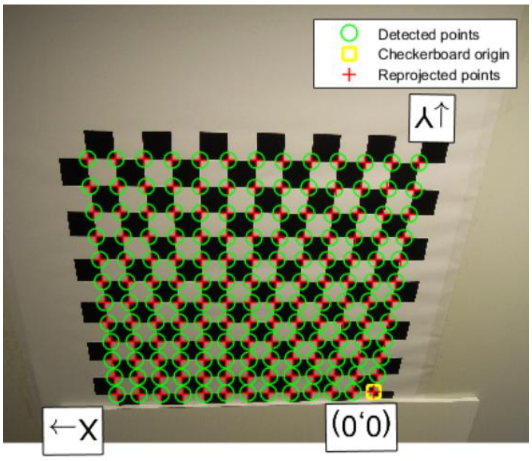
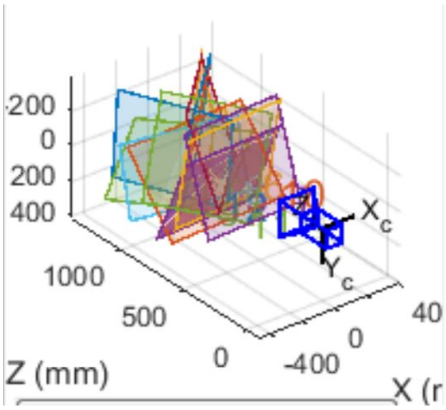
# Undistort radial distortion in an image using checkerboard




מגישים:

טימור ליידרמן 306122631

אסף קסלסי 301467098

# Table of Contents

# Introduction

In geometric optics, distortion is a deviation from rectilinear projection; a projection in which straight lines in a scene remain straight in an image. It is a form of optical aberration. For most lenses, the radial distortion component is predominant, it may appear as a barrel distortion or pincushion distortion. Radial distortion bends straight lines into circular arcs, violating the main invariance preserved in the pinhole camera model, in which straight lines in the world map to straight lines in the image plane. Radial distortion is the most significant type of distortion in today's cameras. With a calibration and some remapping we can correct this. Furthermore, with calibration you may also determine the relation between the camera's natural units (pixels) and the real-world units (for example millimeters). To better understand the world of lens distortion we have to understand the pinhole camera model.

# Pinhole camera Model

A pinhole camera is a simple camera without a lens and with a single small aperture. Light rays pass through the aperture and project an inverted image on the opposite side of the camera. Think of the virtual image plane as being in front of the camera and containing the upright image of the scene.

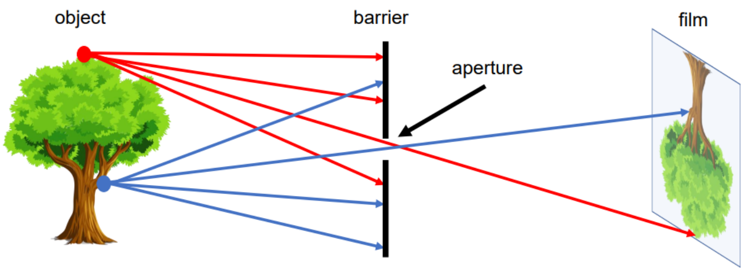



*Figure 1: CS231A Course Notes 1: Camera Models Kenji Hata and Silvio Savarese*

The pinhole camera parameters are represented in a 4-by-3 matrix called the *camera matrix*. This matrix maps the 3-D world scene into the image plane. The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent the location of the camera in the 3-D scene. The intrinsic parameters represent the optical center and focal length of the camera.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R \mid t] \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} \quad (1)$$

where (u, v) is the distortion-free image point on the image plane. The matrix A fully depends on the camera's 5 intrinsic parameters $(f_x, f_y, \gamma, u_0, v_0)$.

$(f_x, f_y)$ the focal length, the distance between the pinhole and the film (a.k.a. image plane), the focal length is measured in pixels and in a true pinhole camera, both $f_x$ $and$ $f_y$ have the same value.

