社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-3 (5/4)

Physical Visualization Sub-League: A New Platform for Research and Edutainment

Rodrigo da Silva Guerra [1], **Joschka Boedecker** [1], **Minoru Asada** [1,2]
[1] Graduate School of Engineering, Osaka University, Osaka, Japan
[2] JST Erato Asada Project
{rodrigo.guerra,joschka.boedecker,asada}@ams.eng.osaka-u.ac.jp

Abstract

This work introduces a novel minirobotics system which is to become a new sub-league of the RoboCup Soccer Simulation League, called Physical Visualization. We incorporate mature technology proven efficient over the years in other RoboCup leagues and we introduce new collaborative development concepts into the games of this new sub-league shifting essential research issues from the playing agents themselves to the development of a new versatile research and educational platform. We describe in detail the technical aspects supporting this multi-agent robotic framework, integrating cutting-edge and low cost watch technology in the form of a miniature multi-robot system which mixes reality and simulation. Competition formats and roadmaps are presented and discussed and the advantages for education and research applications are emphasized. Finally we discuss benefits of this new platform in terms of standardization, flexibility and reasonable price and try to characterize and discuss the future possibilities enabled by this project and the place of this new sub-league within the RoboCup community.

1 Introduction

Physical Visualization (PV for short) is candidate to be a new RoboCup Soccer Simulation sub-league. The project is intended for fostering education, research and development together with the RoboCup community. The PV is based on a miniature multi-robot system which mixes reality and simulation through an Augmented Reality (AR) environment. The project has a two-folded focus: research and education. The main goals of the PV are:

- to gradually improve the platform so that it becomes a powerful and versatile standard for multi-agent research and education.
- to explore educational possibilities and real world applications based either on the system as a whole or on some parts of it (e.g. the robots alone).

Since March of 2006 CITIZEN Co. and Osaka University committed themselves to the endeavor of developing together with RoboCup a new miniature, and yet affordable, robotics platform. This comes against the main stream in robotics, where, in general, the solutions are costly. We focused on versatility and affordability, taking advantage of well established industry technologies to allow the development of an inexpensive platform. In order to do that we used the know-how of the cutting-edge and low cost watch technology as a basis for building an affordable miniature multi-robot system mixing reality and simulation. This allows the employment of a large number of robots in a rather reduced space with a very low budget and amazing portability. Both the robots and the system are to be constantly upgraded and improved, being developed together with PV and CITIZEN exclusively for the competitions. Three dominant characteristics of the project are: (a) affordability, (b) standardization and (c) open architecture. These aspects are explained in detail in the next paragraphs.

Affordability: Generally speaking, doing research on robotics is an expensive task, specially when it comes to multi-agent robotics. Even in the most inexpensive real-robot experiments, it is a common sense that one would expect to spend at least several thousands of dollars in order to have a multi-agent setup. For the main reasons, one could surely account for the unavailability of adequate commercial platforms, thus bringing the need for custom robots. In RoboCup the strong competition forces teams to challenge themselves to come up with new design ideas which quite often are translated into more complex and expensive hardware. This last factor also implies that a wider spectrum of technical fields needs to be covered for the complete design of the machines, including issues which are not always related to the research focus originally in mind. Such difficulties may seem inherent to the research track of some institutions, but they most likely come as an obstacle to those who do not have the man-hour and the money for the journey.

Standardization: Sharing a common standard platform allows the easy comparison of results and concepts into the same grounds. The two-dimensional environment of simulation league [8] is a successful example of such standardization: papers often show comparative results using the same common simulation environment, for example, playing against the code of a good team of former years (e.g. [7, 12, 11]). To a minor degree the four-legged league [15] also shares some of these characteristics as, for instance, the champion teams usually release their source code for the others to build on in the coming year, thus speeding the progress and avoiding the need for newcomer teams to "re-invent the wheel". In our understanding there was still a lack of a standard platform such ours, providing the flexibility of simulations but in real robots and at a reasonable price.

Open architecture: Our platform brings the above standardization in a completely open architecture with room for collaborative improvement. All program codes, including the computer software codes and the robots firmware are being released with the GNU GPL license [10]. Moreover, schematics are already being provided for all electronic circuits of the system.

