

Vision Based Referee Sign Language Recognition System for the RoboCup MSL League

Paulo Trigueiros¹, Fernando Ribeiro² and Luís Paulo Reis³

^{1,2} Departamento de Electrónica Industrial da Universidade do Minho, Campus de Azurém 4800-058, Guimarães, Portugal

³EEUM – Escola de Engenharia da Universidade do Minho – DSI, Campus de Azurém 4800-058, Guimarães, Portugal

`ptrigueiros@gmail.com, fernando@dei.uminho.pt, lpreis@dsi.uminho.pt`

Abstract. In RoboCup Middle Size league (MSL) the main referee uses assisting technology, controlled by a second referee, to support him, in particular for conveying referee decisions for robot players with the help of a wireless communication system. In this paper a vision-based system is introduced, able to interpret dynamic and static gestures of the referee, thus eliminating the need for a second one. The referee's gestures are interpreted by the system and sent directly to the Referee Box, which sends the proper commands to the robots. The system is divided into four modules: a real time hand tracking and feature extraction, a SVM (Support Vector Machine) for static hand posture identification, an HMM (Hidden Markov Model) for dynamic unistroke hand gesture recognition, and a FSM (Finite State Machine) to control the various system states transitions. The experimental results showed that the system works very reliably, being able to recognize the combination of gestures and hand postures in real-time. For the hand posture recognition, with the SVM model trained with the selected features, an accuracy of 98,2% was achieved. Also, the system has many advantages over the current implemented one, like avoiding the necessity of a second referee, working on noisy environments, working on wireless jammed situations. This system is easy to implement and train and may be an inexpensive solution.

Keywords: Hand gesture recognition, Support Vector Machine (SVM), Hidden Markov Model (HMM), Finite State Machine (FSM), RoboCup MSL

1 Introduction

RoboCup is an international joint project to promote Artificial Intelligence (AI), robotics, and related fields since 1996 [1]. RoboCup project has been held every year, and the progress that has been seen in applied technology is amazing, especially in the Soccer Middle Size League (MSL) [2]. RoboCup MSL is one of the most challenging, using real non humanoid robot teams to play with an ordinary soccer ball in an autonomously way. Since the main goal consists of a soccer game in 2050 between the humans world champions team against the robots world champions team, this paper presents a new system, able to interpret dynamic and static gestures of the referee, thus eliminating the need for a second person. In RoboCup MSL the main referee uses assisting technology, controlled by a second referee, to support him, in particular for conveying referee decisions for robot players with the help of a wireless communication system. On this system, the referee's gestures are interpreted by a vision-based system and sent to the Referee Box, which then sends the commands to the robots. The system uses a depth image acquired with a Kinect camera for hand feature extraction (centroid distance signature [3]) and an SVM model to classify hand postures (**Fig. 1**). For dynamic gestures, each gesture to be recognized is scored

against 11 different HMMs that consists our defined language set (**Fig. 2**). The model with the highest score is selected as the recognized gesture. A finite state machine controls the transition between each system state namely: DYNAMIC, STATIC and PAUSE, to build the final referee command (**Table 2**).

The main advantages of using such a system identified to date are:

- The RoboCup gaming environment is quite noisy, so a vision-based system would improve in the identification of commands;
- The communication between referees is sometimes difficult due to ambient noise, and lack of visibility caused by enthusiastic public;
- The response time can be improved, since the command identified by the vision system is automatically transmitted to the robot;
- Reducing the number of referees during a game;
- Reducing the number of indecisions that happen many times during a game;

The system also has disadvantages and one major limitation is currently the restriction on the referee movements, which can reduce the visibility of the game, limitation that can be improved if the referee is placed at a higher level like in tennis games.



Fig. 1. The defined hand postures for the referee static commands.

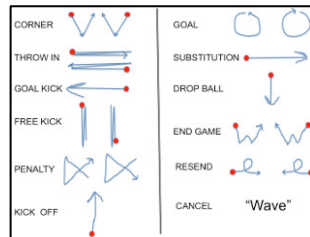


Fig. 2. The defined dynamic gestures used in the Referee CommLang

The rest of the paper is as follows. Firstly, the related work is review in section 2. Section 3 introduces the actual data pre-processing stage, feature extraction and models creation. The *Referee CommLang* is introduced in section 4. Experimental methodology and results are explained in section 5. Conclusions and future work are drawn in section 6.

