
University of Idaho
Department of Computer Science
Coeur d'Alene
Lecture #4
Camera Calibration in OpenCV
Garrett Wells
revised August 25, 2024

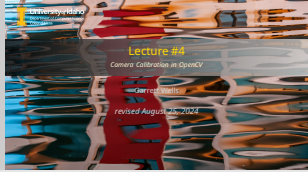


Table of Contents I

Table of Contents I

Table of Contents

Review

Camera Calibration in OpenCV
Calibration Patterns
Images and Image Points
Object Points
Calibration

Distortion Correction

Assignment Preview

Review

$$x = PX$$

Review
$x = PX$

Review

$$x = PX$$

☞ *translates 3D homogeneous to 4D homogeneous*

Review
$x = PX$
☞ translates 3D homogeneous to 4D homogeneous

Review

$$x = PX$$

☞ *translates image to world coordinates*

Review
$x = PX$
☞ translates image to world coordinates

Review

$$P = KR[I|-C]$$

Review
$P = KR[I|-C]$

Review

$$P = KR[I|-C]$$

K	***intrinsic matrix:*** *image plane → camera/virtual image plane*
$R[I\|-C]$	***extrinsic matrix:*** *camera → world coordinate*
R	***rotation***
$-C$	***translation***

Review
$P = KR[I|-C]$
K ***intrinsic matrix:*** *image plane → camera/virtual image plane*
$R[I|-C]$ ***extrinsic matrix:*** *camera → world coordinate*
R ***rotation***
$-C$ ***translation***

Camera Calibration in OpenCV

Overview

1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process:** images of known pattern → find reference points → calculate distortion coefficients

Camera Calibration in OpenCV
Overview
1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process:** images of known pattern → find reference points → calculate distortion coefficients

1. p_x, p_y, are equivalent to c_x, c_y from the OpenCV tutorial.

Camera Calibration in OpenCV

Overview

1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process: images of known pattern** → find reference points → calculate distortion coefficients

Camera Calibration in OpenCV
Overview
1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process: images of known pattern** → find reference points → calculate distortion coefficients

1. p_x, p_y, are equivalent to c_x, c_y from the OpenCV tutorial.

Camera Calibration in OpenCV

Overview

1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process:** images of known pattern → **find reference points** → calculate distortion coefficients

Camera Calibration in OpenCV
Overview
1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process:** images of known pattern → **find reference points** → calculate distortion coefficients

1. p_x, p_y, are equivalent to c_x, c_y from the OpenCV tutorial.

Camera Calibration in OpenCV

Overview

1. **Goal:** find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. **Process:** images of known pattern → find reference points → **calculate distortion coefficients**

Camera Calibration in OpenCV
Overview
1. Goal: find distortion coefficients, intrinsic parameters, extrinsic parameters
   - $(k_1, k_2, p_1, p_2, k_3)$
   - f_x, f_y
   - p_x, p_y
2. Process: images of known pattern → find reference points → calculate distortion coefficients

1. p_x, p_y, are equivalent to c_x, c_y from the OpenCV tutorial.

Calibration Patterns

- OpenCV calibration patterns [1]

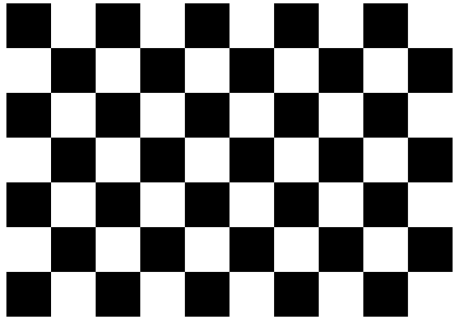


Figure: Chessboard Corners [2]

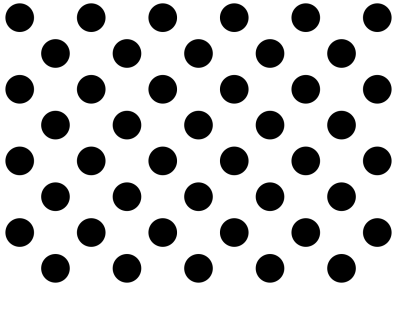
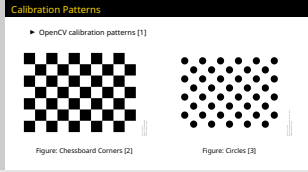


Figure: Circles [3]



1. Note that each calibration pattern includes note(bottom left) describing what the image size should be at $1:1$ scale when used for calibration.
2. The OpenCV tutorial for camera calibration [4], doesn't say much about *why* users might want to use the chessboard pattern over circles. It does say, however, that circles require fewer calibration images to produce good results.

