

# Towards Autonomous Vision Self-Calibration for Soccer Robots

Gerd Mayer

Hans Utz

Gerhard Kraetzschmar

*University of Ulm, James-Franck-Ring, 89069 Ulm, Germany*

## Abstract

*The ability to autonomously adapt to variations in the environmental conditions is a very useful feature for mobile robots. Of particular interest in robotic soccer are self-calibrating vision systems that automatically adapt to local lighting conditions. The paper presents a method for autonomous calibration of a color classifier used for color blob-based image segmentation and landmark and object recognition. The experimental results demonstrate significantly improved robustness of visual processing.*

## 1 Introduction

One of the most obvious differences between human soccer and robotic soccer is the ability of players to deal with variations in the environmental conditions. Human soccer players can cope very well with large differences of relevant objects – ball, players, playing ground (floor), landmark objects surrounding the playing ground – in size, color, shape, material, and dynamics, and under vastly different lighting and weather conditions. In robotic soccer, by contrast, we still have very strict rules and constraints on these issues, including lighting conditions. Despite these constraints, current robots typically require tedious calibration processes, especially for the vision part, and it often takes several hours to adapt the robots to on-site conditions. Research into methods that can cope with a wider range of environmental conditions and are able to autonomously adapt to local environments is an essential step towards the long-range ROBOCUP goal. Furthermore, new technology and methods in this area are of immediate interest for other mobile robot applications.

Like for many other teams, calibrating the vision system for our robots means adapting a color classifier to the situation on-site, which is determined by both the local lighting conditions and the organizers' choice of object colors for particular objects, like the goals and the floor carpet. The goal is to replace the tedious and error-prone manual calibration process by a method, which allows the robot to autonomously self-calibrate its vision system, based on

a number of image samples it collected after being placed into a playing ground. The method applies RETINEX, a biologically-inspired algorithm for improving color invariance, and k-means clustering for the adaptation of color classes. The result is compiled into a color classification table, which is directly used online for color-based image segmentation and landmark recognition.

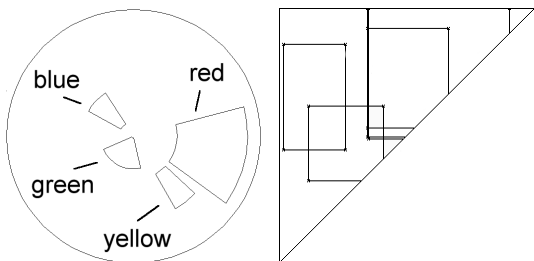
The next section briefly surveys the vision processing architecture used in our robots. Section 3 gives an overview on the autonomous self-calibration procedure, and sections 4 and 5 describe the RETINEX algorithm for improving color constancy and the clustering method used in the adaptation step. Section 6 presents experiments and results. Conclusions and hints on future work are given in the final section.

## 2 Vision Processing for Robotic Soccer

Most teams in ROBOCUP try to visually detect relevant objects like the ball, the goals, teammates and opponents, which follow strict color coding rules: the ball is orange, the goals are yellow and dark blue, robots are black with magenta or light blue color markers. Vision processing in robotic soccer must trade off desirable application of advanced vision algorithms with limited on board computational power and real time constraints (at least 10–15 frames must be processed per second to play reasonably well). Many teams exploit this strong color coding and perform color classification on grabbed images as a first vision processing step. The result can then be searched for contiguous regions of the same color class (color blob detection, e.g. for balls and goals) or transitions between color classes (edge detection, e.g. for field lines and goal posts). In our case, blob detection is simply performed by computing occurrence histograms in both vertical and horizontal direction for the color class of interest. Thresholding is applied to the histogram both to filter out noise and to identify candidate regions for color blobs. For other features, the local neighborhood within the labeled image along lines found by a Canny edge detector on the real picture is regarded to detect specific transitions within the image (e.g. green-white floor lines).