2 Related Work

Hand gestures, either static or dynamic, for human computer interaction in real time systems is an area of active research and with many possible applications. However, vision-based hand gesture interfaces for real-time applications require fast and extremely robust hand detection, and gesture recognition. Support Vector Machines (SVM's) is a technique based on statistical learning theory, which works very well with high-dimensional data. The objective of this algorithm is to find the optimal separating hyper plane between two classes by maximizing the margin between them [4]. Faria et al. [5, 6] used it to classify robotic soccer formations and the classification of facial expressions, Ke et al. [7] used it in the

implementation of a real-time hand gesture recognition system for human robot interaction, Maldonado-Báscon [8] used it for the recognition of road-signs and Masaki et al. [9] used it in conjunction with SOM (Self-Organizing Map) for the automatic learning of a gesture recognition mode. He first applies the SOM to divide the sample into phases and construct a state machine, and then he applies the SVM to learn the transition conditions between nodes. Almeida et al. [10] proposed a classification approach to identify the team's formation in the robotic soccer domain for the two dimensional (2D) simulation league employing Data Mining classification techniques. Trigueiros et al. [11] have made a comparative study of four machine learning algorithms applied to two hand features datasets. In their study the datasets had a mixture of hand features. Hidden Markov Models (HMMs) have been widely used in a successfully way in speech recognition and hand writing recognition [12], in various fields of engineering and also applied quite successfully to gesture recognition. Oka et al. [13] developed a gesture recognition system based on measured finger trajectories for an augmented desk interface system. They have used a Kalman filter for the prediction of multiple finger locations and used an HMM for gesture recognition. Perrin et al. [14] described a finger tracking gesture recognition system based on a laser tracking mechanism which can be used in hand-held devices. They have used HMM for their gesture recognition system with an accuracy of 95% for a set of 5 gestures. Nguyen et al. [15] described a hand gesture recognition system using a real-time tracking method with pseudo two-dimensional Hidden Markov Models. Chen et al. [16] used it in combination with Fourier descriptors for hand gesture recognition using a real-time tracking method. Kelly et al. [17] implemented an extension to the standard HMM model to develop a gesture threshold HMM (GT-HMM) framework which is specifically designed to identify inter gesture transition. Zafrulla et al. [18] have investigated the potential of the Kinect depth-mapping camera for sign language recognition and verification for educational games for deaf children. They used 4-state HMMs to train each of the 19 signs defined in their study. Cooper et al. [19] implemented an isolated sign recognition system using a 1st order Markov chain. In their model, signs are broken down in visemes (equivalent to phonemes in speech) and a bank of Markov chains are used to recognize the visemes as they are produced. In the field of feature selection and statistical techniques, Ribeiro et al. [20] proposed a method that consisted in using a histogram of white-green transitions for robot orientation. This technique does not take much computational time and it proved to be very reliable.

3 Pre-Processing and Feature Extraction

3.1 Static hand gesture classification

For static gesture classification, hand segmentation and feature extraction is a crucial step in computer vision applications for hand gesture recognition. The pre-processing stage prepares the input image and extracts features used later with classification algorithms [3]. In our system we used feature vectors composed of centroid distance values to train a SVM (Support Vector Machine) for hand posture classification.

3.1.1 Centroid distance

The centroid distance signature is a type of shape signature [3]. The centroid distance function is expressed by the distance of the hand contour boundary points, from the centroid (x_c, y_c) of the shape calculated in the following manner:

$$d(i) = \sqrt{[x_i - x_c]^2 + [y_i - y_c]^2}, \quad i = 0, \dots, N - 1 \quad (1)$$

where $d(i)$, is the calculated distance, x_i and y_i are the coordinates of contour points and N is the number of contour points used in the calculation. This way, we obtain a one-dimensional function that represents the hand shape signature as shown in **Fig. 3**. In our system we have used 32 equally spaced points on the contour to build the hand signature.

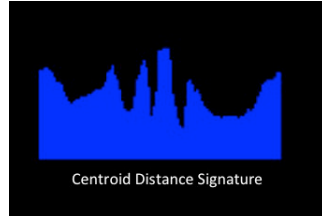


Fig. 3. Hand Centroid Distance Signature

Due to the subtraction of centroid, which represents the hand position, from boundary coordinates, the centroid distance representation is invariant to translation as Rayi Yanu Tara et al. [21] have demonstrated in their paper.

