# Adaptive Field Detection and Localization in Robot Soccer

Yongbo Qian and Daniel D. Lee

GRASP Lab, University of Pennsylvania, Philadelphia PA 19104, USA, {yongbo,ddlee}@seas.upenn.edu https://www.grasp.upenn.edu/

**Abstract.** Major rule updates for the RoboCup Standard Platform League (SPL) in recent years pose significant perception challenges for recognizing objects with similar color. Despite the frequent color changes to goalpost, soccer ball and jerseys, the soccer field itself remains unaffected, which makes green the only reliable color feature that can be exploited. In this paper, we propose an efficient approach for adaptive soccer field detection model utilizing NAO's two-camera system. Building upon real-time image histogram analysis between top and bottom camera frames, the field color classifier is robust under inconsistent lighting conditions, and can be further processed to generate field boundaries. This approach could also be useful for other object detection modules and robot's self-localization.

**Keywords:** image histogram, color segmentation, boundary detection, natural lighting, self-localization

# 1 Introduction

The RoboCup soccer competition requires a robust and efficient perception system which should achieve accurate object detection and self-localization in real time. However, due to the constrained nature of the computation on mobile robots, vision has to run fast enough to provide real-time information while leaving processing resources available to other autonomy algorithms. Moreover, variable lighting conditions, frequent object occlusion and noise makes the problem even more challenging.

Similar to many other teams in the league, our team, the UPennalizers[1], used to handle this problem using a manually defined color look-up table[2]. This method is well suited and efficient for image segmentation when object has its unique color. Since everything except the field has been changed to white, color-based segmentation becomes less effective to distinguish between different objects. Furthermore, we are interested in having the robot eventually play soccer outside with humans under natural lighting. This motivates us to develop a new robust perception framework because the pre-defined color table lacks the robustness when illumination varies.

# 2 Y. Qian and D D. Lee

As green becomes the only unique color cone we can leverage, wisely using it to detect static field becomes an important first step in perception, because it could provide contextual information for other objects. For example, performing ball and line detection only on the field could not only reduce the search region, but also yield to more accurate result. Similarly, goalpost detection could be more precise if the algorithm uses the fact that posts grow vertically onward the field boundary.

Using the field region to aid the detection of other objects is not a novel task in RoboCup. However, most teams generate field boundary with the prelabeled green color. It is among our best interest to investigate real-time green feature analysis so that it could adapt to lighting changes. Berlin United 2015[3] estimated the field color as a cubic area in the YUV color space, based on the assumption that both top and bottom images are mostly covered by the field. HTWK 2014[4] detect field color through a peak detection in a 2D CbCr-Color-Histogram together with a fast search and region growing approach. This is also based on the same assumption which is not necessarily true all the time. To get rid of this assumption, HTWK 2015[5] trained 300 different images from SPL events to extract the field color.

The teams from other leagues were also working towards illumination invariance techniques in RoboCup soccer. In the middle size league, Mayer et. al. [6] suggested to use automated on-site vision calibration routines, along with improved color constancy algorithms and additional visual features aside of color to tackle the lighting variations. Sridharan [7] from the UT Austin Villa team proposed a color constancy method on Sony Aibo robots by comparing the color space distributions with the color cubes in the training samples using the KLdivergence measure.

Unlike the previous approaches, we present a simple yet efficient method in this work. This method considers a stronger assumption and does not require training images. As shown in Figure 1, the two-camera system for NAO robot has its unique advantage allowing each camera performs different task. For example, as noticed in the previous games, the coverage of the bottom camera is always within the field because of the lower pitch angles; Therefore, we only need to implement field detection in the top camera image. This gives us the opportunity to pre-analyze the color distribution of bottom camera frames to find out the statistics information of green pixels and apply it to the top image. Now the task can be simply achieved by basic histogram smoothing, peak picking [8] and back projection techniques [9].



**Fig. 1.** Camera model for Alderbaran NAO robots [10]

To complete the field detection module in our new perception framework, we then analyze the linear relations of the edge points to generate the field boundary. The boundary will re-define the search area for the detection of each object in top image, and also serves as a landmark feature in our particle filter to correct robot's orientation in localization. The object detection will still be performed in the whole bottom image. The detailed object detection mechanism is not within the scope of this paper, but will be mostly based on the edge and shape features of the non-green pixels on the field. The overall of this framework is shown in Figure 2.