The thresholds for noise and candidate selection represent the available signal-to-noise ratio in the color-classified image. If this ratio is low, the likelihood for false positives (mistaking noise for an object) or false negatives (failing to detect objects) increases significantly, resulting in poor play. The signal-to-noise ratio critically depends on the quality of color classifier, because misclassifications of pixels will cause the signal-to-noise ratio to decrease. A comparison of sample images with their ideal color labelings has shown that a perfect classifier is not possible, because there are always pixels with the same RGB values, but different color labels. Nevertheless, maximizing the signal-to-noise ratio by well-adjusted color classifiers is highly desirable in order to produce good overall results.

So far, the color classifier was specified by determining a set of cuboids in HSV space. Before developing an autonomous self-calibration procedure, the cuboids were adjusted by hand using a small GUI-based tool. This tool allows to grab an image (in RGB) and select areas containing objects of interest. The tool maps all pixels into HSV color space and displays their distribution. The user can then adjust the boundary values of the cuboid in each of the HSV dimensions. The hard part of this process is to determine the boundaries in areas where there is overlap between different color classes, and to trade off filtering out noise vs. missing features. Figure 1



**Figure 1:** Projections of cuboids in HSV space.

shows the result of a hand-tuned color calibration. The left image shows hue vs. saturation in the usual polar representation, while the right image plots saturation vs. brightness. The diagonal line in the right image outlines the shape of the HSV cone.

### 3 Autonomous Self-Calibration

The autonomous self-calibration procedure consists of the following steps:

1. Sampling the environment
2. Improving color constancy
3. Clustering on HSV-transformed images

4. Associating class labels
5. Back-projecting class labels to RGB space

*Sampling the environment* is autonomously performed by the robot by driving around the field and grabbing sample images. The robot simply uses a program similar to playing soccer and its vision system, which is not yet calibrated to the new environment, but usually sufficient to determine when all relevant objects have been seen at least once.

*Improving color constancy* is done by applying the RETINEX algorithm to all sampled images prior to the adaptation step. In short, RETINEX transforms the images such that there are fewer color values in the image that belong to different objects, and all color values belonging to the same object have smaller spatial dilatation and a higher probability mass around a center point. We show that the cluster algorithm in the adaptation step can deal much better with images transformed by RETINEX and produce a more accurate segmentation.

The *clustering* step converts all pixels of the sampled images into HSV and applies k-means clustering to find representative centers of HSV subspaces with high concentrations of pixels. The clustering is done in the HSV color space, because the circular geometrical partitioning of this color space nicely fits the neighborhood definition of the clustering mechanism. Note, however, that cluster (and therefore, color class) boundaries do no longer need to be collinear with color space dimensions.

*Associating class labels* means interpreting the results of the adaptive clustering process and assigning color classes, representing features or objects, to cluster centers.

*Back-projecting class labels to RGB space* is done in order to speed up online vision processing. Instead of transforming RGB values to HSV space and determining the nearest cluster center online, this mapping is performed during the calibration process for all RGB values and results are stored with their associated color class label in a large index table. Because the lowest bits in images taken with standard low-cost cameras are mostly noise anyway, they can be omitted. By using only the highest 6 bits in each color dimension, memory requirements for storing the lookup table decreases from 16MB to 256KB bytes. During online processing, color classification is done merely by table lookup.

It is noteworthy that the back-projection step allows to identify pixel values with high likelihood for misclassification and treat them in a special way. Due to RETINEX, it can happen that two pixels with the same RGB value in the original image have differ-

ent values in the RETINEX-processed image and will be assigned to different color classes in the adaptation step. This can be discovered and handled during back-projection, e.g. by assigning them to a catch-all color class that has no special semantics.

