An Unsupervised Approach for Adaptive Color Segmentation

Ulrich Kaufmann[1], Roland Reichle[2], Christof Hoppe[2] and Philipp A. Baer[2]

[1] Institute of Neural Information Processing, University of Ulm
James-Franck-Ring, 89069 Ulm, Germany

[2] Distributed Systems Group, University of Kassel
Wilhelmshöher Allee 73, 34121 Kassel, Germany

Abstract. One of the key requirements of robotic vision systems for real-life application is the ability to deal with varying lighting conditions. Many systems rely on color-based object or feature detection using color segmentation. A static approach based on preinitialized calibration data is not likely to perform very well under natural light. In this paper we present an unsupervised approach for color segmentation which is able to self-adapt to varying lighting conditions during run-time. The approach comprises two steps: initialization and iterative tracking of color regions. Its applicability has been tested on vision systems of soccer robots participating in RoboCup tournaments.

1 Introduction

In recent developments, vision systems more and more emerge as the main sensory component of autonomous mobile robots. In real-world scenarios, one of the key requirements is the ability to deal with natural light and varying lighting conditions. However, this remains a very challenging task and topic of research, as shown, for example, by the efforts of the RoboCup community. RoboCup [7] is an international joint project attempting to foster research in robotics, artificial intelligence and related fields. One of the long-term goals of the RoboCup initiative is to create soccer robots capable of playing on typical soccer playgrounds. These include, but are not limited to, outdoor soccer fields under natural light.

So far, RoboCup tournaments exhibit constant artificial lighting conditions. Only minimal changes in lighting are allowed, such as caused by sunlight coming through windows. The transition to natural light is performed only slowly. The main reason is that in order to detect the color-marked objects like the ball, goals, opponents, or team members, the robot vision systems are based on color segmentation approaches. Color segmentation reduces the number of colors by combining color regions to single colors or filtering out irrelevant colors. Usually color segmentation depends on calibration data gathered before a game which identify color regions of interest in a color space.

Nevertheless, static approaches are not able to deal with natural illumination with varying temporal dynamics. There may be slow changes in lighting during the day

Kaufmann U., Reichle R., Hoppe C. and A. Baer P. (2007).
An Unsupervised Approach for Adaptive Color Segmentation.
In *Robot Vision*, pages 3-12
DOI: 10.5220/0002066200030012

or fast changes caused by passing clouds. In order to be able to deal with such unstable lighting conditions, this paper presents an unsupervised, self-adapting approach for color segmentation, comprising two steps: The first step detects and initializes the color regions of interest. The second step tracks these regions iteratively during runtime. The approach is, as already mentioned, completely unsupervised as the user is not confronted with parameter adjustment at all. The only information required a-priori is a very rough estimation of the problem-specific color regions of interest in the color space.

The remainder of the paper is organized as follows. Section 2 discusses related work facing similar challenges or applying similar techniques. In section 3 the overall approach is introduced and the algorithms for each step are presented in detail. Section 4 presents the results of the experimental evaluation. The last section summarizes the main contributions of the paper and hints at future work.

2 Related Work

In many industrial and research applications, color provides a strong clue for object recognition to be performed by robotic vision systems. Therefore, camera calibration and color indexing are important topics in robotics research. One of the key challenges for such systems is the ability to cope with changing lighting conditions.

In [6], Mayer et al. present a case study which discusses various lighting conditions, ranging from artificial to natural light, and their effect for image processing and vision routines. As a result, they conclude that dynamic approaches for color segmentation are required under these conditions. Jüngel et al. [5, 4] describe a calibration approach which initially looks for regions of a reference color (e.g. green as in the RoboCup 4-Legged League which is used as example scenario) applying simple heuristics. Based on these regions, the regions of the remaining predefined colors are determined in the YUV color space, maintaining their relative placement. However, this approach can be considered as risky, since relative distances of the color regions of interest are not constant and may be stretched by changes in illumination [6]. A somewhat different approach is presented by Gönner et al. in [2]. They calculate chrominance histograms representing the frequency of color values of specific objects. The relative frequency of the color values corresponds to the conditional a-priori probability of a certain color value, assuming a certain object is present. The a-posteriori probability of a color value being assigned to some specific object is derived from a Bayesian combination of these chrominance histograms. However, for creating the initial a-priori probability distribution the approach relies on elaborate object recognition mechanisms. Very similar to our approach, a contribution by Anzani et al. [1] describes a method for initial estimation of color regions and their tracking to cope with changes in illumination. The color regions are represented as a mixture of 3D-Gaussians (ellipsoids) in the HSV color space. The tracking of color regions is realized by applying the EM algorithm. In contrast to our approach, however, this method has to deal with the problem of determining the optimal number of ellipsoids representing the color region, in order to avoid overfitting of noise and to prevent a too rough representation. In our approach, noise elimination mechanisms are integrated for the initialization step and also for the tracking. This allows us to represent the relevant color regions in a very fine-grained manner without the risk of overfitting noise.