$(u_0, v_0)$ the coordinates of the principal point, and γ describing the skewness of the two image axes. $[X^c\ Y^c\ Z^c]^T$denotes a point in the camera frame that is related to the corresponding point $[X^w\ Y^w\ Z^w]^T$ in the world reference frame by $P^c = RP^w + t$, with (R, t) being the rotation matrix and the translation vector

The world points are transformed to camera coordinates using the extrinsic parameters. The camera coordinates are mapped into the image plane using the intrinsic parameters.

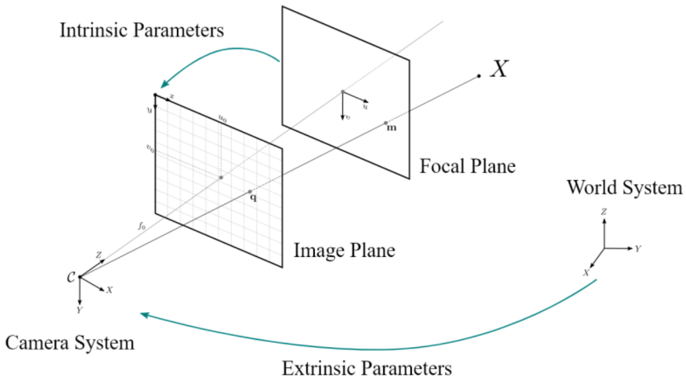



*Figure 2. Intrinsic and Extrinsic Parameters (http://lfa.mobivap.uva.es/~fradelg/phd/tracking/camera.html)*

# Camera Calibration Parameters

The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.

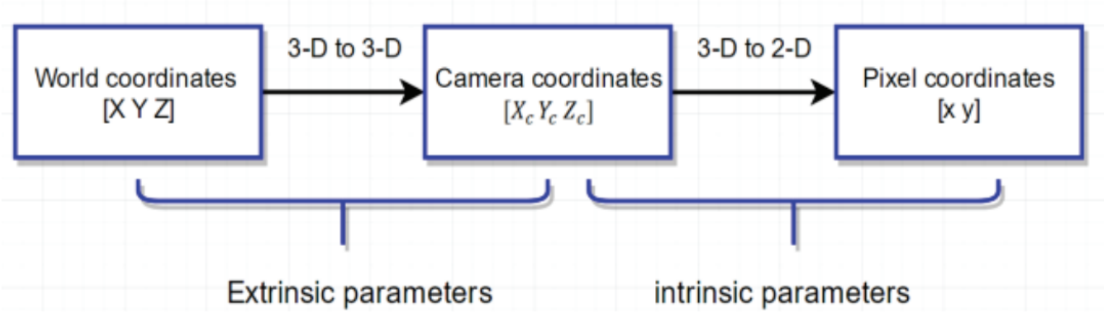



*Figure 3. World to Camera Transform Conference Proceedings, Zurich Switzerland Jan 15-16, 2018, 20 (1) Part III*

## Extrinsic Parameters

The extrinsic parameters consist of a rotation, *R*, and a translation, *t*. The origin of the camera's coordinate system is at its optical center and its *x*- and *y*-axis define the image plane.

## Intrinsic Parameters

The intrinsic parameters include the focal length, the optical center, also known as the *principal point*, and the skew coefficient.

# Polynomial and Rational distortion functions

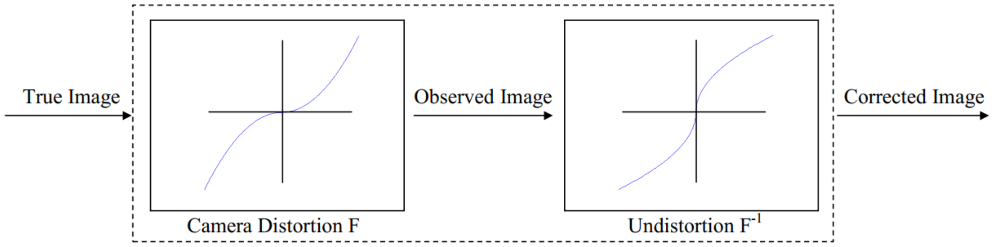



*Figure 4. Lens Distortion A Family of Simplified Geometric Distortion Models for Camera Calibration*