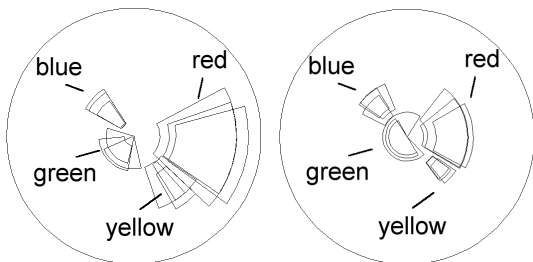
#### 4 Color Constancy with Retinex

Some test images made under three different lighting conditions illustrate the need for improved color constancy. In Figure 2, the first image is taken under a combination of floodlight and neon light and represents conditions typical for official ROBOCUP competitions. The second image, taken under neon light only, exhibits a very diffuse illumination and a slightly darker image. The third image is taken under special spotlight lamps, which are used for indoor television recording. They emit a very warm and yellowish light and throw hard shadows. Left



**Figure 2:** Example images taken under floodlight, neon light and spotlight conditions.

Picture 3 shows the superposition of the color calibrations made by hand for these three lighting conditions. Note, that the individual boundaries differ substantially and that the areas of different color classes sometimes even overlap. The right Figure 3 shows the superposition of color calibrations after pre-processing the images with a color constancy algorithm. The color areas now vary much less between different lighting conditions and the spatial extension decreases for most colors.



**Figure 3:** Superposition of color calibrations under different lighting conditions. On the left, calibrations for unprocessed images, on the right, calibrations for images pre-processed with the RETINEX algorithm.

#### Color Constancy Algorithms

Algorithms for improving color constancy have been already researched for some time. Funt et al. sug-

gest that the relationship between an image and the chromaticity of that illumination can be learned by a neural network [1][4]. This is a model-free method that makes no assumption about surfaces or spectral power distribution of the illumination. However, it needs the chromaticity of the illumination during its learning phase. Barnard and Funt also present an approach on combining the color indexing method of Swain and Ballard [8] with a color constancy method. Forsyth [2][3] views color constancy as a Bayesian inference problem, which can be solved by Markov chain Monte Carlo methods. This approach uses many different, complex models of spatial surface reflectance and illumination which have to be modeled first by hand. Rahman et al. [5][7] introduce RETINEX, a color constancy approach based on ideas from human perception. The RETINEX algorithm is based on the well-known work of Land and McCann [6] and achieves good color rendition on most real-life sceneries.

The exact illumination and reflectance parameters of the objects on the field are usually not known in ROBOCUP. Methods that require to model these parameters are therefore not adequate in this domain. Because it needs no further knowledge about the scenery, the RETINEX algorithm seems to present a good choice.

#### The Retinex Algorithm

For clarity, the RETINEX algorithm is briefly reviewed here. For a full explanation of the RETINEX method, see [5][7][6]. The output of RETINEX is defined by

$$R_i(x, y) = \log I_i(x, y) - \log[S(x, y) * I_i(x, y)]$$

where  $I_i(x, y)$  is the color value in a specific color band  $i$  at the position  $(x, y)$  which is convolved with the surround function  $S(x, y)$  centered at the same position. The surround function in turn is a standard Gaussian function  $S(x, y) = e^{-(x^2+y^2)/c^2}$  which is scaled with the  $c$  value to an overall integral sum of one. Finally, a specific amount of the color values for each color band is clipped by a gain offset (some of the lower- and some of the uppermost values), before the resulting values are scaled back to the usual value range. We did not discover significant improvement for the multiscale RETINEX variation, so we use further on the computationally less expensive singlescale version.

The RETINEX algorithm not only improves color constancy between images taken under different lighting conditions, it also makes the colors within single objects more similar. This results in a higher sample concentration to the center of each color segment. This means that all colors that are assigned to one

label in the segmented image vary less in their spatial distribution and/or in their maximal spatial extension in the color space.

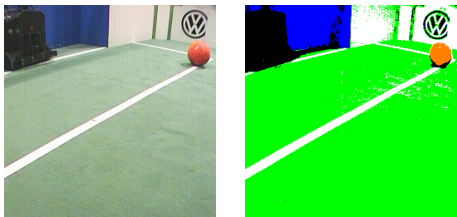
## 5 K-means Clustering

After preprocessing the sample images with RETINEX, pixels are mapped to HSV space. Pixels belonging to the same color class build dense clouds, or clusters, in HSV space. K-means clustering is applied to find the centers of these clusters, which are then used to segment color space and determine color classification tables.