OpenCV Calibration Patterns

Key Factors

OpenCV Calibration Patterns
Key Factors

OpenCV Calibration Patterns

Key Factors

1. **We know the size of the pattern and number of elements in the pattern.**

OpenCV Calibration Patterns
Key Factors
1. We know the size of the pattern and number of elements in the pattern.

OpenCV Calibration Patterns

Key Factors

1. We know the size of the pattern and number of elements in the pattern.
2. **Approx. 10 images are needed for best results [4].**

OpenCV Calibration Patterns
Key Factors
1. We know the size of the pattern and number of elements in the pattern.
2. **Approx. 10 images are needed for best results [4].**

OpenCV Calibration Patterns

Key Factors

1. We know the size of the pattern and number of elements in the pattern.
2. Approx. 10 images are needed for best results [4].
3. **Pattern is easily distinguishable, not naturally occurring in environment.**

OpenCV Calibration Patterns
Key Factors
1. We know the size of the pattern and number of elements in the pattern.
2. Approx. 10 images are needed for best results [4].
3. **Pattern is easily distinguishable, not naturally occurring in environment.**

OpenCV Calibration Patterns

Key Factors

1. We know the size of the pattern and number of elements in the pattern.
2. Approx. 10 images are needed for best results [4].
3. Pattern is easily distinguishable, not naturally occurring in environment.
4. **Either pattern or camera needs to move between frames, but not both.**

OpenCV Calibration Patterns
Key Factors
1. We know the size of the pattern and number of elements in the pattern.
2. Approx. 10 images are needed for best results [4].
3. Pattern is easily distinguishable, not naturally occurring in environment.
4. **Either pattern or camera needs to move between frames, but not both.**

Mounting the Calibration Pattern

8.5”x11” Cardboard! *or any rigid & flat surface*

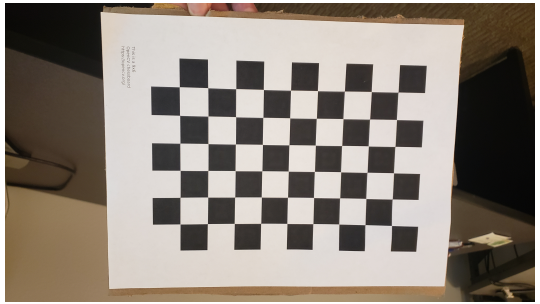


Figure: Example Setup

Mounting the Calibration Pattern
8.5”x11” Cardboard! *or any rigid & flat surface*
Figure: Example Setup

Image Capture

1. Use camera to capture multiple images of the calibration pattern.
2. Use OpenCV to get the *image points(2D)* and *object points(3D)*

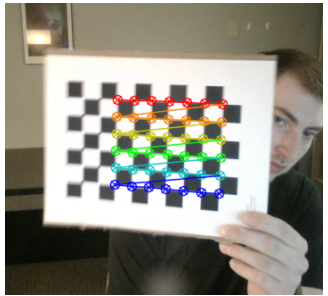
Image Capture
1. Use camera to capture multiple images of the calibration pattern.
2. Use OpenCV to get the *image points(2D)* and *object points(3D)*

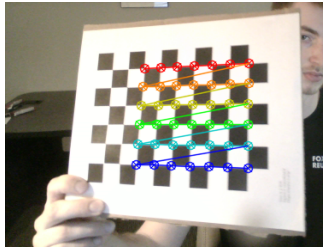
1. Remember, keep the camera **or** the pattern stationary, move the unfixed one between images
2. If using a webcam, likely easiest to have the camera be stationary

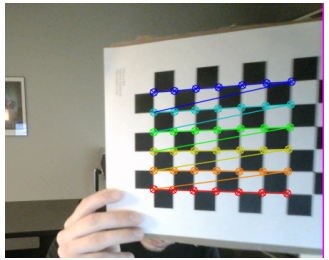
Finding Image Points

Example

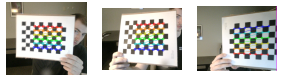
```
cv.findChessboardCorners(image, (x_size, y_size), None)
```







Finding Image Points
Example
cv.findChessboardCorners(image, (x_size, y_size), None)



Finding Image Points

Example

```
cv.findChessboardCorners(image, (x_size, y_size), None)
```

1. 1 set per calibration image
2. Chessboard Pattern Image Point → contact point between two black squares

Finding Image Points
Example
cv.findChessboardCorners(image, (x_size, y_size), None)
1. 1 set per calibration image
2. Chessboard Pattern Image Point → contact point between two black squares

Finding Image Points

Example

```
cv.findChessboardCorners(image, (x_size, y_size), None)
```

1. 1 set per calibration image
2. Chessboard Pattern Image Point → contact point between two black squares
3. Save points for calibration

Finding Image Points
Example
cv.findChessboardCorners(image, (x_size, y_size), None)
1. 1 set per calibration image
2. Chessboard Pattern Image Point → contact point between two black squares
3. Save points for calibration

Object Points

Object Point

An object point is has fixed location and size based on our calibration pattern.