2 Technical Aspects

The technical aspects of the main system are illustrated in the block diagram of figure 1 and on the simplified drawing of figure 3-a. Robots obey commands sent by a central server through an IR beam, while their actual position and orientation is fedback to the server by a camera located on the top. Meanwhile a number of visual features are projected onto the field by using a flat display. This system merges characteristics and concepts from two of the most mature RoboCup leagues, Simulation and Small-Size [5], and adds a new key-feature: augmented reality.

All the robots are centrally controlled from one CPU but their decision making algorithms run on networked clients, making the robots behave autonomously virtually isolated from each other just like in simulation league. Position feedback is based on colored markers placed on top of the robots which are detected through a vision system in the same way used in small-size league. Robot control is based on strings of commands sent by modulated infrared signals (in this sense resembling U-league to some extent [1]).

One characterizing feature of the system is the unmodelled embodiment dependent representation of the robots. Contrary to the misleading impression the term "physical visualization" might imply, the robots *are not* mere physical visualizations of some sort of internally simulated mechanism of any kind. On the contrary, the system blindly sends client commands to the robots which may (or may not) respond by performing arbitrary movements. In other words, changes on the physical body of the robot would not require changes on the server internal representation of the robot's mechanisms for there is no such a thing.

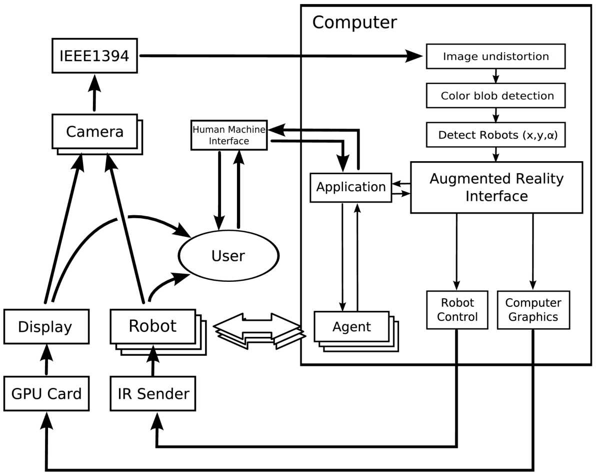


Figure 1: Block diagram illustrating the main system components

2.1 The position feedback

The position of the robots (and eventually other objects, such as ball) is detected from the processing of high-resolution camera images. The computer vision system currently implemented can be divided into three main subsystems: (a) undistortion, (b) blob detection, and (c) identification & orientation. Each one is described in the following paragraphs.

Undistortion: The vast majority of consumer cameras are known to have no significant lens distortion, therefore it is common practice to assume a linear pinhole model. Despite the fact of the PV robots being real three-dimensional objects occupying volume in space, the domain of possible locations for their bodies over the plane of the flat screen is known to be confined into a two-dimensional space. Because of that the calibration problem can be reduced, without loss of generality, to a plane-to-plane linear transformation from the plane of the captured image to the plane of field itself. This transformation is a single linear 3×3 matrix operator which defines a homography in the two-dimensional projective space (see figure 2). In the presence of significant lens distortion the simple addition of a prior step for radial lens undistortion, such as in Tsai's method [14], is likely to be sufficient. Refer [3] for a more extensive review on the projective geometry approach to computer vision.

Blob detection: After undistorted, the image is segmented into blobs of certain colors of interest. These colors are defined by a mask in the three-dimensional $Y \times U \times V$ space. Adjacent pixels, in a 8-neighborhood, belonging to the same color mask configure a single blob. The area (total amount of pixels) and center of mass (average (x, y) coordinates) of the blobs are extracted. Blobs whose mass values are not within a tolerance range from the expected are discarded. This procedure is used

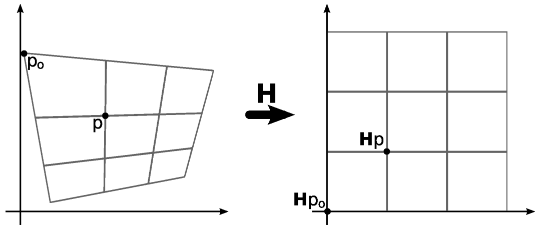


Figure 2: Plane-to-plane projective undistortion based on homography transformation, where **H** is a 3×3 matrix operator and p and p_o are 3-dimensional vectors representing points in the two-dimensional projective space

