Benchmarks for Robotic Soccer Vision

Ricardo Dodds[1], Luca Iocchi[1], Pablo Guerrero[2], and Javier Ruiz-del-Solar[2]

[1] Dipartimento di Informatica e Sistemistica, Universita "La Sapienza", Rome, Italy
{dodds,iocchi}@dis.uniroma1.it
[2] Departament of Electrical Engineering, Universidad de Chile, Chile
{pguerrer,jruizd}@ing.uchile.cl

Abstract. Robotic soccer vision has been a major research problem in RoboCup and, even though many progresses have been made so that, for example, games now can run without many constraints on the lighting conditions, the problem has not been completely solved and on-site camera calibration is always a major activity for RoboCup soccer teams. While different robotic soccer vision and object perception techniques continue to appear in the RoboCup Soccer League, there is a lack of quantitative evaluation of existing methods.

Since we believe that a quantitative evaluation of soccer vision algorithms will led to significant advances in the performance on perception and on the entire soccer task, in this paper we propose a benchmarking methodology for evaluating robotic soccer vision systems. We discuss the main issues of a successful benchmarking methodology: (i) a large and complete data base or data sets with ground truth; (ii) a public repository with data sets, algorithms and implementations that can be dynamically updated and (iii) evaluation metrics, error functions and comparison results.

Keywords: Benchmarking and Evaluation, Color-based Object Recognition, Robotic Soccer Vision.

1 Introduction

We can generally agree that the camera is the most important sensor to interact with the outer world. The main objective of the perception module is to process the raw image and extract useful information. The inputs for a vision system are images of the camera and sensor readings from internal robot sensors, the outputs are the relative positions of all recognized objects to the robot.

Over all the visual tasks that a computer may perform, recognizing an object present in an image could be the most challenging one. Not only a large number of object categories exist, but also each class can reach a wide number of configurations. In a single image one could find a large number of objects. In the case of video analysis, and tracking several objects in a dynamic environment, things can get worse.

T. Röfer et al. (Eds.): RoboCup 2011, LNCS 7416, pp. 427–439, 2012.

In the RoboCup Soccer League environment, the number of objects of interest is small and there is enough a priori information about their size, color and position (in the case of static landmarks), but still recognizing them remains a complex task. Over the years, in Robocup Soccer competitions, there have been several proposals to solve the problem of object perception. It is possible to find a wide variety of publications about the main stages in object recognition in color coded environment. Since objects of interest have characteristic colors, most of these attempts are based on color vision. Moreover, although code sharing in RoboCup (specially in the Standard Platform League) allows in principle to use and test approaches made by other research groups, this did not occur as desirable.

Indeed, regardless the large amount of work in the area and the fact that a quantitative evaluation is essential to ensure progress, very little has been done on performing a systematic comparison among these approaches. This is mainly due to the lack of a benchmark methodology and of common data sets. Consequently, almost every paper (see next section) provides its own results on its own sequence of images.

Conversely, many other research areas have defined standard benchmarks that are commonly used when introducing new methods. For example, within the SLAM community reference data sets and the most important algorithms are available on-line[1] (often as open-source projects), so that it is easy to directly evaluate a new algorithm with respect to existing ones. Also in the Computer Vision community, many data sets are available for method evaluation: for example, PETS[2] is used for people tracking and human activity recognition.

The main objective of this work is thus to define a benchmark methodology for robotic soccer vision systems, developing both a database and an evaluation methodology. The benchmark can be useful for any RoboCup Soccer League and also for general color-based object recognition. Although at the moment only object recognition and localization is considered, the benchmark methodology can be extended to other relevant issues in robotic soccer vision (like activity recognition, anomally detection, etc.).

Having a challenging data set helps to track progresses made by current algorithms, stimulates new ideas, improves code sharing and the overall progress of the league. In order to realize such an effective benchmark methodology for soccer robots, the data set must be well organized and sufficiently large (i.e., in use by most/all research groups), the evaluation methodology should take into account different performance metrics, the repository should contain also algorithms (and their implementation), possibly in open-source, with a clear input/output definition, so that can be actually used by the research community.