Which is equivalent to $x_d = x\, f(r, K), y_d = y\, f(r, K)$ (2)

| # | $f(r, \mathbf{k})$ | # | $f(r, \mathbf{k})$ |
|---|---|---|---|
| 1 | $1 + k_1\, r$ | 6 | $1/(1 + k_1\, r^2)$ |
| 2 | $1 + k_1\, r^2$ | 7 | $(1 + k_1\, r)/(1 + k_2\, r^2)$ |
| 3 | $1 + k_1\, r + k_2\, r^2$ | 8 | $1/(1 + k_1\, r + k_2\, r^2)$ |
| 4 | $1 + k_1\, r^2 + k_2\, r^4$ | 9 | $(1 + k_1\, r)/(1 + k_2\, r + k_3\, r^2)$ |
| 5 | $1/(1 + k_1\, r)$ | 10 | $(1 + k_1\, r^2)/(1 + k_2\, r + k_3\, r^2)$ |

*Figure 5. Table of K functions : A Family of Simplified Geometric Distortion Models for Camera Calibration*

Notice that all the functions in the Table satisfy the following properties:
1) The function is radially symmetric around the center of distortion and is expressed in terms of the radius r only;
2) The function is continuous and $r_d = 0\ \ if\ r = 0$;
3) The approximation of $X_d$is an odd function of x.
The above three properties act as the criteria to be a candidate for the radial distortion function

Function [4] which is the most commonly used for conventional radial distortion in the polynomial approximation category and the other 9 functions in table are studied specifically with the goal to achieve comparable performance. function [4] is the best using the least amount of model complexity and as few distortion coefficients as possible.

# Distortion

The camera matrix does not account for lens distortion because an ideal pinhole camera does not has a lens. To accurately represent a real camera, the camera model includes the radial and tangential lens distortion.

## Radial Distortion

Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center. The smaller the lens, the greater the distortion.

The radial distortion coefficients model this type of distortion. The distorted points are denoted as ($x_{\text{distorted}}$, $y_{\text{distorted}}$):

$$x_{\text{distorted}} = x\,(1 + k_1{*}r^2 + k_2{*}r^4 + k_3{*}r^6) \qquad (3)$$

$$y_{\text{distorted}} = y\,(1 + k_1{*}r^2 + k_2{*}r^4 + k_3{*}r^6) \qquad (4)$$

*x*, *y* — Undistorted pixel locations. *x* and *y* are in normalized image coordinates. Normalized image coordinates are calculated from pixel coordinates by translating to the optical center and dividing by the focal length in pixels. Thus, *x* and *y* are dimensionless.

$k_1$, $k_2$, and $k_3$ — Radial distortion coefficients of the lens.

$r^2$: $x^2 + y^2$

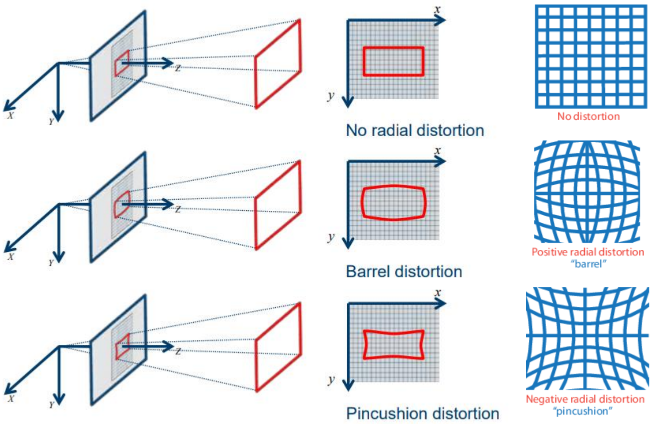



*Figure 5. Distortion types*

For the distortion we consider the radial and tangential factors. For the radial factor one uses the following formula:

$$X_{corrected} = X(1 + K_1 r^2 + K_2 r^2 + K_3 r^6 \ldots) \qquad (5)$$

$$Y_{corrected} = Y(1 + K_1 r^2 + K_2 r^2 + K_3 r^6 \ldots) \qquad (6)$$

For an old pixel point at (X,Y) coordinates in the input image, its position on the corrected output image will be $(X_{corrected}, Y_{corrected})$. The presence of the radial distortion manifests in form of the "barrel" or "fish-eye" effect.