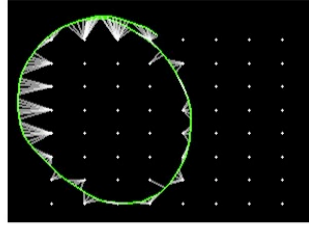
3.1.2 SVM

A SVM is used to learn the pre-set hand postures (**Fig. 1**). The SVM is a pattern recognition technique in the area of supervised machine learning, which works very well with high-dimensional data. When more than two classes are present, there are several approaches that evolve around the 2-class case [22]. The one used in our system is the one-against-all, where c classifiers have to be designed. Each one of them is designed to separate one class from the rest. One drawback with this kind of approach is that after the training phase there are regions in the space, where no training data lie, for which more than one hyperplane gives a positive value or all of them result in negative values [23].

3.2 HMM dynamic gesture classification.

Dynamic hand gestures are time-varying processes, which show statistical variations, making HMMs a plausible choice for modeling the processes [24]. So, for the recognition of dynamic gestures a HMM (Hidden Markov Model) model was trained for each possible gesture. A Markov Model is a typical model for a stochastic (i.e. random) sequence of a finite number of states [25]. A human gesture can be understood as a Hidden Markov Model where the true states of the model are hidden in the sense that they cannot be directly observed. HMMs have been widely used in a successfully way in speech recognition and hand writing recognition [12]. In our system the 2D motion hand trajectory points are labeled according to the distance to the nearest centroid based on Euclidean distance, resulting in a discrete feature vector like the one shown in **Fig. 4**, and translated to origin. In our case the 2D features are sufficient for hand gesture recognition.

The feature vectors thus obtained are used to train the different HMMs and learn the model parameters: the initial state probability vector (π), the state-transition probability matrix ($A = [a_{ij}]$) and the observable symbol probability matrix ($B = [b_j(m)]$). In the recognition phase an output score for the sample gesture is calculated for each model, given the probability that the corresponding model generated the underlying gesture. The model with the highest output score represents the recognized gesture. In the proposed system, a left-right HMM (**Fig. 5**) was used [26, 27]. This kind of HMM has the states ordered in time so that as time increases, the state index increases or stays de same. This topology has been chosen, since it is perfectly suitable to model the kind of temporal gestures present in the system.



[1 2 2 2 3 3 4 4 4 5 6 13 13 14 14 14 14 14 14 14 13 13
20 19 19 18 18 17 24 23 22 22 22 22 22 29 29 29 29
29 29 29]

Fig. 4. Gesture path with respective feature vector

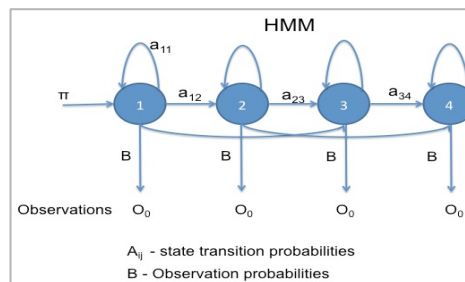


Fig. 5. Example of a left-right HMM

The model with the highest output score represents the recognized gesture. In the proposed system, a left-right HMM (Fig. 5) was used [26, 27]. This kind of HMM has the states ordered in time so that as time increases, the state index increases or stays the same. This topology has been chosen, since it is perfectly suitable to model the kind of temporal gestures present in the system.

4 The Referee CommLang.

The Referee CommLang is a new and formal definition of all the commands that the system accepts. As in [28], the language must represent all the possible gesture combinations (static and dynamic) and at the same time be simple in its syntax. The language was defined with BNF – BakusNaur form [29], and has three types of commands: **Team commands**, **Player commands** and **Game commands**.

