Retrieving and reusing qualitative cases: an application in the humanoid-robot soccer

Thiago P. D. Homem [a,b,*], Danilo H. Perico [b] Paulo E. Santos [b] Reinaldo A. C. Bianchi [b] and Ramon L. de Mantaras [c]

[a] *Instituto Federal de São Paulo, São Paulo, Brazil*

[b] *Centro Universitário FEI, São Paulo, Brazil*

[c] *Artificial Intelligence Research Institute, CSIC, Barcelona, Spain*

Abstract. This paper proposes a new Case-Based Reasoning (CBR) approach, named Q-CBR, that uses a Qualitative Spatial Reasoning theory to model, retrieve and reuse cases by means of spatial relations. Qualitative relations between objects, represented in terms of the $\mathcal{EOPRA}$ formalism, are stored as qualitative cases that are applied in the definition of new retrieval and reuse algorithms. The retrieval algorithm uses a Conceptual Neighborhood Diagram to compute the similarity between a new problem and the cases in the case base, and to select the most similar case. The reuse algorithm uses a composition algorithm to calculate the adapted position of the agents based on their frame of reference. The proposed approach was evaluated on simulation and on real humanoid robots. Results suggest that this proposal is faster than using a quantitative model with numerical similarity measurement such as the Euclidean distance. As a result of running Q-CBR, the robots obtained a higher average number of goals than those obtained when running a metric CBR approach.

Keywords: Case-Based Reasoning, Qualitative Spatial Reasoning, Humanoid Robots

Introduction

Case-Based Reasoning (CBR) is a paradigm of Artificial Intelligence (AI) that uses the knowledge obtained in past situations, defined in terms of *cases*, to solve new problems. It combines processes for solving a problem and learning from this experience in a process cycle known as the four REs: *retrieve, reuse, revise* and *retain* [1]. The retrieval step searches the case base for the most similar cases to a given problem, retrieving the candidate cases; the reuse step selects the most similar case to be used as a solution to the problem; the revise step analyses the proposed solution and the retain step decides if the reused case is useful to solve the new problem.

In general, when CBR is applied in problems where the objects' positions in space is relevant, a metric coordinate system is used to represent case similarity. Consequently, there is a large number of distinct similarity measurement strategies based on numerical distances or other metric information [6].

*Corresponding Author: Instituto Federal de São Paulo - IFSP, Av. Mutinga, 951, Jardim Santo Elias, CEP 05110-000, São Paulo, SP, Brazil; E-mail: thiagohomem@ifsp.edu.br.

In some domains, however, a metric representation is not the most effective. For instance, in a humanoid-robot domain, where a video camera is the main source of information, the use of a metric coordinate system to represent object's position generates a high error rate. In this context, qualitative relations between spatial entities can provide a more appropriate representation of the robot's environment. From the distance and direction information obtained by the robot's sensors, qualitative spatial regions can be defined, allowing for reasoning about, and comparison of, relations between domain objects, the regions in which the objects are located and their occupancy regions.

This paper proposes a novel CBR approach using Qualitative Spatial Reasoning (QSR) to model cases and to serve as the basis for retrieval and reuse algorithms. The idea is to use $\mathcal{EOPRA}$ [24] for domain modeling, whereby instead of representing cases using the Cartesian coordinate system, we represent them as qualitative orientation and distance relations. The proposed algorithms use Conceptual Neighborhood Diagrams (CND) [7, 14] and a cost function to compute the similarity between a new problem and the cases in the case base, to retrieve the most similar case and to reuse its solution to solve the new problem. This work was evaluated in the robot-soccer domain, as defined by the RoboCup Federation Humanoid League [34]. In this domain, a team of humanoid robots plays a soccer game against an opponent team on an artificial grass-soccer field. Three categories separate the teams according to the robots' heights. The robots must have a human-like body, with two legs, two arms and one head attached to a trunk. Two types of experiments were performed: the first was conducted in a simulation software, in which the proposed approach was compared to the metric-based method presented in Ros et al. [35] and to a reactive approach; the second experiment was executed with real robots, where the present proposal was compared with a reactive approach. In both experiments, the number of goals scored and the retrieval time were analyzed.

Ros et al. [35] applies CBR for the coordinated action selection in the robot-soccer domain, using the Cartesian coordinate system to represent the position of objects in the field. The present work differs from Ros et al. since it discretizes the world following a qualitative spatial reasoning formalism and proposes a faster retrieval algorithm that can be used in robots with limited processing power. Finally, by running the algorithms proposed in this paper, the robots performed a higher average number of goals than running a metrical-based CBR.

In the remainder of this work we present the CBR and QSR, which are the foundations of this work, (Section 1), the proposed Qualitative Case-Based Reasoning method (Section 2), the results obtained during the retrieval and reuse steps (Section 3) and the related work (Section 4).

1. Research Background

This section presents the two methodologies that are used in this work, Case-Based Reasoning (CBR) and Qualitative Spatial Reasoning (QSR).

1.1. Case-Based Reasoning

Case-Based Reasoning (CBR) [1] can be summarized by means of two principles: the existence of real-world regularities (i.e., similar problems have similar solutions) and the tendency to encounter similar problems [17].

Given a new problem, CBR uses the knowledge of previous situations (cases) to find a similar case that can be reused to solve the new problem.

A case in the robot-soccer domain can be defined as the following triple [35]:

$$case = (P, A, K), \tag{1}$$

where P is the problem description, A is the solution description and K represents the case scope.

According to [35], the problem description (P) corresponds to the situation in which the case can be used, representing the global coordinates of the objects in the case. For instance, in the robot-soccer domain, the problem description of a case can include the position of any object in the soccer field. Considering a case with u objects, where each object is represented with the symbol R_i ($i \in \{1, \dots, u\}$), P can be defined as:

$$P = \{R_1 : (x_1, y_1), \dots, R_u : (x_u, y_u)\}. \tag{2}$$

The solution description (A) is composed of a sequence of actions that each agent (that is part of the solution) must perform to solve the problem. Let v be the number of agents that are part of the solution and p, the number of actions that each agent can perform. A solution description (A) can be defined by means of one set of actions a_{i_j} (for $i \in \{1, \dots, v\}$ and $j \in \{1, \dots, p\}$) assigned to each agent R_i:

$$A = \{R_1 : \{a_{1_1}, \dots, a_{1_p}\}, \dots, R_v : \{a_{v_1}, \dots, a_{v_p}\}\}. \tag{3}$$

The case-scope representation (K) is defined as elliptic regions around the object's positions, where the objects should be positioned in order to retrieve that case. In other words, K defines the applicability boundaries of the cases.

