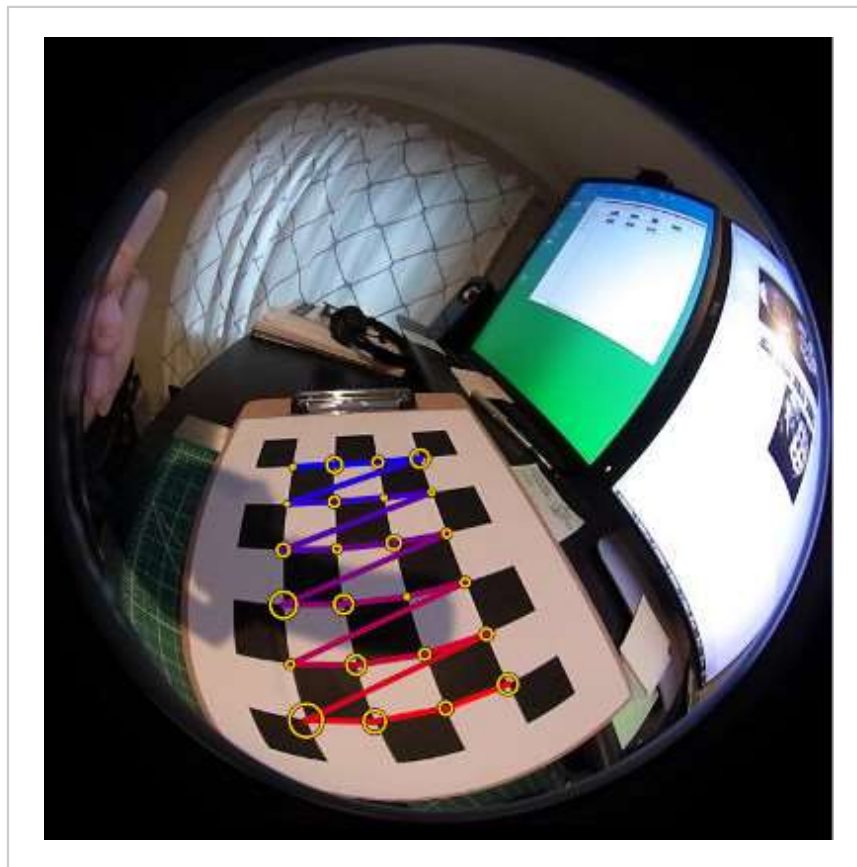


Example Calibrate Planar Fisheye

From BoofCV



Detected Calibration Target in a Fisheye Camera.

This example demonstrates how to compute the intrinsic camera parameters for a fisheye camera lens. Fisheye lenses exhibit significantly more distortion than regular lenses with a more narrow field of view. It's not unusual for a fisheye lens to have a FOV of 185 degrees. The calibration process is very similar to regular cameras. A planar calibration target is shown at different angles across the entire field of view. The main difference is the camera model.

Example File: ExampleCalibrateFisheye.java (<https://github.com/lessthanoptimal/BoofCV/blob/v1.1.0/examples/src/main/java/boofcv/examples/calibration/ExampleCalibrateFisheye.java>)

Calibration Tutorial: Wikipage

Concepts:

- Camera calibration
- Fisheye Lens distortion
- Intrinsic parameters

Relevant Videos:

- YouTube Video (<https://youtu.be/TGg-xgTyaU8?t=69>)

Related Examples:

- Calibrate Regular Lenses

Example Code

```
/**
 * Example of how to calibrate a single (monocular) fisheye camera using a high level interface. This example
 * for the most part follows the same routine as {@link ExampleCalibrateMonocular}. Fisheye cameras tend to require
 * more images to properly calibrate. Often people will use larger calibration targets too that are easier to
 * see at a distance and cover more of the fisheye's camera large FOV.
 *
 *
 * @author Peter Abeles
 * @see CalibrateMonoPlanar
 */
public class ExampleCalibrateFisheye {
    public static void main( String[] args ) {
        DetectSingleFiducialCalibration detector;

        // Circle based calibration targets are not recommended because the sever lens distortion will change
        // the apparent location of tangent points.

        // Square Grid example
        detector = FactoryFiducialCalibration.squareGrid(null, new ConfigGridDimen(/*rows*/ 4, /*cols*/ 3, /*size*/ 30, /*space*/ 30));
        images = UtilIO.listAll(UtilIO.pathExample("calibration/fisheye/square_grid"));

        // Chessboard Example
        detector = FactoryFiducialCalibration.chessboardX(null, new ConfigGridDimen(/*rows*/7, /*cols*/5, /*size*/30));
        List<String> images = UtilIO.listAll(UtilIO.pathExample("calibration/fisheye/chessboard"));

        // Declare and setup the calibration algorithm
        var calibrator = new CalibrateMonoPlanar();

        // Specify the camera model to use. Here are a few examples.
        //
        calibrator.configureUniversalOmni( /*zeroSkew*/ true, /*radial*/ 2, /*tangential*/ false);
        // it's also possible to fix the mirror offset parameter
        // 0 = pinhole camera. 1 = fisheye
        calibrationAlg.configureUniversalOmni( /*zeroSkew*/ true, /*radial*/ 2, /*tangential*/ false, /*offset*/ 1.0);
        // Another popular model is Kannala-Brandt. Most people just use the symmetric terms.
        calibrationAlg.configureKannalaBrandt( /*zeroSkew*/ true, /*symmetric*/ 5, /*asymmetric*/ 0);

        var usedImages = new ArrayList<String>();
        for (String n : images) {
```

```

BufferedImage input = UtilImageIO.loadImage(n);
if (input == null)
    continue;
GrayF32 image = ConvertBufferedImage.convertFrom(input, (GrayF32)null);
if (detector.process(image)) {
    // Need to tell it the image shape and the layout once
    if (usedImages.isEmpty())
        calibrator.initialize(image.getWidth(), image.getHeight(), List.of(detector.getLayout()));
    calibrator.addImage(detector.getDetectedPoints().copy());
    usedImages.add(n);
} else {
    System.err.println("Failed to detect target in " + n);
}
}
// process and compute intrinsic parameters
CameraModel intrinsic = calibrator.process();

// save results to a file and print out
CalibrationIO.save(intrinsic, "fisheye.yaml");

System.out.println(calibrator.computeQualityText(usedImages));
System.out.println();
System.out.println("--- Intrinsic Parameters ---");
System.out.println();
intrinsic.print();
}
}

```

Retrieved from "http://boofcv.org/index.php?title=Example_Calibrate_Planar_Fisheye&oldid=3272"

This page was last edited on 9 September 2023, at 19:04.