This way, a language is defined to be a set of commands that can be a TEAM_COMMAND, a GAME_COMMAND or a PLAYER_COMMAND. The TEAM_COMMAND is composed of the following ones: KICK_OFF, CORNER, THROW_IN, GOAL_KICK, FREE_KICK, PENALTY, GOAL or DROP_BALL. A GAME_COMMAND can be the START or STOP of the game, a command to end the game (END_GAME), cancel the just defined command (CANCEL) or resend the last command (RESEND). For the END_GAME command, we have to define the PART_ID (identification of the part of the game ending) with one of four commands – 1ST, 2ND, EXTRA or PEN (penalties).

```
<LANGUAGE> ::= {<COMMAND>}
<COMMAND> ::= <TEAM_COMMAND> | <GAME_COMMAND> | <PLAYER_COMMAND>
<TEAM_COMMAND> ::= <KICK_OFF> | <CORNER> | <THROW_IN> | <GOAL_KICK> |
<FREE_KICK> | <PENALTY> | <GOAL> | <DROP_BALL>
```

```

<GAME_COMMAND> ::= <START> | <STOP> | <END_GAME> | <CANCEL> | <RESEND>
<PLAYER_COMMAND> ::= <SUBSTITUTION> | <PLAYER_IN> | <PLAYER_OUT> |
<YELLOW_CARD> | <RED_CARD>

```

For the TEAM_COMMANDS we have several options: KICK_OFF, CORNER, THROW_IN, GOAL_KICK, FREE_KICK, PENALTY and GOAL that need a TEAM_ID (team identification) command, that can be one of two values – CYAN or MAGENTA, and finally the DROP_BALL command.

```

<KICK_OFF> ::= KICK_OFF <TEAM_ID>
<CORNER> ::= CORNER <TEAM_ID>
<THROW_IN> ::= THROW_IN <TEAM_ID>
<GOAL_KICK> ::= GOAL_KICK <TEAM_ID>
<FREE_KICK> ::= FREE_KICK <TEAM_ID>
<PENALTY> ::= PENALTY <TEAM_ID>
<GOAL> ::= GOAL <TEAM_ID>
<DROP_BALL> ::= DROP_BALL

```

For the PLAYER_COMMAND, first we can have a SUBSTITUTION command with the identification of the player that leaves (PLAYER_OUT) and the one that enters (PLAYER_IN) the game with the PLAYER_ID command. The PLAYER_ID can take one of seven values (PL1, PL2, PL3, PL4, PL5, PL6, PL7). For the remaining commands, PLAYER_IN, PLAYER_OUT, YELLOW_CARD or RED_CARD there exists the need to define the TEAM_ID as explained above, and the PLAYER_ID.

```

<SUBSTITUTION> ::= SUBSTITUTION <PLAYER_IN> <PLAYER_OUT>
<PLAYER_IN> ::= PLAYER_IN <TEAM_ID> <PLAYER_ID>
<PLAYER_OUT> ::= PLAYER_OUT <TEAM_ID> <PLAYER_ID>
<YELLOW_CARD> ::= YELLOW_CARD <TEAM_ID> <PLAYER_ID>
<RED_CARD> ::= RED_CARD <TEAM_ID> <PLAYER_ID>
<START> ::= START
<STOP> ::= STOP
<END_GAME> ::= END_GAME <PART_ID>
<CANCEL> ::= CANCEL
<RESEND> ::= RESEND
<TEAM_ID> ::= CYAN | MAGENTA
<PLAYER_ID> ::= PL1 | PL2 | PL3 | PL4 | PL5 | PL6 | PL7
<PART_ID> ::= 1ST | 2ND | EXTRA | PEN

```

5 Experimental Methodology and Results

The experimental methodology was divided into three parts, each one with its own application built and tested in an Intel Core i7 (2,8 GHz) Mac OSX computer with 4GB DDR3. First, a feature extraction phase and SVM model training of the defined static hand postures. Second, acquisition of hand motion sequences for each of the defined gestures, and HMM model training. The final one, the Referee CommLang implementation, is the system with posture identification and gesture recognition, controlled by a event driven Finite State Machine that controls the transition between the three possible states: DYNAMIC, STATIC and PAUSE, and builds the final command that is transmitted to the robots. All the datasets were built with data collected from four people making the defined postures and gestures in front of the Kinect. For feature extraction and SVM training, the application built in C++ (**Fig. 7**) uses openFrameworks (<http://www.openframeworks.cc/>), OpenCV [30] (computer vision library), OpenNI [31] (Open Natural Interface) that enables real-time hand tracking and skeleton tracking, and the machine-learning library dlib (<http://dlib.net/>). The dataset was built from features extracted from four users performing the set of commands in front of the camera, and was analyzed with the help of RapidMiner [32], in order to find the best parameters for the learner. The experiments were performed with a *10-fold cross validation*, and we were able to achieve an accuracy of 98,2% for the given features,