The paper is organized as follows. In Section 2 we describe the main methods that have been proposed. We give the specifications of the Data Set in Section 3 and introduce the Data Set Description concept. In Section 4 we describe the evaluation methodology for soccer vision systems and its implementation in Section 5. We summarize our work and discuss future directions in Section 6.

---

[1] `radish.sourceforge.net` and `openslam.org`

[2] `www.cvg.rdg.ac.uk/PETS2010/`

2 Overview of Robotic Soccer Vision

Even if the scope of this paper is not to give a full review of soccer vision algorithms, we should describe the most important stages and approaches of a general perception module. Without losing generality, a perception module can be decomposed into two stages: (i) low-level vision, that takes as input a raw image and outputs a set of region candidates and (ii) high-level vision, which uses the regions to determine the 3D relative position of available objects. Despite their differences, many of the proposed techniques can be viewed conceptually in terms of these two stages of processing[3].

In the low-level stage, a bottom-up processing of the image is performed, where image pixels are analyzed in order to extract useful information. This is generally the most computationally expensive task, and at each stage the amount of information that will be processed further is reduced. The first important step is the color segmentation, which uses a color table (look-up table) to map pixels from raw image values to a class of symbolic colors, considerably reducing the amount of information per pixel from 256^3 to the number of classes $|C|$ (normally 8 or 9). Several attempts has been made in order to improve this task. In [3] they use a evolutionary algorithm to convert YUV color space into a easy-to-separate one facilitating the classification task. Following previous work, in [16] colors are not only mapped to unambiguous but also to ambiguous color classes. The ambiguous color classes are resolved based on their unambiguous neighbors in the image. In contrast to other approaches, the neighborhood is determined on the level of regions, not on the level of pixels. Another interesting method is proposed in [6] where they use knowledge of spatial relationship between color classes to calibrate them starting from another classes.

In the early stage of a color-based vision system, the color constancy problem is tackled. As mentioned in [13], variations on the light conditions have a strong effect on the spatial distribution of target color classes. This is known as the color constancy problem. We can agree that manual calibration is a time consuming task and may lead to errors. As lighting will always be different on each playground (even at different times of the day in the same place), teams started developing automated vision calibrations routines. In the literature, there are described two ways for improving color constancy dynamically: (i) adjusting camera parameters or (ii) modifying the color table. These algorithms try to stabilize color recognition, so that the same colors look almost the same under different illuminations.

One of the simplest solutions, presented in [9], is to calculate in each frame the intensity of the image and accumulate an average value in order to detect if there are variations and adjust the gain parameter if necessary. In [5] a genetic algorithm is used to optimize the camera parameters instead of modifying color table.

[3] This decomposition assumes that it is possible to define a valid interface between the two levels, general enough to describe all possible configurations of a perception module.

Modifying the color table is the most used approach. In [17] is proposed a simple solution that defines three different lighting conditions: bright, intermediate and dark, and using KL-divergence compares the actual illumination condition and determines which color table to use. The most common method, described in [10] is to adapt the color table using statistics of recognized objects. Similarly, [7] and [1] use geometrical models of field and objects to identify landmarks independently of color classifications. Color information from these landmarks is used build color classes. An adaptive color calibration based on the Bayes Theorem, using chrominance histograms and object shapes to update them is proposed in [4]. Also [11] uses different layers of color representation and updates the color look-up table for each layer using information about recognized objects. A novel approach is presented in [8], where an adaptive transformation of the color distribution of the image is used to adjust static thresholds.

On the other hand the high-level vision module performs a top-down image analysis, using features provided by low-level vision. The main objective is to find objects of interest and estimate their properties. In this stage contextual information and expectations of objects that might be in the image could be used. Usually starting from a list of region candidates of the appropriate color, binary rules are applied to discard false perception. Physical features as size and shape are used to quickly filter wrong candidates. Subsequently deeper analysis are performed. In [14] a context-based vision system is presented. In this work, perception of static objects is improved using a bayesian framework that considers different context-coherence instances. Also, machine learning is a widely used technique in soccer vision. A decision tree learning algorithm, used for compute the pose of visible robot, is described in [18].

So far, we have described algorithms strongly based on a robust color segmentation of the image. Nevertheless there have been proposed different approaches that focus its efforts on high level processing. In [12], assuming that a color core exists independently of the illumination condition, they only use sparse color classification, i.e. name a color blue only if you are sure it is going to be blue for all illumination conditions. Sequently, using edge detection from blobs and counting the number of color pixels inside, the blob color is defined.