# Experiment

for unit conversion we use the following formula:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{8}$$

Here the presence of w is explained by the use of homograph coordinate system and (W = Z). The unknown parameters are $f_x$ and $f_y$ (camera focal lengths) and $(c_x, c_y)$ which are the optical centers expressed in pixels coordinates. Note that $\gamma = 0$ we assume the we have no skew.If for both axes a common focal length is used with a given "α" aspect ratio (usually 1), then $f_y = f_x * a$ and in the upper formula we will have a single focal length F. The matrix containing these four parameters is referred to as the *camera matrix*. While the distortion coefficients are the same regardless of the camera resolutions used, these should be scaled along with the current resolution from the calibrated resolution. The process of determining these two matrices is the calibration. Calculation of these parameters is done through basic geometrical equations. The equations used depend on the chosen calibrating objects. We will use for this a classical black and white chessboard (checkerboard)

The recommended calibration procedure is as follows:

1. Print a pattern and attach it to a planar surface;
2. Take a few images of the model plane under different orientations by moving either the plane
or the camera;
3. Detect the feature points in the images;
4. Estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution.
5. Estimate the coefficients of the radial distortion by solving the linear least-squares.

# Analysis

We took a checkerboard with the square dimensions of 315mm and took several photos in order to cover the whole space, we then gave our code these images as inputs and ran our algorithm, as some of the images we took were blurry because of motion blur and shake because of the camera movement not all of the images were able to be calibrated so we used only some of the inputs.

The next step was to detect the checkerboard pattern in our images

Then we Read the first image to obtain image size

Later on we generated world coordinates of the corners of the squares, as we already know that each square size in millimeters is 315mm

We then calibrate the camera and go over the reprojection errors to make sure our error is not too big,
We then visualize pattern locations and display parameter estimation errors
The final step is to export the distortion parameters and with that data and the calibration data we can remove effects of lens distortion.