Fig. 2. Overview of the field detection module in perception framework

# 2 Field Color Detection

This section provides an overview of the field color detection method proposed in the perception framework. The approach utilizes the prior analysis of the color histogram of the robot foot area to segment the soccer field in top camera image using histogram back projection technique. Since the histogram analysis and projection is performed in real time, this method could adapt to the lighting changes during the game.

## 2.1 Robot Foot Area Projection

In any soccer games, players should stay on the field if not being penalized. Same rule also applies to robot soccer, meaning robot's feet should always be surrounded by the field, which is green carpet in this case. This provides a valid assumption which makes the foot area as our region of interest:

Assumption: the majority color within robot's foot area is green.

The correctness of this assumption depends on the definition of the foot area. It holds true for most of the positions on the field; however, when robot approaches to the field line, a large portion of white will also appear in the image, as shown in the Figure 3 left. Assuming the resolution of the image  $I_{btm}$  from bottom camera is  $w \times h$ , based on camera's projective transformation[11], the

width of line (around 5 cm) should not occupy more than  $15\% \times h$  pixels. Here the robot foot area *ROI* is defined in  $I_{btm}$  as follows:

$$I_{btm(i,j)} \in ROI, \text{ if } 0.625h \leqslant i \leqslant h, \text{ and } 0.125w \leqslant j \leqslant 0.875w \tag{1}$$

The equation basically chooses the bottom 37.5% region of the image from bottom camera to be the foot area. This percentage is estimated by the logging data of a mock game. Larger values may include more white components, mostly ball and other robots' feet, while smaller foot area may be completely occupied by field lines, both risk holding against the assumption.

In addition, robot's head angle is constantly changing in order to keep track of the ball during the game. The defined ROI in the image may be projected to the ball, robot's own feet and jersey when kicking the ball (Figure 3 middle), or robot's shoulder when head yaw value increases. Therefore, to complete the definition of foot area, the ROI will be discarded if when head pitch  $\theta < -24.6^{\circ}$ or head yaw  $|\psi| > 65^{\circ}$  in the corresponding frame. All the other cases are safe since robot will adjust its head angles to align the ball in the center of the image  $I_{btm}$  when the ball is getting close, so that the ball will not appear in the defined foot area if  $\theta > -24.6^{\circ}$ .



Fig. 3. Left: the field of view of bottom camera when  $\theta = 0$ ,  $\psi = 0$ ; and the defined foot area *ROI* in  $I_{btm}$ . Middle: when  $\theta = -24.6^{\circ}$ , *ROI* need to be discarded since field green may not be the majority color. Right: when  $|\psi| = 65^{\circ}$ , *ROI* need to be discarded since robot's shoulder occupies the most of the region.

## 2.2 Histogram Analysis

The green color of the soccer field may ideally have unique distribution over the G channel of the RGB color space. However, the field itself may not be lit evenly, for instance if lit by different spotlights or natural lighting, it could cause inconsistent apparent green across the field. Therefore, using an illumination invariant color space is important in order to eliminate the effect of varying green intensities. Here the RGB color space is transformed into normalized chromaticity

coordinates, such that g chromaticity is:

$$g = \frac{G}{R+G+B} \tag{2}$$

The 1D histogram of the g chromaticity space will be used to extract the field feature from robot foot area ROI. Specifically, g can be quantized into n bins and the histogram bin set can be expressed as Bin = [1, 2, ..., n]. Each bin has a number of  $h_b$  pixels, where  $b \in Bin$ .

If the assumption that the majority color in the ROI is green stays true, the histogram should have a peak value  $h_{b,max}$  which indicates that bin b consists of most pixels of field green. In order to further solidify the assumption, the histograms of five previous ROIs are combined. The values for the same bin can be simply added together. This essentially extends the ROI over the frames to minimize the non-green pixel distribution.

Note that five previous ROIs are not equivalent to five previous frames, since some frames might not pass the head angle check to have the valid ROI. Also, the images of ROIs for the previous frames will not be stored; instead, only the histograms of valid ROIs will be saved in a queue of size 5 for future processing. In this way, the algorithm can still run in a fast and efficient manner.