Notice that our review and further analysis is only restricted to regular cameras. Although systems running with omnidirectional cameras, share common elements in the perception process, they use techniques that go beyond the scope of this paper.

3 Data Set

The benchmark we are proposing is composed by two main topics: the specification and the gathering of the data sets and the methodology for evaluating and comparing different approaches. The former is discussed in this section, while the latter in the next one. A data set is given by a continuous sequence of data taken with the on-board robot sensors (including the camera) and possibly with additional external sensors that are used to provide ground truth. These data are taken in a specific setting that is described through a Data Set Description.

3.1 Data Set Descriptions

In order to motivate future progresses in robotic soccer vision, well-organized and challenging data sets are required. Data sets must contain both input data to be processed by the algorithms and output data (ground truth) to evaluate the results. Data sets should also consider the many different variabilities that occur in real applications. In this work, we propose to define the characteristics of each data set with a formal *Data Set Description* (DSD) formed by the following items:

- **ID of the robot**, defines from which robot data were taken (e.g., NAO-48),
- **sensor used**, describes which sensors (both on the robot and external sensors) are used to collect data (e.g., robot camera, robot joints, external camera),
- **scene**, describes the environment where the data set has been taken (e.g., Singapore 2010 SPL-Field A),
- **action**, describes the game situation referring to soccer actions (e.g., penalty kick against blue goal),
- **light**, describes the light conditions and whether videos where taken with natural or artificial illumination (e.g., Singapore 2010 venue illumination),
- **occlusions**, describes additional occlusion if any (e.g., other robots or people in the field),
- **shadows**, describes presence of shadows (e.g., due to people around the field),
- **robot dynamics**, specifies if the robot capturing the images is fully, partial (i.e., only the head) or not moving (e.g., still robot tracking the ball with its head),
- **environment dynamics**, describes other elements moving in the field (e.g., ball and other robots moving around).
- other information, such as date and time of capture, duration, author, etc.

Each data set is associated to a DSD that will be useful also for partial analysis: for example, evaluating methods in particular conditions by using only the corresponding data sets. Moreover, DSD are extensible, thus additional features and relevant information can be added in the future, responding to additional uses of the benchmark.

3.2 Data Set Acquisition

Once the characteristics of the data set have been defined, a careful data acquisition procedure must be performed. Since data will be used to evaluate performance of soccer vision methods, three kinds of information must be collected: (i) the images acquired by the robots, (ii) some internal states of the robots (e.g., joints configuration for humanoid robots) in order to determine the pose of the camera with respect to the body), (iii) external sensors used to measure ground truth (in particular the pose of the robot in the environment).

The difficulties in data set acquisition are: 1) acquisition of precise ground truth, 2) data synchronization. Since the field of action of the robots is quite limited, a precise ground truth can be obtained by placing sensors around the field. Both top-view vision cameras and laser range finders can be used for this task (see for example [2]). In case of external vision cameras, we can either place markers on the robot and use an automatic robot pose estimation, or label images by hand. With laser range finders, the use of two or three of them allows for dealing also with multiple robots and occlusions. For data synchronization, internal clocks of the robots can be used to synchronize images and robot states, while if external videos are taken for ground truth, a flashing light pattern, visible from both the robot camera and the external camera, can be used to determine synchronous frames.

In this paper we show some examples of data sets including robot camera videos, robot joints and top-view camera, as shown in Figure 1.

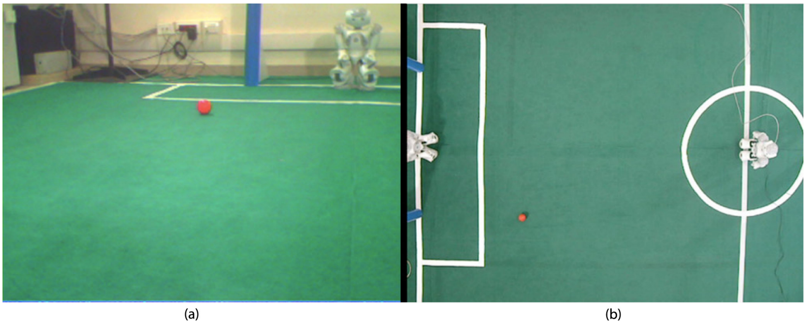

Fig. 1. *Video acquisition* from NAO camera (a) and from a top view camera (b)