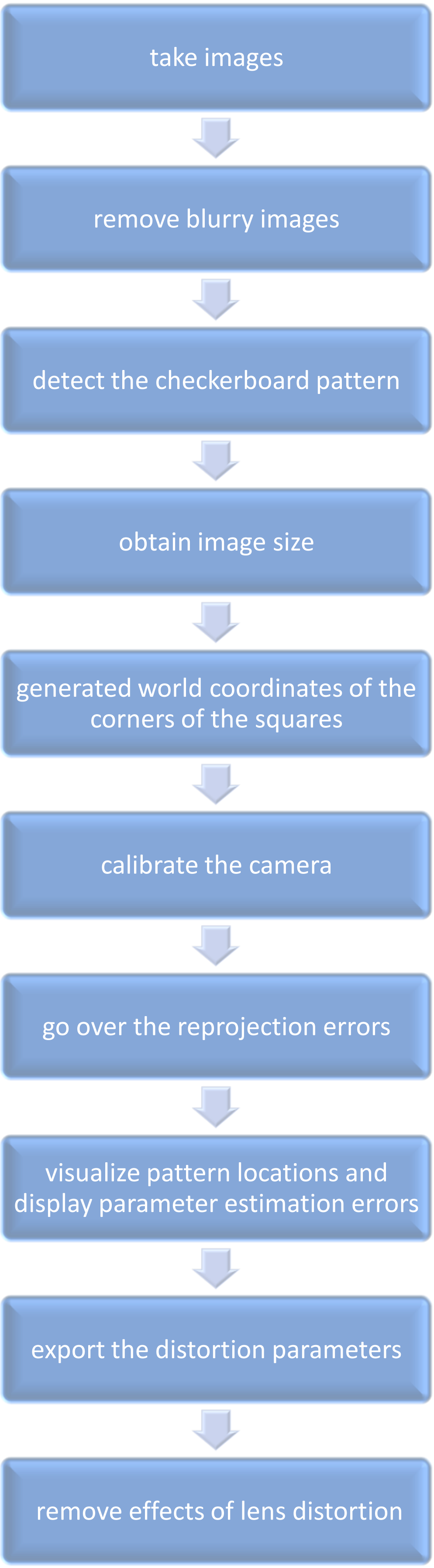



*Figure 6. Top-down Flow Chart diagram (self-Made)*

# Comparisons

To demonstrate the feasibility of the results obtained from the proposed system, they were compared to detection techniques that are already well established. This analysis was carried out for both the detection of the board pattern and the calibration values.
The Python's OpenCV lib method first performs a thresholding, then followed by an erosion process, which is used to break the connectivity of the squares; after this it finds their number and position in the image. Since information regarding the number of rows and columns is supplied, the erosion process is repeated as many times as is necessary to break the connectivity and find the corresponding numbers. Next, the squares are brought back together to form the board pattern. The number of points must be passed through each calibration and the pattern must be completely visible in the image, as is opposite to the technique proposed in this project. The results of the calibration with the proposed method were compared to those obtained with the MATLAB Toolbox, the main difference found when using this technique is that the method proposed here does not require each corner of the chess board pattern to be marked, neither the knowledge of the number of rows and columns.