with an RBF (radial basis function) kernel with a C value equal to 6. The obtained confusion matrix can be seen in **Table 1** where can verify the existence of some high values between command number four and three, between number five and four and between number six and seven, and that contributed to the 1,8% of false positives. For dynamic gesture learning and training, we built another application (**Fig. 8**) using openFrameworks with OpenNI [31] and a personal HMM class library for gesture learning and classification. In the case of the HMM the number of centroids used to label the hand path points, is a predefined value, equally spaced in a 2D grid as was shown in **Fig. 4**. For each gesture, a HMM is trained and the model parameters (the initial state probability vector, the state-transition probability matrix and the observable symbol probability matrix) are learned. The number of observation symbols (alphabet) was defined to be 64 with 4 hidden states. Several values for the number of observations and hidden states were tried out, without significant improvements for values greater than the defined ones.

Table 1. Centroid distance confusion matrix

		Actual class						
		1	2	3	4	5	6	7
Predicted class	1	588	0	0	0	0	2	0
	2	2	706	0	0	1	0	1
	3	0	1	578	1	0	0	0
	4	0	0	12	715	3	0	0
	5	0	1	1	13	536	1	3
	6	1	8	0	1	5	693	12
	7	2	0	0	2	6	9	751

The Referee CommLang prototype (**Fig. 9**), uses the SVM model and the HMM models controlled by a FSM (Finite State Machine) to recognize the defined command language.

5.1 Finite State Machine

Since the Referee CommLang uses a combination of dynamic and static gestures, we needed to model the command semantics. A Finite State Machine is a usually employed technique to handle this situation [33, 34]. Our system uses a FSM, as shown in **Error! Reference source not found.**, to control the transition between the three possible states: DYNAMIC, STATIC and PAUSE. By using a rest or pause state it is possible to identify changes in gestures and somehow eliminate all unintentional actions between dynamic-static or static-static gestures. The combination of a dynamic gesture and a static or set of static gestures gives us a command as defined in the Referee CommLang and listed in **Table 2**.

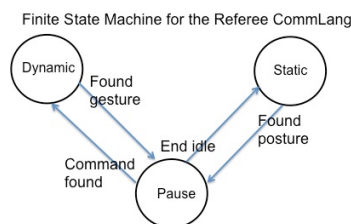


Fig. 6. The Referee CommLang Finite State Machine

6 Conclusions and Future Work

This paper presents a system able to interpret dynamic and static gestures of the referee, thus eliminating the need for a second person. The referee's gestures are interpreted by the vision-based system and sent to the Referee Box, which sends then the proper commands to the robots. The set of dynamic gestures defined is a first approach to the problem, and future adjustments or definition of new ones is always possible. For static posture recognition an SVM model was learned from centroid distance features and a good recognition rate was achieved. For temporal gesture recognition one HMM was trained for each gesture and it was also possible to achieve a good recognition rate, proving that this kind of models, as was already seen in other references, works very well for this type of problem. A vision-based sign language was defined – *the Referee CommLang* – that is able to represent all the possible gesture combinations (static and dynamic) and at the same time be simple in its syntax. The experimental results showed that the system works very reliably, being able to recognize the combination of gestures and hand postures in real-time, although with some limitations due to the nature of the camera used. Despite these limitations, for the hand posture recognition, with the SVM model trained with the selected features, an accuracy of 98,2% was achieved. The system has many advantages over the current implemented one, where there exists the need for two referees, and is an easy and inexpensive solution to implement and train. As further work it is intended to keep improving this system in order to eliminate some of the limitations, like the referee movements restrictions. The current system only allows 2D gesture paths, although as further work it is thought to test and include not only the possibility of 3D gestures but also to work with several cameras to thereby obtain a full 3D environment and achieve view-independent recognition. Also, the existence of different new cameras, with improved depth resolution and with a higher frame frequency, now available on the market will be tested, to improve some of the limitations of the Kinect, thus leading to better static gesture recognition rates.

