Language

- C++ 11

Dependencies

- Opencv version 4.x
- Cmake version >3.15

Build and install

1. cd into the source directory
2. Create build folder and cd into it
3. Run ‘cmake ../’
4. Run ‘cmake --build .’
5. Run ‘cmake -- install .’

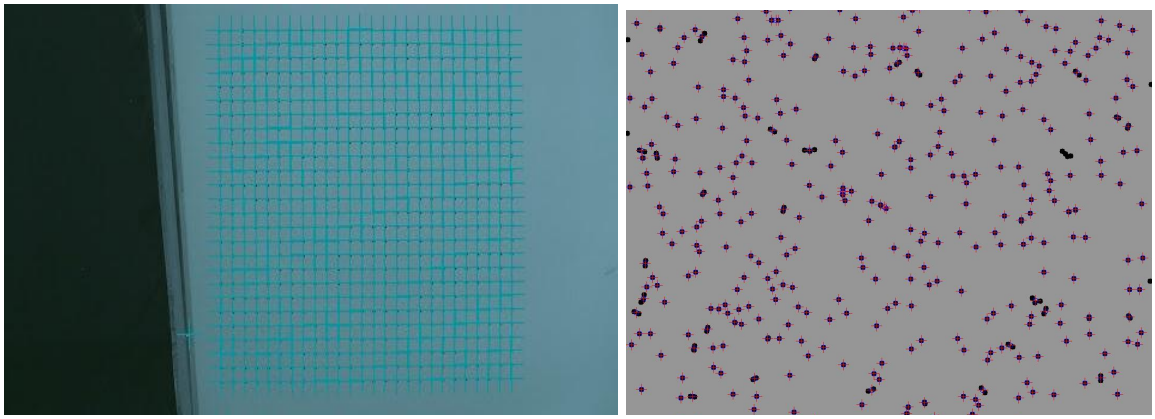
Usage

LocateHoles --test -n <number of holes>
LocateHoles -f <file> -a <required accuracy> -w <width of hole>

Algorithm

Image to be processed is loaded and initial filters are applied to remove some white gaussian noise. Thresholding and Gaussian blurring are applied in this regard. The Canny edge detection procedure is called on the blurred image to detect all available blobs. This is followed by contour generation. The generated contours are checked against circularity and size shape criterion to filter out non circular contours. Finally centroids of the remaining circular contours are calculated using their moments.

To achieve sub-pixel accuracy, opencv’s getRectSubPix function is used on sections of the centroid pixel identified in the previous steps. The number of these sections is determined by the parameter d, which is input as a command line sub-pixel accuracy option, -a. The centroid pixel is divided into grids with side length equal to this parameter. The intensities of these sections are calculated iteratively and the maximum is taken as the new centroid of the contour. This is done for all contours. The result is shown in the following image.



Known Issues

The provided solution has a number of issues. The following are but some of the notable ones.

1. The algorithm fails to detect any of the holes when a moderate gaussian, pepper and salt noise is introduced
2. Filters do not employ dynamic thresholding
3. Sub-pixel detection algorithm is very inefficient, O(n^3)
4. Detection rate and accuracy are not satisfactory (for instance about 450/500 dots detected)
5. A number of user requirements are not met (Python (numpy) interoperability, mm part dimension,)