The k-means cluster algorithm is based on two relatively simple rules. One is that each class has a center which is the mean position of all the samples in that class. The other rule is, that each sample is in the class whose center it is closest to. The algorithm is calculated iteratively by assigning all samples to that center for which the distance is minimal. After that, the mean of the class is recalculated and the center shifted to this new result. The shifting depends on the distance measure that is used. In our case, we use the normal Euclidian distance if the HSV color space is viewed as a cone. The assignment of the image pixel to the classes is defined by  $k_j = \{p : \|p - c_j\| < \|p - c_i\|, i \neq j\}$ , where an image point  $p$  is assigned to the class  $k$  whose center  $c$  has the lowest Euclidian distance to this point. The new spatial position of each center point  $c_j$  is then calculated by

$$c_j = \frac{\sum_{p \in k_j} p}{|k_j|}$$

Figure 4 demonstrates how a recorded image is segmented after autonomous self-calibration. The association of cluster centers to feature colors is currently still done by hand. However, usually it is possible to forecast the right class assignment with a rough spatial pre-initialization of the center positions. The



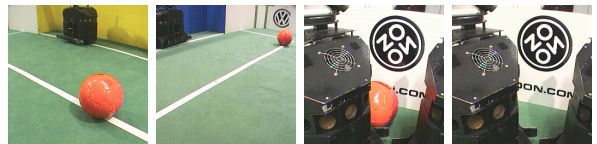
**Figure 4:** Example of how an recorded image (left picture) is segment with an autonomous calculated calibration (right).

number of classes (respectively cluster centers) that are used is always a trade-off between the accuracy of the clustering result on the one hand and the computational effort and the difficulty to interpret the

result on the other hand. In our experiments we find it a good compromise to use a number of nine centers to detect the six occurring colors in our image (Yellow and blue for the goals, green and white for the playing field and the lines on it, orange for the ball, black for the robots, and sponsor logos on the wall) with black also serving as catch-all class.

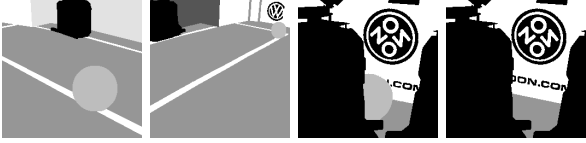
## 6 Experiments and Evaluation

Preliminary experiments in the lab indicate that autonomous self-calibration yields color classifiers that are about as good, and in many cases better, than those obtained by manual calibration. In-depth evaluation and comparison will be performed based on experimental data obtained during the next major ROBOCUP tournament. However, assessing the value of applying color constancy algorithms can be done by comparing results obtained by applying the self-calibration procedure to a small set of typical, but critical sceneries recorded from past tournaments. Figure 5 shows the different scenarios used for the following experiments. These are typical situations as they appear during a ROBOCUP soccer match. All these images are taken under three different lighting conditions as described in Section 4. The rightmost figure, with two robots competing to get to the ball – a scene occurring very often in ROBOCUP – is intentionally included because its low portion of relevant object colors. Because the clustering is done in the



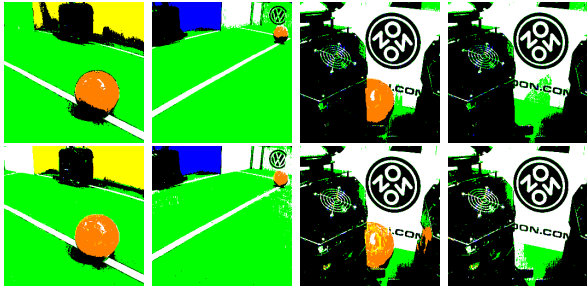
**Figure 5:** The four scenes used for experiments.