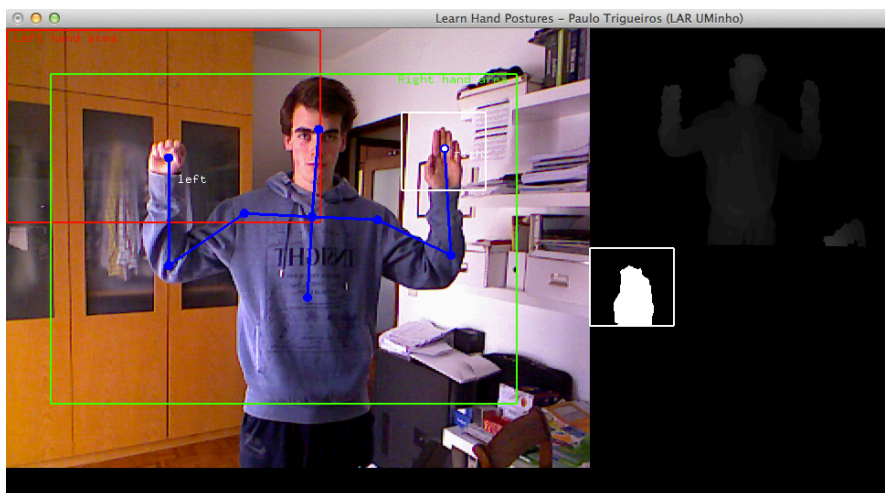


Fig. 7. Application interface for the posture learning and training system

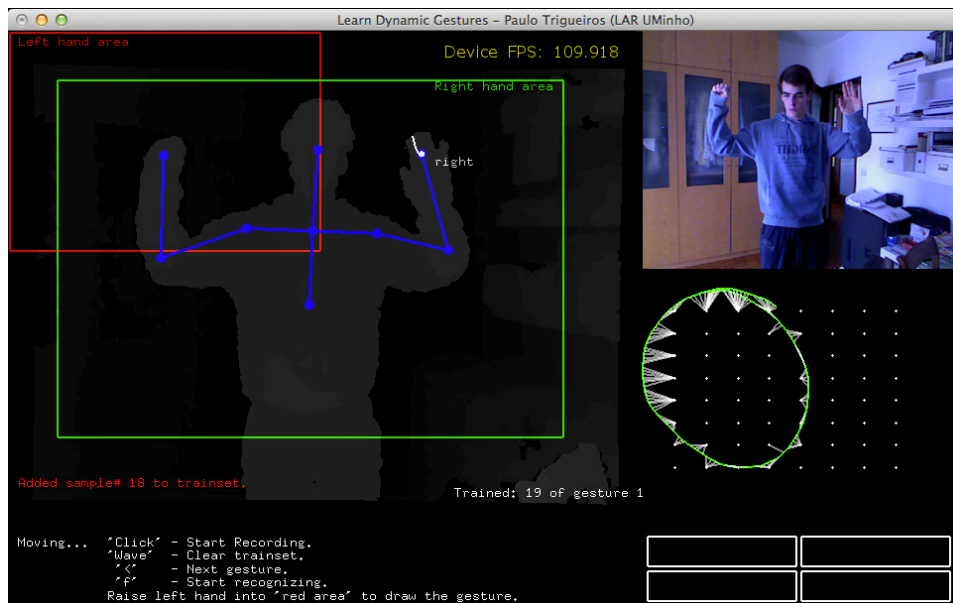


Fig. 8. The application interface for the gesture learning and training system

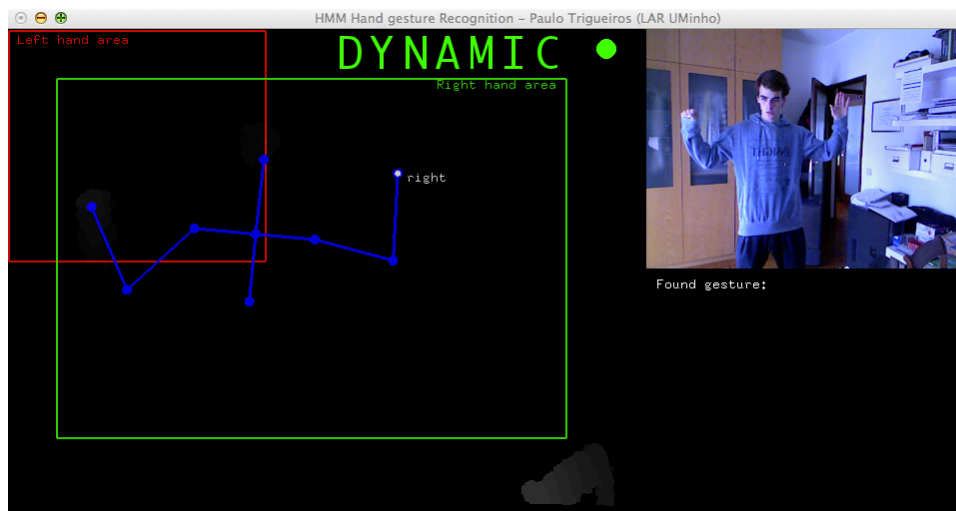



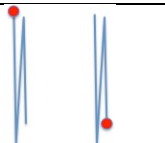
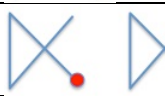
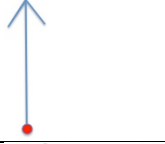


