5,5) - Window size = 11x11 (Note: To reset the window size, run script clearwin)

Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...

Size of each square along the X direction: dX=30mm

Size of each square along the Y direction: dY=30mm (Note: To reset the size of the squares, clear the variables dX and dY)

If the guessed grid corners (red crosses on the image) are not close to the actual corners, it is necessary to enter an initial guess for the radial distortion factor kc (useful for subpixel detection)

Need of an initial guess for distortion? (n=no, other=yes)

Corner extraction...

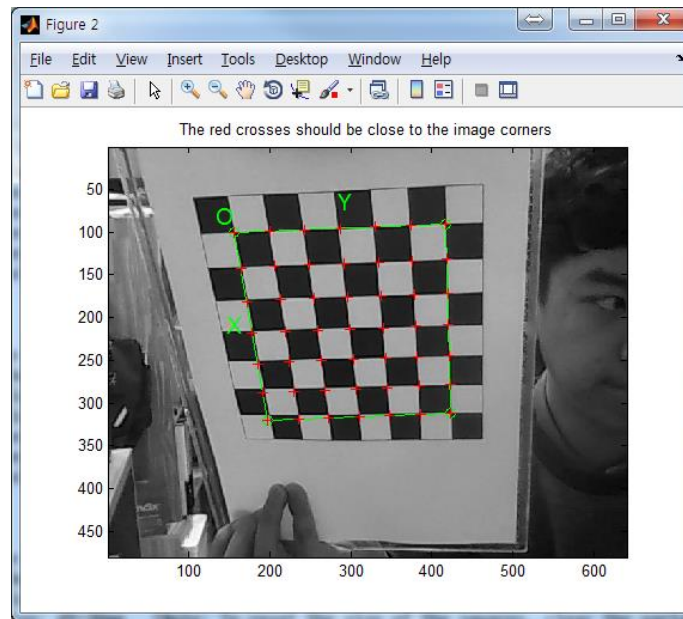
done

01 MATLAB Camera Calibration Toolbox

Manual

5. Select the **Extract grid corners** button.

Caution



Number of squares along the X direction ($N_x=6$) = Number of squares along the Y direction ($N_y=6$) = 6

01 MATLAB Camera Calibration Toolbox

Manual

6. Select the **Calibration** button.

Calibration results after optimization (with uncertainties):

Focal Length: **fc** = [**797.63941** **795.14342**] ? [**6.58758** **6.59608**]

Principal point: **cc** = [**330.89928** **250.97817**] ? [**8.55274** **8.58204**]

Skew: **alpha_c** = [0.00000] ? [0.00000] => angle of pixel axes = 90.00000 ? 0.00000 degrees

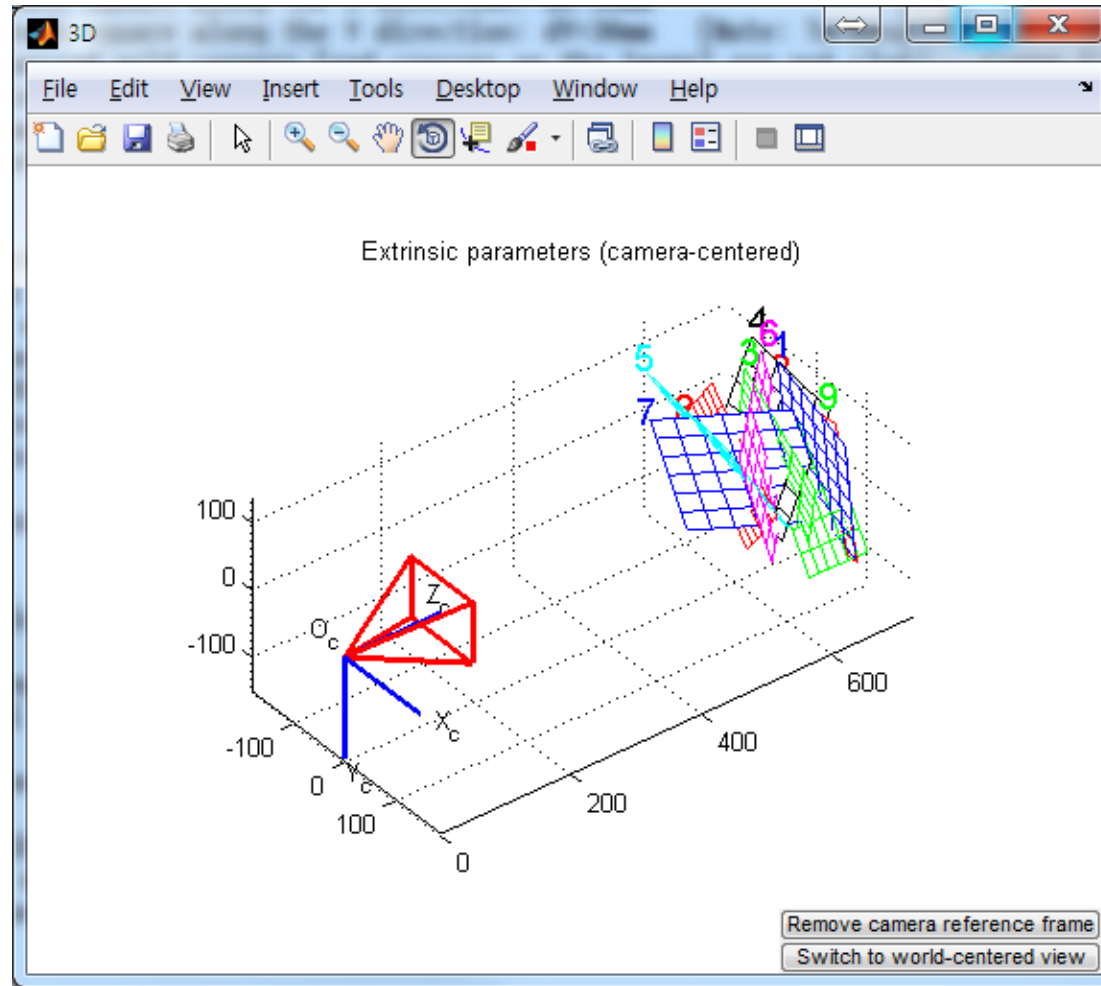
Distortion: **kc** = [0.02543 0.38418 0.00201 -0.00615 0.00000] ? [0.04303 0.42102 0.00400
0.00431 0.00000]