HSV color space, we look how the spatial distribution of the color values in this color space looks like after processing them with the RETINEX algorithm. The smaller the spatial dilatation of the color values for each label are, and the higher the probability density of the color values of this label is within a vicinity of the center of a segment, the better the cluster algorithm should be able to find these centers. While the probability density is expected to be more concentrated, the spatial dilatation is not expected to decrease, simply because dark and uniform images are brightened up and get a significantly higher variance in their color distribution. For evaluation, the maximum distance between all color values indexed into the same color class and the variance (respectively, the standard deviation) in HSV color space were calculated. The quality of color classification



**Figure 6:** Ideal segmentation for example scenes.

is determined against artificial images with idealized color segmentation (see Figure 6). This segmentation is strongly idealized, because in the original images e.g. the robots are not completely totally black and parts of the goals are actually colored differently because of scratches. Due to blooming effects in the recorded images, even a human often cannot determine exactly, which pixels are part of an object or of the background. For this reason, the pixels in the immediate vicinity of object boundaries are ignored for all further tests. Tables 1 lists the standard de-



**Figure 7:** Result of color classification on the original images (above) and for RETINEX processed images (below). Images were taken under floodlight conditions.

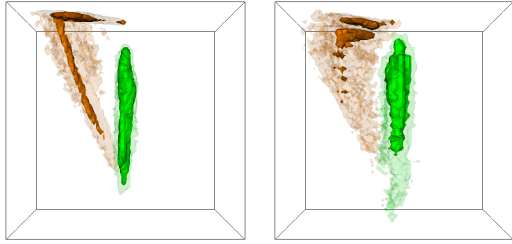
viation and the maximum spatial distance for each of the three lighting conditions. The color names in the leftmost column indicate the segment label taken from the idealized color classification of Figure 6. In almost all cases standard deviation decreases, whereas spatial dilatation vary strongly. For the blue color class both standard deviation and maximal distance increase. One possible explanation is, that for our specific test images (blue should only be seen in one scenario) the robots throws a hard shadow under some lighting conditions. This shadow is filtered out by the RETINEX algorithm, so the difference between this shadowed region and the rest of the blue goal increases. Note that the values for the color class black also increase. But this is not surprising, because many details are only seen in the RETINEX processed images (the original images are way too dark). Also, the idealized color classifications with black as catch-all class makes this class a pool of strongly varying colors. For example the fan on the robots that have a totally different color, but are classified as black.

**Table 1:** The standard deviation and the maximal spatial distance for the pixel values of original and the RETINEX processed images for the three lighting conditions.

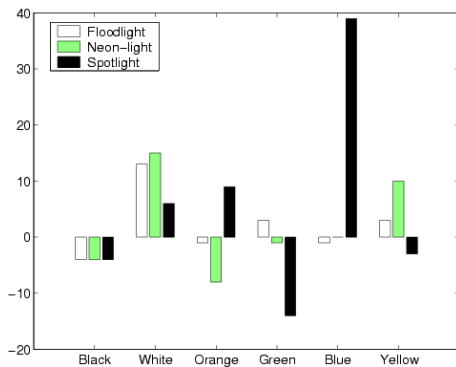
	Std. Deviation		Max. Distance	
	Original	Retinex	Original	Retinex
<b>Floodlight</b>				
Black	39.8	60.4	243.1	269.4
White	21.4	17.4	193.3	220.1
Blue	21.7	33.3	146.6	250.2
Yellow	37.3	29.0	133.4	131.1
Green	25.9	22.2	139.2	179.6
Orange	56.4	52.1	232.7	203.1
<b>Neon-Light</b>				
Black	37.3	61.6	248.2	276.3
White	28.8	18.4	176.2	207.8
Blue	29.6	55.2	148.9	250.7
Yellow	37.3	31.6	123.8	125.4
Green	31.3	24.9	137.7	181.7
Orange	69.6	55.9	206.5	206.0
<b>Spotlight</b>				
Black	79.1	86.6	249.1	256.6
White	28.8	17.8	170.9	173.9
Blue	13.9	26.9	122.3	247.2
Yellow	35.9	33.7	107.3	129.5
Green	41.2	31.6	143.7	187.3
Orange	84.0	76.4	206.7	240.7