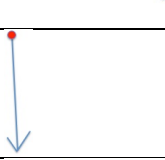




Fig. 9. The Referee CommLang interface

Table 2. Command set definition with associated codes and text description

Nr.	Command	1 st Gesture	2 nd Gesture	3 rd Gesture	4 th Gesture	Code (TEXT)
1	CORNER		TEAM			11 - CORNER, TEAM 1 12 - CORNER, TEAM 2
2	THROW_IN		TEAM			21 - THROW_IN, TEAM 1 22 - THROW_IN, TEAM 2
3	GOAL_KICK		TEAM			31 - GOAL_KICK, TEAM 1 32 - GOAL_KICK, TEAM 2
4	FREE_KICK		TEAM			41 - FREE_KICK, TEAM 1 42 - FREE_KICK, TEAM 2
5	PENALTY		TEAM			51 - PENALTY, TEAM 1 52 - PENALTY, TEAM 2
6	KICK_OFF		TEAM			61 - KICK_OFF, TEAM 1 62 - KICK_OFF, TEAM 2
7	GOAL		TEAM	PLAYER		71(1-7) - GOAL, TEAM1, PLAYER(1-7) 72(1-7) - GOAL, TEAM2, PLAYER(1-7)
8	SUBSTITUTION		TEAM	PLAYER_IN	PLAYER_OUT	81(1-7)(1-7) - SUBSTITUTION, TEAM 1, PLAYER_IN(1-7), PLAYER_OUT(1-7) 82(1-7)(1-7) - SUBSTITUTION, TEAM 2, PLAYER_IN(1-7), PLAYER_OUT(1-7)
9	DROP_BALL					9 - DROP_BALL
10	END_GAME					101 - END_GAME, PART 1 102 - END_GAME, PART 2
11	RESEND					11 - CANCEL
12	RESEND	"Wave"				12 - RESEND

Acknowledgment.

The authors wish to thank all members of the Laboratório de Automação e Robótica (LAR), at University of Minho, Guimarães. We would like to thank IPP (Instituto Politécnico do Porto) for the contributions that made all this work possible. Also a special thanks to all others who contributed to making this work possible.

References

- [1] Y. Kitazumi and K. Ishii, "Survey of cooperative algorithm in robocup middle size league," in *World Automation Congress (WAC), 2010*, 2010, pp. 1-6.
- [2] M. T. Committe. (2013, April 2013). *Middle Size Robot League Rules and Regulations for 2013*. Available: [http://wiki.robocup.org/wiki/Middle_Size_League - Rules](http://wiki.robocup.org/wiki/Middle_Size_League_-_Rules)
- [3] P. Trigueiros, F. Ribeiro, and L. P. Reis, "A Comparative Study of different image features for hand gesture machine learning," in *5th International Conference on Agents and Artificial Intelligence*, Barcelona, Spain, 2013.
- [4] A. Ben-Hur and J. Weston, "A User's Guide to Support Vector Machines," in *Data Mining Techniques for the Life Sciences*. vol. 609, ed: Humana Press, 2008, pp. 223-239.
- [5] B. M. Faria, N. Lau, and L. P. Reis, "Classification of Facial Expressions Using Data Mining and machine Learning Algorithms," in *4ª Conferência Ibérica de Sistemas e Tecnologias de Informação*, Póvoa de Varim, Portugal, 2009, pp. 197-206.
- [6] B. M. Faria, L. P. Reis, N. Lau, and G. Castillo, "Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations and Opponent Teams," presented at the IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore, 2010.
- [7] W. Ke, W. Li, L. Ruifeng, and Z. Lijun, "Real-Time Hand Gesture Recognition for Service Robot," pp. 976-979, 2010.
- [8] S. Maldonado-Báscon, S. Lafuente-Arroyo, P. Gil-Jiménez, and H. Gómez-Moreno. (2007, June 2007) Road-Sign detection and Recognition Based on Support Vector Machines. *IEEE Transactions on Intelligent Transportation Systems*. 264-278.
- [9] M. Oshita and T. Matsunaga, "Automatic learning of gesture recognition model using SOM and SVM," in *International Conference on Advances in Visual Computing*, Las Vegas, NV, USA, 2010, pp. 751-759.
- [10] R. Almeida, L. P. Reis, and A. M. Jorge, "Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain," presented at the Proceedings of the 14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence, Aveiro, Portugal, 2009.
- [11] P. Trigueiros, F. Ribeiro, and L. P. Reis, "A comparison of machine learning algorithms applied to hand gesture recognition," in *7th Iberian Conference on Information Systems and Technologies*, Madrid, Spain, 2012, pp. 41-46.
- [12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.
- [13] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *Computer Graphics and Applications*, *IEEE*, vol. 22, pp. 64-71, 2002.
- [14] S. Perrin, A. Cassinelli, and M. Ishikawa, "Gesture recognition using laser-based tracking system," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, South Korea, 2004, pp. 541-546.
- [15] N. D. Binh, E. Shuichi, and T. Ejima, "Real-Time Hand Tracking and Gesture Recognition System," in *Proceedings of International Conference on Graphics, Vision and Image Cairo - Egypt*, 2005, pp. 362--368.

