Documentation

June 12, 2016

1 Algorithm

Firstly, the image is processed with the Canny edge detector after histogram equalization. Then, the idea of the algorithm is to minimize the function which is defined along the approximiate goal post position. The domain of the function is the sample points of the goal position, the function return the area of white pixels around the goal post. See Figure 1.

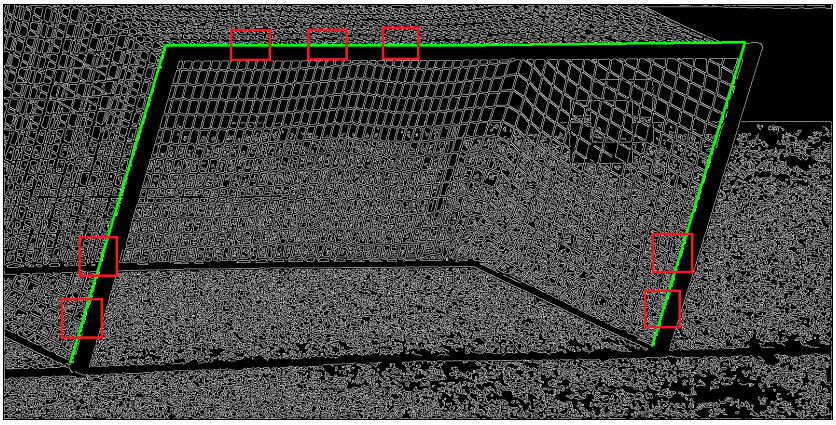

Figure 1: image0001.png. Green line - goal post position, red squares - defines the region, where the area of white pixels will be computed .

Sample points are distrubuted equally along the post, Figure 1 shows only an example of a few sample points. So, the idea is to minimize the area of white pixels along the goal post. The minimizer (CMA-ES was used in my case), will naturally move the position towards the space where less white pixels along the post are present, i.e. to the needed position. See Figure 2.

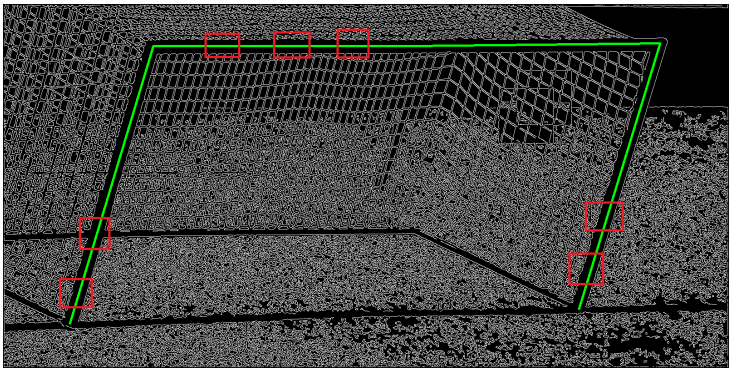

Figure 2: After finding global minimum, the goal post is found.

The function also penalizes trivial solutions, i.e. if the area of the red square is less than some epsilon. We penalize such case, so that the goal post won't move to some wrong direction.

2 Finding best parameters

In the TestingSrc.zip located the source code used for testing the quality of the algorithm and finding the best parameters. We define 3 parameters which are required for the algorithm: numberOfInnerSqaures, LineStep, SquareSize.

numberOfInnerSqaures - is the parameter which defines how many additional squares are inside per one square, which are smaller than the original. The idea was to see if that would benefit in better results. But, it turned out that 1 sqaure is enough, i.e. numberOfInnerSqaures = 1.

LineStep - is the sample step along the goal post for our function.

SquareSize - the size of the square.

I found that numberOfInnerSqaures =1, LineStep = 10, SquareSize = 30.0f; are the most stable parameters, with MSE $\approx$ 18.0.

However, with numberOfInnerSqaures =1, LineStep = 10, SquareSize = 20.0f; MSE $\approx$ 11.0, but it fails to find the correct position for 0019 image sometimes. But, with SquareSize = 30.0f it produces robust results for all images.

3 Results

Some results of the algorithm.

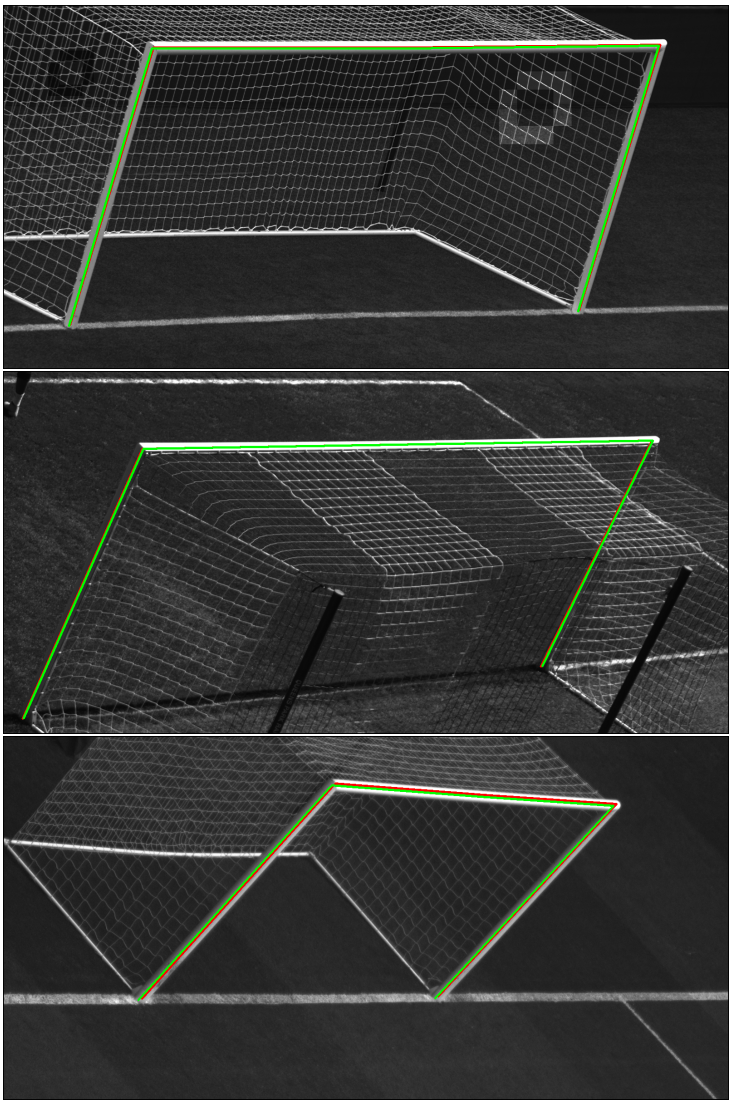

Figure 3: Green line - exact goal post position, red line - algorithm computed position.