Ros et al. [35] also proposed a retrieval method where cases are evaluated along three important aspects: the similarity between problem and case; the cost of adapting a problem to a case; and, the applicability of a solution to a case. The similarity function was defined measuring the distance between robots and the ball in a problem and in a case:

$$Sim(p, c) = dist(B^c, B^p) + \sum_{i=0}^{u} dist(R_i{}^c, R_i{}^p), \tag{4}$$

where B^c is the ball's position in the case c, B^p is the ball's position in the problem p, $R_i{}^c$ is the i robot's position in the case c, $R_i{}^p$ is the i robot's position in the problem p and $dist(R_i{}^c, R_i{}^p)$ is the Gaussian distance between the robot $R_i{}^c$ to the robot $R_i{}^p$.

The adaptation cost was defined in [35] measuring the distance the robots have to move from their current positions to their adapted positions:

$$Cost(p,c) = \sum_{i=1}^{v} dist(r_i, adaptPos_i), \quad (5)$$

where v is the number of robots that take part in the case solution, $dist$ is the Euclidean distance, r_i is the current position of robot i and $adaptPos_i$ is the adapted position for robot i.

Finally, the applicability measure take into account the adversarial component of the domain, i.e., one solution retrieved in the case depends on the opponents' positions. Ros et al. [35] combined the opponent similarity function, which measures the opponent's threat to accomplishing the task, with a function that computes if the trajectory of the ball indicated in the case is free of opponents.

In addition to the work of Ros et al., [35], other works have used CBR in the robot-soccer domain. Lin, Liu and Chen [19] presented one of the first architectures that includes a deliberative CBR system for soccer-playing agents; Karol et al. [16] proposed high-level planning strategies, which included a CBR system. In Marlin et al. [21], three case-based reasoning prototypes were developed for a team in the RoboCup small-size league, where CBR was used to position the goalie, select team formations and recognize game states for the team.

Floyd, Esfandiari and Lam [12] used CBR in the RoboCup Soccer Simulation League, where the agents perceive the objects in the field, convert this perception into a case structure and retrieve the k-most similar cases, using the k-nearest-neighbor search. This work proposed two similarity functions and allows an agent to imitate the actions of a player.

The work presented in Davoust, Floyd and Esfandiari [5] proposed the use of fuzzy histograms to represent the objects in the field and a similarity metric, based on the Jacard Coefficient, that matches scenes in a given problem to cases in a case base, retrieving the action related to the most similar case. Altaf et al. [2] proposed an architecture to control more complex soccer behaviors such as dribbling and goal scoring applied to humanoid multi-robot scenarios.

The main difference between the work cited in this section and the present proposal is the use of a qualitative formalism to model, retrieve and reuse cases. Also, the work described in this paper was tested in a real-robot domain, considering robot failures and noises, whereas in much previous works, experiments were conducted in simulated environments, under optimal conditions, with a global knowledge of the environment and using numerical values. More specifically, in our proposal the agents have local vision and use qualitative spatial representations to retrieve and reuse cases. Even if the qualitative position of an object is different from the precise object location, the retrieval algorithm proposed in this paper retrieves the case with the lowest adaptation cost.

The next section introduces the field of qualitative spatial reasoning in AI, describes the $\mathcal{EOPRA}_m$ formalism and the idea of Conceptual Neighborhood Diagrams.

1.2. Qualitative Spatial Reasoning

Qualitative Spatial Reasoning (QSR) is a subfield of knowledge representation in AI that formalizes qualitative spatial relations between objects, aiming at modeling the human common sense understanding of the world [39]. QSR has been applied in distinct fields, such as robot navigation and self-localization, geographic information systems and computer vision [3]. Formalisms in QSR verse on various spatial modalities, such as mereotopology [31], qualitative directions [13, 33], occlusion [22, 30, 36] and so forth [3, 18].

Among the several proposed formalisms in the QSR literature, the Oriented Point Relation Algebra ($\mathcal{OPRA}$) [23] has been the major formalism for representing and reasoning about objects with intrinsic fronts, such as cars, boats [11, 25] and robots [23]. $\mathcal{OPRA}_m$ refers to the Oriented Point Algebra with granularity m, used in order to obtain the angular resolution, which is equal to $\frac{2\pi}{2m}$ [25]. The objects are represented as oriented points, that refer to Cartesian coordinates (x and y) and orientation (θ). Each point defines a relative reference frame of granularity m ($m \in \mathbb{N}$).

In $\mathcal{OPRA}_m$ a relation between two oriented points A and B is represented as $A_m\angle_i^j B$, which means: given the granularity m, the relative position of B with respect to A is described by i and the relative position of A with respect to B is j. The $\mathcal{OPRA}$ formalism describes only the orientation between objects, however in several domains the distance is an important spatial information that must be considered.

In order to represent distances, Moratz and Wallgrüm [24] proposed a definition of relative distance based on local references called *elevations*. Elevations are defined by the height of objects, whose projection in the 2D plane defines a circle around the object's locations, that is used as a distance reference. The size of this projection is represented by δ, and all distance ratios are calculated taking into consideration n and δ, where n is the distance granularity [8]. The granularity also applies to elevations in order to provide an appropriate level of abstraction for distance relations. Equation 6 calculates the boundaries of qualitative distances around an elevated point A, where $0 \leqslant ep \leqslant 2n$ and ep must be an even number [24].

$$b_A(ep) = \begin{cases} \infty & \text{if } ep = 2n, \\ \frac{ep\delta_A}{n} & \text{if } ep \leqslant n, \\ \frac{n\delta_A}{2n-ep} & \text{otherwise.} \end{cases} \tag{6}$$

In this context, the distance relations between two points A and B is represented as $A\ {}_{m}\bigcirc_{e}^{f}\ B$, where e represents the relative distance of B with respect to A and f, the relative distance of A with respect to B.

The idea of elevated points can be combined with a directional calculus, enhancing its expressiveness. An example is $\mathcal{EOPRA}$ [8, 25] that combines the concepts of directional relations of $\mathcal{OPRA}$ with qualitative distance as elevated points, describing the positions of objects (distance and orientation) from the point of view of an agent.

The $\mathcal{EOPRA}_m$ notation is derived from $\mathcal{OPRA}_m$ and it allows a joint representation of qualitative direction and distance between two points as: $A^n_m\angle^j_i\,{}^f_e B$, where m is the orientation granularity, n is the distance granularity, i and j are orientation relations, and e and f are distance relations. A granularity parameter m allows the definition of angular zones used to represent a world discretization. Given the granularity parameter m, the world is partitioned into $4m$ regions for each oriented object.

Figure 1 shows an example of a $\mathcal{EOPRA}$ relation between two elevated points (A and B): $A_4\angle^1_{13}\,{}^5_3 B$ representing that both A and B have been discretized into 16 orientation relations ($4m$) and 8 distance relations ($2m$). For relative orientation, A is in the sector 1 of B and B is in the sector 13 of A, and for relative distance, A is in the sector 5 of B and B is in the sector 3 of A.