# Conclusions

In this Image Processing Project, a semi-automatic camera calibration method has been presented. This is achieved by detecting an artificial pattern located within an image. The only information available in the calibration pattern used is that it is composed of a chess board configuration. The process can determine the number of rows and columns of the pattern. Using different examples, it has been shown that this process is a valuable tool for camera calibration and that its use will make this task less tedious, allowing the process to be easily performed by non-specialized personnel.

# Future improvement suggestion

A 3D straight line in space can be distorted to a circle arc. We can find all "straight lines" in the image and fit every line to a circle arcs then we look for the distortion parameters that minimize the curvature.

The algorithm to estimate the parameter of the radial distortion works as follows:

1. Extract as many "straight line" from the image using Hough transform;

2. Determine parameter by fitting every "straight line" with an arc of a circle to calculate the center of the radial distortion.

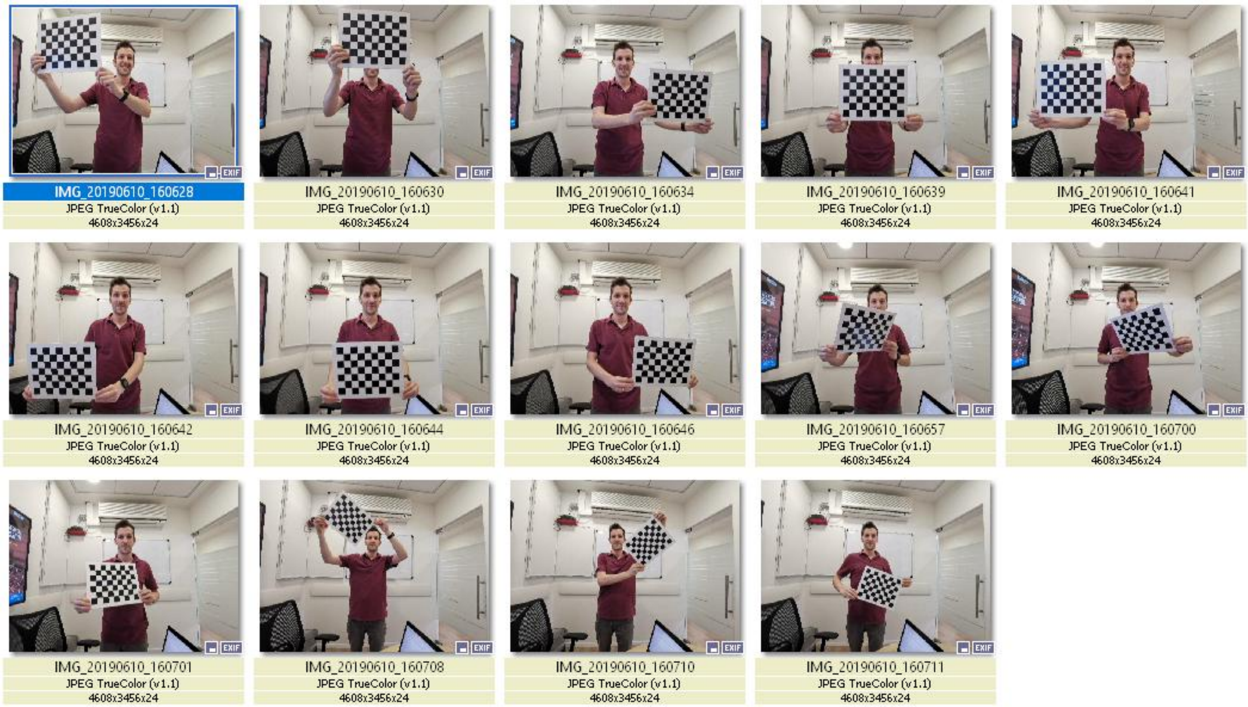
4. Calculate the parameter of radial distortion with least square algorithm.

