Checkerboard Tracker for Camera Calibration

Andrew DeKelaita
EE368

Abstract

The checkerboard extraction process is an important pre-preprocessing step in camera calibration. This project attempts to implement the checkerboard extraction processes using methods learned from EE368. This includes the isolation of the checkerboard using morphological operations. The benefit of the approach is that it is efficient since morphological operations can be implemented efficiently on a DSP. In addition, it will be shown that it is possible to obtain a camera calibration by combining the code developed for this project with the Camera Calibration Toolbox for Matlab.

Motivation and Introduction

Camera calibration is an important first step for computer vision applications. While checkerboard extraction is considered a solved problem, The Camera Calibration Toolbox for Matlab [1] requires one to manually enter the edges of the checkerboard in order for the tool to both: define an origin and guide the checkerboard extraction algorithm. [5] The goal of this project is to automatically determine the extreme corners of the checkerboard and track an arbitrarily selected origin such that the camera calibration process may be automated.

Method

Initial Screening Process

Step 1: Harris Corner Detection

The first step involved detecting corners using the Harris corner detector. The goal of this step is to produce enough corners to obtain the outline of the checkerboard. However, producing too many corners is overbearing to the initial screening process. It was empirically found that a Gaussian filter with $\sigma=1$ yields usable results. As σ approaches .5 it was found that a great deal of corners were produced. This result became overbearing for the initial screening process as more clutter was introduced into the image.

Figure-1: The image illustrates the original image in grayscale with the corners produced by the Harris corner detector.

Step 2: Outlier Detection

Outliers were detected and removed using the Thomson Tau method. For this project 5% of the initial corners obtained using the Harris Corner detector were considered outliers and removed immediately. Also, the histogram of the original RGB image was used to detect if the corners that passed screening in Step 2 resided in a region belonging to the checkerboard.

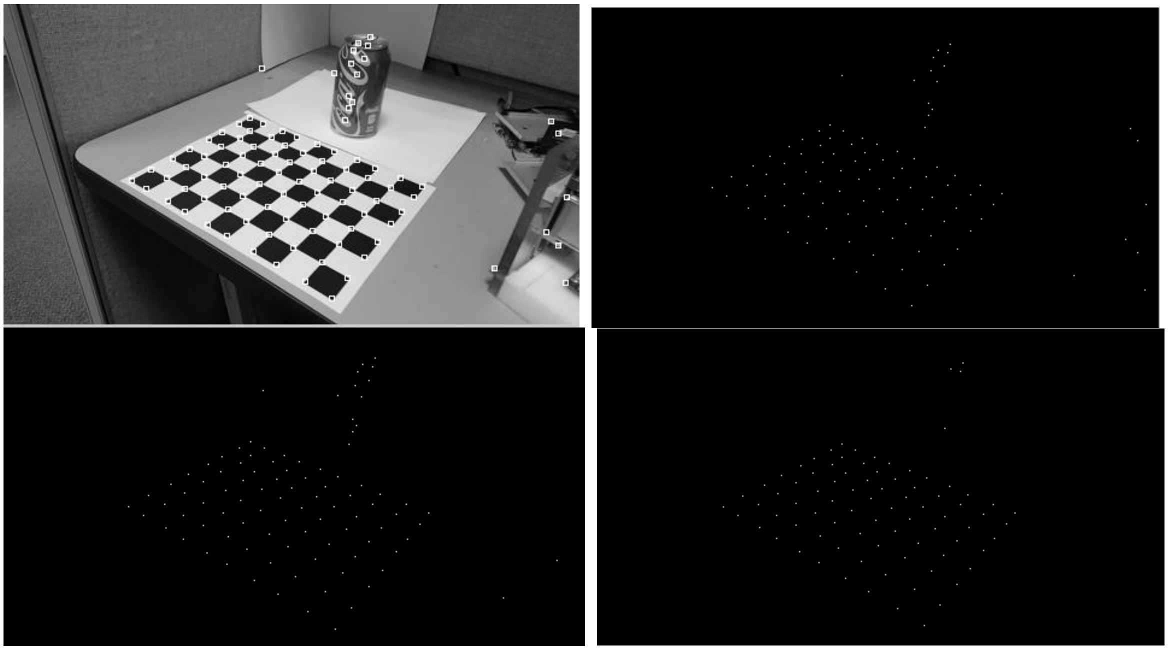

Figure-2: The **top left** image illustrates the original image with the corners produced by the Harris corner detector overlaid. The **top right** image illustrates the binary image of the corners. The **bottom left** image illustrates the image of the corners after 5% of the corners have been removed via Thomson Tau. The **bottom right** image illustrates the corners in which RGB pixel value seem most likely to makeup the checkerboard. These are the corners that passed the initial screening process.

