Towards Automatic Colour Segmentation for Robot Soccer

Donald Bailey, Miguel Contreras, Gourab Sen Gupta
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
D.G.Bailey@massey.ac.nz

***Abstract*—Tuning the colour thresholds within robot soccer is laborious, and sensitive to changes in illumination. An algorithm for automatic gain control and white balancing within the camera is described. This significantly reduces the effects of the variations in lighting on the thresholds. Next, a new colour space is proposed which maximises the hue separation of the different coloured regions to improve the colour discrimination. Finally several automatic segmentation techniques are briefly discussed. All of the algorithms are designed to operate on data directly streamed from the camera, enabling a low latency FPGA implementation.**

Keywords-colour transformation; colour thresholding; colour balancing; robot soccer

I. INTRODUCTION

Robot soccer has often been used as a test-bed for research into robotics, robot vision, and multi-agent collaboration. It has also been used as a teaching platform for a wide range of mechatronics, electronics, and computer systems engineering courses. Recently, we have been developing a field programmable gate array (FPGA) based smart camera for the vision processing for robot soccer [1, 2].

Within the small robot leagues for robot soccer, a global vision system is usually used to track the robots and the ball. Each team mounts a camera over the playing area, with the field of view covering the complete playing field. Individual robots are identified by coloured patches on the top surface of the robots (visible to the global camera, see Figure 1). While there are many different coding schemes used by different teams, in principle the identity, position and orientation of each robot is determined from the patch configuration encoded within the coloured patches.

Usually software based image processing is used to locate the robots and ball; this data is then passed onto a strategy processor which determines the desired behaviour, and sends commands to the individual robots. Rather than performing the image processing on a conventional computer, a smart camera performs all of the processing within the camera itself [3]. Their portability makes smart cameras attractive for robot soccer [4].

Most smart cameras store the captured images inside frame buffers within the camera before beginning processing. One of the advantages of using smart cameras is that only the data extracted from the images needs to be transferred to the strategy processor and not the entire image, greatly reducing the communication overhead.

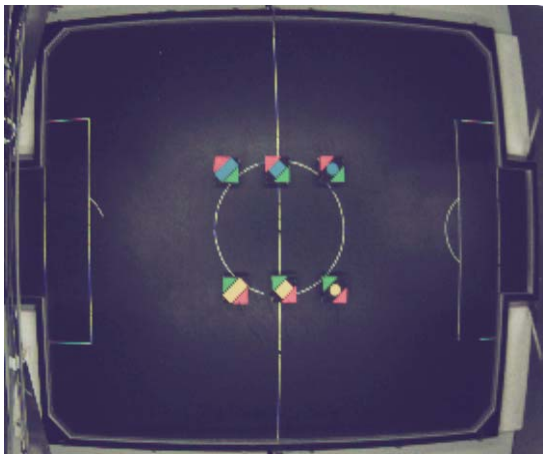

Figure 1. View of robots as seen from the global camera.

In many applications using an FPGA based smart camera allows the pixel stream to be processed directly from the image sensor without the need for a frame buffer. Direct interface between the FPGA and the sensor can significantly increase both the spatial and temporal resolution, and reduce the latency [1], all of which lead to improved control of the robots. However, stream processing imposes strict timing constraints (one pixel per clock cycle) and requires all processing to be performed on the pixels in raster scan order. These place significant constraints or limitations on the algorithms used within the processing.

One common problem encountered within robot soccer is segmenting and identifying each individual coloured patch. While to the human eye, the colours appear distinct with good separation, changes in the light, in particular changes in light intensity, colour temperature and variations in illumination across the playing area often play havoc for machine vision. This is especially so with the commonly used fixed global thresholding for segmenting the colours. Manual setup of thresholds is tedious and time consuming, and must be repeated every time the lighting environment changes. This problem is sometimes called colour constancy – estimating the true colour in the presence of variations in illumination [5, 6]. For this reason there has been much research into colour calibration.