The histogram normalization is then performed on the new combined histogram  $H_b$ . This will obtain the probability that a pixel's g value is in bin b, given it is a pixel in the extended ROI. This distribution can be abbreviated as  $P_b$ :

$$P_b = \frac{H_b}{\sum_{1}^{n} H_b} \tag{3}$$

The bottom left histogram in Figure 4 shows this probability distribution. Here n is set to be 32 in order to simplify the model. However, this histogram needs to be further processed to be representative for the whole field. First, a high pass filter, shown as the horizontal yellow line, is used to filter out the bins with low probability. The bins are discarded if the value is less than 30% of the peak value. The vertical yellow line is to remove the local maxima in the distribution. Based on the results of the proposed histogram model, the bins represent green pixels should be consecutive and the distribution of their values should be unimodal for global maxima, any local maxima is most likely to be another color.

$$P_b = 0 \quad if P_b < 0.3 P_{b,max} \tag{4}$$

$$P_b = 0 \quad if \ P_b \neq P_{b,max} \quad and \quad P_{b-1} < P_b, P_{b+1} < P_b$$
(5)

After post-processing with filtering and thresholding techniques, the new histogram  $\bar{P}_b$  with fewer bins (Figure 4 bottom middle) becomes a valid model to represent field feature, since it should only contain the pixel information of field green.



**Fig. 4.** Top: five consecutive valid ROIs in queue. Bottom Left: combined histogram for each ROI on top, further thresholding and filtering by yellow lines. Bottom Middle: histogram model for field feature. Bottom right: green classification on  $I_{top}$ .

## 2.3 Green Classification

The histogram model  $\bar{P}_b$  will then be used to find the green pixels in the image  $I_{top}$  from top camera. Same quantization technique for the color space is performed. Essentially, a 32 bin 1D histogram of the g chromaticity space is calculated on  $I_{top}$ .

Here, a fast and efficient binary green classifier is more desired then the probabilistic green model, so the back projection process can be simplified. For all the non-zero bin b in  $\bar{P}_b$ , the pixels on  $I_{top}$  which are also in bin b are green. Bottom right plot in Figure 4 masked the pixels classified as green on  $I_{top}$ . It is acceptable that the classification result is not completely precise, as long as it does not affect the formation of the field boundary.

Note that in this task, the parameters of both cameras need to be set the same so that the green pixels of two images could generally match. Although occasionally there were small inconsistencies between the cameras, but those slight differences did not affect the classification results in the experiments we performed.

As shown in Figure 5, this field color detector approach is robust under inconsistent lighting conditions between multiple images. We also tested the algorithm under the natural light, and since the normalized g chromaticity channel is illumination invariant, the pixel values for green carpet under shadow and sunshine are similar; therefore, the method still works fairly well as shown in Figure 6.

## 2.4 Experiments & Results

For the purpose of evaluating the perception rate and quality, different log files were recorded. Each log files contain 100 frames and was created when the basic robot behavior was performed in our lab environment. In order to simulate different lighting conditions, two different scenarios created, as seen in Figure 5 (left). For comparison, the same set of log files was also evaluated upon other



Fig. 5. When lighting changes during the game (top and bottom left), the peak value for histogram has a one-bin shift (top and bottom middle), and the green classification results are as expected (top and bottom right).



Fig. 6. The top and bottom scenes show that this field color detection approach works with inconsistant lighting conditions within the field, specifically under natural lighting which could cast shadows.

two methods: using the G channel in RGB (unnormalized G) as color space; and our traditional color look-up table method using Gaussian Mixture Model for color segmentation. The camera parameters and configuration of robot are set the same during the comparison.

One visualization example of those three methods can be seen in Figure 7. It shows the green classification results for each method under two lighting conditions. Since green is classified in real time from the true green color around robot's foot area, both our proposed method (left) and the method using unnormalized G channel (middle) provide consistent results in various illuminations; however, without using g chromaticity space, the classification cannot handle the inconsistent light within the field as well as the greenish background.



Fig. 7. The comparison of green classification results for three different methods on  $I_{top}$ , evaluated upon both dark (top) and bright (bottom) scene.

The traditional colortable-based method works well if the lighting condition does not change after the color was manually labeled (Figure 7 top right). Given static nature of pre-defined color-table, it cannot work when lighting changed (bottom right). In that condition, Our method significantly out-performs the colortable based techniques.