Step 3: Outlier Detection using RGB Image

Using the RGB image, a histogram was created and pixels closest to the colors comprising the checkerboard were considered to pass screening. Almost always, ignoring the case of non-uniform illumination, a checkerboard is comprised of two colors, usually black and white. In an image of a black and white checkerboard, the values of RGB for the pixels that comprise the checkerboard pattern should be close to the same value (i.e. very close to black or very close to white). Following this line of reasoning, the standard deviation of the RGB values for each pixel location returned by the Harris Corner detector should be close to zero for a black and white checkerboard. As the standard deviation between the RGB values increase, the pixel begins to take on a different color. Since the assumption of the project is that the checkerboard is black and white, any corners in a region that do not comprise the checkerboard can be eliminated using thresholding the standard deviation of RGB.

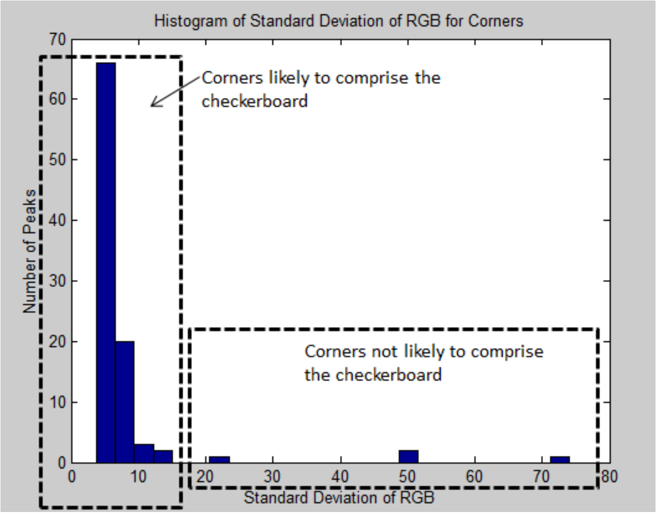

Figure-3: The image illustrates the histogram of the standard deviation of the RGB pixel locations of the corners returned by the Harris corner detector for a black and white checkerboard..

For a black and white checkerboard, it is most likely the standard deviation of the RGB values of the corners comprising the checkerboard will be close to zero. Figure-3 illustrates the difference in standard deviation between the RGB values of pixel locations classified as corners by the Harris corner detector.

Checkerboard Isolation

Step 1: Dilation

Since the corners that comprise the checkerboard are in close proximity to one another, the spatial information for corners could be used to detect the checkerboard. However, a centroid method, that is attempting to predict the center of the checkerboard and classify points based on distance from the center, proved to not be the best method because it failed when the checkerboard was oriented in a fashion such that corners making up the checkerboard were considered outliers.

An iterative dilatation method was adopted. The thought process behind a dilation/region growing method is: spurious corners become penalized during the region-growing process as they will not grow to combine with nearby corners (like those making up the checkerboard). The corners comprising the checkerboard should dilate into a large region while the other regions remain much smaller.

The dilation process continues until five regions remain. At this point, the area of the regions is calculated during each iteration until the ratio of the area of the largest region to the sum of the area of all other regions is greater than some threshold. At this point, the smallest regions are automatically removed.

The rate of dilation is a function of the number of regions present within the image. As the number of regions is reduced, the dilation rate is reduced. The reasoning behind this is: when the checkerboard is at an angle, the corners of the squares at the far corners of the checkerboard are at different distances from one another. In order to prevent the region from growing from becoming too aggressive, the rate was reduced to accommodate for warping of the checkerboard at different vantage points. The dilation rate is defined by the radius of the structuring element. The structuring element was chosen to be a disk since a disk grows at the same rate in all directions during dilation.