For each jointly exhaustive and pairwise disjoint set of QSR relations there is a specific Conceptual Neighborhood Diagram (CND) [14]. A CND is a graph with nodes corresponding to a relation between spatial entities and edges corresponding to a pair of conceptual neighbors (i.e. there is no other relation from the set

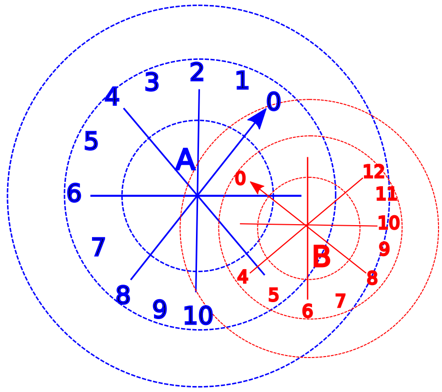

Fig. 1. $\mathcal{EOPRA}_4$ relation $A\ {}_4\angle^1_{13}\,{}^5_3\ B$.

that represents the transition from one relation to the other in the pair). Randell and Witkowski [32] have used CND and similarity matrix as a tool to compare and measure the distance between sets of spatial regions. The work of Weghe and Maeyer [7] used a QSR formalism and its CND to represent and reason about movements of objects, measuring the distance between the relations of two objects. In this paper, we apply a Conceptual Neighborhood Diagram as a tool to measure the distance between a new problem and items in a case base in order to retrieve the most similar case to the problem. This idea is developed in the next section.

2. Problem Formulation

This section presents the Qualitative Case-Based Reasoning (Q-CBR) method, the qualitative spatial modeling for the cases, the CND of $\mathcal{EOPRA}_m$ and the description of the use of CND as a tool for similarity measuring, defining a new retrieval algorithm for CBR.

2.1. Qualitative direction and distance

This work uses $\mathcal{EOPRA}_m$ to represent the relations between any two objects in the RoboCup domain as a tuple of orientation and distance. Based on the work of Moratz and Wallgrün [24] and Dorr, Latecki and Moratz [8], we have considered the viewpoint orientation as being the front of the agent and the granularity parameter $m = 6$, creating 24 direction sectors. These direction sectors are grouped into 8 regions: *left (l)*, *right (r)*, *front (f)*, *back (b)*, *left-front (lf)*, *right-front (rf)*, *left-back (lb)* and *right-back (rb)*. Figure 2 shows the direction sectors and regions defined, where each region is composed of three sectors. We have considered that *left-front*, *right-front*, *left-back* and *right-back*

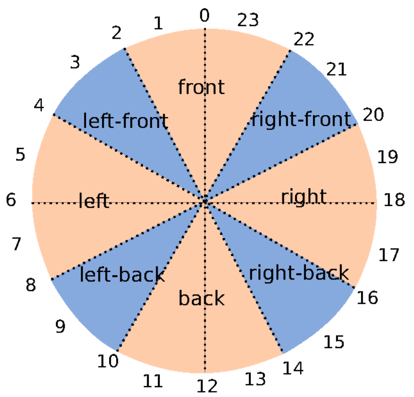

Fig. 2. Qualitative orientation representation.

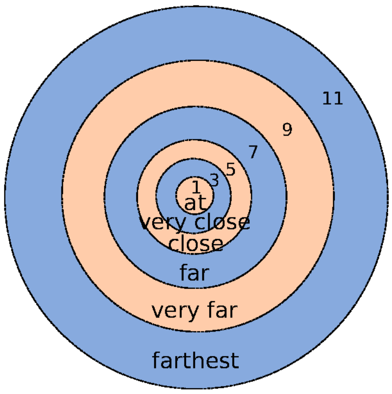

Fig. 3. Qualitative distance representation.

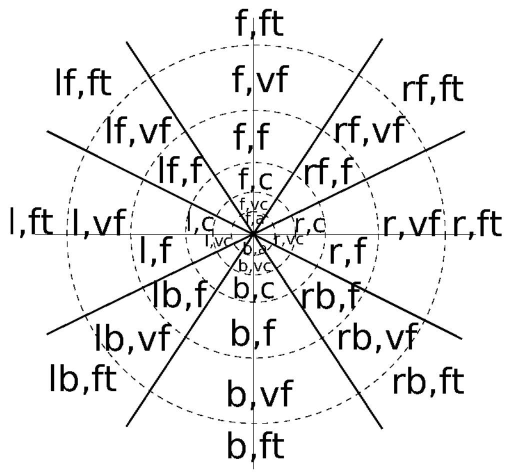

Fig. 4. Qualitative relations for distance and orientation.

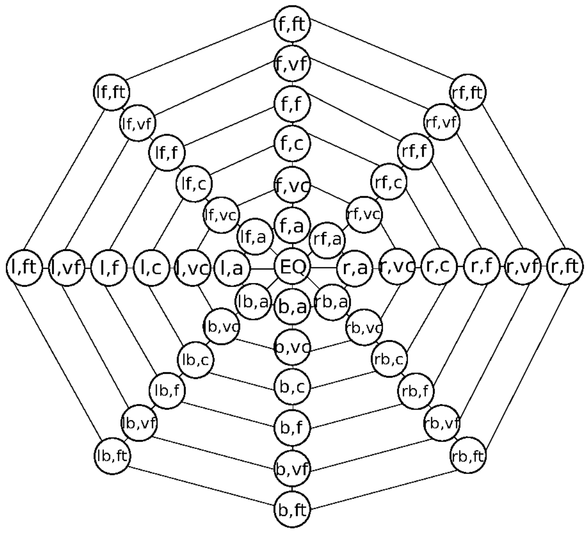

Fig. 5. CND of the proposed $\mathcal{EOPRA}_6$ representation.

are transition regions, so they have a smaller angular region.

Considering an environment of length 9.00 meters and width 6.00 meters (a robot-soccer field), a granularity parameter of $n = 6$ was assumed defining 12 distance sectors, numbered from 0 to 11. These distance sectors are then grouped into 6 categories: *at (a)*, *very close (vc)*, *close (c)*, *far (f)*, *very far (vf)* and *farthest (ft)*. Figure 3 shows the distance regions created. Based on Moratz and Wallgrün [24], and considering the agent's height (0.55 meters), the regions were defined with respect to the observer, such that *at* refers to an object placed closer than 0.33 meters, *very close* refers to an object placed between 0.33 and 0.66 meters, *close* represents an object placed between 0.66 and 1.00 meter, *far* refers to an object placed between 1.00 and 1.50 meters, *very far* is related to an object placed between 1.50 and 3.00 meters, and *farthest* refers to an object at more than 3.00 meters.

In this work we consider a granularity parameter of $m = n = 6$ (to both the distance and direction partitions), defining $\mathcal{EOPRA}_6$, as presented in Figure 4. $\mathcal{EOPRA}_6$ is composed of 48 qualitative regions plus one, labeled *equal (EQ)*, at the center of the relations, corresponding to the observer's position. A Conceptual Neighborhood Diagram for $\mathcal{EOPRA}_6$ is shown in 5. In this paper we consider a cost of 1 for each transition between adjacent relations in this CND.