Pixel error: **err** = [0.21182 0.17411]

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

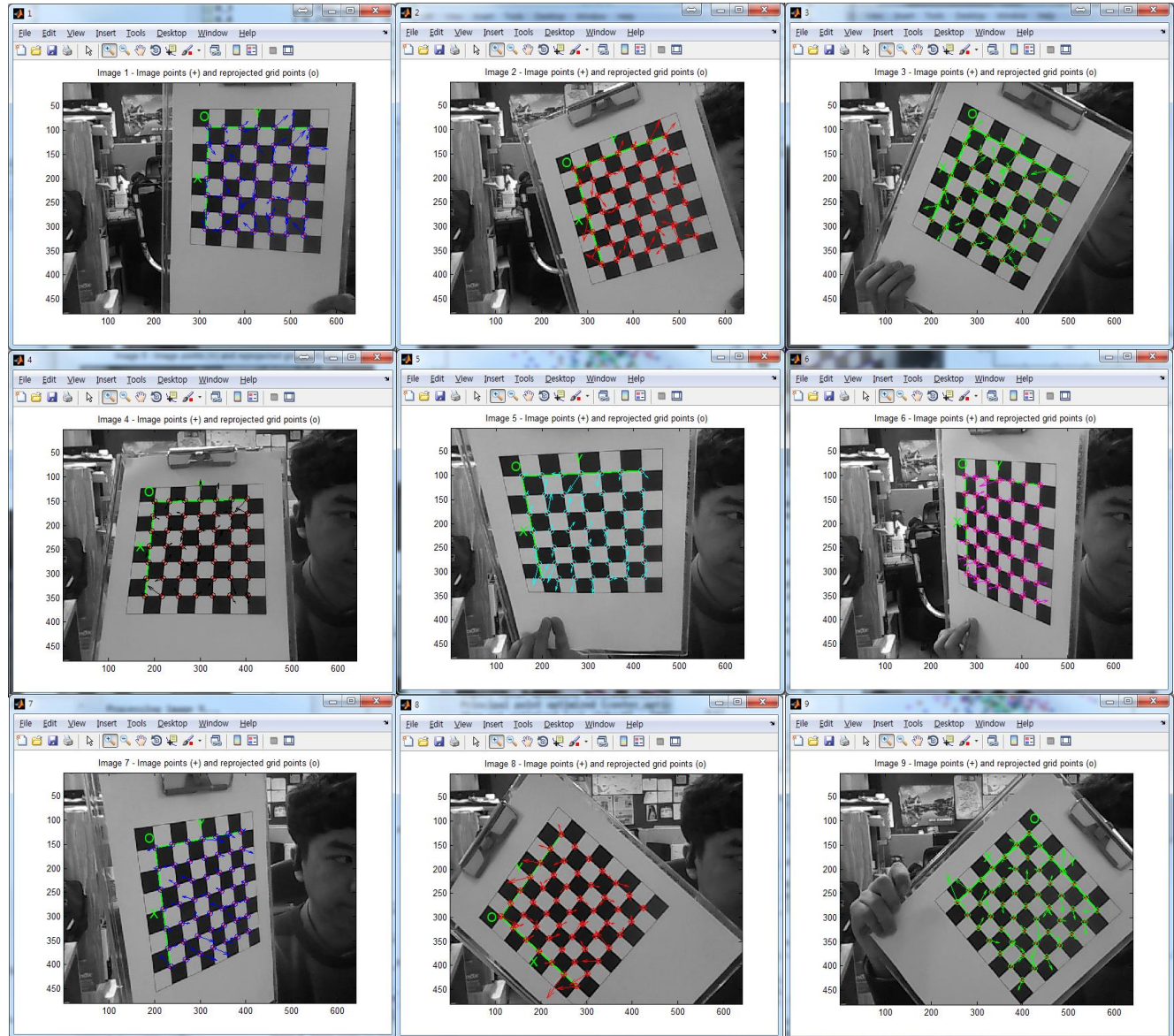
01 MATLAB Camera Calibration Toolbox

cf. Show Extrinsic



01 MATLAB Camera Calibration Toolbox

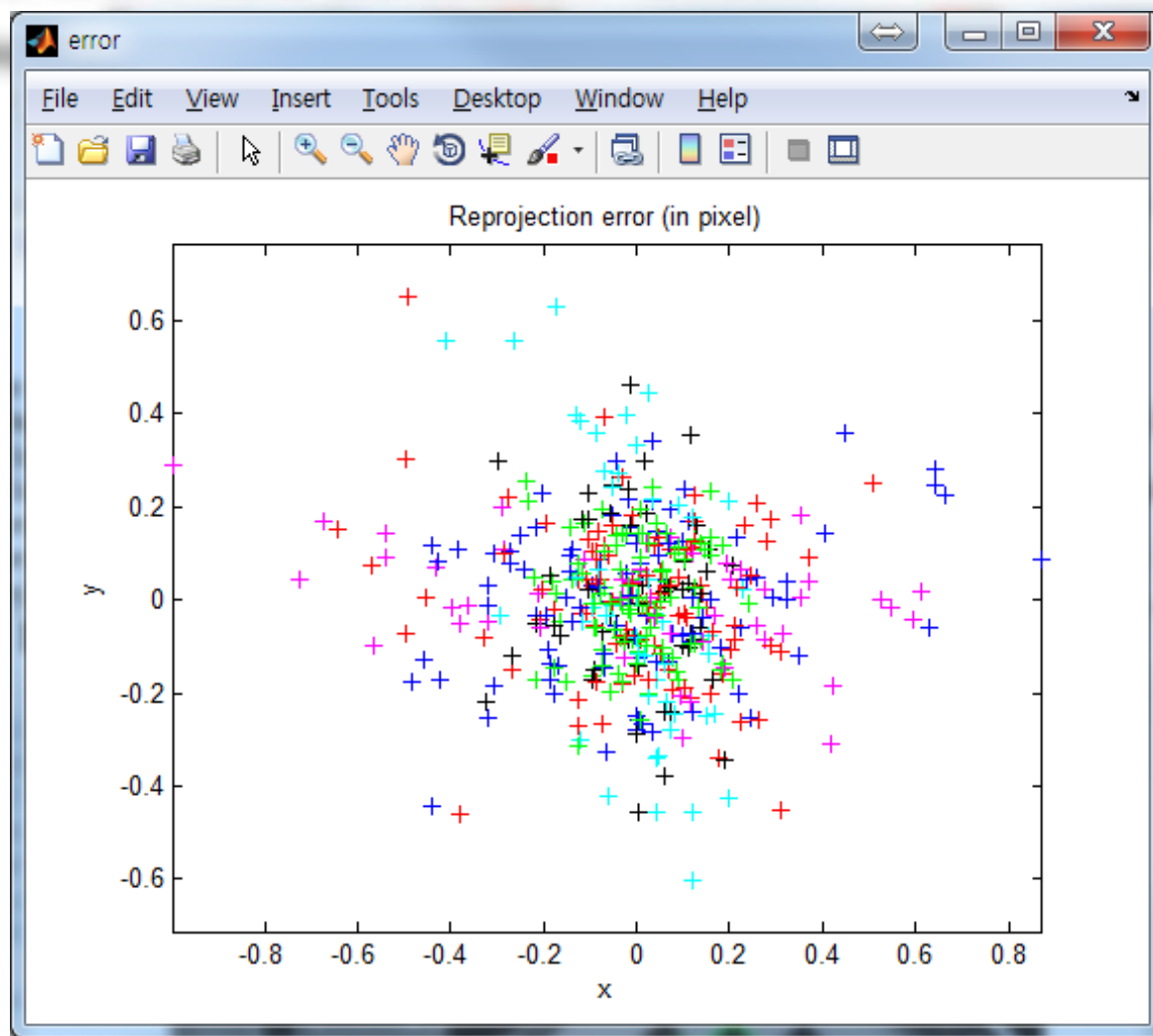
cf. Re-project on Images



01

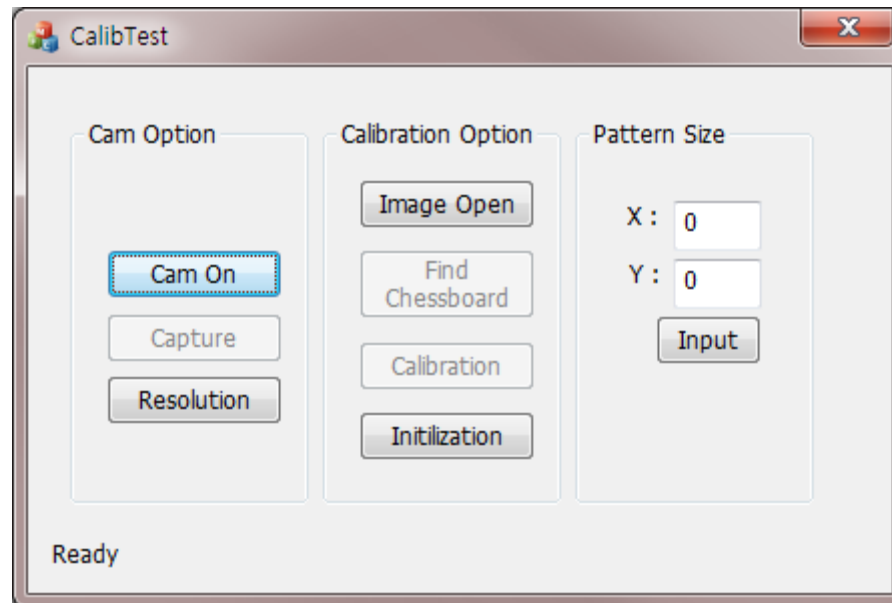
MATLAB Camera Calibration Toolbox

of Reprojection error



02 Camera Calibration Using the Open CV

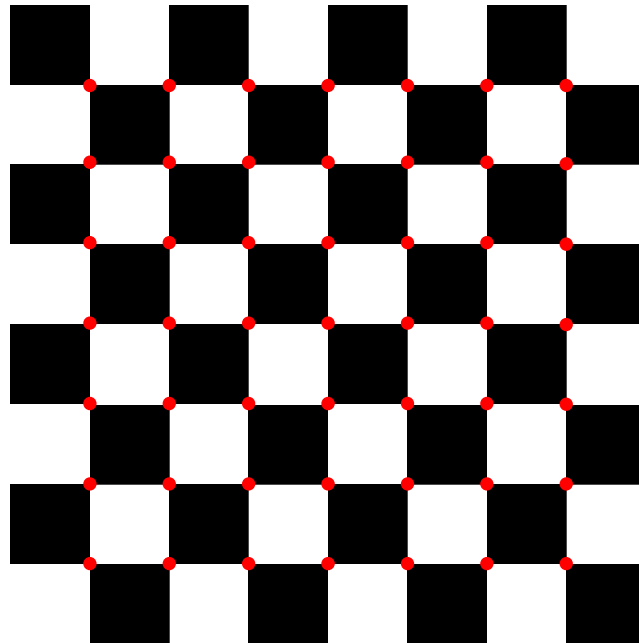
▀ Structure



02 Camera Calibration Using the Open CV