3.3 Data Set Labeling

The third step in the Data Set collection consists in providing the ground truth that is needed to evaluate algorithm performance. Since our main objective is the evaluation of performance of soccer vision methods, whose main objective is to recognize field elements (ball, goals, lines, robots,...) and determine their position with respect to the robot, it is necessary to label in each frame all the objects of interest. On the other hand we are not interested in a pixel based recognition, thus we believe it is not necessary to label single pixels in the image.

Therefore, we propose to proceed as follows.

First, we define the elements of interest: *ball*, *goal post*, *goal cross-bar*, *entire goal*, *field line*, *field corner*, *robot*. Then we decide to associate to each of these elements a planar shape. For simplicity, we will use ellipses for the *ball* and quadrilaterals for all the other elements.

Second, a manual annotation of the frames is performed. To this end we developed a tool to simplify this process: it allows the user to move frame by frame within the data set, declare an element type and click over four points that will be used to determine a bounding box (an ellipse in case of the ball) for that element. Examples of such labeling are shown in Figure 2.

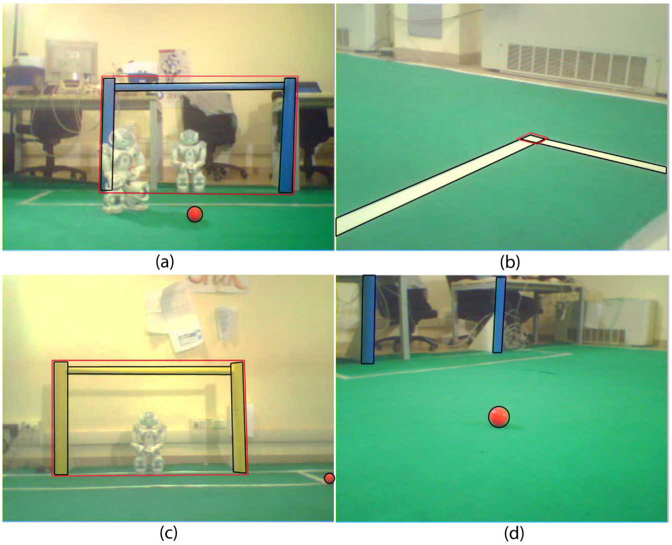

Fig. 2. *Labeling objects of interest.* (a),(c) both entire goals and it parts are labeled using quadrilaterals, as well as lines and corners (b); (d) ellipses are used to label the ball

In case an external camera is used to get videos for the ground truth and automatic detection is not available, also the videos taken from this external camera need to be labelled. For simplicity, we decided to label only the moving objects: the ball and the robots players, including the one that is taken the data set. For the robots we also determine the orientation, although this information will not be very precise. Thus the relative position of the ball with respect to the robot taking the data set can be computed by transforming the ball location from a field reference system to the robot local reference system. While we do not label static elements, thus the distance from the robot taking the data set and the static elements in the field (if needed) may be computed afterwards by knowing the geometry of the field.

Positions and poses extracted by the external camera are provided in metric units within a field reference system. It is also desirable that the external camera introduces as less distortion as possible. In case of high lens distortion, a pre-calibration procedure is suggested, and images in the data set should be provided undistorted.

It is also important to notice that it is unavoidable that this measure introduces small errors when the human operator clicks over an object. In order to estimate this error, we asked several people to repeat the experiment of annotating the ball and a robot within a data set. Results are shown in Table 1.

From these results we can compute standard deviations of human errors in labeling that will be taken into account in the evaluation.