Again, the dilation continues until the ratio of the area of the largest region to the sum of the area of all other regions is greater than some threshold T. For this project T is 4.5. The idea behind this is that the checkerboard size could be different at various vantage points. The metric for detecting the checkerboard mask needed to be robust to changes in vantage points. In this case of this project the ratio of the area of the largest region to the sum of the area of all other regions proved to be robust enough. Finally, all the smaller regions are eliminated and the largest region serves as a mask that is logically and with the corners that passed the initial screening.

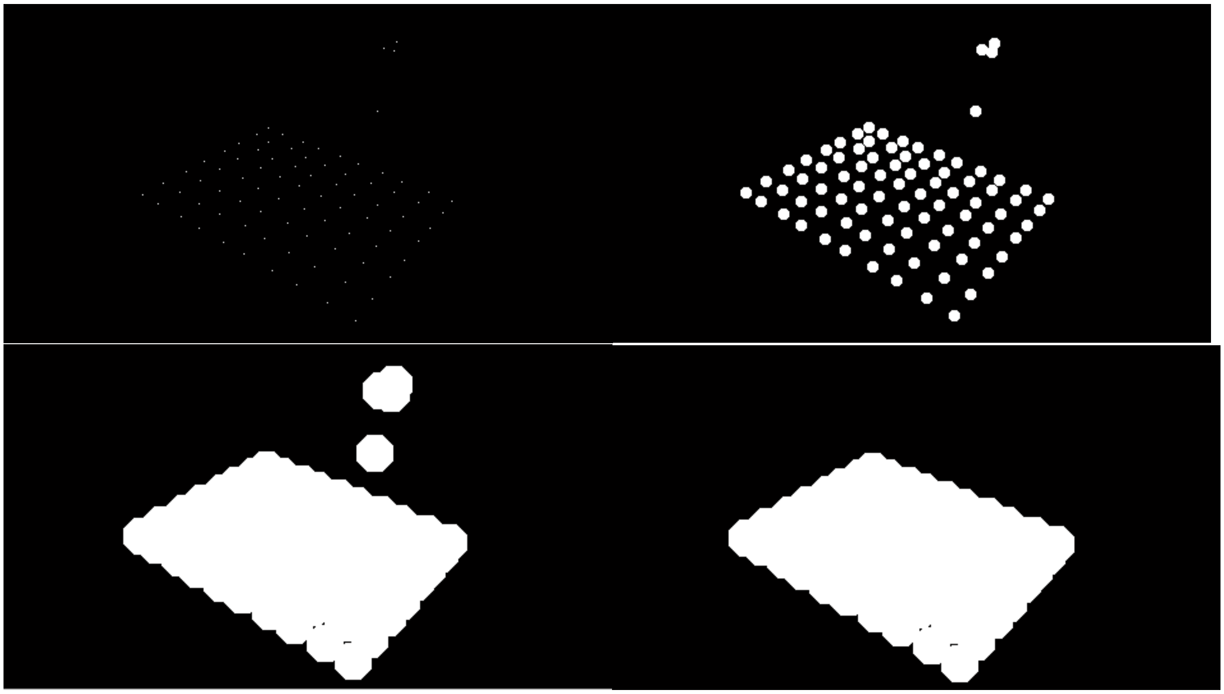

Figure-4: The **top left** image illustrates a binary image of the corners returned by the Harris corner detector. The **top right** image illustrates the dilated image of corners after a single iteration. The **bottom left** image illustrates the image of the corners after N iterations in which the area of the largest region is at least 4.5 times greater than the sum of the area of all other regions. The **bottom right** image illustrates the region to be used as a mask to isolate the corners comprising the checkerboard.

Step 2: Checkerboard Detection

The mask obtained from the previous step is now logically anded with the corners that pass initial screening. The result is a binary image of corners that comprise the checkerboard. This image is then dilated by a disk (size 1). The convex hull is then found. This convex hull is assumed to cover the entire checkerboard.