Function Description

```
bool findChessboardCorners(InputArray image, Size patternSize, OutputArray corners, int  
flags=CALIB_CB_ADAPTIVE_THRESH+CALIB_CB_NORMALIZE_IMAGE )
```



Pattern size: 7x7

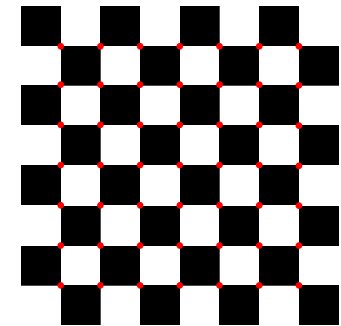
02 Camera Calibration Using the Open CV

Function Description

```
bool findChessboardCorners(InputArray image, Size patternSize, OutputArray corners, int flags=CALIB_CB_ADAPTIVE_THRESH+CALIB_CB_NORMALIZE_IMAGE)
```

[flags]

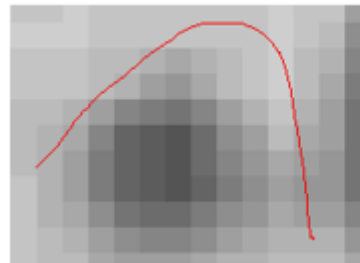
CV_CALIB_CB_ADAPTIVE_THRESH	Using the average image brightness