The quantitative results are summarized in Table 1. The true positive rate is calculated from the percentage of correctly classified green pixels in the total green pixels for all the logging data, while the false positive rate is the percentage of incorrectly classified green pixels in total non-green pixels. Note that the log images were down-sampled to simplify the process of manually selecting and labeling the green area to obtain the ground truth. The results clearly show the necessity of normalizing G channel in color space, and the advantage of our proposed method over the color-table method.

# 3 Field Boundary Detection

Knowing the field boundaries helps robot limiting the search region for the objects of interest, which lead to improved detection speed and accuracy. The field

Perception Rate	Proposed method	Unnormalized G	Color- table
True Positive Rate	87.6%	68.1%	70.4%
False Positive Rate	1.4%	27.2%	7.5%

Table 1. Field color detection rate of the proposed method compared to other two approaches.

boundary detection algorithm utilizes the field color classifier from the previous section to analyze the vertical scanlines and search linear relation of the green to non-green class transitions. HTWK [4] used RANSAC [13] algorithm to match the model of two straight lines, while B-Human 2013 [12] estimated the boundary by successively calculating the convex hull [14]. Our method here is a hybrid model which combines the advantages of both techniques.

# 3.1 Field Boundary Points

The first step of field boundary detection is to search for the possible points on the field edge. Since the valid field boundary points should always be below robot's horizon, we calculate the horizon line through robot's head angle transformation and top camera's projection. A topdown approach from the horizon to the bottom of the image is then adopted to build a score  $S_{i,j}$  for each pixel *i* on the corresponding vertical scanline j. The policy is as



Fig. 8. Left: horizon line on the binary green classifier . Right: selected field boundary points on top camera image.

follows: the score is initialized to 0 on the horizon line. A reward is added to  $S_{i-1,j}$  for each  $S_{i,j}$ . If the Pixel<sub>i,j</sub> is classified as green, reward is set to be 1; otherwise reward is -1. Since the scan is downwards from non-field region to field, the pixel where the score is the lowest is then selected as the boundary point on that scanline.

The algorithm continues to perform on the next vertical scanline in the top camera image. Figure 8 left shows the horizon line on the binary green classifier. The selected field boundary points are marked as yellow spots in Figure 8 right.

## 3.2 Convex Hull Filtering

In most cases, the field boundary points extracted from minimum score do not show clear linear relations. There are lots of false positive spots either due to

#### 10 Y. Qian and D D. Lee

the inaccuracy of green classification; or the objects on the field such as robots and ball which occlude part of the boundary. Those points need to be filtered out for further processing.

A filtering technique utilizing convex hull is performed on the raw field boundary points. Since most of the false boundary points are from the objects on the field, which are below the actual boundary, we calculate the upper convex hull from the raw boundary point set. For points not on the upper hull, associate them with upper hull edge formed by their 2-nearest neighbor vertices. The point far away from its corresponding edge is then removed. Figure 9 left shows the upper hull of the raw points; figure 9 middle shows the filtered boundary points.

# 3.3 RANSAC Line Fitting

In order to represent boundary points as boundary lines, a variant of RANSAC algorithm is implemented to fit the best line first. The algorithm randomly chooses two points in the filtered boundary points to form a line and check the distance between all the points and that line. If the distance is below a certain threshold, the corresponding point can be considered as an inlier. This process runs iteratively to maximize the number of inliers and find the best fit line to be a field boundary.



**Fig. 9.** Field boundary detection for both single boundary line case (top) and two lines case(bottom). The detection follows the sequence of building upper convex hull (left), filtering raw boundary points (middle) and line fitting using RANSAC (right).

The boundary points are not fitted by the first line might either because of noise, or the existence of a second boundary line. Therefore, the decision needs to be made carefully whether the second round of line fitting should be performed. If the percentage of the points left is above certain threshold, and nearly all of those points are distributed on the same side of the point set used to fit the first line, they are less likely to be just noise. A second round of RANSAC with smaller distance threshold is then performed on those points to match a secondary boundary line. The final step is to remove the lines above the intersection of two boundary lines to form a concrete field boundary. Figure 9 right shows the accurate line fitting for both single and two field boundary line(s).

# 4 Localization

The field boundary could be added as another vision-based landmark feature for robot's self-localization. As proposed from Schulz and Behnke [15], the approach utilizing the structure of field boundary and lines could be quite useful in robot's self-localization.