978-1-4799-6466-6/15/$31.00 ©2015 IEEE

The remainder of this paper is structured as follows. Section 2 reviews prior work towards automatic colour calibration methods within the context of robot soccer. Section 3 describes our implementation of automatic gain control and automatic white balancing within an FPGA based smart camera. This is followed in section 4 by a brief comparison of the approaches used to make the segmentation adaptive.

II. Prior Work

There have been several approaches to the colour constancy problem. These fall into three broad categories: selection of a colour space which is relatively invariant to illumination; removing the effects of illumination through colour correction giving the reflectance of targets in the scene; and using adaptive approaches to colour segmentation.

A. Colour Space Selection

For computational simplicity, predefined or precalculated thresholds are commonly used. A cuboid in the chosen colour space reduces the logic to fixed thresholds on each of the colour axes.

The RGB colour space is a very common output used by many image sensors. However, one problem with the RGB colour space is that variations in light intensity will have a strong effect on all three components. Therefore, the RGB space is not robust for colour segmentation in the presence of lighting variation.

A significant improvement results from separating the luminance from the chrominance. Many cameras provide YCbCr output (because of its dominance in television) so this is another commonly used colour space. Although the Cb and Cr components still scale within intensity, this has a significantly smaller effect than the RGB components because these are colour differences.

For an RGB camera, the matrix multiplication for conversion to YCbCr, while not complex, is time consuming when implemented on every pixel. To avoid transforming every pixel, a common alternative is to back-map the segmentation to RGB space [7, 8], resulting in a large lookup table. While this lookup table is expensive to calculate, it only needs to be done once, and results in a relatively quick segmentation.

With the distinct colours within robot soccer, the Y component provides little discrimination, so simple colour difference signals (R-G, R-B and B-G) can be used [9]. Variations on YCbCr which use power of 2 coefficients enable the computation to be performed more efficiently [7].

Another commonly used colour space is HSL or similar hue based colour space [10, 11]. These spaces eliminate the scaling of the chrominance components (hue and saturation) with intensity. Again because of the computational complexity, fixed thresholds in HSL space can also be back-mapped to RGB to enable segmentation by simple lookup. Hue related colour spaces are sensitive to white balance, with any imbalance resulting in hue errors, especially for low saturation colours. This requires white balancing before hue segmentation [12].

The two biggest problems with fixed thresholds are the variation in illumination across the playing field, and the time-consuming process of re-tuning the thresholds when the conditions change. One approach to the variation across the field is to divide the field into a grid, and have separate thresholds for each cell within the grid [13]. However these still require calibration and adjustment for temporal variation in lighting.

B. Colour Correction

The sensed RGB values depend not only on the reflectance, but also the spectral sensitivity of the receiver and the spectral composition of the illumination source. Colour constancy involves transforming the received RGB values to remove the effects of varying illumination. This 3×3 transform requires knowing the characteristics of the illumination, although this can be approximated simply by scaling each of the red, green and blue components [5]. The resultant scaling is sometimes called white balancing, and may be performed either in software, or within the camera itself.

Several authors attempt to optimise the camera parameters (gain, exposure, gamma, and colour channel gains). This is easiest if a white or grey reference patch is present within the image [5]. However, other metrics can be used to assess the image quality [14], and these have been optimised using linear (proportional and integral) control [14] or fuzzy control [15]. Many techniques assume a neutral background. If not, other techniques can be used for identifying the dominant background colour to enable its removal as a preprocessing operation [16].

In software, more complex methods can be used to achieve colour constancy. For example, [8] uses a centre-surround retinex algorithm to reduce the variation of the illumination. This is effectively using the local context to provide the normalisation for each colour, mimicking aspects of the human visual system.

C. Adaptive Approaches

The third approach is to perform adaptive segmentation. Techniques in this category adjust the colour segmentation thresholds based on what was seen in recent images. These effectively track any slowly varying changes in the illumination (either globally, or as the robot moves from one part of the field to another where illumination may differ). [17] treats each colour class as a sphere in colour space, and adaptively adjusts the centre and radius of the sphere. In a similar manner, [18] models each class as a 3D Gaussian. Bayesian statistics have been used by the authors in [10], and a mixture of Gaussians model has been used by the authors in [11] to update the colour model for each class.