CV_CALIB_CB_NORMALIZE_IMAGE	Equalizes the histogram.
CV_CALIB_CB_FILTER_QUADS	Filter out false quads extracted
CV_CALIB_CB_FAST_CHECK	speed up the call in the degenerate condition (no chessboard)



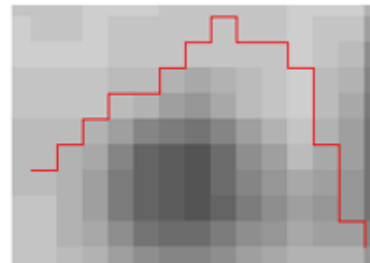
02 Camera Calibration Using the Open CV

Function Description

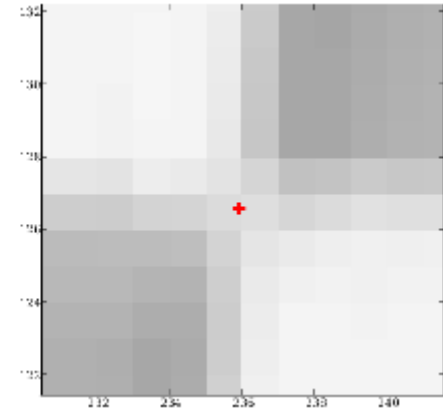
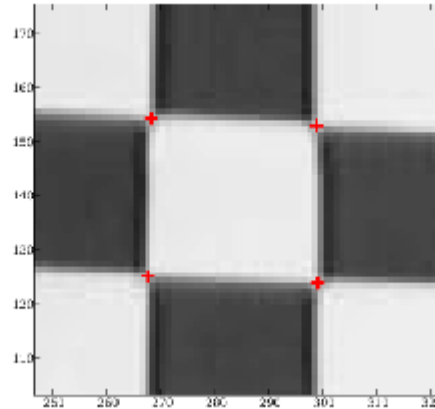
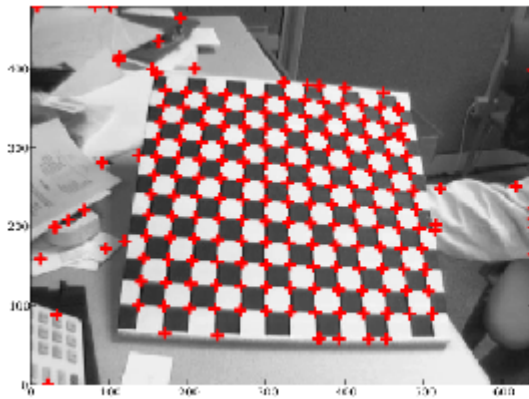
void `cornerSubPix`(InputArray `image`, InputOutputArray `corners`, Size `winSize`, Size `zeroZone`, TermCriteria `criteria`)