Heinemann et al. [3] propose another technique which also allows the modification of the color-mapping function over time using a set of scenario-dependent assumptions. For this approach, the position of the robot is required a-priori. Another method for generating segmentation tables is to use the fixed positions and the shaping of known objects. Before the start of a mission, these objects are scanned and their color data are used for calibration [8, 9]. Similar as for the approach of Heinemann, a great amount of problem-sepcific knowledge is required.

3 Approach

As sketched in the RoboCup scenario described in section 1, many vision systems face the challenge to detect colored objects. However, natural lighting conditions cause even very homogeneously colored objects to exhibit a high number of different shadings. Thus, segmentation approaches are commonly used to simplify object recognition: Color regions within a color space are mapped to an ideal color or representing color labels; irrelevant colors are filtered out. Each region is minimal, as larger regions would include shadings that do not belong to the objects in question. Under natural ligthting conditions, traditional static segmentation approaches are likely to fail: Shadings mapped to a single color label may change. Thus, the mappings need to be adjusted.

As already introduced above, our approach is able to initialize the color regions and to adapt the regions with regard to the changing lighting conditions in a completely unsupervised manner. It is divided into two different steps: The first step determines the initialization, the second step tracks the color regions. As it is an unsupervised approach, the user does not have to deal with parameter adjustment or an extensive calibration process at all. Only three prerequisites must be met:

- The colors of interest must be known with regard to a very rough rectangular estimation in the UV dimensions of the YUV color space.
- Objects of interest have to be colored fairly homogeneously.
- Changes in lighting conditions are not abrupt (e.g. turning on and off floodlights), but provide a kind of smooth transition.

The initialization and the tracking are working on two different color spaces: (i) UV as the projected subspace from the YUV color space and (ii) H-RGB as RGB color space enhanced by the H dimension of the HSV space. These color spaces have proven to be very suitable for these two problems. UV is used by the initialization algorithm providing a kind of partitioning of the color space with regard to the identification of dense regions, i.e. regions that represent a major number of pixels. Therefore, the two-dimensional (yet the full color information containing) UV color space is chosen, as dense regions emerge with higher probability in a low-dimensional color space. For the tracking algorithm the situation is different. As the regions for the different colors are tracked separately, it is necessary to optimize the spatial distribution of the pixels in the color space, in order to avoid melting of one color region into another. Therefore, the four-dimensional H-RGB color space is used which proved to provide a sufficient spatial distribution. The next two paragraphs present the algorithms in detail.

3.1 Initialization

To establish the initial mapping of color regions to color labels, the following algorithm is executed before the game or mission. It consists of six steps applied only to the UV dimensions of the YUV color space, as mentioned above. The Y dimension is discarded to be as independent from illumination influences as possible. It is only used in a preprocessing step to discard too light or dark pixels, i.e. pixels with an Y value beyond certain thresholds.

1. A 256×256 UV-histogram is created from a number of images that contain objects and the colors of interest. About two to five images are required here.
2. The histogram is logarithmized and smoothened with a gaussian low-pass filter ($\sigma = 1,7$, mask size $= 5$ pixel). This suppresses local maxima which are not relevant for the further processing steps.
3. Each of the remaining local maxima represents an initial color region. The color values are assigned to a region represented by a local maxima using a Hill Climbing algorithm. If the hill climbing path ends at a point not assigned to a color region, a new initial color region is created. Thus, no method is required to detect the local maxima upfront.
4. The previous steps usually produce $10 - 20$ initial color regions. Many of these regions contain irrelevant colors and thus have to be considered as noise. They can be eliminated by applying very simple heuristics:
   (a) Color regions consisting of only a very small number of pixels are said to be noise.
   (b) Color regions formed by pixels which are widely distributed over the images can be discarded as well. This assumption holds, because the surface of an object most probably extends on a quite compact area within the image. The standard deviation of the distances of the pixels in the image to the center of the surface is one measure for compactness we use. (Different others are also possible, though.)
5. Color labels are assigned to the remaining initial color regions. These labels are determined by a very rough rectangular estimation of the colors of interest in the UV space. The label is the ideal color lying next to the center of the color region.