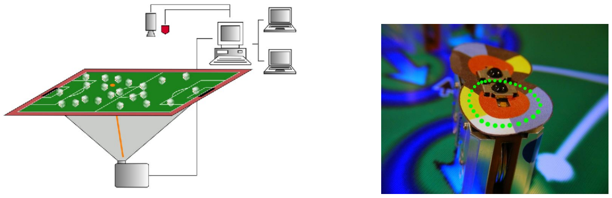
for finding the center of the colored marking patterns on the top of each robot – the red shape seen on figure 3-b.

Identification and orientation: The process here described is inspired on [9]. Once a potential blob is found, a radial pattern of colors is sampled within a pre-defined radius of its center. In figure 3-b these sampling locations are artificially illustrated by a closed path of little green dots. This pattern is cross correlated with a database of stored patterns, each of which uniquely defining a robot's identity. Let's denote $x(i)$ to be the color in the pattern x at the angle i. The cross-correlation r_{xy} is calculated accordingly to the equation 1 for each pattern y the database, and for each $\Delta\alpha$ in the interval $[0^o, 360^o)$. If, for a pattern x, the minimum value of $r_{xy}(\Delta\alpha)$, for any y and $\Delta\alpha \in [0^o, 360^o)$, exceeds a minimum threshold, then the corresponding y gives the identity of a robot, and $\Delta\alpha$ gives its orientation.

$$r_{xy}(\Delta\alpha) = \frac{\sum_{i=0^o}^{360^o} \left[(x(i) - \bar{x}) \cdot (y(i - \Delta\alpha) - \bar{y})\right]}{\sqrt{\sum_{i=0^o}^{360^o} (x(i) - \bar{x})^2} \cdot \sqrt{\sum_{i=0^o}^{360^o} (y(i - \Delta\alpha) - \bar{y})^2}} \tag{1}$$

2.2 Augmented reality

The idea about the augmented reality setup is an extension of a previously published similar concept where robot ants would leave visually coloured trails of "pheromones" by the use of a multimedia projector on the ceiling of a dark room in a swarm intelligence study [13]. Huge improvements in versatility, flexibility, and standardization can be introduced by applying that concept into a more customizable system. The figure 3-a shows an illustrative drawing and figure 3-b shows an actual picture of our system in action. Given the reduced size and weight of the PV sub-league robots the application of a conventional flat display as the field becomes feasible – depending on the application, displays as small as 20-inches are more than enough. This adds much versatility to the system without adding much costs and without complicating the required setup. The mixture of reality and simulation enables projections of environmental features surrounding the real robots. By doing so, not only the environment becomes more visually appealing, but also allows for an enormous variety of new applications which would be otherwise impractical extending the possibilities to the limit of one's imagination.



(a) (b)

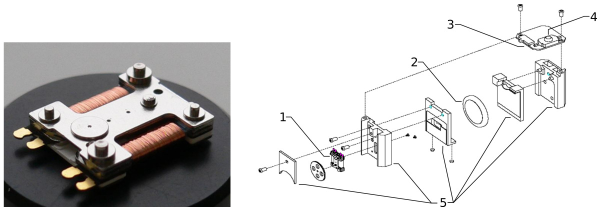
Figure 3: On the left an illustration of the overall system including the feedback control loop (infrared transmitter, camera, server) and the augmented reality screen. On the right an actual close-up picture of two robots playing using such setup. (See last paragraph of subsection 2.1 for explanation about the green dots)

2.3 The miniature robot

Until now, a few developments have been made on very small sized robots, being ALICE one of the most prominent names (see [2] for a survey). In terms of hardware the (current) robot here described is not much different from those many other mini-robots that have been developed so far. We emphasize that it is the unique features brought together by our proposed framework allied to the low cost, robustness and simplicity of the architecture that make this system so attractive.