- [16] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and Vision Computing*, vol. 21, pp. 745-758, 2003.
- [17] D. Kelly, J. McDonald, and C. Markham, "Recognition of Spatiotemporal Gestures in Sign Language Using Gesture Threshold HMMs," in *Machine Learning for Vision-Based Motion Analysis*, L. Wang, G. Zhao, L. Cheng, and M. Pietikšinen, Eds., ed: Springer London, 2011, pp. 307-348.
- [18] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," presented at the 13th International Conference on Multimodal Interfaces, Alicante, Spain, 2011.
- [19] C. Helen and B. Richard, "Large lexicon detection of sign language," in *11th International Conference on Human-Computer Interaction*, Rio de Janeiro, Brazil, 2007, pp. 88-97.
- [20] F. Ribeiro, G. Lopes, B. Pereira, J. Silva, P. Ribeiro, J. Costa, *et al.*, "Robot Orientation with Histograms on MSL," in *RoboCup 2011: Robot Soccer World Cup XV*. vol. 7416, T. Rafer, N. M. Mayer, J. Savage, and U. Saranle, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 507-514.
- [21] R. Y. Tara, P. I. Santosa, and T. B. Adji, "Sign Language Recognition in Robot Teleoperation using Centroid Distance Fourier Descriptors," *International Journal of Computer Applications*, vol. 48, June 2012 2012.
- [22] S. Theodoridis and K. Koutroumbas, *An Introduction to Pattern Recognition: A Matlab Approach*: Academic Press, 2010.
- [23] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 4th Edition*: Elsevier, 2009.
- [24] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," presented at the Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, 1999.
- [25] G. A. Fink, *Markov Models for Pattern recognition - From Theory to Applications*: Springer, 2008.
- [26] F. Camastra and A. Vinciarelli, *Machine Learning for Audio, Image and Video Analysis*: Springer, 2008.
- [27] E. Alpaydin, *Introduction to Machine Learning*: MIT Press, 2004.
- [28] L. P. Reis and N. Lau, "COACH UNILANG - A Standard Language for Coaching a (Robo) Soccer Team," in *RoboCup 2001: Robot Soccer World Cup V*. vol. 2377, A. Birk, S. Coradeschi, and S. Tadokoro, Eds., ed: Springer Berlin Heidelberg, 2002, pp. 183-192.
- [29] J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, *et al.* (1960, January 1963) Revised Report on the Algorithmic Language ALGOL 60. *Communications of the ACM*. 1-17. Available: <http://doi.acm.org/10.1145/366193.366201>
- [30] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed.: O'Reilly Media, 2008.
- [31] OpenNI. (2013). *The standard framework for 3D sensing*. Available: <http://www.openni.org/>
- [32] R. Miner. (December 2011). *RapidMiner : Report the Future*. Available: <http://rapid-i.com/content/view/181/196/>
- [33] M. Buckland, *Programming Game AI by Example*: Wordware Publishing, Inc., 2005.
- [34] I. Millington and J. Funge, *Artificial Intelligence for Games*, second edition ed.: Elsevier, 2009.