3.2 Iterative Tracking of Color Regions

The iterative color tracking adapts the pre-initialized color regions over time to overcome changes in the lighting conditions. On the one hand it must be possible to modify a region's size and position in the color space. On the other hand, color regions with different labels must not be merged. The algorithm must be able to adapt to color displacement which is, for example, caused by clouds that are passing by. Abrupt changes in lighting conditions are not considered here. To optimize the spatial separation of the color regions, a new dimension is added to the RGB color space: The hue value (H dimension) provided by the HSV color space. It represents the angle of the color in the HSV color circle and thus introduces a linearly independent component. In this four-dimensional color space our algorithm performs the computations described below.

1. A set of preliminary color regions of interest, as provided by the initialization step, is assumed. Several images, e.g. five, are taken to form data pools for each color of interest. All pixels of the images with a color value contained in a color region are inserted into the corresponding data pool. All elements of the data pools are then transformed into the H-RGB space. The resulting data cloud for a data pool is then examined by its location and size and represented through uniformly distributed centers. The locations of the centers are calculated hierarchically. For each data cloud a sum-histogram of all H-values is generated. Adjacent H-values with a relative frequency above a given threshold form ranges within this histogram. These ranges are divided into equidistant bins with a predefined width. The last bin in each range may be smaller. In the next step, sum-histograms of the R-values are generated for each bin. This procedure is applied to each dimension of the H-RGB color space. As a result, hypercubes within the H-RGB space are formed. In order to reduce noise, the density of data points within a hypercube must exceed a given threshold. The distance between two centers is given by the bin width for each direction. The result of this first step is an aggregation of hypercubes in the H-RGB color space for each color label.
2. To determine the new color segmentation, the location of each pixel is now examined: If it is in the proximity of a center, it is assigned the color label of this center. The proximity of a center is defined as the surrounding hypercube.
3. The following procedure is applied to every n-th image. n depends on the probability of changes in the lighting conditions, $n = 50$, for example.
   New hypercubes are defined for all centers of a color label. They are chosen somewhat larger than for the color segmentation in the previous step. This allows to take pixels into account which are not represented yet. The size of the hypercubes must not be chosen too large in order to keep separate color regions disconnected.
   All pixels of one or more images are examined whether they are represented by a label's hypercube. If this is true, they are stored in a data vector for this label. To retain the history of past images to some degree, at most 60% of the old vector elements are overwritten. As in the first step, an aggregation of hypercubes in the H-RGB space is created for each data vector. The segmentation of the next image is based on these new aggregrations.

4 Experimental Evaluation

For the experimental evaluation of our approach we use the RoboCup scenario and the vision systems of a team of soccer robots that participated in the RoboCup World Championships 2006. The basic challenge of these vision systems is to detect the color-marked objects, like the ball (red) and the goals (blue and yellow). These systems are commonly used for more elaborate tasks like detection of teammates and opponents or the extraction of features used for self-localization as well. In our evaluation, however, we only focus on tracking the color regions for the ball, the goals, and for the green playing ground.

In order to be able to assess the applicability of our approach, we use a set of test images from different locations with completely different lighting conditions and different dynamics of the changes in the lighting. Corresponding to the two separate parts of our approach, the evaluation of the initialization and the tracking is presented separately, as well.

For the evaluation of the **Initialization** of the color regions, we use a test set of 35 images from three different situations: (i) Playing ground at the RoboCup world championships 2006 in Bremen, Germany, (ii) a lawn in front of a building of the University of Kassel with natural lighting conditions, and (iii) the same lawn on another day with a more closed aperture. First we performed a manual segmentation of the images with the help of a calibration tool. Afterwards the images are also processed with our unsupervised initialization approach. In addition, for each of the images masks are provided that only include the pixels of the objects of interest. In order to estimate the quality of our initialization approach, three different values are calculated:

1. Percentage of pixels of an object mask that are segmented correctly (**Coverage**)
2. Percentage of pixels associated to a color label by the manual segmentation that are assigned to the same color label by the unsupervised approach (**Agreement**)
3. Percentage of pixels associated to a color label by the unsupervised approach but not assigned to the same color label by the manual segementation (**Disagreement**)

Coverage indicates how well the objects of interest are covered by the segmentation. *Agreement* and *Disagreement* allow a comparison between the manual segmentation and the unsupervised approach. A high *Agreement* value and a low *Disagreement* value indicate that the manual and unsupervised approach provide very similar segmentation results. The average values for the 35 images are shown in table 1.