Where there are one or two dominant colour classes that can be detected unambiguously, these can be used as a reference for mapping the remaining detected colours to known classes [19].

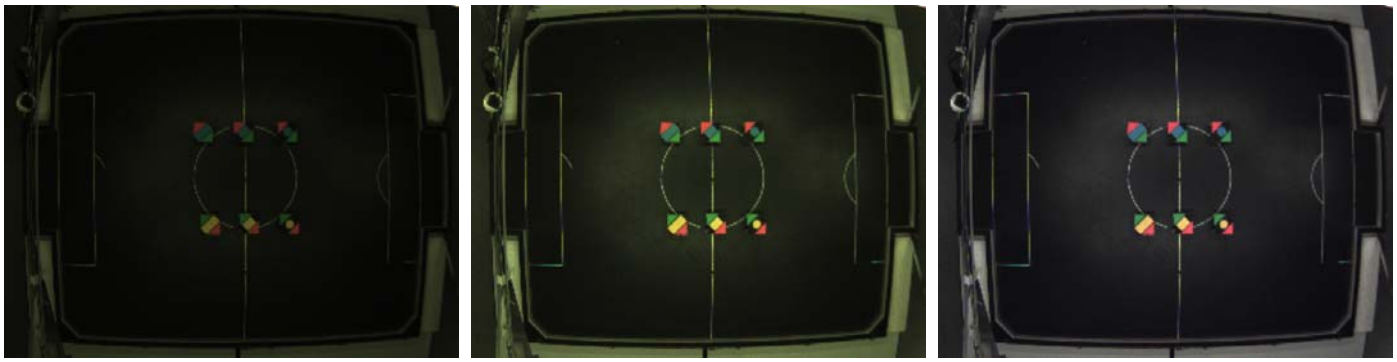

Figure 2. Effect of gain control and white balancing. Left: raw image. Centre: after adjusting gain control. Right: after white balancing

III. Gain Control and White Balancing

In our experiments on an FPGA, we want to operate on streamed images to limit the resources used by our design. To retain the latency advantage of streaming from an FPGA we want to process each pixel only once as it is streamed from the camera. Any pre-processing that can be accomplished by adjusting the camera reduces the resources required, and the delay associated with processing in the FPGA. Therefore, the exposure and gain can be adjusted directly on the camera to maximise the dynamic range of the streamed images.

A. Gain Control

With our implementation of robot soccer, we have increased the frame rate to about 120 frames per second. At this rate, the exposure has been maximised, and any further increase in dynamic range requires adjusting the gain within the camera.

The available dynamic range is maximised when the brightest pixels are just saturating in the image. When highlights are not saturated, it is trivial to calculate the gain directly

$$G_{n+1} = G_n \frac{2^k - 1}{P_{\max}} \tag{1}$$

where k is the number of bits per pixel, $P_{\max}$ is the maximum pixel value in the image, and G_n and G_{n+1} are the gains for frames n and n+1.

Unfortunately, when the image is saturated, the true value of $P_{\max}$ is lost, and the gain must be decreased. Therefore, rather than calculate the gain directly, the value programmed into the camera gain register is incremented or decremented with each frame. Hysteresis is provided between the adjustments to prevent flickering from unnecessarily adjusting the gain with each frame. It also allows small fluctuations (for example the mains beat frequency of the lights) without making unnecessary adjustment.

For 8-bit pixel data, the thresholds have been set at 224 and 252. If fewer than 64 pixels are greater than 224, then the image is under-exposed, and the camera gain is increased. If 64 or more pixels are greater than 252, then the image is over-exposed, and is likely to have too many saturated pixels, and the camera gain is reduced. Requiring a count of 64 allows a few saturated pixels while still maximising the dynamic range. The result of applying this can be seen in Figure 2 (centre).

B. White Balancing

The colour segmentation algorithm uses adaptive hue thresholds in a modified HSV colour space. The hue of colours with low saturation is particularly sensitive to small differences in the balance between the red, green and blue components of the light. To be effective, use of an HSV colour space requires colour balancing.