Table 1. Statistics on clicks over objects

	#	1	2	3	4	5	6	7	8	9	10	
Ball	x[m]	1.081	1.074	1.081	1.070	1.062	1.074	1.074	1.081	1.085	1.074	
	y[m]	1.986	1.982	1.974	1.982	1.982	1.986	1.982	1.982	1.986	1.978	
Robot	x[m]	3.155	3.144	3.155	3.152	3.175	3.140	3.163	3.163	3.167	3.224	
	y[m]	1.365	1.353	1.361	1.353	1.361	1.361	1.350	1.342	1.353	1.361	
	theta[o]	181.2	180.6	180.3	180.9	182.0	181.5	179.8	179.3	181.5	180.1	
	#	11	12	13	14	15	16	17	18	19	20	σ
Ball	x[m]	1.081	1.070	1.074	1.081	1.085	1.074	1.070	1.070	1.074	1.074	0.006
	y[m]	1.971	1.982	1.994	1.982	1.974	1.997	1.986	1.982	1.982	1.986	0.006
Robot	x[m]	3.244	3.175	3.190	3.186	3.140	3.201	3.198	3.198	3.224	3.178	0.029
	y[m]	1.342	1.350	1.353	1.342	1.342	1.350	1.353	1.350	1.353	1.350	0.007
	theta[o]	179.7	179.1	180.4	182.2	181.3	179.5	182.0	180.4	180.7	179.3	0.967

3.4 Data Set Statistics and Data Base

Finally, once labeling is done, it is useful to extract statistics from each Data Set: for example, the percentage of frames in which a ball is visible, etc. To this end we have realized a tool that can extract different statistics from a data set.

It is important to notice that this formalization of the data sets allows for creating a real data base of data sets, that can be effectively queried with a simple query language (for example, through a Web interface). In this way it will be possible to retrieve, for example, all the data sets regarding penalty kicks in natural light conditions, with a high percentage of frames containing both the ball and the yellow goal.

4 Evaluation Methodology

In order to evaluate the performance of soccer vision algorithms in a quantitative way, their outputs have to be compared to the ground truth using pre-defined error metrics. Here we propose a *per-image* measure methodology, frequently used in object detection. The evaluation determines the following measures:

– *false positives* and *false negatives* in object detection
– *object detection precision* in the image space
– *object detection precision* in the world space
– *computational efficiency*

We have written an automatic measurement tool that can perform the evaluation of a method, provided that its output is given in a pre-defined format (which is described in the benchmark web site).

In particular, the evaluation compares, for each object of interest that the algorithm declares to detect and for each frame in the data set, the output of the algorithm with the ground truth. For objects, like goal posts and field lines, that may appear multiple times in an image, we evaluate them for each occurrence in the frame. For example, if a frame contains two goal posts and the algorithm returns only one, the missing one will be counted as a false negative.

4.1 False Positives and False Negatives

The first measure we consider is to evaluate the quality of the vision method in detecting the objects of interest in the images.

Let us denote with O the set of objects of interest, and with $o \in O$ a specific object, I_t the frame taken at time t, T the set of all time-stamps, $N = |T|$ the total number of frames, $\delta_G^o(t)$ the number of occurrences of o in I_t in the ground truth, $\delta_A^o(t)$ the number of occurrences of o in I_t in the output of the algorithm, N_G^o the number of frames in which there is at least one occurrence of o in the ground truth, N_A^o the number of frames in which there is at least one occurrence of o in the output of the algorithm.

Many measures can be used to measure the quality of object detection, and while we believe the following ones are the most significant, others can be considered as well in the evaluation methodology.

True positive rate (or detection rate) and *false positive rate* measure accuracy of object detection for a given object o and are defined as

$$TPR^o = \frac{\Sigma_{t \in T}(\min(\delta_A^o(t), \delta_G^o(t)))}{N_G^o}$$

$$FPR^o = \frac{\Sigma_{t \in T, \delta_A^o(t) > \delta_G^o(t)}(\delta_A^o(t) - \delta_G^o(t))}{N}$$

4.2 Object Detection Precision in Image Space

For correctly detected objects, i.e. for all those frames where $\delta_G^o(t) > 0$ and $\delta_A^o(t) > 0$, precision measures will be performed. For objects occurring multiple times a data association problem must be solved in order to match outputs of the method with the ground truth. This must be done even in presence of false positive or false negatives detection. For example, if a frame contains three lines and the algorithm return two of them, one element is counted as a missing detection, but for the other two precision must be computed by comparing them with two of the lines in the ground truth. Data association is solved with a best scoring approach, so the association that returns the best score is chosen. Note that, although this process requires the evaluation of $\delta_G^o(t) > 0 \times \delta_A^o(t) > 0$ combinations, this is in fact a small number, since $\delta_G^o(t)$ is typically limited.