# Matlab code:

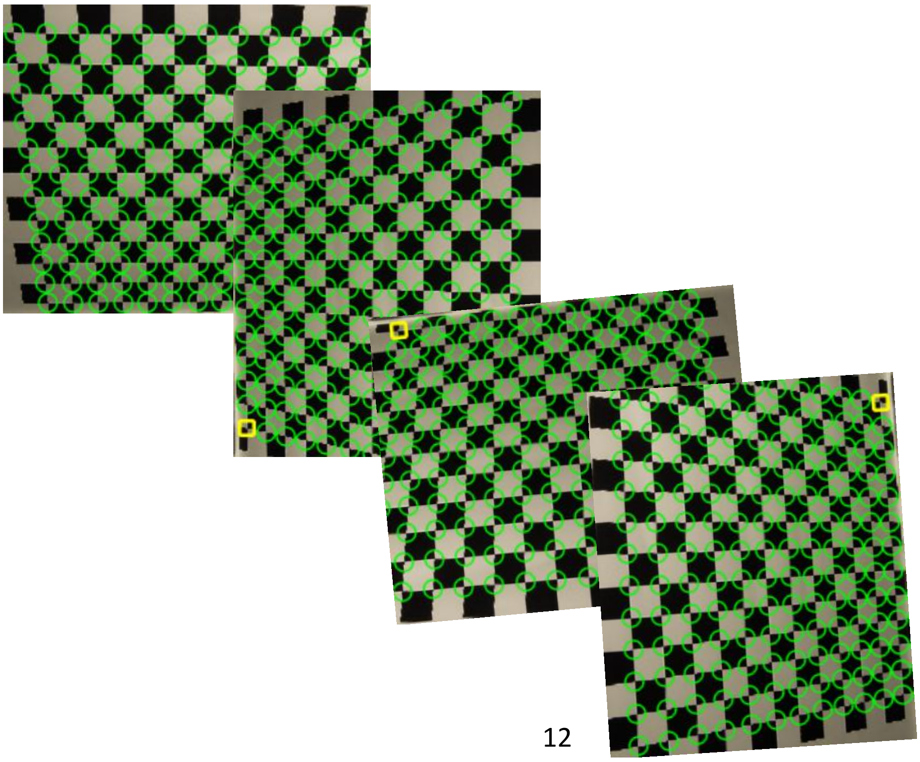
```
% Define images to process
imageFileNames = {'E:\developer\undistort\1.bmp',...
    'E:\developer\undistort\2.bmp',...
    'E:\developer\undistort\3.bmp',...
    'E:\developer\undistort\4.bmp',...
    'E:\developer\undistort\5.bmp',...
    'E:\developer\undistort\6.bmp',...
    };
% Detect checkerboards in images
[imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(imageFileNames);
imageFileNames = imageFileNames(imagesUsed);

% Read the first image to obtain image size
originalImage = imread(imageFileNames{1});
[mrows, ncols, ~] = size(originalImage);

% Generate world coordinates of the corners of the squares
squareSize = 315;  % in units of 'millimeters'
worldPoints = generateCheckerboardPoints(boardSize, squareSize);

% Calibrate the camera
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints,
worldPoints, ...
    'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
    'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...
    'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...
    'ImageSize', [mrows, ncols]);

% View reprojection errors
h1=figure; showReprojectionErrors(cameraParams);

% Visualize pattern locations
h2=figure; showExtrinsics(cameraParams, 'CameraCentric');

% Display parameter estimation errors
displayErrors(estimationErrors, cameraParams);

% For example, you can use the calibration data
% to remove effects of lens distortion.
undistortedImage = undistortImage(originalImage, cameraParams);

figure(1)
imshow(undistortedImage)
hold on
imshow(originalImage)
```