The first versions of the miniature robot here used were originaly developed by CITIZEN as merchandize devices for demonstrating their new solar powered watch technologies [16]. Since March of 2006 three new prototype versions were already developed for matching the requirements of this project. The most current version of the robot has dimensions of $18 \times 18 \times 22mm$, no sensors, an infrared receiver and is driven by two differential wheels. This first robot was purposely designed to have rather simplistic hardware configuration as a starting point, a seed, to be followed by numerous upgrades in the long term. The main robot components are (numbers in accordance to figure 4-b):

1. Motor – Customized from wristwatch motor unit. See further details in the dedicated sub-section 2.4.
2. Battery – Miniature one-cell rechargeable $3.7V$ lithium ion polymer battery with capacity of $65mAh$.
3. Control board – Currently based on the Microchip $8bit$ PIC18 family of microcontrollers, each robot comes equipped with a PIC18LF1220 which features 4kb of re-programmable flash memory.
4. IR sensor – An IR sensor is used in order to listen for commands from the PC. The sensor operates at the $40kHz$ bandwidth modulation (same of most home-appliance remote controls).
5. Body – The resistant durable body of the robot is micro-machined in aluminum using CITIZEN's high precision CNC machines.



(a) (b)

Figure 4: On the left a close-up picture of the step motor, on the right an exploded view of revealing the robot parts.

Feature	**Value**
Dimensions (mm)	$7.0 \times 8.5 \times 1.9$
Configuration	2 coils × 1 rotor
Gear ratio	1 : 240
Torque ($gf \cdot cm$ at $2.8V$)	between 2.0 and 4.0
Current at $200rps$ (mA)	between 4 and 12
Nominal rotation (rpm)	12.000
Direction	standard and reverse

Table 1: Technical specifications of the step motors used in the miniature robots

2.4 The micro step motor

CITIZEN Co. is renowned in what regards to the manufacture of miniature devices. The motors are one of the main features without which it would be hardly possible to achieve such elevated degree of miniaturization. The miniature robots use two motors to drive its differential wheels. For couping with the torque requirements CITIZEN developed a new special class of step motors combining high-speed rotation and nano-scaled geared reduction. The result is highly reliable motor with very low power consumption fitting into the same thin packaging.

2.5 Robot's firmware and control protocol

The current control protocol was programmed in C and compiled using the proprietary MPLAB C18 compiler – sadly, a code port to the open source SDCC compiler has been delayed due to temporary instabilities in that compiler's support for the PIC18 architecture. Eventually the code might become supported by the GNU C compiler if the robot's microcontroller changes in the next comming years (see sub-section 3.1).

All robots share the same firmware but dynamic IDs are be assigned so that commands to an individual robot can be discriminated. Each of the two wheels can be controlled to run at two different speeds, in both directions or stopped (total of 5 possible values). These two speeds can be fine tunned by infrared commands. An extra set of fast ballistic movements is also provided, with duration customizable, again, by infrared commands. Additional firmware features include low-level battery check and special software reset and sleep commands. Because of the physical nature of the infrared light beam, commands have to be sent by the server to one robot at a time, in an ordered fashion. This implies that bigger number of robots result in longer control lags. Therefore the protocol format was designed so that the command could be sent in a very short time. The current command protocol has a length of $12bits$: ID ($5bits$), left command ($3bits$), right command ($3bits$), and bit parity check ($1bit$). Less frequently used instructions are multiplexed from a sequence of two or more commands.

3 Competitions with cooperation

RoboCup is not only a place for robot tournaments and competitions, but also an international effort towards a bigger common goal [6]. Therefore it is crucial for the survival of the new sub-league that its conceptual bases are not redundant with other leagues and in accordance with the RoboCup long term road map. In the strict technical sense, the autonomous playing agents developed for the PV system, just by themselves add no new challenge if compared to other existing RoboCup competitions and *are not* the central aspect around which teams should concentrate their effords. The original and dominant point of the proposed sub-league is its concept of collaboration towards the development of a central platform for the benefit of all. While in other leagues essential research issues are traditionally faced in the playing agents themselves (AI, biped walking, vision, etc.), in the PV the research issues are in the improvement of the system – in the development of the platform and its robots.

Therefore, the PV sub-league multi-agent interface should be simple to use and adequate for educational purposes. Agent code is expected to be developed by seasonal students while permanent members (professor, staff and graduates) pursue longer term projects that contribute for the sub-league itself and for its platform versatility in a variety of fields.