The convex hull is then eroded, depending on the average size of the square. Then the eroded mask is logically anded with the corners that passed the initial screening process. The eroding of the convex hull allows the corners at the edges of the checkerboard to be masked out. The resulting image is the inner corners of the checkerboard. These are the corners required as inputs to the Camera Calibration Toolbox.

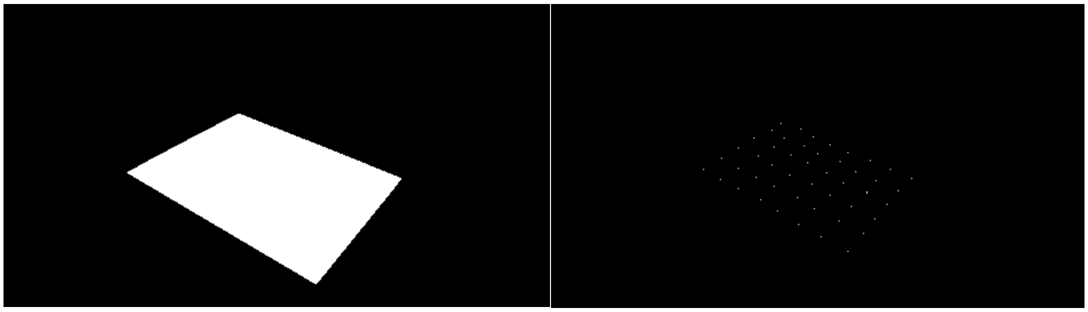

Figure-5: The **left** image illustrates the convex hull of the points masked by the process illustrated in Figure-5. When one erodes the convex hull and performs a logical and with the corners returned by the Harris corner detector, the inner corners of the checkerboard are extracted. The **right** illustrates the result of extracting the inner corners of the checkerboard.

Results/Testing

The code was integrated into Camera Calibration Toolbox for Matlab by Bouguet. For this toolbox, the goal is to extract grid corners for a group of images without the need to manually inform the tool on the whereabouts of the corners, more specifically, the origin of the checkerboard. The tool arbitrary chooses an origin and keeps state regarding the position of the origin frame to frame.

When calibrating using the Camera Calibration Toolbox there is a manual step in which the toolbox expects user input. The toolbox expects the user to click on the four extreme corners of the inner checkerboard as the tool queries the user for input: “Click on the four extreme corners on the rectangular checkerboard pattern. The clicking locations are shown on the four following figures (WARNING: try to click accurately on the four corners, at most 5 pixels away from the corners. Otherwise some of the corners might be missed by the detector).” [5]

The idea is: rather than manually selecting the extreme corners, the code developed in this project can be integrated into the Camera Calibration Toolbox to automatically perform the extreme corner selection process. The trick with this process is that the automation tool needs to keep track of the origin. In this case, the code arbitrarily selects an origin in the first frame and keeps state. The assumption is that the checkerboard will not be rotated more than 90 degrees between frames. Using this assumption the origin is tracked by finding the corner that is nearest to the origin of the last frame. The results can be viewed in Figure-6.