# Results

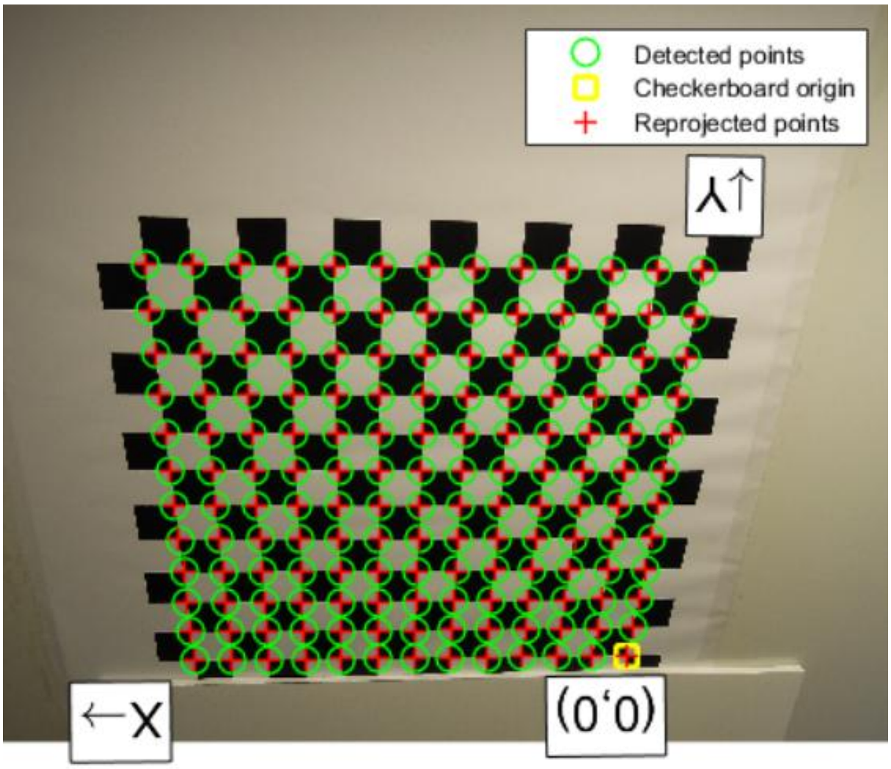
## Input images



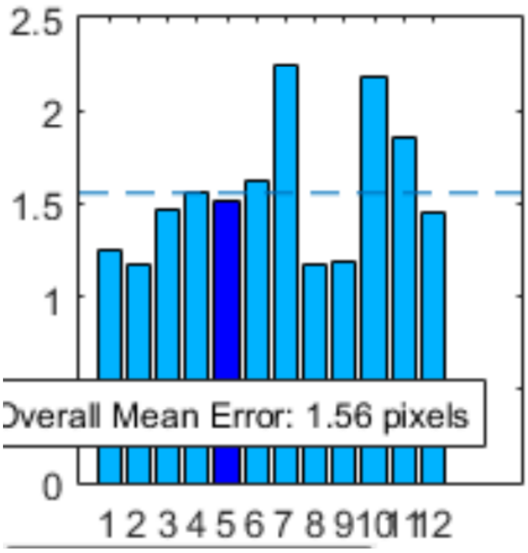

# Detected points



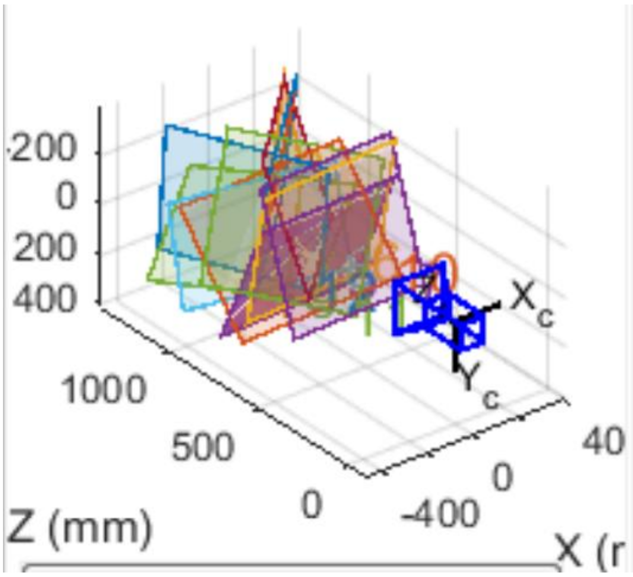
# Detected points, origins and re-projected points




# Reprojection errors and mean error [in pixels]



# Checkerboard Position in mm related to the camera




# Properties and values

| Property | Value |
|---|---|
| RadialDistortion | [-0.0826 0.2631 -0.3568] |
| TangentialDistortion | [0 0] |
| WorldPoints | *156x2 double* |
| WorldUnits | 'mm' |
| EstimateSkew | 0 |
| NumRadialDistortionCoefficients | 3 |
| EstimateTangentialDistortion | 0 |
| TranslationVectors | *12x3 double* |
| ReprojectionErrors | *156x2x12 double* |
| RotationVectors | *12x3 double* |
| NumPatterns | 12 |
| IntrinsicMatrix | [2.8325e+03 0 0;0 2.8191e+03 0;2.0123e+03 1.5272e+03 1 |
| FocalLength | [2.8325e+03 2.8191e+03] |
| PrincipalPoint | [2.0123e+03 1.5272e+03] |
| Skew | 0 |
| MeanReprojectionError | 1.5566 |
| ReprojectedPoints | *156x2x12 double* |
| RotationMatrices | *3x3x12 double* |

# References

[1] A Family of Simplified Geometric Distortion Models for Camera Calibration, Lili Ma, YangQuan Chen, Kevin L. Moore , (Submitted on 2 Aug 2003)

[2] Duane, C. Brown. "Close-range camera calibration." Photogramm. Eng 37.8 (1971): 855-866.

[3] Wikipedia contributors. (2019, May 24). Distortion (optics). In *Wikipedia, The Free Encyclopedia*. Retrieved 18:27, June 10, 2019, from https://en.wikipedia.org/w/index.php?title=Distortion_(optics)&oldid=898595830

[3] Zhang, Z. “A Flexible New Technique for Camera Calibration.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22, No. 11, 2000, pp. 1330–1334

[4] https://www.mathworks.com/help/vision/ug/camera-calibration.html

[5] CS231A Course Notes 1: Camera Models Kenji Hata and Silvio Savarese

[6] Self-Calibration of Fish-Eye Camera for Advanced Driver Assistance Systems Atef Alaaeddine Sarraj, Brendan Jackman, Frank Walsh , Conference Proceedings, Zurich Switzerland Jan 15-16, 2018, 20 (1) Part III

[7] http://lfa.mobivap.uva.es/~fradelg/phd/tracking/camera.html