Football detection on Aldebaran Nao

Kert Pjatkin
Institute of Computer Science
University of Tartu
b15275@ut.ee

Abstract—RoboCup is a worldwide movement that has a clear goal set: the target is to be able to win a FIFA World Cup winner by 2050. In order to be able to fulfill the task, the RoboCup Humanoid Soccer World Cup is organized every year. The organizers make the task of playing football more difficult year-by-year. During the last few years, the goals changed from blue and yellow to white. Considering that the robots from both competing teams are also white, the change made the detection of goals more difficult, as the robots had to make distinction between their, their opponent’s goal and the robots. The most important change in the rules of this year’s competition is the change of traditional orange ball, used in different robotics competition, to a ball that is more like a real football ball. As the goals, the robots and the lines are already white, the decision adds a layer of complexity to the robot. The main target of the paper is to research and implement the white ball detection in different situations on a traditional Aldebaran Nao.

Index Terms—RoboCup, Aldebaran Nao,

I. INTRODUCTION

RoboCup is a worldwide movement that tries to introduce the problems of playing football into the world of robotics. Their target is to beat the FIFA world champion by 2050. This is a really difficult task considering the level of football played in the real world. In order to move to the target, the rules of the competition are getting tougher and tougher every year. The most significant change in 2016 is the change of orange ball to a more traditional white ball with black pentagons. Figure 1 depicts the difference.

Fig. 1. Orange and white ball

The main objective of the paper is to develop an algorithm that is able to successfully develop an algorithm that is able to detect a traditional football ball real time on Aldebaran Nao.

II. ALDEBARAN NAO

Aldebaran Nao is a humanoid robot produced by a French company Aldebaran Robotics. It is used for several purposes: for example teaching the autists or doing research. However, the platform has gained the biggest recognition from RoboCup SPL. It has been as the standard hardware platform in the competition from year 2007. The technical details of Aldebaran Nao can be found from Table 1.

<table>
<thead>
<tr>
<th>TABLE I ALDEBARAN NAO TECHNICAL SPECS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Intel Atom 1.6GHz</td>
</tr>
<tr>
<td>RAM 1GB</td>
</tr>
<tr>
<td>Storage 2GB &amp; supports Micro SDHC</td>
</tr>
<tr>
<td>Operating System NAOQI</td>
</tr>
<tr>
<td>Camera 1 x 640x480 &amp; 1 x 1280x960</td>
</tr>
</tbody>
</table>

III. ROBOCUP SPL

RoboCup is an international robotics competition that first took place in 1997. University of Tartu has been taking part of the competition since 2014, achieving 11th place last year. The idea of the competition is to play 5 vs 5 football between two teams on a green carpet of size 6m X 9m. The team that scores more goals during 2 10-minutes period is the winner.

IV. ORANGE BALL DETECTION

The current codebase of team Philosopher is adapted from the RoboCup team of the University of New South Wales in Australia. The ball detection consists of several steps that are described in the sections below.

A. Locating ball candidates

The core of the algorithm is to get the candidates right. Removing the balls and accepting non-balls as an input means that from one side the next steps have no chance to recognize the real ball and on the other hand waste a large amount of the limited resources on trying to find the ball where it is not in reality.

The algorithm starts by generating horizontal and vertical histograms of orange color. After the generation, the peaks are regarded as the edges of a possible ball.

B. Foveating In

After having detected the possible areas of ball, the algorithm tries to normalize the possible area of the ball to be from 6-12 pixels and the edge intensities have to have the peaks around the ball. Hence, the first task is to fix a sampling rate to be able to fit the ball radius to the limit. The second step is to generate a fovea of the edges taken from the blue channel,
because the research done by the team showed that the blue channel depicts the difference between orange ball and the other objects the best.[4]

C. Locating edge points

The task of this step is to find 16 edge points of the object in the fovea. Hence, the team has implemented an algorithm that starts from the center of the fovea and scans the pixels in a specific direction until and edge is located. This kind of strategy makes sure that the detected edge points are equally distributed over the circle. The process is depicted on figure 2. [4]

D. RANSAC

RANSAC (Random Sample Consensus) is an algorithm for parameter estimation introduced by Fischler and Bolles. The general steps are the following:

1) pick some random points. In the case of the algorithm introduced by UNSW, it should be 3.
2) A model is created on the 3 points picked previously.
3) The algorithm tries to fit the other points to the predefined model.
4) If there are enough points that fit to the model, then the search is successful.