A simple white balance can be accomplished by adjusting the individual red, green and blue gains within the camera after first setting the global gain. Obviously, when operating directly on streamed data, we cannot adjust the camera gains for the data we have just received. Therefore, it is necessary to assume that the conditions are slowly varying relative to the frame rate of the camera, and use data derived from the current frame to adjust the gains for the following frame.

Since the image is dominated by the dark grey background of the playing area, we propose to use a "grey world assumption" [20], which assumes that the average pixel value over the whole image is a neutral grey. The perturbations caused by the colour patches is negligible because of their small size and on average, the patches also average out to approximately grey.

Conceptually, we measure the average pixel value in each of the red, green, and blue channels. The gains of the red and blue channels are adjusted to make the average for those channels the same as that of the green channel:

$$G_{Red} = \frac{\frac{1}{N}\sum Green}{\frac{1}{N}\sum Red} \quad G_{Blue} = \frac{\frac{1}{N}\sum Green}{\frac{1}{N}\sum Blue} \tag{2}$$

Since the camera gain has discrete steps, the red and blue gains are again adjusted incrementally. Rather than perform the division, we make the observation that when balanced (for the red channel)

$$\sum Red \approx \sum Green \tag{3}$$

This enables us to analyse the raw Bayer image directly as it is streamed from the camera. The Bayer pattern has alternating red and green pixels on every second line, and

alternating blue and green pixels on the other lines. On the red lines, we accumulate

$$S_{R-G} = \sum Red - Green \tag{4}$$

and similarly for the blue lines:

$$S_{B-G} = \sum Blue - Green \tag{5}$$

At the end of the frame, if S_{R-G} or S_{B-G} is greater than a threshold, it indicates that there is too much red or blue respectively, and the corresponding gain is reduced one step for the next frame. Conversely if S_{R-G} or S_{B-G} is smaller than a negative threshold, there is insufficient red or blue, and the corresponding gain is increased by one step for the following frame. This usually converges to a white balanced image within a few frames, and is able to track changes in lighting. The effectiveness of the white balancing algorithm can be clearly seen in figure 2 (right).

IV. Colour Segmentation

A. Colour Space Mapping

To further reduce the effects of illumination variation (for example a non-uniformity of intensity across the playing area), the RGB values are converted to a YUV-like colour space.

True YUV (or YCbCr) is not required, because the resulting images are not intended for television viewing. Therefore a power of 2 variation is used to reduce the computation required [21]

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix}_1 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6}$$

From the YUV_1 space, the U and V components are converted from rectangular to polar coordinates to give hue, H, and colour strength, S:

$$H = \arctan\frac{V}{U} \tag{7}$$

$$S = \sqrt{U^2 + V^2} \tag{8}$$

Note that the hue here is not the same as the hue from the standard HSL colour space – the angle may be rotated and distorted depending on the weights used. Colour strength is related to saturation, although the normalisation by dividing it by the lightness is not necessary here because the gain control within the camera effectively performs a form of intensity normalisation.

The effect of applying this mapping on the colours in our robot soccer system is shown in Figure 3. The circular region in the centre corresponds to the bulk of the grey background that can be eliminated from consideration.

The main problem is that the angles are very close around the orange colour associated with the ball. This makes discrimination of the ball challenging, making the thresholds quite sensitive. To improve the discrimination here, it is necessary to separate the range between the pink and the yellow. This leads to the following weights:

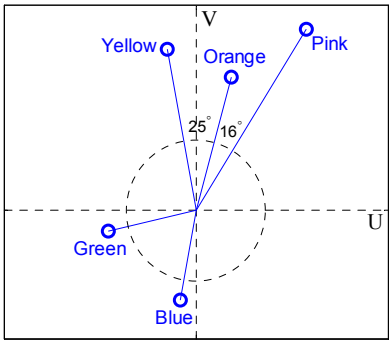

Figure 3. Mapping of colours in YUV_1 colour space.