Table 1. Comparison of the unsupervised initialization approach with a manual segmentation.

	red	**blue**	**yellow**	**green**
Coverage	53%	77%	93%	89%
Agreement	40%	78%	81%	88%
Disagreement	22%	23%	6%	10%

The numbers show that big parts of the blue and yellow goals and the green playing ground are classified correctly, and also that the results provided by the unsupervised and the manual approach are very similar for these colors. The only exception is the red ball. Here only about fifty percent of the surface is covered and the manual and unsupervised approach differ quite a lot. This can be explained by an overexposure of the ball surface in a number of images which even makes the manual segmentation very difficult, and the quite small number of red pixels in comparison with the other colors of interest. However, for the purposes of our vision systems these values are very sufficient.

In order to illustrate the results of a segmentation which is purely based on the initial color regions, figure 1 shows three examples. In the first row, the original images are shown. It is obvious that they exhibit completely different lighting conditions. The second row shows the initial color regions in the UV space determined with our approach, and the third row shows the resulting segmented images.

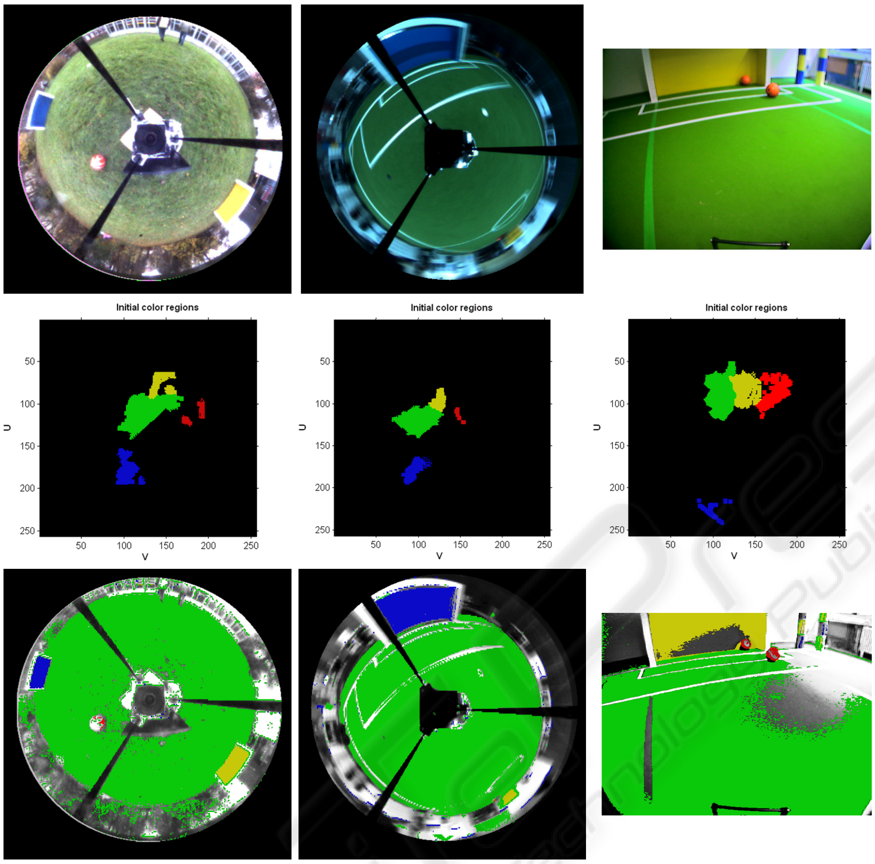

Fig. 1. Image segmentation based on the initially determined color regions.

The initialization is performed once before the game or the mission. Therefore, the performance of our algorithm is not relevant. However, the approach proved to be quite efficient. The average processing time for performing the initialization is about 170ms.