For illustration, the class assignments are visualized directly in HSV color space in Figure 8 which shows the three dimensional histogram of color values from two specific classes from the images taken under floodlight conditions. Dimensions are the same as in Figure 1. The vertical axis is the brightness value, the other two axes are hue and saturation values in polar representation. The left illustration shows the values without, the right image with applying the RETINEX algorithm. All parts are dyed in the color of the color class. The figures describe the spatial distribution of the differently labeled regions as color blobs. Different levels of transparency in the plot describe different iso-levels of the density distribution, with a convex hull drawn over all points with the same histogram value. Note, that the opaque part of the orange distribution decreases strongly between the two figures as a result of the decreasing variance, and the transparent hull of the green color class increases slightly in the vertical direction. The RETINEX algorithm modifies the images in a way that should be very useful for the successive clustering process. To test this, all test sets, consisting of all scenarios under one specific lighting condition, are clustered under all lighting conditions, once with the original images and once with RETINEX processed images. This is repeated five times and the results are averaged. The cluster centers are initialized with random values. Results are again compared to the idealized segmentations from Figure 6. The bar plot in Figure 9 shows the different





**Figure 8:** Three-dimensional histogram of the spatial distribution of two classes within the HSV color space, based on the idealized segmentation. Shown are the color distributions for the color classes orange (for the ball, on the left) and the green (for the playing field, near the center) under floodlight conditions (original image on the left, RETINEX processed on the right)



**Figure 9:** Bar plots of differences in percentage of unambiguously classifiable colors for original and RETINEX processed images for different illuminations.

of the percentage of unambiguously classifiable pixels between original and RETINEX processed images for each color class and lighting condition. Overall performance increases by about 5% between the original and the RETINEX processed images. A remarkable difference occurs for the blue color class in the spotlight configuration. The reason is that in two of the five runs with original images, the blue goal and the green floor are assigned to the same cluster center. With these calibrations, the blue goal would never be detected by the robot. Calibration based on RETINEX processed images easily deal with this situation. In all scenarios the value for the white class increases, which is very important for teams, who apply self localization methods that rely on detecting the white lines on the floor. When interpreting these results, it must be taken into account that the lighting conditions for all situations were quite good. The floodlight scenario especially is used for ROBOCUP competitions, because it is known to

be more or less optimal for color indexing methods. For less controlled illuminations like normal daylight, even better result can be expected.

## 7 Conclusions

For most robotic soccer teams, the quality of landmark and object recognition critically depends on a color classifier, which is precisely tuned to on-site conditions and, preferably, exhibits some robustness against noise in lighting conditions. The method described in this paper combines three ideas in order to avoid tedious manual tuning procedures: i) representation of the classifier as indexed color class tables, ii) k-means clustering in HSV color space for adaptation, and iii) RETINEX preprocessing of training images for increased color constancy. Experimental results show, that the method provides improved color classifiers even if RETINEX processing is currently not applied during online processing. Research in speeding up the RETINEX algorithm for application in real time applications is currently ongoing and will further improve robustness of color classification. Thus, the method presents a significant step towards autonomous self-calibration of robotic vision systems.

## References

- [1] V. C. Cardei, B. V. Funt, and K. Barnard. Modeling color constancy with neural networks. In *Proceedings of the International Conference on Vision, Recognition and Action: Neural Models of Mind and Machine*, May 1997.
- [2] D. Forsyth. A novel approach for color constancy. *International Journal of Computer Vision*, 5(1):5–36, 1990.
- [3] D. Forsyth. Sampling, resampling and colour constancy. In *Proceedings Computer Vision and Pattern Recognition*, pages 300–305, 1999.
- [4] B. V. Funt, V. C. Cardei, and K. Barnard. Learning color constancy. In *Proceedings IS&T/SID Fourth Color Imaging Conference*, pages 58–60, November 1996.
- [5] D. J. Jobson, Z. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. March 1997.
- [6] Edwin H. Land and John J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1), January 1971.
- [7] Z. Rahman, D. J. Jobson, and G. A. Woodell. Multi-scale retinex for color image enhancement. In *International Conference on Image Processing*, 1996.
- [8] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:1:11–32, 1991.