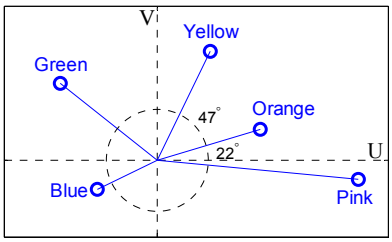

Figure 4. Mapping of colours in YUV_2 colour space.

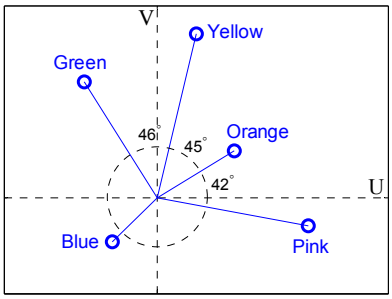

Figure 5. Mapping of colours in YUV_3 colour space.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix}_2 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9}$$

As seen in Figure 4, this gives some improvement, especially between the yellow and orange. By observing that the orange and pink have opposite values for V, the angle between them can be increased by scaling the U component relative to the V:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix}_3 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{10}$$

The result of this is shown in Figure 5. This shows a significant improvement in the separation, particularly in the pink to yellow range.

On an FPGA, the required calculation of U and V have each been reduced to a single subtraction. A CORDIC operation is able to convert from U and V components to give hue and colour strength within a single clock cycle. A simple fixed threshold of the colour strength is able to remove much of the background, as can be seen in Figure 6. Apart from the coloured regions, all that remains are the lines of the field, and some edges. These are the result of colour artefacts from the Bayer filter and using simple bilinear interpolation used for demosaicing.

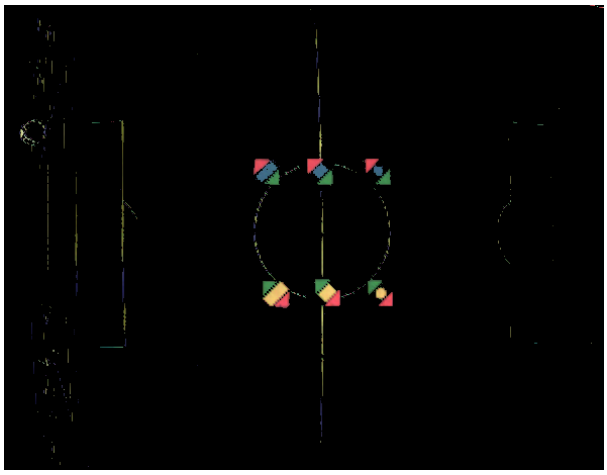

Figure 6. Image after masking out regions of low colour strength (within the dashed circle in Figure 5).

B. Threshold Selection

The final step in segmentation is to select the thresholds which define each colour class. We are currently investigating a number of methods for automating this process. They are described here in order of increasing complexity.

1) Fixed thresholds

Since colour balancing removes much of the effects of changing illumination, it may be possible to fix the thresholds. Although scaling each of the RGB primaries is reasonably effective in removing most of the effects of varying light colour, it is not able to provide complete colour constancy. The effectiveness of this approach requires examining the hue shifts over a wide range of illumination conditions to determine the viability of fixed thresholds.

The problem still remains, however, of initially setting the threshold levels. This must either be done manually, or by using one of the other methods during the setup phase.

2) Adaptive thresholds

Starting with an initial set of thresholds, the mean hue is calculated of each colour class detected. These mean hues are then used to determine adjusted thresholds for the following frame.

Two approaches for setting the thresholds are:

- Midway between the currently detected hue angles;
- A fixed angle on either side of the detected hue angles

This approach still requires an initial set of target colours, although these could be set at the angles determined in the previous section. As long as the starting point is reasonably close, and the lighting conditions do not change rapidly (both of which are reasonable assumptions for robot soccer), this approach should be able to lock onto and track the different colour classes.

3) Dynamic thresholding

This extends the previous approach, only rather than adjusting the thresholds from the previous frame, this determines appropriate thresholds directly for each frame.