With sub-pixel resolution



Without

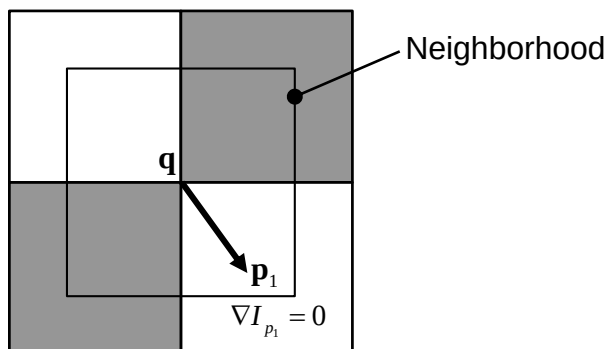


02 Camera Calibration Using the Open CV

Function Description

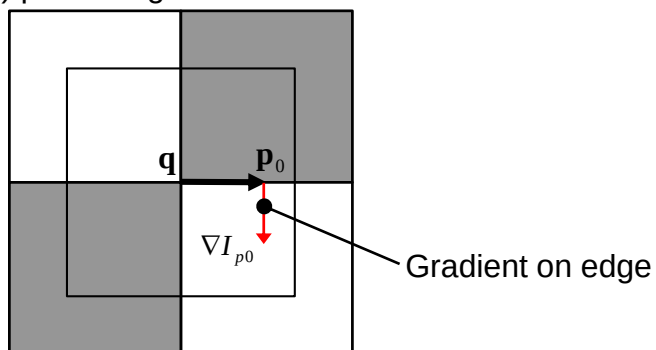
void `cornerSubPix`(InputArray `image`, InputOutputArray `corners`, Size `winSize`, Size `zeroZone`, TermCriteria `criteria`)

i) p in a region



In either case the dot product is **zero**.

ii) p on a region

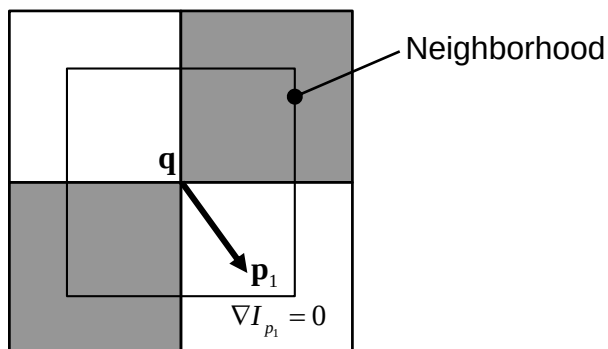


02 Camera Calibration Using the Open CV

Function Description

void `cornerSubPix`(InputArray `image`, InputOutputArray `corners`, Size `winSize`, Size `zeroZone`, TermCriteria `criteria`)

i) p in a region

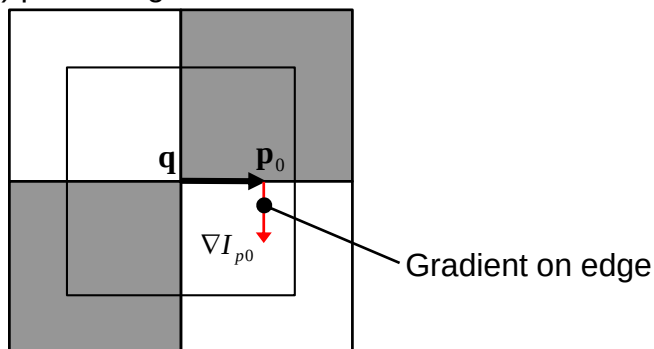


$$\varepsilon_i = \nabla I_{p_i}^T \cdot (q - p_i)$$

$$\sum_i (\nabla I_{p_i} \cdot \nabla I_{p_i}^T) q - \sum_i (\nabla I_{p_i} \cdot \nabla I_{p_i}^T \cdot p_i) = 0$$

$$q = G^{-1} \cdot b \quad \left(G = (\nabla I_{p_i} \cdot \nabla I_{p_i}^T), \quad b = \nabla I_{p_i} \cdot \nabla I_{p_i}^T \cdot p_i \right)$$

ii) p on a region



02 Camera Calibration Using the Open CV

Function Description

void `cornerSubPix`(InputArray `image`, InputOutputArray `corners`, Size `winSize`, Size `zeroZone`, TermCriteria `criteria`)

winSize : Half of the side length of the search window.

ex) if `winSize=Size(5,5)`, then $(5*2+1) \times (5*2+1) = 11 \times 11$

zeroZone : Half of size of dead region in the middle of the search zone

ex) `Size(-1,-1)`

criteria: Criteria for termination of the iterative process of corner refinement.

ex)

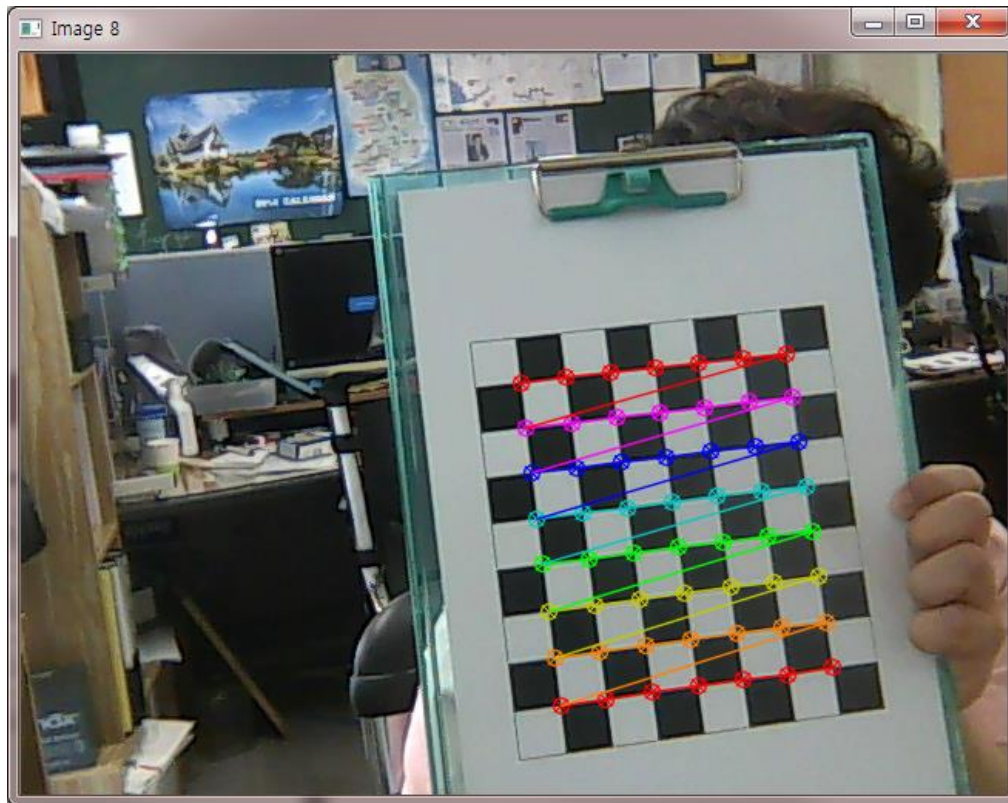
`CV_TERMCRIT_EPS` The maximum number of iterations