The notion of similarly and CND in the work of Randell and Witkowski [32] can be used to define a distance function $Dmin_\phi(X_1, X_2)$ that takes two spatial relations X_1 and X_2 and maps them to the minimum CND (node to node) distance. In other words, $Dmin_\phi(X_1, X_1) = 0$, $Dmin_\phi(X_1, X_2) > 0$ if and only if $X_1 \neq X_2$ [32]. Using the distance function, the CND can be presented as a similarity matrix, in which the minimal CND distance can be computed using any algorithm to find the shortest path between nodes in a graph, such as Dijkstra's algorithm [4]. In this work, CND of $\mathcal{EOPRA}_6$ resulted in a similarity matrix with the minimal path (distance) between each 49 relations, allowing a fast case comparison during the retrieval step[1].

2.2. Qualitative case representation

Inspired by the work of Ros et al. [35], we define a case (C) as the problem description (P) and the solution description (A), represented by:

$$C = (P, A). \quad (7)$$

The problem description (P) corresponds to the qualitative spatial relations between an agent and the objects in the environment, given by the qualitative direction and distance to each object, from the agent's viewpoint. P is given by:

$$P = \{R_1 : [O_1, \ldots, O_u], \ldots, R_v : [O_1, \ldots, O_u]\}, \quad (8)$$

where v is the number of robots that take part in the case solution, u is the total number of objects that each agent can perceive, R_i is the agent's label and $O_1, O_2, ..., O_u$ are the qualitative relations between an object and the current agent R_i (each relation is an orientation and distance tuple). By objects, in the domain considered, we mean the ball and the other robots that can be perceived by the agent.

The solution description (A) is that defined in Section 1 above.

In contrast to the work presented in [35] (described in Section 1), the use of the case scope (K) is not necessary in the qualitative case representation proposed here, since in the present paper objects are located in qualitative regions. In this work K is only necessary in the experiments (Section 3), where the results obtained with our method is compared to those of Ros et al. [35].

2.3. Qualitative case retrieval

The present work uses the distance between relations in the CND to compute the similarity between the new problem and the cases in the case base. This can be done by means of a distance function based on the matrix described in Section 2.1. This qualitative distance function can be defined as:

[1]The distance matrix for $\mathcal{EOPRA}_6$ is available at the URL https://goo.gl/photos/nJ83KngMH6i789xz7

$$Dist_Q(p,c) = \quad (9)$$
$$\sum_{i=1}^{v} Dmin_\phi(R_i{}^c, R_i{}^p) + \sum_{j=1}^{u} Dmin_\phi(O_j{}^c, O_j{}^p),$$

where v is the number of robots that take part in the case solution, u is the number of objects that each agent can perceive, $R_i{}^c$ is the qualitative position of each robot i in the case and $R_i{}^p$ is the qualitative position of robot i in the given problem, $O_j{}^c$ is the qualitative position of each object j in the case and $O_j{}^p$ the object j qualitative position in the problem.

The distance function is used to calculate case similarity by means of the qualitative similarity function defined in Equation 10.

$$Sim_Q(p,c) = \quad (10)$$
$$\frac{CND_{MaxDist} \times (v+u) - Dist_Q(p,c)}{CND_{MaxDist} \times (v+u)},$$

where v and u are as defined in the qualitative distance function and $CND_{MaxDist}$ is the maximum distance between two objects in the CND. The result is normalized.

A retrieved case is not always directly applicable to the problem at hand without some adaptation. If this is the case, the qualitative adaptation cost function (shown in Equation 11) is applied.

$$Cost_Q(p,c) = \sum_{i=1}^{v} Dmin_\phi(R_i{}^c, R_i{}^p), \quad (11)$$

where v is the number of robots that take part in the case solution, $R_i{}^c$ is the qualitative position of each robot i in the case and $R_i{}^p$ is the qualitative position of robot i in the problem. The adaptation cost function includes only robots that are in the agent's team, meaning that their position can be controlled (i.e., adapted). The adaptation cost is the cost to move the robots of the team to the position that is described in the most similar candidate case, and it reflects how much this adaptation costs.

Algorithm 1 represents the proposed retrieval method based on the CND distance measure and adaptation cost. This algorithm has two lists: *sim_candidates* which contains cases whose similarity value is greater than a *threshold*; and the list *adapt_candidates* that is used to compute the adaptation cost of the candidate cases, sorted in ascending cost order. Lines 2-11 of Algorithm 1 search for candidate cases in the entire case base. Line 3 measures the qualitative similarity from

Algorithm 1 Retrieval step using CND similarity measure.

```
 1: function RETRIEVE(Problem p, Case base CB)
 2:     for each case c ∈ CB do
 3:         sim_value = Sim_Q(p, c)
 4:         if sim_value = 1 then
 5:             return c
 6:         else
 7:             if sim_value > threshold then
 8:                 insert(sim_value, c, sim_candidates)
 9:             end if
10:         end if
11:     end for
12:     if empty(sim_candidates) then
13:         return reactive_case
14:     end if
15:     for each case c ∈ sim_candidates do
16:         adapt_value = Cost_Q(p, c)
17:         insert(adapt_value, c, adapt_candidates)
18:     end for
19:     sort(adapt_value, adapt_candidates)
20:     return first(c, adapt_candidates)
21: end function
```

problem to case using Equation 10. In lines 4-5, if there is a case equal to the problem, the function returns the case and ends the search. If no case is found within the similarity range allowed, a pre-defined reactive case is returned (lines 12-13). A reactive case consists of a naïve behavior, in which the robot searches for the ball, walks toward it, aligns itself with respect to the opposing goal and kicks the ball forward. Lines 15-20 compute the cost of adaptation of each case found in the previous steps. The list of cases is sorted by the adaptation cost, and the case with the lowest adaptation cost is returned (*sim_value* is the second sort criteria).

Figure 6 shows an example of a qualitative retrieval task with three stored cases. In this example, one agent is the reference (positioned as *EQ*) and two other objects are randomly positioned in the environment. Consider one teammate robot ($u = 1$) and three objects, for instance two opponent robots and one ball ($v = 3$), where the teammate is the only object that can be adapted in the case. According to the distance matrix, $CND_{MaxDist}$ is equal to 8.

Figure 6 (top-left) shows the CND of the new problem, representing a snapshot of the objects' position in the environment, where the ball is placed *very close* and in front of the robot (*f,vc*), the teammate is placed to the left and far from the robot (*l,f*), one opponent positioned on the left and close to the robot (*l,c*) and

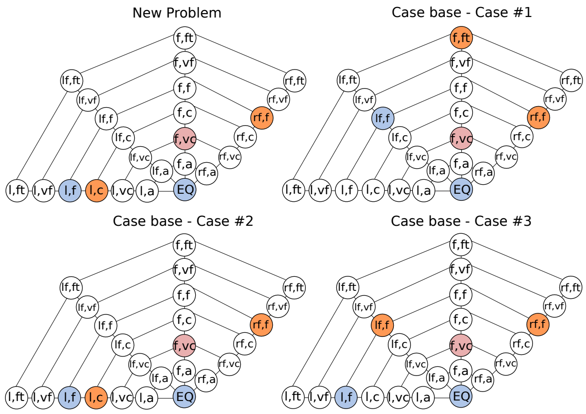

Fig. 6. Example of stored cases and case retrieval.

another opponent positioned on the right-front and far from the reference robot (*rf,f*).