Within a hue based colour space, one approach would be to obtain a hue histogram, and identify the peaks within this corresponding to the different colours within the image. The thresholds are then placed in the valleys between the peaks. Such a histogram based approach cannot be applied directly to streamed images because it would require saving the image within a frame buffer while accumulating the histogram used to segment the image. The streamed alternative assumes that the change from one frame to the next is relatively slow, so the valleys between peaks in the previous frame would be good thresholds for the current frame.

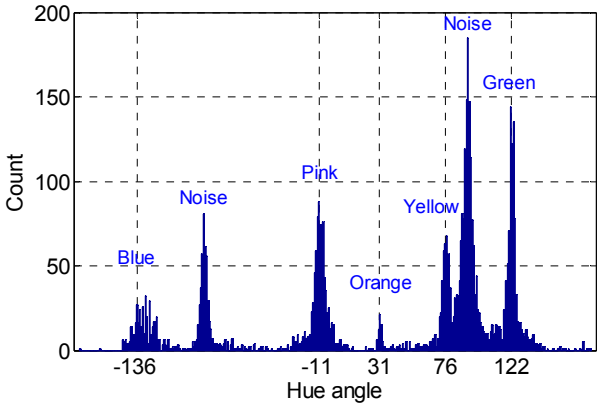

Figure 7. Histogram of detected pixels.

There are two complicating factors with this approach, as can be seen in Figure 7. The first is the presence of two strong noise peaks, next to the blue, and yellow classes. These result from the Bayer interpolation artefacts, and correspond to the detected lines (as shown in Figure 6), and Bayer interpolation artefacts around the edges of the yellow region. Most of the line artefacts can be eliminated using appropriate morphological filtering, although those around the edge of the yellow region are harder to eliminate.

The second problem is that the peak associated with the orange ball is quite small (reflecting the small size of the ball relative to the area of the other colour regions). As a result, this peak could easily be missed, and requires specifically searching for this.

4) Region growing

Rather than directly threshold the image, cluster adjacent pixels which have similar hue angles. Adjacent pixels with sufficiently different hue will be assigned to different regions [22]. This can be integrated within the connected components

analysis stage, operating directly on the hue without performing the colour classification first. This approach finds blobs of similar colour first, and identifies the actual colour of the blobs afterwards.

This approach requires edge enhancement first, to prevent the blurring between adjacent coloured regions from having a path of connected pixels that are all within the hue difference threshold.

V. Summary and Discussion

Much of the variation in lighting can be removed from captured images by implementing gain control and white balancing directly within the camera. White balancing is particularly important when using a hue based colour space, because small variations in the light colour can have significant variations on the hue angle, especially for weakly saturated colours. Pragmatic incremental algorithms for achieving gain control and white balancing have been described and implemented on an FPGA. The result of these is a more consistent range of images, significantly simplifying the colour segmentation problem.

The next stage is to transform the images into a colour space which maximises the discrimination between the different colours. Using a hue based colour space reduces the colour classification to a single dimension, requiring only simple thresholding to detect each colour. To change the separation between colours, hue was redefined as a polar coordinate representation of the chrominance components. Therefore, by changing the definition of the chrominance components, improved hue discrimination was obtained.

The final stage in automating the segmentation process is to automatically determine the threshold levels. Several methods have been proposed, which are the subject of future investigation.

References

[1] M. Contreras, D. G. Bailey, and G. Sen Gupta, "FPGA implementation of global vision for robot soccer as a smart camera," in *2nd International Conference on Robot Intelligence Technology and Applications*, Denver, Colorado, USA, 2013, pp. 657-665.

[2] D. Bailey, G. Sen Gupta, and M. Contreras, "Intelligent camera for object identification and tracking," in *1st International Conference on Robot Intelligence Technology and Applications*, Gwangju, Korea, 2012, pp. 1003-1013.

[3] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *IEEE Computer,* vol. 39, no. 2, pp. 68-75, 2006.

[4] P. Wills, "The hardware design of a smart camera for the robot soccer environment," in *Department of Computer Science and Electrical Engineering*. vol. BE (Hons): University of Queensland, 1999.