Currently, our localization algorithm[1] utilizes 200 particles to estimate the position state of the robot. A particle filter is implemented to track the continuous changes on the position, orientation and weight of each particle. Field boundary information can be included in the measurement update phase to adjust the particle state. Since the boundaries can be detected from far away, the position calculation may have large variance; therefore, field boundary detection will only correct particles' orientations and weights. If two boundaries can be detected, there are only four hypothesis of robot's orientation. If only one boundary can be seen and robot cannot see goal post at the same time, it is fair to assume that robot is facing the sideline. Combining with the body yaw value in the motion update phase, field boundaries can be extremely useful in determining robot's orientation. Figure 10 shows how field boundaries correctly update the particles' orientations.



Fig. 10. Field boundary detection (top) and the corresponding positions and orientations of the particles (bottom). Using two boundary lines in the measurement update, and robot's body yaw in the motion update, particle filter tracks robot's direction (Left and Middle). Single boundary line could also serve as an additional landmark feature besides goal post to achieve more accurate localization (right).

# 5 Conclusion

We have presented an efficient approach for soccer field detection. Unlike other approaches that assume green is the majority color in both top and bottom im-

#### 12 Y. Qian and D D. Lee

ages, we decrease the assumption region to robot's foot area, and utilizing head angles and previous frames to enhance the appearance of the green pixels. We analyze the histogram of g chromaticity space to find the threshold for top camera green classification. The binary field color classifier is then used to generate the field boundary using convex hull filtering and RANSAC line-fitting algorithms. We also briefly described how field boundary could help robot's self-localization.

The results indicate that our approach works adaptively on the field under variable lighting conditions and dynamic environment. Although it has been tested in our lab, we expect to fully examine the new perception framework built upon this field detection approach in real game scenario in RoboCup U.S Open 2016, and have the new system ready for the outdoor games in RoboCup 2016.

## References

- Qian, Y., He, Y., Han, Q., Poudel, S., Small, A., Lee, K., Lee, D.: RoboCup 2015 Standard Platform League Team Description Paper. Technical report (2015)
- McGill, S., Yi, S., Zhang, Y., Lee, D.: Extensions of a RoboCup Soccer Software Framework. RoboCup 2013: Robot World Cup XVII, Volume 8371 of the series Lecture Notes in Computer Science pp 608-615 (2013)
- Mellmann, H., Krause, T., Ritter, C., Kaden, S., Hubner, T., Schlotter, B., Tofangchi, S.: Berlin United - Nao Team Humboldt Team Report 2015 - RC2. Technical report (2015)
- 4. Tilgner, R., Reinhardt, T., Kalbitz, T., Seering, S., Wunsch, M: Team Research Report. Technical report (2014)
- Tilgner, R., Reinhardt, T., Kalbitz, T., Seering, S., Wunsch, M: Team Research Report. Technical report (2015)
- Mayer, G., Utz, H., Kraetzschmar: Playing Robot Soccer under Natural Light. RoboCup 2003: Robot Soccer World Cup VII (2003)
- Sridharan, M., Stone, P: Towards Illumination Invariance in the Legged League. RoboCup 2004: Robot Soccer World Cup VIII (2004)
- Kurugollu, F., Sankur, B., Harmanci, A: Color image segmentation using histogram multithresholding and fusion. Image and Vision Computing 19 (2001) 915-928 (2001)
- 9. Swain, M., Ballard, D: Indexing Via Color Histograms. In: Third International Conference on Computer Vision (1990)
- 10. NAO Video camera specification, http://doc.aldebaran.com/2-1/family/ robots/video\_robot.html
- Kofinas, N: Forward and Inverse Kinematics for the NAO Humanoid Robot. thesis (2012)
- Rofer, T., Laue, T., Bockmann, A., Muller, J., Tsogias, A: B-Human 2013: Ensuring Stable Game Performance. RoboCup 2013: Robot Soccer World Cup XVII (2013)
- 13. Fischler, M., Bolles, R: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Thesis (2012)
- 14. Kriegel, H., Kroger, P., Zimek, A: Outlier Detection Techniques. In: Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (2009)
- Schulz, H., Behnke, S: Utilizing the Structure of Field Lines for Efficient Soccer Robot Localization. Advanced Robotics, 26(14): 1603-1621 (2012)