After running the retrieval algorithm the result is: (a) case #1 (top-right) with $Dist_Q = 6$, $Sim_Q = 0.8125$ and $Cost_Q = 1$; (b) case #2 (bottom-left) with $Dist_Q = 0$, $Sim_Q = 1$ and $Cost_Q = 0$; (c) case #3 (bottom-right) with $Dist_Q = 2$, $Sim_Q = 0.9375$ and $Cost_Q = 0$. In this example, case #2 will be retrieved because it has $Sim_Q = 1$, but if case #2 would be discarded, case #3 could be retrieved since it has the lowest adaptation cost.

2.4. *Qualitative case reuse*

The reuse step consists of adapting the position of the robots in the problem to the qualitative position of the retrieved case. Basically, this step contains three agents: the coordinator robot (R_{coord}), which coordinates the retrieval and reuse steps, the executor robot (R_{exe}), a robot that is part of the solution, and a retrieved robot (R_{ret}), a virtual robot which represents the R_{exe}'s position of the retrieved case.

The reuse step focuses on calculating how the R_{exe} can reach the R_{ret}'s position, and the actions it must perform to reach this position. But before sharing the retrieved case to the agents, an intermediate step is necessary: the adaptation step.

The adaptation step is performed by the Composition Algorithm (CA) proposed by Perico et al. [27] which infers the qualitative orientation and distance from R_{exe} to R_{ret}. The CA (presented in Algorithm 4) uses two other algorithms to infer the qualitative distance and direction: Algorithm 2 infers a set of $\mathcal{OPRA}_m$ relations, and Algorithm 3 restricts the set of relations by means of triangulation.

Algorithm 2 uses the $\mathcal{OPRA}_m$ algorithm proposed by Mossakowski and Moratz [25] and returns a set of possible direction relations. Lines 5 and 14 compute the composition of $\mathcal{OPRA}$ relations as defined by Mossakowski and Moratz [25]:

$$
\begin{aligned}
& opra({}_m\angle_i^j, {}_m\angle_k^k, {}_m\angle_s^t) \text{ iff} \\
& \exists 0 \leqslant u, v, w < 4m . turn_m(u, -i, s) \wedge \\
& turn_m(v, -k, j) \wedge turn_m(w, -t, l) \wedge \\
& triangle_m(u, v, w)
\end{aligned} \tag{12}
$$

According to [25], $turn_m(a,b,c)$ detects complete turns, while $triangle_m(a,b,c)$ detects a triangle by means of the sum of the angles and the comparison of the sign of the angles ($sign_m(d)$). A triangle is verified as:

$$
\begin{aligned}
& triangle_m(a,b,c) \text{ iff} \\
& turn_m(a,b,c-2m) \wedge (a,b,c) \neq \\
& (2m,2m,2m) \wedge \\
& sign_m(a) = sign_m(b) = sign_m(c)
\end{aligned} \tag{13}
$$

Equation 14 verifies a complete turn and Equation 15 returns the sign of an angle, as:

$$
\begin{aligned}
& turn_m(a,b,c) \text{ iff} \\
& |((a+b+c+2m) \bmod 4m) - 2m| \leqslant \\
& (a \bmod 2m) \times (b \bmod 2m)
\end{aligned} \tag{14}
$$

$$
sign_m(d) = \begin{cases} 0, & \text{if } (d \bmod 4m = 0) \vee \\ & \quad (d \bmod 4m = 2m) \\ 1, & \text{if } (d \bmod 4m < 2m) \\ -1, & \text{otherwise} \end{cases} \tag{15}
$$

$\mathcal{OPRA}_m$ algorithm and the used functions are well-defined by referred authors.

Let i, j, k and l be known object relations and s and t be unknown relations. Given a set of relations between the objects A, B and C, where $A_m\angle_i^j B$, $B_m\angle_k^l C$ and $A_m\angle_s^t C$, the algorithm infers the set of possible directions, i.e., it checks which values s can assume when t is given; or which values of s and t can assume when t is not given, and returns all compositions that hold. Using triangulation, Algorithm 3 reduces the number of possible relations in the disjunction, resulting in a qualitative direction.

Finally, CA calculates the rough distance between R_{coord}, R_{exe} and R_{ret} and discretizes it into qualitative distances, as presented in Section 2.1.

Algorithm 2 $\mathcal{OPRA}_m$ - Inferring the set of relations $\hat{s}$ or $\hat{s}$ and $\hat{t}$ of $\mathcal{OPRA}_m$ for non-coincident points.

1: **function** OPRAINFERENCE(Granularity m, Relations i, j, k, l, t)
2:   **if** relation $t = \emptyset$ **then**
3:     **for** each $s_{test} \in relations$ **do**
4:       **for** each $t_{test} \in relations$ **do**
5:         **if** $opra(m, i, j, k, l, s_{test}, t_{test})$ **then**
6:           $insert(s_{test}, \hat{s})$
7:           $insert(t_{test}, \hat{t}$
8:         **end if**
9:       **end for**
10:     **end for**
11:     **return** $\hat{s}$, $\hat{t}$
12:   **else**
13:     **for** each $s_{test} \in relations$ **do**
14:       **if** $opra(m, i, j, k, l, s_{test}, t)$ **then**
15:         $insert(s_{test}, \hat{s})$
16:       **end if**
17:     **end for**
18:     **return** $\hat{s}$
19:   **end if**
20: **end function**

Algorithm 3 Triangulation - Restricting the set of $\hat{s}$ relations by triangulation.

function RESTRICTINGOPRA(Granularity m, Relations $\hat{s}$, Angle α)
  $i_{aux} = DiscretizeToOpra(m, i, \alpha)$
  $i_{restr} = sum(i, i_{aux})$
  if i_{restr} = even number **then**
    $\hat{c} = [i_{restr} + 1, i_{restr} - 1]$
  else
    $\hat{c} = [i_{restr}]$
  end if
  for each $n \in \hat{s}$ **do**
    if $n \subset \hat{c}$ **then**
      $insert(n, \hat{a})$
    end if
  end for
  return $\hat{a}$
end function

Algorithm 5 presents the proposed reuse method. As the retrieved case contains the qualitative position of the coordinator robot's point of view, it needs to be converted to the executor robot's point of view, that has its own qualitative relations about the world. The algorithm receives the problem and the retrieved case and, for each robot that is part of the solution, an adapted position is generated based on the executor robot's point of view (line 3). Line 4 shares with the executor robot the adapted positions and line 5 shares the actions it must perform to solve the problem.