Keeping these ideas in mind we prepared, three competitions:

- Electronics & Firmware Competition;
- Educational Games Competition;
- Undergrad Team Development Competition.

These three competitions form a kind of self-sustainable "ecological" cycle like shown in the figure 5. Arcs 1 and 2 represent respectively the new needs that inspire hardware development and the new possibilities these developments tend to provide (in future years, if they are incorporated into the the standard system). Arcs 3 and 4 work in a similar way, but within the official hardware of the system to which all teams shall have access. Arc 3 represents a selection of interesting ideas from educational games presented in previous years that are put in practice in a big tournament among undergraduates from various teams, bringing again new ideas for even more interesting games in arc 4. Arc 5 and 6 represent the volunteer contribution of the technical committee for "cluing" all pieces together and making the system work. The idea is to maximize the flow in arcs 1, 2, 3, and 4 while minimizing the contributions through arcs 5 and 6, in the form of a self-sustained evolution cycle for the sub-league.

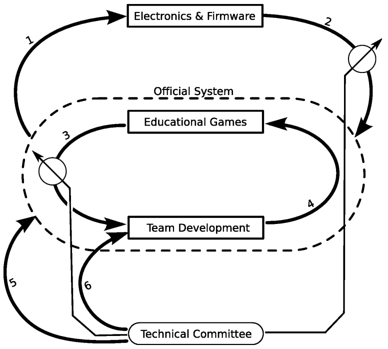


Figure 5: Typical self-sustained development cycle expected

These competitions are indroduced in the next subsections.

3.1 Electronics & Firmware Competition

Goal Allow the evolution of the robot's technology and improve all non-software related aspects of the system.

Summary Teams would have the opportunity to contribute with new ideas for the electronic aspects of the system as well as robot's firmware. Those with background in fields more closely related to the hardware would be able to include in their projects the improvement of certain aspects of the system either for didactic purposes (e.g. class on microcontrollers) or for research. Meanwhile, teams with background in fields more related to computer science would be able to acquire valuable experience by accompanying or even contributing to these projects.

Entries for this competition will consist of documents describing in detail the proposed modification, along with CAD drawings, source code, schematics, etc. Developments could be made on any of the current elements of the system, including the robot, or by introducing a new electronic element to the system. All must be made available to other teams so that they can use and improve at their own. Closed portions will be allowed as part of an entry, but but those parts will not be officially considered during judgement.

Developments on the control circuit of the robots will be required to meet several restrictions regarding the position of mounting holes and pads, the size and shape of the board, the maximum bounding volume, the weight limit of the robot and place of certain components in order to ensure compatibility with current micro-mechatronic architecture. Within those constraints completely new architectures could be proposed. Optionally, entries could even consist of firmware development only.

3.2 Educational AI Games Competition

Goal Create a pool of interesting didactic software applications in the form of games using the system for educational purposes.

Summary Entrants would come up with different game ideas using the system in which they teach concepts related to common subjects ranging from basic computer programming to very specialized topics related to multi-agent systems and artificial intelligence.

The entries would consist of the proposed games along with their source code, supporting tools or API (if any), documentation and accompanying teaching materials. In order to ensure that other teams could easily profit from these contributions the entries would need to be necessarily based on the current official system only. While the eventual introduction of accessories such as balls maze walls or colored objects would, in general, be permitted, no external specialized electronic devices would be allowed. Live demonstrations and poster presentations would be performed during the game event, and together with prior qualified reviewing would rank the entrant.

3.3 Rapid (Soccer) Team Development Competition

Goal Allow undergraduate students to develop complete teams of their own within the typically limited time window of their courses.

Summary The teams would be based on a simplified didactic game framework allowing easy development requiring only a very limited amount of knowledge. All contestants would have an equally limited amount of time for the development of their teams, thus giving similar advantages to teams with limited time to spare. Game rules and supporting software would be officially released just a predefined amount of months before the games.

This comes to fill the gap between RoboCup Junior and the other RoboCup Senior leagues. Typically RoboCup Junior focus mainly on primary and secondary school children, making its challenges less interesting for the more mature undergraduates of courses more related to specialized subjects. The undergraduation curricula are generally composed of a number of classes that last half a year or less and are taken simultaneously over the course of several years. This makes it very difficult for projects based on RoboCup Senior leagues to be included into the main curriculum. Refer to [1, 4] for a previous account to some of above points, where a new league directed exclusively toward undergraduate students was proposed (the U-league).