Object Points
Object Point
An object point is has fixed location and size based on our calibration pattern.

Object Points

Object Point

An object point is has fixed location and size based on our calibration pattern.

- ☞ OpenCV makes this easy, just pass in list of points

Object Points
Object Point
An object point is has fixed location and size based on our calibration pattern.
☞ OpenCV makes this easy, just pass in list of points

Object Points

Object Point

An object point is has fixed location and size based on our calibration pattern.

- ☞ OpenCV makes this easy, just pass in list of points
- ☞ Since calibration pattern doesn’t change, same points apply to each image

Object Points
Object Point
An object point is has fixed location and size based on our calibration pattern.
OpenCV makes this easy, just pass in list of points
Since calibration pattern doesn't change, same points apply to each image

Calibration

Example

```
cv.calibrateCamera(obj_pts, img_pts, ...)
```

- ☞ Returns:
  - ➥ camera matrix (P)
  - ➥ distortion coefficients
  - ➥ rotation params
  - ➥ translation params

Undistorting An Image

The final step…

OpenCV provides two methods for doing this:

- using camera matrix, dist. coefficients, etc, to correct image (easy)
- using remapping (more complex)

1. Remap specifies a pixel location (x, y) to map every input pixel to in the output.

Validation

- Is there potential error?
  - yes

Validation
- Is there potential error?
  - yes

Validation

- Is there potential error?
- Reprojection Error

Validation
- Is there potential error?
- Reprojection Error

Validation

- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points

Validation
- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points

Validation

- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points
  - Calculates the difference between an image generated from our **object points** and the distortion corrected image

Validation
- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points
  - Calculates the difference between an image generated from our **object points** and the distortion corrected image

Validation

- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points
  - Calculates the difference between an image generated from our object points and the **distortion corrected image**

Validation
- Is there potential error?
- Reprojection Error
  - We can use transform/**reproject** object points into image points
  - Calculates the difference between an image generated from our object points and the **distortion corrected image**

Assignment #3 Preview

Calibrate a Camera

Use an OpenCV calibration pattern to calibrate your camera.

1. Print out and mount a calibration pattern
2. Obtain 10 images of the pattern using your camera
3. Save your parameters to JSON file
4. Load parameters from file
5. Use them to correct distortion in an image from your camera

Assignment #3 Preview
Calibrate a Camera
Use an OpenCV calibration pattern to calibrate your camera.
1. Print out and mount a calibration pattern
2. Obtain 10 images of the pattern using your camera
3. Save your parameters to JSON file
4. Load parameters from file
5. Use them to correct distortion in an image from your camera

Extra Credit

50-75 pts for building Camera Obscura/Pinhole Camera
Due before December 1st.

Extra Credit
50-75 pts for building Camera Obscura/Pinhole Camera
Due before December 1st.

Bibliography I

[1] “OpenCV: Camera Calibration and 3D Reconstruction,” (), [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga93efa9b0aa890de240ca32b11253dd4a (visited on 08/24/2024).

[2] “Opencv/doc/pattern.png at 4.x · opencv/opencv,” GitHub. (), [Online]. Available: https://github.com/opencv/opencv/blob/4.x/doc/pattern.png (visited on 08/24/2024).

[3] “Opencv/doc/acircles_pattern.png at 4.x · opencv/opencv,” GitHub. (), [Online]. Available: https://github.com/opencv/opencv/blob/4.x/doc/acircles_pattern.png (visited on 08/24/2024).

[4] “OpenCV: Camera Calibration,” (), [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (visited on 08/24/2024).

Bibliography I

[1] “OpenCV: Camera Calibration and 3D Reconstruction,” (), [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga93efa9b0aa890de240ca32b11253dd4a (visited on 08/24/2024).

[2] “Opencv/doc/pattern.png at 4.x · opencv/opencv,” GitHub. (), [Online]. Available: https://github.com/opencv/opencv/blob/4.x/doc/pattern.png (visited on 08/24/2024).

[3] “Opencv/doc/acircles_pattern.png at 4.x · opencv/opencv,” GitHub. (), [Online]. Available: https://github.com/opencv/opencv/blob/4.x/doc/acircles_pattern.png (visited on 08/24/2024).

[4] “OpenCV: Camera Calibration,” (), [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (visited on 08/24/2024).