Let us denote with $\Gamma_t^{o_i}$ the bounding box of the i-th occurrence of o in I_t in the ground truth, and $B_t^{o_i}$ the bounding box of the i-th occurrence of o in I_t in the method output. Let $f(i)$ be the association function between the occurrence of an object in the method output and in the ground truth, that returns the best score. Then precision in the detection of the i-th occurrence of o in I_t by the method, which is associated of the $f(i)$-th occurrence of o in I_t in the ground truth, is given by following the PASCAL ([15]) measure

$$\alpha_t^{o_i} = \frac{area(B_t^{o_i} \cap \Gamma_t^{o_{f(i)}})}{area(B_t^{o_i} \cup \Gamma_t^{o_{f(i)}})} \tag{1}$$

The average precision in detecting o in all the data set is given by

$$\alpha^o = \frac{\Sigma_{t \in T} \Sigma_{i=1,..,\delta^o_A(t)} \alpha^{o_i}_t}{\Sigma_{t \in T} \delta^o_A(t)}$$

4.3 Object Detection Precision in Field Space

The third evaluation stage is made using the real positions of the object, provided in the data set. Again for each object for which the algorithm returns a position in the world (in robot local ground coordinate frames), the precision of such measure is evaluated by comparing the values in the data set ground truth.

As mentioned in the previous section, we have decided to label only the position of the ball and the poses (position and orientation) of the robots.

Let x^{ball}_t be the relative position of the ball with respect to the robot in the frame I_t computed by the method under evaluation, and χ^{ball}_t the relative position of the ball with respect to the robot in the frame I_t computed with the ground truth.

The evaluation of the location of the ball is based on the Euclidean distance between the two positions x^{ball}_t and χ^{ball}_t. We denote the distance between the tow positions with $||x^{ball}_t - \chi^{ball}_t||$ and the distance of the ball to the robot (i.e., the norm of χ^{ball}_t) as $||\chi^{ball}_t||$.

When this distance is within the standard deviation of the human error in annotating the ground truth σ_d, then this error is fixed to 0.

$$\epsilon^{ball}_t = \begin{cases} 0 & \texttt{if } ||x^{ball}_t - \chi^{ball}_t|| \leq \sigma_d \\ \left| \frac{||x^{ball}_t - \chi^{ball}_t||}{||\chi^{ball}_t||} \right| & \texttt{otherwise} \end{cases}$$

A similar measure is defined for poses, by considering both the position and the orientation error. Then, the average error over the entire data set is obtaining by averaging ϵ^o_t over time $t \in T$.

4.4 Computational Efficiency

The last performance measure is computational time of the method. Here it would be important to evaluate it in the actual CPU on which the method will run, thus on the robot CPU. In the case of standard platforms, this is very easy and all the methods can be directly compared. In case of self-built soccer robots, the specification of the CPU, memory and other characteristics of the processing unit on board the robot needs to be provided as well.

5 Benchmark Implementation

The benchmarking methodology described in this paper has been implemented by developing a collection of tools and data sets and by making them available through the web site
`http://labrococo.dis.uniroma1.it/RobotSoccerVisionBenchmark/`

The web site contains a full description of the components of the data sets, so that it is possible and easy to contribute to it by extending its scope.

At the moment this paper is completed, the benchmark contains 20 data sets, corresponding to about 40 minutes of labeled videos taken with the NAO robots. Additional data sets will be added in the next future, provided also by other teams. Moreover, in order to simplify the use of the benchmarking methodology, we released the following software tools: (i) a tool for data acquisition for the NAO robots, (ii) a tool for labeling sequences of images or videos and (iii) a shared library that facilitates evaluating vision methods.

6 Discussion

It is well known that defining standard benchmarks and common performance metrics is a very important issue in order to improve the general performance of a research topic. In this paper, we propose a standard benchmark and performance metrics for one of the major RoboCup soccer problems: vision and object recognition.

Although we propose a specific methodology and performance metrics that we believe of general interest for evaluating the research in this field, the benchmark can be easily extended in order to capture other specific needs. For example, adding a new performance metric does not affect the collection and labeling of data sets; adding a new object to recognize or additional labels does not affect previous collections.