In the Rapid Soccer Competition alumni woulb be able to experiment their ideas into a RoboCup environment regardless of their time constraints (i.e. having more time to spare would post no advantage). Competitions would take the form of a tournament which would spam over the duration of the RoboCup event.

4 Discussion

This paper introduced the main technical and conceptual characteristics of a new miniature robotic platform. In particular, it was emphasized in the beginning of section 3 the advantages of shifting of focus from the play-

Figure 6: People get very attracted to the small robots

ing agents to the shared system. Furthermore, the three competitions showed in a more clear way how this collaboration shall be fostered toward the constant development of a versatile system for education and research.

In the last trimester of 2006 we already started some undergradutation class experiments where inexperienced second year students of engineering courses could learn to program and develop whole soccer playing teams in only five sessions in three weeks. Figure 6 gives an idea of the kind of entertaining atmosphere the mini robots produce on people, attracting crowds in the public demonstrations of the system. People feel much more attracted to the miniature robots with their limited behavioral skills than for the virtual agents such as those used in RoboCup soccer simulation league. The small robots seem to attract specially the kids. This is likely to help the use of the system in studies that require interaction of small children in the loop (see figure 1), typical in fields related to developmental cognitive studies such as developmental psychology and cognitive neuroscience.

References

[1] John Anderson, Jacky Baltes, David Livingston, Elizabeth Sklar, and Jonah Tower. Toward an undergraduate league for robocup. In *Proceedings of RoboCup-2003: Robot Soccer World Cup VII*, Lecture Notes In Artificial Intelligence. Springer, 2003.

[2] G. Caprari. *Autonomous Micro-Robots: Applications and Limitations*. PhD thesis, Federal Institute of Technology Lausanne, 2003.

[3] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[4] Fredrik Heintz. Robosoc, a system for developing robocup agents for educational use. Master's thesis, Dept. of Computer and Information Science, Linkopings Univ., 2000.

[5] Harukazu Igarashi, Shougo Kosue, Yoshinobu Kurose, and Kazumoto Tanaka. A robot system for robocup small size league:js/s-ii project. In *Proceeding of RoboCup Workshop (5th Rim International Conference on Artificial Intelligence)*, pages 29–38, 1998.

[6] Hiroaki Kitano and Minoru Asada. The robocup humanoid challenge as the millennium challenge for advanced robotics. *Advanced Robotics*, 13(8):723–736, 2000.

[7] Jelle R. Kok and Nikos A. Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In *RoboCup 2005: Robot Soccer World Cup iX*, Lecture Notes in Artificial Intelligence, pages 1–12. Springer, 2006.

[8] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[9] Shoichi Shimizu, Tomoyuki Nagahashi, and Hironobu Fujiyoshi. Robust and accurate detection of object orientation and id without color segmentation. In *Proceedings of RoboCup-2005: Robot Soccer World Cup IX*, Lecture Notes In Artificial Intelligence, pages 408–419. Springer, 2005.

[10] Richard Stallman and Roland McGrath. *GNU make : a program for directing recompilation*. Free Software Foundation, 1996.

[11] Frieder Stolzenburg, Oliver Obst, and Jan Murray. Qualitative velocity and ball interception. In *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, Lecture Notes In Computer Science, pages 283 – 298, 2002.

[12] Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press, 2000.

[13] Ken Sugawara Toshiya Kazama and Toshinori Watanabe. Traffic-like movement on a trail of interacting robots with virtual pheromone. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, pages 383–388, 2005.

[14] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Lecture Notes In Artificial Intelligence, pages 364–374, 1986.

[15] Manuela Veloso, William Uther, Masahiro Fujita, Minoru Asada, and Hiroaki Kitano. Playing soccer with legged robots. In *Proceedings of IROS-98, Intelligent Robots and Systems Conference*, Victoria, Canada, October 1998.

[16] Kazuhiko Yoshikawa. Eco-be!: A robot which materializes a watch's life. *Nature Interface*, (8):56–57, 2002.