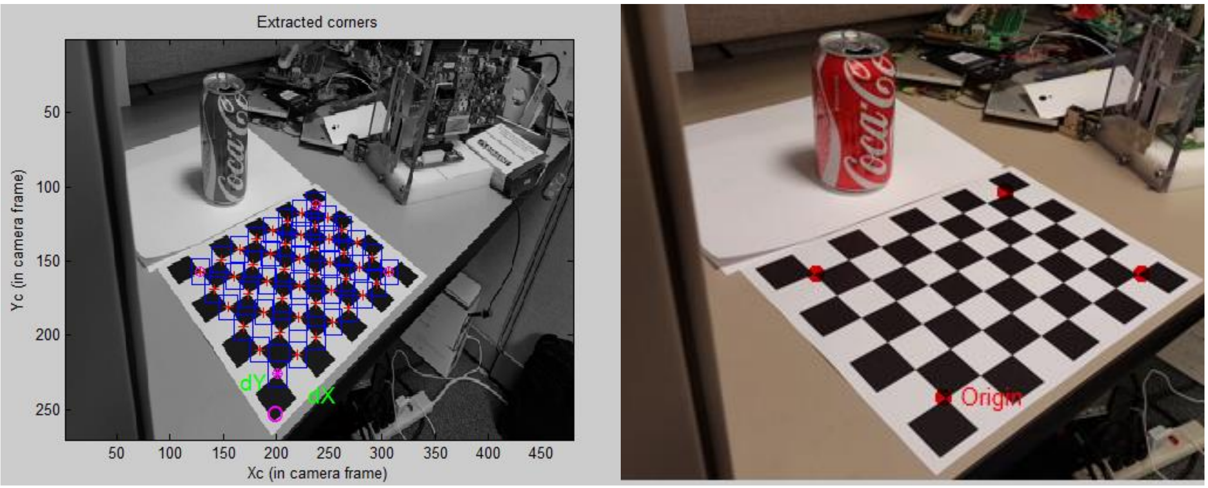

Figure-6: The **left** image illustrates the Camera Calibration utility determining the corners of the checkerboard with aid from the corner extraction code developed for this project. Note: the origin (indicated by O) in the image. The image to the **right** illustrates the color image with the extreme corners and origin labeled in red.

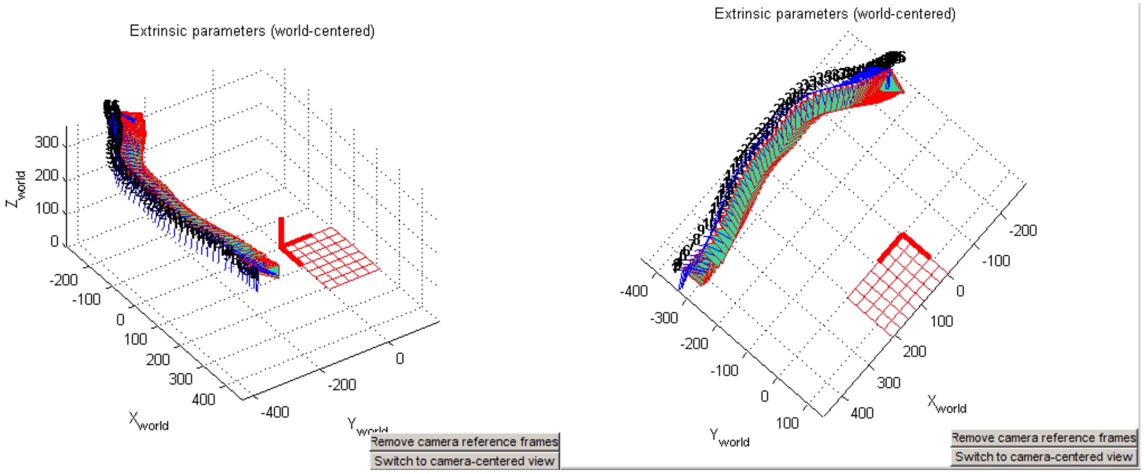

Figure-7: Using the extracted extreme edge corners from the code developed in this project, the calibration utility can show the path of the video camera as the camera is moved around the checkerboard.

Conclusion

In conclusion, relatively simple processing can be used to automate the manual extreme corner extraction needed by the Camera Calibration Toolbox. The key issues to be aware of are: many of the assumption used by this tool require uniform lighting. In the case of calibrating a camera, lighting conditions are usually controlled. Furthermore, the corner extraction utility does not deal well with extreme distortion.

Sources

[1] http://www.vision.caltech.edu/bouguetj/calib_doc/

[2] Automatic Chessboard Detection for Intrinsic and Extrinsic Camera Parameter Calibration. Arturo de la Escalera, Jose María Armingol. *Sensors* 2010, *10*, 2027-2044; doi:10.3390/s100302027

[3] ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration. Simon Placht, Peter Fursattel, Etienne Assoumou Mengue,Hannes Hofmann, Christian Schaller, Michael Balda, Elli Angelopoulou

[4] Automatic Detection of Checkerboards on Blurred and Distorted Images. Martin Rufli, Davide Scaramuzza, and Roland Siegwart. Autonomous System Lab, ETH Zurich, Switzerland.

[5] http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html