The **Iterative Tracking** was evaluated using 326 pictures taken in our laboratory, one every 100 seconds. The robot was equipped with a directed camera that was aligned to one fixed scenery on the field. The lighting conditions changed over the day since both, artificial and natural light sources were present. Natural light came through a window front nearby the field. As an example, we analyzed the effect of changing lighting conditions with regard to the deviation of all yellow pixels in the H (color) and V (brightness) dimensions of the HSV color space. Within a time span of 100 sec the maximum deviation of H was 2° and 13 units in the H dimension, within 300 sec 4° in the H dimension and 19 units in the V dimension. The iterative tracking approach was able to follow the changes for the whole day. We started with initial color regions provided by our initialization approach and all the pictures got segmented appropriately.

In order to have an exact evaluation of what our tracking approach is capable of, we manually modified pictures of the directed camera and pictures taken by RoboCup robots equipped with an omnidirectional camera. The test set consisted of 15 pictures and we considered both indoor and outdoor sceneries:

First, we shifted the H-value of the HSV color space until the iterative tracking produced wrong or unusable results. The same was done for the V-value. With our test set, displacements of up to $\pm 10^\circ$ in the H-value and up to ± 20 units in the V-value are compensated. Colors of very homogeneous surfaces are not tracked anymore in case of higher derivations. Figure 2 illustrates the benefits of our tracking approach when shifting the H dimension of the HSV color space by $+10^\circ$. The original picture is shown in the left, the middle picture is the segmented version of the artificially modified one without iterative tracking. The blue and yellow goals are not covered completely. In the right picture, iterative tracking is enabled; both goals are now covered completely. The results are nearly the same for changes in the V-value.

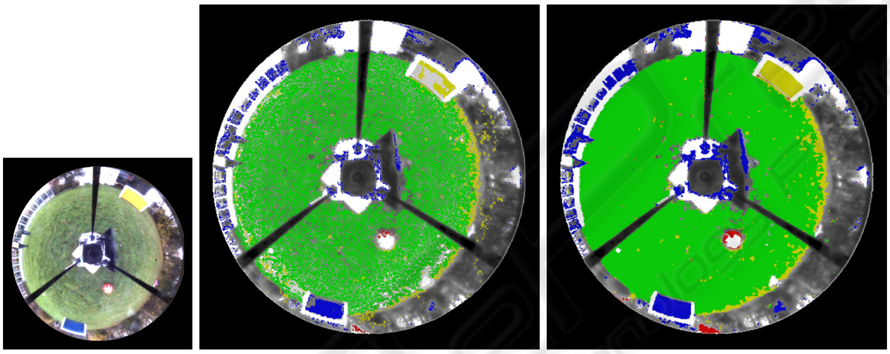

Fig. 2. Effect of the iterative tracking when shifting the H dimension of an image by $+10^\circ$.

We evaluated the pictures with regard to the three measures *Coverage*, *Agreement* and *Disagreement* in the same way as already presented with the evaluation of the initialization step. Tables 2 and 3 show the values for the segmentation results with disabled and enabled tracking.

Table 2. No tracking.

	red	blue	yellow	green
Coverage	14%	79%	43%	63%
Agreement	6%	64%	41%	65%
Disagreement	55%	75%	81%	1%

Table 3. Iterative tracking.

	red	blue	yellow	green
Coverage	43%	99%	97%	90%
Agreement	14%	83%	92%	93%
Disagreement	30%	80%	69%	1%

The numbers show that with enabled tracking the object *Coverage* is notably increased for all four colors. This can be observed for the *Agreement* value as well. In addition, the values for *Disagreement*, in particular for yellow and blue, are quite high. However, this only indicates the fact that the unsupervised tracking approach selects more pixels of the environment in comparison with the manaul segementation. Due to the same reasons as already mentioned with the initialization approach the results for the ball are not of the quality that can be observed for the other colors. It has also to be considered that in the tracking approach, parts of the color space with a very few number of pixels are regarded as noise. Of course, this effect is more prominent with color regions which in total contain only a few number of pixels, as e.g. red.