[5] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?," in *5th European Conference on Computer Vision (ECCV'98)*, Freiburg, Germany, 1998, pp. 445-459.

[6] R. Gershon, A. D. Jepson, and J. K. Tsotsos, "From [R,G,B] to surface reflectance: Computing color constant descriptors in images," in *International Joint Conference on Artificial Intelligence*, Milan, Italy, 1987, pp. 755-758.

[7] G. Sen Gupta and D. Bailey, "Discrete YUV look-up tables for fast colour segmentation for robotic applications," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2008)*, Niagara Falls, Canada, 2008, pp. 963-968.

[8] G. Mayer, H. Utz, and G. Kraetzschmar, "Towards autonomous vision self-calibration for robot soccer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, pp. 214-219.

[9] J. Baltes, "Practical camera and colour calibration for large rooms," in *RoboCup-99: Robot Soccer World Cup III*, New York, USA, 2000, pp. 148-161.

[10] C. Gonner, M. Rous, and K. F. Kraiss, "Real-time adaptive colour segmentation for the RoboCup middle size league," in *RoboCup 2004: Robot Soccer World Cup VIII*, Lisbon, Portugal, 2005, pp. 402-409.

[11] F. Anzani, D. Bosisio, M. Matteucci, and D. G. Sorrenti, "On-line color calibration in non-stationary environments," in *RoboCup 2005: Robot Soccer World Cup IX*, Osaka, Japan, 2006, pp. 396-407.

[12] P. Jonker, J. Caarls, and W. Bokhove, "Fast and accurate robot vision for vision based motion," in *RoboCup 2000: Robot Soccer. World Cup IV*, P. Stone, T. Balch, and G. Kraetzschmar, Eds.: Spinger Verlag 2001, pp. 149 -158.

[13] A. Egorova, M. Simon, F. Wiesel, A. Gloye, and R. Rojas, "Plug and play: Fast automatic geometry and color calibration for cameras tracking robots," in *RoboCup 2004: Robot Soccer World Cup VIII*, Lisbon, Portugal, 2005, pp. 394-401.

[14] A. J. R. Neves, A. Trifan, and B. Cunha, "Self-calibration of colormetric parameters in vision systems for autonomous soccer robots," in *RoboCup 2013: Robot World Cup XVII*, Eindhoven, The Netherlands, 2014, pp. 183-194.

[15] A. Espinola, A. Romay, T. Baidyk, and E. Kussul, "Active color calibration for a vision system in a semi-structured environment," *Transactions on Control and Mechanical Systems,* vol. 1, no. 2, pp. 65-72, 2012.

[16] G. Wyeth and B. Brown, "Robust adaptive vision for robot soccer," in *Mechatronics and Machine Vision in Practise*, Hervey Bay, Queensland, Australia, 2000, pp. 41-48.

[17] P. Heinemann, F. Sehnke, F. Streichert, and A. Zell, "Towards a calibration-free robot: the ACT algorithm for automatic online color training," in *RoboCup 2006: Robot Soccer World Cup X*, Bremen, Germany, 2007, pp. 363-370.

[18] M. Sridharan and P. Stone, "Towards eliminating manual color calibration at RoboCup," in *RoboCup 2005: Robot Soccer World Cup IX*, Osaka, Japan, 2006, pp. 673-681.

[19] P. Guerrero, J. Ruiz-del-Solar, J. Fredes, and R. Palma-Amestoy, "Automatic on-line color calibration using class-relative color spaces," in *RoboCup 2007: Robot Soccer World Cup XI*, Atlanta, Georgia, USA, 2008, pp. 246-253.

[20] G. D. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by correlation: A simple, unifying framework for color constancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1209-1221, 2001.

[21] D. G. Bailey, *Design for embedded image processing on FPGAs*. Singapore: John Wiley and Sons (Asia) Pte. Ltd., 2011.

[22] D. G. Bailey, "Raster based region growing," in *6th New Zealand Image Processing Workshop*, Lower Hutt, NZ, 1991, pp. 21-26.