`CV_TERMCRIT_ITER` The desired accuracy

02 Camera Calibration Using the Open CV

Function Description

```
void drawChessboardCorners(InputOutputArray image, Size patternSize, InputArray corners, bool patternWasFound)
```



02 Camera Calibration Using the Open CV

Function Description

```
double calibrateCamera(InputArrayOfArrays objectPoints, InputArrayOfArrays imagePoints, Size imageSize,
InputOutputArray cameraMatrix, InputOutputArray distCoeffs, OutputArrayOfArrays rvecs, OutputArrayOfArrays
tvecs, int flags=0, TermCriteria criteria=TermCriteria( TermCriteria::COUNT+TermCriteria::EPS, 30,
DBL_EPSILON) )
```

objectPoints : The corners are at 3D location (X,Y,Z)

ex) (0,0,0), (0,1,0), ..., (1,0,0), (2,0,0), ..., (width-1,height-1,0)

imagePoints : The corners are at image plane (x,y)

imageSize : The size of image

cameraMatrix : The floating-point camera matrix

ex)
$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

distCoeffs : The distortion coefficients

ex)
$$\begin{bmatrix} k_1 & k_2 & p_1 & p_2 \end{bmatrix}^T \quad \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix}^T \quad \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 & k_4 & k_5 & k_6 \end{bmatrix}^T$$

rvecs, tvecs: The rotation vector & translation vector

rotation angle about each axes (CCW)

cf. Rotation matrix

```
void Rodrigues(InputArray src,
OutputArray dst,
OutputArray jacobian=noArray())
```

02 Camera Calibration Using the Open CV

Function Description

```
double calibrateCamera(InputArrayOfArrays objectPoints, InputArrayOfArrays imagePoints, Size imageSize,
InputOutputArray cameraMatrix, InputOutputArray distCoeffs, OutputArrayOfArrays rvecs, OutputArrayOfArrays
tvecs, int flags=0, TermCriteria criteria=TermCriteria( TermCriteria::COUNT+TermCriteria::EPS, 30,
DBL_EPSILON) )
```

flags

CV_CALIB_USE_INTRINSIC_GUESS Optimization using initial values of fx, fy, cx, cy

CV_CALIB_FIX_PRINCIPAL_POINT Image center or initial value

CV_CALIB_FIX_ASPECT_RATIO The ratiofx/fy stays the same as in the input cameraMatrix

CV_CALIB_ZERO_TANGENT_DIST Tangential distortion coefficients $(p_1, p_2) = 0$

CV_CALIB_FIX_K1, ... CV_CALIB_FIX_K6 Radial distortion coefficient

CV_CALIB_RATIONAL_MODEL Coefficients k4, k5, and k6 are enabled

The algorithm is based on Zhang2000* and BouguetMCT*

03 GML C++ Camera Calibration Toolbox

Set up

Browser

← → ↻ <http://graphics.cs.msu.ru/en/node/909>



Graphics and Media Lab

Languages

- English
- Русский

Lab

- Main
- News
- About lab
- People

Research

- Publications
- Grants and contracts
- Projects
 - Collaboration with Microsoft Research
 - Joint research with Intel
 - Video surveillance
 - Image recognition
 - 3D reconstruction

GML C++ Camera Calibration Toolbox

Contact person: Alexander Velizhev (avelizhev@graphics.cs.msu.ru)

Downloads

Latest binaries:

[Download GML C++ Camera Calibration Toolbox 0.75](#)

Latest sources:

[Download GML C++ Camera Calibration sources 0.71](#)

[Download GML C++ Camera Calibration sources 0.4](#)

Image samples:

[Download images](#)

ChangeLog

16.07.2013 v0.75

- A critical bug was fixed.