Another benefit of our tracking approach is the ability to cope with sub-optimal initial color regions and to improve the color regions within some tracking steps. As shown in figure 1 the initialization failed to provide optimal initial color regions for the image of the directed camera. Parts of the yellow goal are missing, the ball is not covered completely and particularly big parts of the green field are not segmented appropriately. Figure 3 illustrates the segmentation results after some tracking steps. The left picture shows the original image again. In the middle the segmenation results based on the initialization is presented. The results after 2 tracking steps are shown on the right: all three features, the yellow goal, the red ball and the green field are now covered almost completely. The average processing time for one iterative tracking step (executed on every 50th image, for example, which roughly means every 2 seconds assuming a camera capturing pictures with 30Hz) is between 50 ms and 100 ms in our test. It depends on the number of calculated centers. So the processing time is very short if the colors are homogeneous.

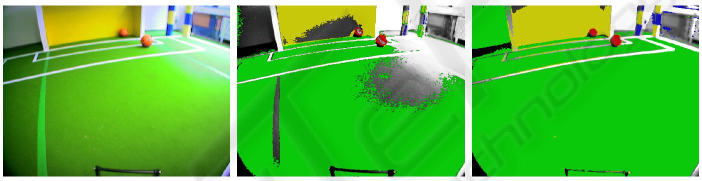

Fig. 3. Improvement of sub-optimal initial color regions through iterative tracking.

5 Conclusion and Future Work

In this paper we have presented an unsupervised approach for adaptive color segmentation which is able to deal with varying lighting conditions. The approach comprises two different steps: An initialization step provides initial regions for the colors of interest. These regions are iteratively tracked during run-time to be adjusted to changes in illumination. As presented in section 2 there are some other approaches that are able to provide calibration data for color segmentation automatically. However, some of these approaches are static and not able to deal with varying lighting conditions. Others provide this ability but are coupled with object recognition approaches, rely on form information, or a number of scenario-dependent assumptions. In contrast, our approach only needs three prerequisites to be fulfilled: a rough rectangular estimation of the color regions in the UV space, homogeneously colored objects, and fairly smooth transitions in the lighting conditions. However, these very basic prerequisites can be assumed in most cases. Our approach has revealed to be very powerful and is applicable for omnidirectional vision systems and for vision systems with a directed camera as well. The experimental evaluation has also shown that the initialization provides appropriate initial color regions for a number of different lighting conditions. The iterative tracking is able to follow the changes in lighting conditions that can be observed during a whole day. Several different methods might be suitable to improve our approach to be able to deal with abrupt changes in lighting conditions. One possibility is to run the initialization step in some time intervals during run-time and to compare the resulting color regions with the tracked ones. If the differences are too big, the color regions provided by the initialization algorithm are used for further tracking. This would also help to make the algorithm more stable and would prevent situations where the tracking algorithm fails because of two or more melted regions in the H-RGB color space.

References

1. Federico Anzani, Daniele Bosisio, Matteo Matteucci, and Domenico G. Sorrenti. On-line color calibration in non-stationary environments. In *RoboCup 2005 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence, pages 396–407. Springer, 2006.
2. Claudia Gönner, Martin Rous, and Karl-Friedrich Kraiss. Real-time adaptive colour segmentation for the RoboCup middle size league. In *RoboCup 2004 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence, pages 402–409. Springer, 2005.
3. P. Heinemann, F. Sehnke, and A. Zell. Towards a calibration-free robot: The act algorithm for automatic online color training. In *RoboCup 2006 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence. Springer, 2007. to appear.
4. Matthias Jüngel. Using layered color precision for a self-calibrating vision system. In *RoboCup 2004 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence, pages 209–220. Springer, 2005.
5. Matthias Jüngel, Jan Hoffmann, and Martin Lötzsch. A real-time auto-adjusting vision system for robotic soccer. In Daniel Polani, Brett Browning, and Andrea Bonarini, editors, *RoboCup 2003 - Proceedings of the International Symposium*, volume 3020 of *Lecture Notes in Artificial Intelligence*, pages 214–225, Padova, Italy, 2004. Springer.
6. Gerd Mayer, Hans Utz, and Gerhard K. Kraetzschmar. Towards autonomous vision self-calibration for soccer robots. *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2002)*, 1:214–219, September-October 2002.
7. RoboCup Official Site. http://www.robocup.org/.
8. M. Sridharan and P. Stone. Autonomous planned color learning on a legged robot. In *RoboCup 2006 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence. Springer, 2007. to appear.
9. Mohan Sridharan and Peter Stone. Towards eliminating manual color calibration at RoboCup. In *RoboCup 2005 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence, pages 673–681. Springer, 2006.