Algorithm 4 Composition Algorithm.

```
function CA(Problem p, Case c, Integer coord, Integer r)
    i, j, k, l, s, t = ObtainRelation(p, c)
    ŝ = OpraInference(m, i, j, k, l, t)
    qDirect = RestrictingOpra(m, ŝ, alpha)
    qDist = CalculateDistance(p, c, coord, r)
    return qDirect, qDist
end function
```

Algorithm 5 Reuse step using Composition Algorithm.

```
1: function REUSE(Problem p, Case c)
2:     for each robot r ∈ executors_robot do
3:         adapt_pos = CA(p, c, coord, r)
4:         send_positions(adapt_pos, r)
5:         send_actions(c)
6:     end for
7: end function
```

In order to exemplify the reuse step using CA, Figure 7 presents the coordinator robot's (R_{coord}) point of view about the executor robot's (R_{exe}) qualitative position, the robot's position on the retrieved case (R_{ret}), and the executor robot's point of view about the coordinator robot's qualitative position. R_{coord} can easily obtain the angle β, so it can calculate the angle α using the law of cosines. After obtaining α, the angle is discretized according $\mathcal{OPRA}_6$ definitions, representing the R_{exe}'s qualitative orientation to the R_{ret} position. The R_{exe}'s qualitative distance is calculated by the Pythagorean theorem and the distance is discretized according $\mathcal{EOPRA}_6$. In Figure 7 (left) the R_{coord} searches for the objects' position on the environment and finds the R_{exe}'s position in *left,farthest*; it retrieves a case and selects the most similar case where the robot's position in the case is *front,very far* (R_{ret}). Figure 7 (center), by running the Composition Algorithm, the adapted position to the R_{exe}'s point of view is obtained, returning the regions *right-front,farthest* that are shared among the agents. Figure 7 (right) shows that R_{exe} executes the movements to reach R_{ret}'s position and performs the actions to solve the problem[2].

3. Experiments and Results

This section presents the experiments and results obtained applying the algorithms introduced in this work to the humanoid-robot soccer environment. Two types of experiments are performed: (1) in a simulator: where we compared our qualitative case-based algorithms with the metric approach proposed by Ros et al. [35] and with a reactive agent; (2) in a real humanoid-robot domain: where our qualitative case-based algorithms were compared with a reactive agent.

The experiments in this section aim at analyzing which of the approaches resulted in more goals scored and fewer errors, and to compare the retrieval time of cases between metric and qualitative methods. The following sections present the software architecture used in the experiments, describe the two experiments performed as well as the results obtained.

3.1. RoboFEI Humanoid Soccer Simulator

Both simulation and real robot experiments were conducted using a software developed with the purpose of enabling the reproduction of experiments and performance comparison of different algorithms: the RoboFEI Humanoid Soccer Simulator. This software uses the Cross architecture described in Perico et al. [29], which is based on low-level tasks, such as vision, control and communication processes, allowing users to develop and test high-level decision-making algorithms in simulation and transfer them to real robots without the need of much software modifications.

The Cross architecture (Figure 8) is a hybrid architecture, because there are some aspects of reactive and hierarchical paradigms. The processes are completely independent from one another, and they can be grouped into vision, localization, decision-making, planning, communication, perception and control systems, each of which communicate to each other using a shared memory. A major process, named management process, is responsible to launch, synchronize and monitor all the other processes.

[2]The direction and distance labels were hidden to allow clear visualization.

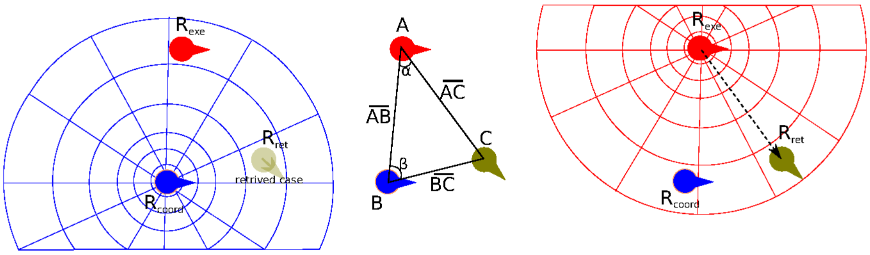

Fig. 7. Example of Reuse step using Composition Algorithm: changing frame of reference

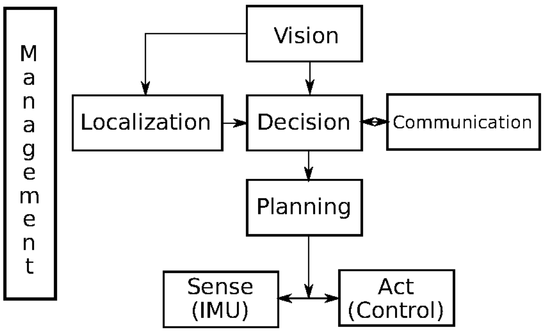

Fig. 8. RoboFEI Humanoid Soccer Simulator software architecture.

The RoboFEI Humanoid Soccer Simulator [28] is an open-source simulator, written in Python, but it allows the integration with other programming languages like C and C++. This simulator has distinct control and vision processes for simulation and real robots experiments, but decision, planning and localization code is shared between both types of experiments. The simulator environment is a soccer field that obeys the RoboCup Humanoid Kidsize rules [34].

In this work, experiments were performed using an Intel NUC i5 with 8GB SDRAM running Ubuntu 14.04 LTS. For reproducibility reasons, the simulator, along with the source code used in the experiments reported below, are available at the URL http://fei.edu.br/~rbianchi/software.html. Some videos of simulation and experiments with real robots can be found at https://goo.gl/nquuKc.

3.2. *Experiments in the simulator*

In order to perform the experiments in the simulator, two scenarios were created, as shown in Figure 9. In the first scenario (Figure 9a) a ball and a robot (B1) are positioned in the center of the field and the teammate (B2) is positioned on the left and in the middle of the field. There are also a goalkeeper (B3) and three opponents (B4, B5 and B6) positioned as defenders. In the second scenario (Figure 9b) the ball, the robot (B1) and one teammate (B2) are positioned in the attacking field and four opponent robots are positioned as in scenario #1.

A centralized case base was used in both scenarios, in which the robot that is the closest to the ball assumes the position of *coordinator*, being responsible for the retrieval process (in both, the qualitative and the metric approaches) and for the coordination of collective actions in the reuse process. The coordinator robot transmits wirelessly the adapted positions and actions that the other robots must perform, which are received and executed by the *executor* agents.

Two case bases were created and populated: (1) a metric case base: with 20 real cases and 180 random cases, with random positions and three actions for each robot, and (2) a qualitative case base: with the same 200 cases represented as qualitative relations. The 20 real cases represent specific positions of the robots and the actions that each robot must perform to solve a problem, such as a soccer set play. In the reactive approach, only reactive actions were implemented, in which the robot looks for the ball, walks toward it, aligns itself with respect to the ball and kicks it, without any team coordination.