23.12.2011 v0.73

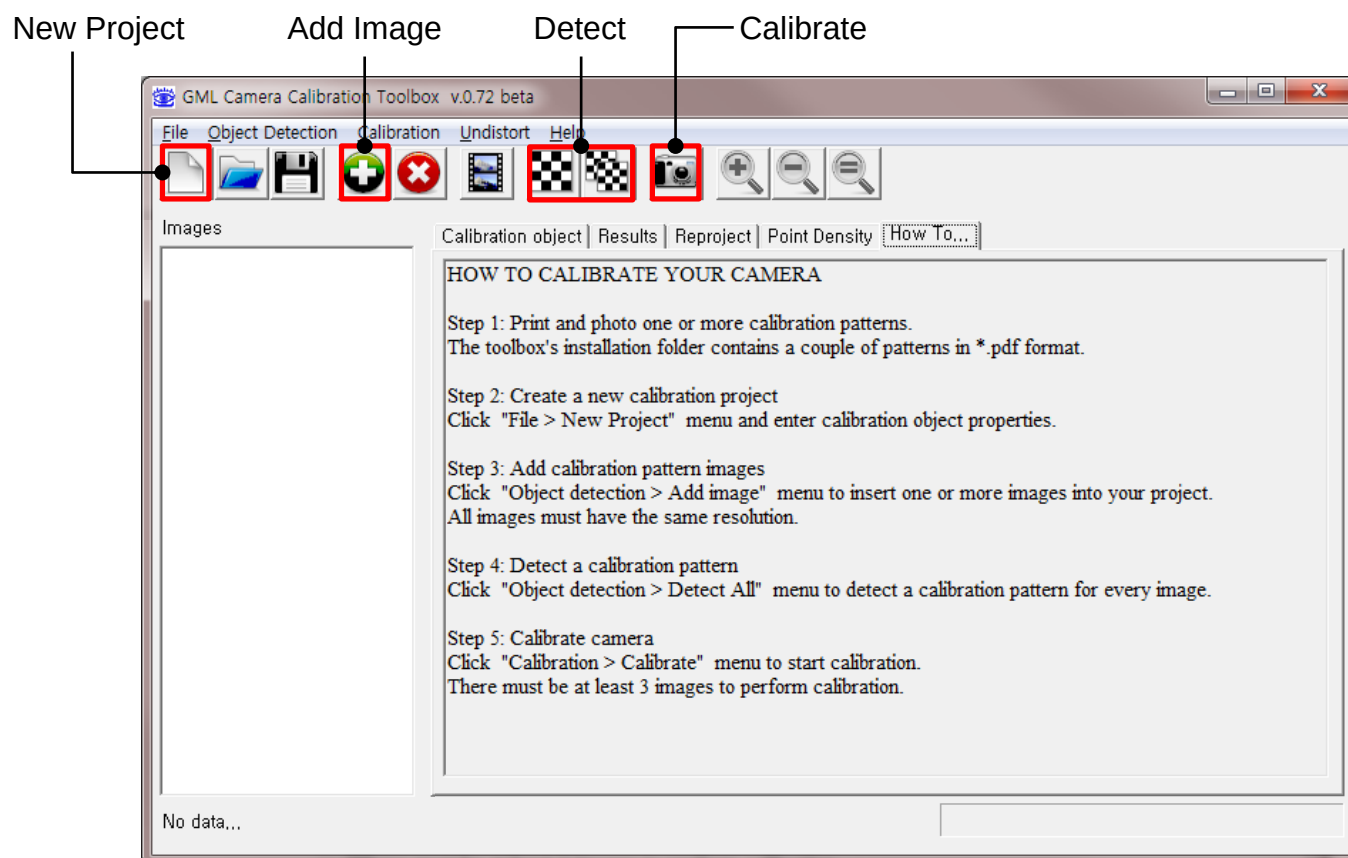
[Two critical bugs were fixed. Many thanks to Evgeny...](#)

Requirements

[.Net Framework 3.5 SP1](#)