[5] [4]

V. WHITE BALL DETECTION

As mentioned previously, the task of this paper is to implement an algorithm that is able to detect a traditional soccer ball. As the method described in the previous chapter, worked quite well, we decided that the basis of the algorithm will not be removed in the first place. Instead, we would try to fit the system as much as possible.

A. Candidate detection

The first method that we tried, generated edge histograms in both horizontal and vertical directions. The next step was similar to the previous one: find the peaks in the histogram and draw a bounding box around it. The strategy is really simplistic and enables us to detect the ball perfectly as Figure 9 indicates. However, the issues arise when the robot sees a lot of noise and as figure 10 indicate, the detection method automatically tends to fail. Hence, there is a need to enhance the algorithm such that it wouldn’t be susceptible to the noise. In addition, we also have to take the performance issue into account, because it is possible to write all sorts of different algorithms to detect the possible ball candidates, but because they consume a lot of resources it is not possible to run the real time on a robot that is as powerful as a standard smartphone. In addition, it is also really important to use as less color information as possible, because it leaves us a possibility to drop the colour configuration part. Different venues with different lightning conditions tend to require different settings. Hence, the traditional colour configuration tends to add a layer of complexity to the process of setting up the robots for the competition.

A possible solution that fulfils the previous requirements would be to take the black and white image of the image of size 60 x 80 pixels and try to see whether the bright and dark spots of a ball form a distinctive pattern. The process iterates over both the horizontal and vertical direction of the image and for every row and column the algorithm marks down the minimum and maximum pixels values. Figures 3 and 4 depict the histograms generated based on Figure 5. Histograms depicted on figures 6 and 7 are generated based on figure 8. The green bars on the histograms depict the white values and the red ones the dark values. As it can be seen from all 4 figures, that the ball seems to have a rather distinctive pattern on the histogram. This means that, when encountering a ball, there will be some darker values (i.e. the minimum value is smaller than usual) and some brighter values (i.e. the maximum value is larger than usual). Hence, our team created an hypothesis that processing the histograms in such a way is able to return the bounding box of a potential ball. The algorithm that tests the hypothesis works in the following way:

1) Iterate over the horizontal black histogram.
2) If the bin in the horizontal black histogram is smaller than 1.3*median of the same histogram, then remember the index.
3) Check the horizontal white histogram at the same index and see if it is more than the predefined limit.
4) If it is, then remember the index.
5) Merge the remembered neighbouring indexes into one region.
6) Repeat the previous steps for vertical histograms.
7) Combine the horizontal and vertical regions. One of the combinations has to be a ball.

Even though the algorithm might also generate some false positives, it is not regarded as a fatal flaw of the algorithm, as
all the bounding boxes are later processed and as it is explained in the coming chapters, detecting whether a bounding box has a ball in it or not, works well. Figure 5 shows an interpolated image without noise with one of the bounding boxes. Figure 8 shows an interpolated image with noise and one of the bounding boxes. As it can be seen from the figures, the algorithm does detect the possible bounding boxes of a ball in both noiseless and noisy environments. Hence, we can conclude that the main target of finding the correct bounding boxes in a noisy environment is achieved and the bounding boxes can be passed to the other components of the software described in the coming paragraphs.

**B. Edge detection**

The process of edge detection is the opposite of the previous description such that it will start from the edges of the bounding box and upon encountering an edge, the algorithm adds it to an array. The reason, why we changed the concept, was that the new ball has the pentagons in it and hence, there will be edges also in the ball as well.

**C. RANSAC**

The last step is to feed the edge points into RANSAC, as mentioned the previous chapter. Figure 9 displays a successful output of RANSAC, where the circle is detected inside a bounding box.

**VI. CONCLUSION**

The main target of the paper is to introduce an algorithm that is able to detect a traditional football ball on Aldebaran Nao real time. Currently we are able to detect the balls, but there is a need to test the algorithm in various conditions in order to fix the possible bugs that might affect the performance during the competition.
Fig. 6. Horizontal histogram without noise

Fig. 7. Vertical histogram with noise

Fig. 8. Interpolated 60x80 black-and-white image with noise

Fig. 9. Ball detected successfully

Fig. 10. Ball not detected

REFERENCES