Although the world discretization presented in Section 2.1 defines 8 qualitative regions of direction, during the experiments only 7 qualitative regions were used due to the RoboCup rules that limits camera panning movement to $\pm 135°$, discarding the region named as *back*. This limitation does not change the discretization of the world and the problem formulation presented in Section 2 remains the same.

For comparison purposes, 40 trials of 10 minutes were performed for each scenario and for each algorithm tested. In each trial, the number of goals scored was considered, as well as the number of near misses and the number of errors. A near miss happens when the ball is kicked and goes out of the field, but passing near one of the goalposts. An error is a situation where a robot cannot find the ball or when the sequence of

(a) Simulated scenario #1 (b) Simulated scenario #2.

Fig. 9. Two scenarios created for simulated experiments

coordinated actions do not result in a goal or in a near miss.

Table 1 shows the results obtained for each of the algorithms tested. Q-CBR had a higher average number of goals when compared to the metric algorithm. Both Q-CBR and the metric algorithm outperform the reactive agent in the scenarios considered. Student's t-test [26] was applied in each scenario and the results indicate that Q-CBR is statistically better, with a certainty of at least 90%.

Another advantage of using Q-CBR is in the case-retrieval time. The results presented in Table 2 show that Q-CBR is about 3 times faster than the metric algorithm. The improvement in the retrieval time is due to the strategy of the qualitative similarity measurement, as shown in Algorithm 1. Student's t-test was also used in order to compare the computational performance of Q-CBR to the metric algorithm. The results shown in Table 2 indicate that Q-CBR is statistically better than the metric CBR with a certainty of at least 99%.

3.3. Experiments with real robots

Experiments with real robots were conducted with two humanoid robots based on the Darwin-OP robot, adapted to use a computer with the same configuration of the simulation experiments. The robots have 0.55 centimeters of height and walk an average of 70 cm/min. They have 22 degrees of freedom, and are equipped with an UM6 Ultra-Miniature Orientation Sensor and a Logitech HD Pro Webcam C920. The main difference between these robots and Darwin-OP is in their chest, where the FEI humanoids carry an Intel NUC i5 with 8GB SDRAM running Ubuntu 14.04 LTS responsible to run the code.

The scenario was similar to the second scenario used in the simulation experiments (Section 3.2), with the same case base and the same implementation of the qualitative and reactive approaches in the simulator. The implementation on real robots did not require many code re-writing, only the vision and the control modules of the Cross architecture had to be changed. The robots were able to recognize the ball and other robots, communicate with each other and perform basic tasks such as walking, turning, kicking and passing the ball.

The experiments consist of 5 trials of 10 minutes and, as in the previous experiments, the average number of goals, the number of near misses and the number of errors were considered. Table 3 presents the results of Q-CBR and the reactive algorithm. Q-CBR algorithm outperforms the reactive agent, scoring a average of 1.20 goals and 2.00 near misses, per trial. Experiments were not conducted with the metric CBR algorithm due to the fact that, in contrast to the simulator, coordinates of the robots and the ball in the field are not given in the real-robot scenario. The average retrieval time was similar to the simulation (about 0.0076 seconds), although the number of goals scored could be higher with the improvement of some aspects of the robot, such as the control of walk, kick or pass. Student's t-test was used in order to compare the performance of the proposed approach with the reactive algorithm. In this experiment, the Q-CBR is statistically better for scored goals than the reactive approach with a certainty of at least 95%.

Figure 10 presents a sequence of actions performed by the robots during the experiments. As it can be seen, the coordinator robot walks to the ball while the executor robot waits until it receives an action. When the co-

Table 1

Results of simulation experiments (mean and standard deviation for 40 trials of 10 minutes).

Scn.	Method	Goals	T-value	Near Misses	T-value	Errors	T-value
#1	CBR	2.58 ± 1.18	1.63 (90%)	2.45 ± 1.22	7.72 (99%)	2.93 ± 1.23	5.39 (99%)
	Q-CBR	2.98 ± 1.01		0.73 ± 0.71		1.58 ± 0.99	
	Reactive	0.33 ± 0.52	-	2.08 ± 1.15	-	3.88 ± 1.54	-
#2	CBR	2.55 ± 1.53	1.69 (95%)	2.65 ± 1.24	2.43 (99%)	2.78 ± 1.06	4.92 (99%)
	Q-CBR	3.10 ± 1.37		2.03 ± 1.06		1.73 ± 0.84	
	Reactive	0.48 ± 0.71	-	1.78 ± 1.06	-	3.53 ± 2.09	-

ordinator robot has the possession of the ball, it scans for objects in the field until it finds a teammate (the executor robot). Then, the coordinator robot retrieves a case, shares the adapted position and action and, finally, passes the ball to the executor robot. The executor robot walks to the adapted position and kicks the ball to the goal.

4. Related Work

Several CBR references can be found in the literature using cases with qualitative representation but with no relation to QSR approaches. For instance, Du, Liang and Sun [9] present an algorithm that integrates spatial relations in CBR, extracting the similarity coefficient of cases and problem, matching each other with respect to some characteristic. The work of Liu, Fu and Zhou [20] proposes a CBR algorithm based on qualitative causality, combining metric and qualitative similarity methods to measure and retrieve a case. The work reported in the present paper uses qualitative spatial reasoning to represent the objects' position and retrieves the most similar case based on a Conceptual Neighborhood Diagram. So, the neighborhood diagram facilitates a definition of distance between qualitative relations, that is used to calculate an adapted position for the agent.

The work presented in Jære, Aamodt and Skalle [15] uses temporal reasoning and CBR, where the cases are represented as temporal graphs and the retrieval step is performed matching the graphs and creating a similarity degree. Dufour-Lussier et al. [10] propose a method for adaptation of spatial and temporal cases during the reuse step of CBR, where the similarity between two

Table 2

Performance of the CBR and Q-CBR retrieval step time in seconds, averaged over 40 trials; absolute t-value and confidence interval (in %)).

Scenario	Method	Retrieval (seconds)	T-value
#1	CBR	0.0218±0.0042	19.902 (99%)
	Q-CBR	0.0076±0.0017	
#2	CBR	0.0228±0.0040	22.746 (99%)
	Q-CBR	0.0075±0.0014	

Table 3

Results with real robots experiments (mean and standard deviation for 5 trials of 10 minutes).