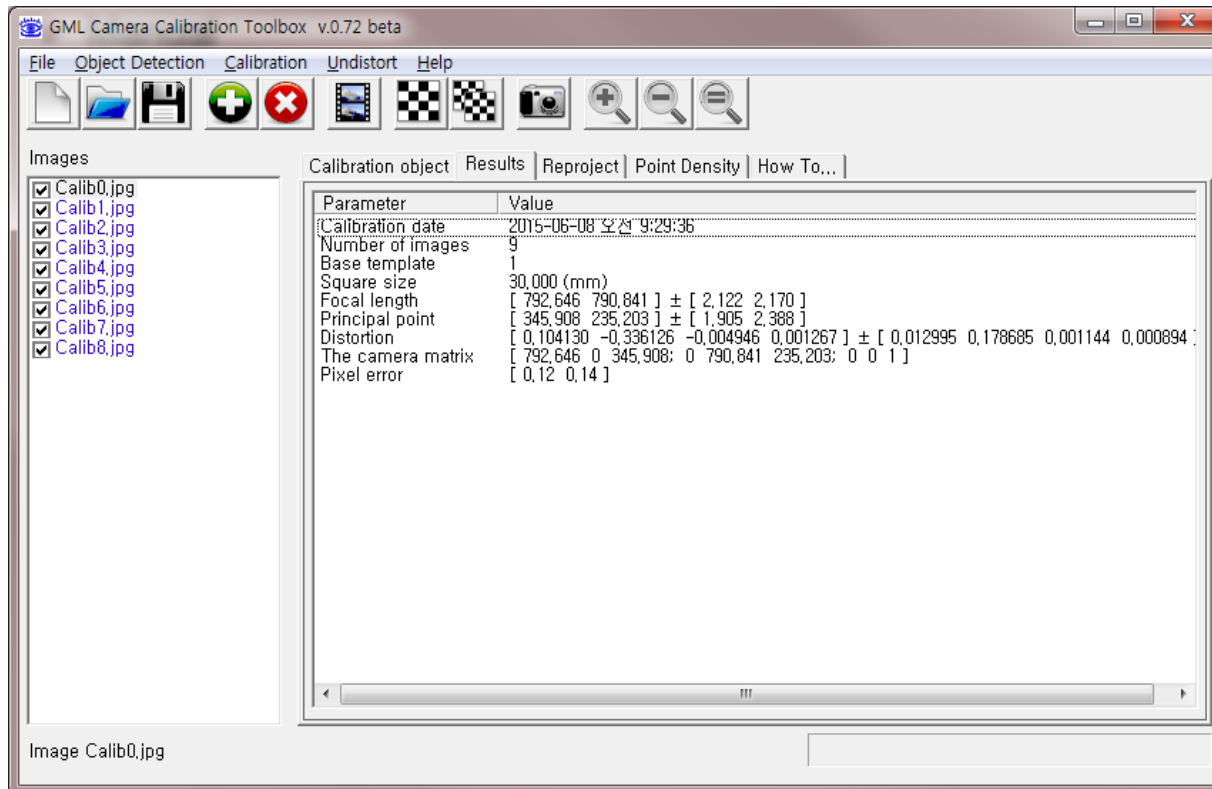
03 GML C++ Camera Calibration Toolbox

Manual



03 GML C++ Camera Calibration Toolbox

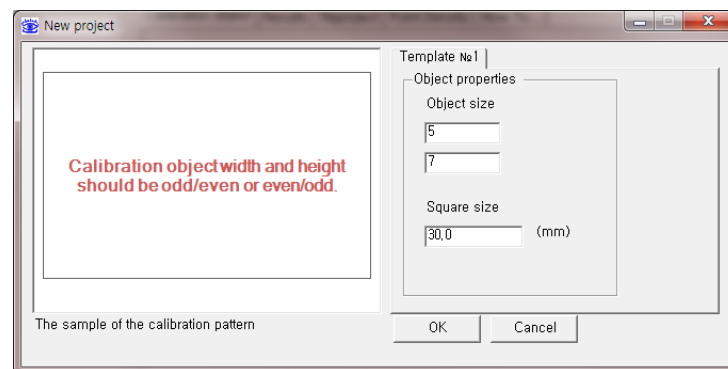
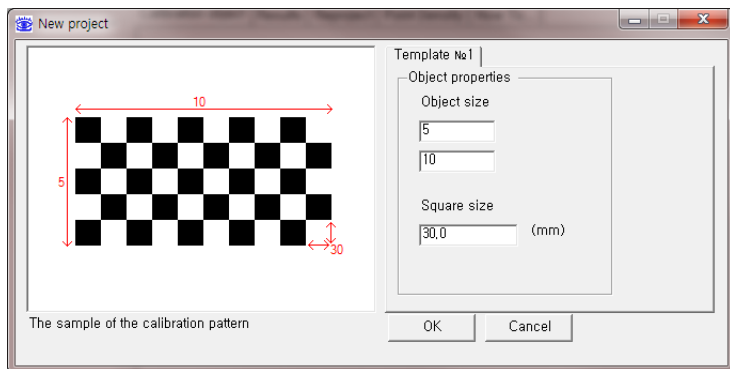
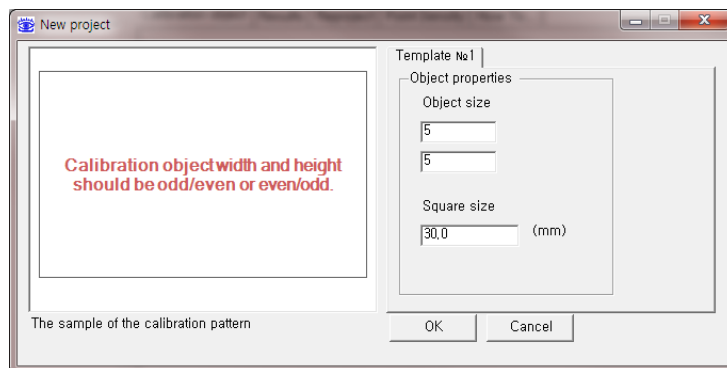
Manual



03 GML C++ Camera Calibration Toolbox

Manual

Caution – Pattern Size



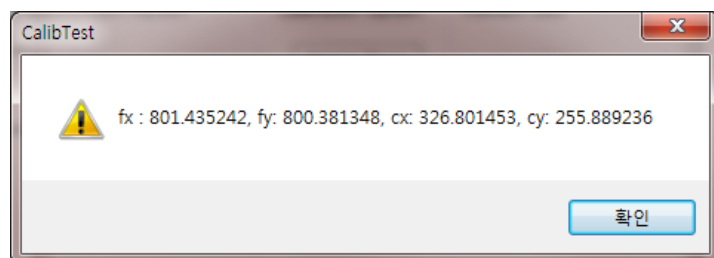
04 Comparison of results

Matlab

Focal Length: $f_c = [800.89664 \quad 799.80606] \diamond [6.54824 \quad 6.25045]$

Principal point: $cc = [325.27471 \quad 256.64234] \diamond [6.35228 \quad 4.94925]$

Open CV



GML

Calibration object		Results	Reproject	Point Density	How To...
Parameter	Value				
Calibration date	2015-06-17 오후 4:32:55				
Number of images	21				
Base template	1				
Square size	30,000 (mm)				
Focal length	[801.093 800.002] ± [2.226 2.126]				
Principal point	[325.311 256.732] ± [2.121 1.654]				
Distortion	[0.079888 -0.143751 0.004856 -0.006727] ± [0.013934 0.257314 0.000740 0.001179]				
The camera matrix	[801.093 0 325.311; 0 800.002 256.732; 0 0 1]				
Pixel error	[0.14 0.14]				

Q & A

Thank You