Moreover, the collection and the use of data sets must not be limited to a single research group, but should be extensively used ideally by all the groups doing research in the field. Therefore the main aim of this work is to involve as many research groups as possible to contribute to the collection of data sets and evaluation of methods, through the benchmark web site that will be constantly updated.

So far we have collected only a few data sets with the methodology presented above in our laboratories. This activity confirmed both the correctness of the proposed approach and of the developed tools and the effectiveness and easy-of-use of the methodology.

Future work is to collect contributions to this benchmark by other research groups, make it widely accepted and use it as a standard evaluation methodology for robotic soccer vision.

References

1. Cameron, D., Barnes, N.: Knowledge-Based Autonomous Dynamic Colour Calibration. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 226–237. Springer, Heidelberg (2004)
2. Ceriani, S., Fontana, G., Giusti, A., Marzorati, D., Matteucci, M., Migliore, D., Rizzi, D., Sorrenti, D.G., Taddei, P.: Rawseeds ground truth collection systems for indoor self-localization and mapping. Autonomous Robots 27(4), 353–371 (2009)

3. Dahm, I., Deutsch, S., Hebbel, M., Osterhues, A.: Robust color classification for robot soccer. RoboCup 2003: Robot World Cup VII, Lecture Notes in Artificial Intelligence (2004)
4. Gönner, C., Rous, M., Kraiss, K.-F.: Real-time Adaptive Colour Segmentation for the Robocup Middle Size League. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 402–409. Springer, Heidelberg (2005)
5. Grillo, E., Matteucci, M., Sorrenti, D.G.: Getting the Most from Your Color Camera in a Color-Coded World. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 221–235. Springer, Heidelberg (2005)
6. Guerrero, P.E., Ruiz-del-Solar, J., Fredes, J., Palma-Amestoy, R.: Automatic On-Line Color Calibration Using Class-Relative Color Spaces. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 246–253. Springer, Heidelberg (2008)
7. Gunnarsson, K., Wiesel, F., Rojas, R.: The Color and the Shape: Automatic On-Line Color Calibration for Autonomous Robots. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 347–358. Springer, Heidelberg (2006)
8. Iocchi, L.: Robust Color Segmentation Through Adaptive Color Distribution Transformation. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006. LNCS (LNAI), vol. 4434, pp. 287–295. Springer, Heidelberg (2007)
9. Jamzad, M., Lamjiri, A.K.: An Efficient Need-Based Vision System in Variable Illumination Environment of Middle Size RoboCup. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 654–661. Springer, Heidelberg (2004)
10. Jüngel, M., Hoffmann, J., Lötzsch, M.: A Real-Time Auto-Adjusting Vision System for Robotic Soccer. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 214–225. Springer, Heidelberg (2004)
11. Jüngel, M.: Using Layered Color Precision for a Self-Calibrating Vision System. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 209–220. Springer, Heidelberg (2005)
12. Lovell, N.: Illumination Independent Object Recognition. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 384–395. Springer, Heidelberg (2006)
13. Mayer, G., Utz, H., Kraetzschmar, G.K.: Playing Robot Soccer Under Natural Light: A Case Study. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 238–249. Springer, Heidelberg (2004)
14. Palma-Amestoy, R., Guerrero, P., Ruiz-del-Solar, J., Garretón, C.: Bayesian Spatiotemporal Context Integration Sources in Robot Vision Systems. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 212–224. Springer, Heidelberg (2009)
15. Ponce, J., Berg, T.L., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B.C., Torralba, A., Williams, C.K.I., Zhang, J., Zisserman, A.: Dataset Issues in Object Recognition. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 29–48. Springer, Heidelberg (2006)

16. Röfer, T.: Region-Based Segmentation with Ambiguous Color Classes and 2-D Motion Compensation. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 369–376. Springer, Heidelberg (2008)
17. Sridharan, M., Stone, P.: Towards Illumination Invariance in the Legged League. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 196–208. Springer, Heidelberg (2005)
18. Wilking, D., Röfer, T.: Realtime Object Recognition Using Decision Tree Learning. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 556–563. Springer, Heidelberg (2005)