Method	Goals	Near Misses	Errors
Q-CBR	1.20 ± 0.75	2.00 ± 1.41	2.80 ± 1.16
Reactive	0.40 ± 0.49	1.16 ± 1.16	2.60 ± 1.02

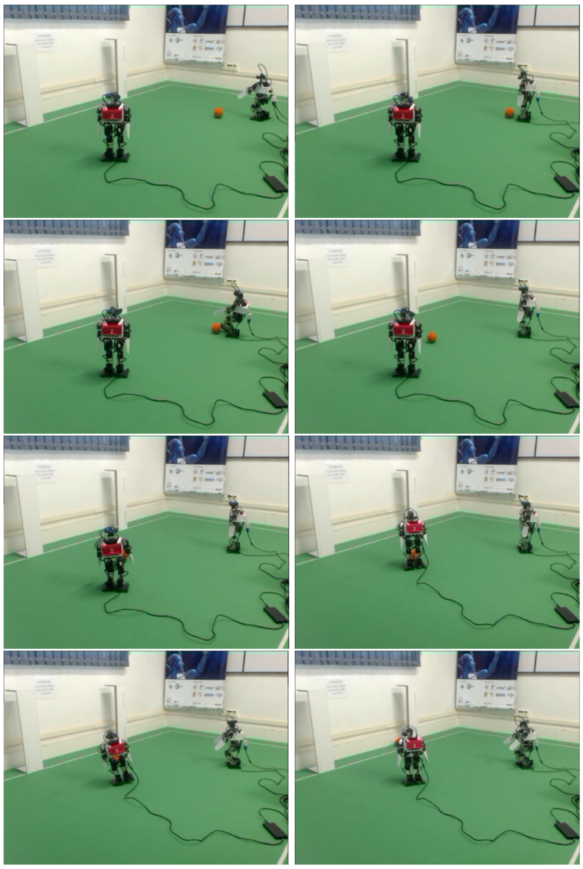

Fig. 10. Sequence of actions performed by the robots

scenarios is measured based on the distance between the considered relations. It differs from the retrieval algorithm presented in this paper since, here, each qualitative position of the objects in the cases is compared with the objects in the problem, retrieving the cases that have the minimal cost of adaptation among the cases that have the CND that is the most similar the problem's CND.

Young and Hawes [40, 41] applied the *Star Calculus* to represent the qualitative direction between entities on the RoboCup Soccer Keepaway [38]. In another environment, Southey and Little [37] applied QSR to games, where the objects' position were modeled as qualitative spatial relations. The results of these papers show that the use of QSR is an interesting way to generalize the objects' position representation. Our work uses $\mathcal{EOPRA}$ and compares its retrieval time with respect to a metric-based algorithm. We also performed experiments on real robots, with limited view of the environment.

5. Conclusion

This work introduced and analyzed an algorithm called Q-CBR, a case-based reasoning method assuming a qualitative spatial representation of the domain. By modeling cases in a CBR system as qualitative spatial relations, and using the notion of Conceptual Neighborhood Diagram and cost functions as similarity measure, a faster case-retrieval method was obtained when compared with a metric algorithm. Besides, in some domains, qualitative representation is more appropriate than numerical. The humanoid-robot soccer is one of these domains, as the robots are not capable of computing the precise positions of objects in the field.

Aiming at evaluating the method proposed in this paper, experiments were performed in a simulated environment with a small case base, using two distinct scenarios. The proposed method was also evaluated in a real humanoid-robot scenario. The results show that the teams that used Q-CBR had a higher number of scored goals and lower (more efficient) retrieval time. In all experiments executed in this work, the algorithm introduced in this paper (Q-CBR) was three times faster than the metric algorithm tested, which allows the execution of Q-CBR in robots with a limited processing power and hardware.

Future work shall consider the implementation of the complete Q-CBR cycle and the investigation of the performances related to the revision and retention processes. We also propose to implement Q-CBR as a multi-agent system, where each robot has its own case base and cooperates with the other team members to define which case would be better to solve the problem. Given the interesting results obtained by Q-CBR, we propose to implement this work following the Goal-Driven Autonomy (GDA) model, where the agents will be able to learn, plan and reuse goal policies.

6. Acknowledgements

Thiago P. D. Homem acknowledges support from CAPES and PRP/IFSP. Danilo H. Perico acknowledges support from CAPES. Paulo E. Santos acknowledges support from CNPq grant 307093/2014-0 and FAPESP grant 2016/18792-9. Ramon L. de Mantaras acknowledges support from Generalitat de Catalunya Research Grant 2014 SGR 118 and CSIC Project 201550E022.

References

[1] Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7:39–59.

[2] Altaf, M. M., Elbagoury, B. M., Alraddady, F., and Roushdy, M. (2015). Extended case-based behavior control for multi-humanoid robots. *International Journal of Humanoid Robotics*.

[3] Cohn, A. G. and Renz, J. (2008). Handbook of knowledge representation. chapter Qualitative Spatial Representation and Reasoning, pages 551–596. Elsevier.

[4] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press, 2nd edition.

[5] Davoust, A., Floyd, M. W., and Esfandiari, B. (2008). *Use of Fuzzy Histograms to Model the Spatial Distribution of Objects in Case-Based Reasoning*, pages 72–83. Springer Berlin Heidelberg, Berlin, Heidelberg.

[6] de Mántaras, R. L., McSherry, D., Bridge, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M., Forbus, K., Keane, M., Aamodt, A., and Watson, I. (2006). Retrieval, reuse, revision, and retention in CBR. In *The Knowledge Engineering Review*, volume 20, pages 215–240.

[7] De Weghe, N. V. and De Maeyer, P. (2005). Conceptual Neighbourhood Diagrams for Representing Moving Objects. *Perspectives in Conceptual Modeling, LNCS*, 3770:1–11.

[8] Dorr, C. H., Latecki, L. J., and Moratz, R. (2015). Shape Similarity Based on the Qualitative Spatial Reasoning Calculus eOPRAm. In Fabrikant, S. I., Raubal, M., Bertolotto, M., Davies, C., Freundschuh, S., and Bell, S., editors, *12th International Conference, COSIT 2015, Proceedings*, pages 130–150. Springer International Publishing, Santa Fe, NM, USA.

[9] Du, Y., Liang, F., and Sun, Y. (2012). Integrating spatial relations into case-based reasoning to solve geographic problems. *Knowledge-Based Systems*, 33:111–123.

[10] Dufour-Lussier, V., Le Ber, F., Lieber, J., and Martin, L. (2012). Adapting Spatial and Temporal Cases. In Agudo, B. D. and Watson, I., editors, *20th International Conference, ICCBR 2012, Lyon, France, September 3-6, 2012. Proceedings*, chapter Adapting S, pages 77–91. Springer Berlin Heidelberg.

[11] Dylla, F. (2009). Qualitative Spatial Reasoning for Navigating Agents: Behaviour Monitoring and Interpretation - Ambient Assisted Living. In *Ambient Intelligence and Smart Environments*, chapter 3, pages 98–128. IOS Press Amsterdam.

[12] Floyd, M. W., Esfandiari, B., and Lam, K. (2008). A case-based reasoning approach to imitating robocup players. In Wilson, D. and Lane, H. C., editors, *Proceedings of the Twenty-First FLAIRS, 2008, Florida, USA*, pages 251–256. AAAI Press.

[13] Frank, A. U. (1991). Qualitative spatial reasoning about cardinal directions. *Proceedings of Auto-Carto 10 ACSM-ASPRS*, 6:148.

[14] Freksa, C. (1991). Conceptual neighborhood and its role in temporal and spatial reasoning. *Decision Support Systems and Qualitative Reasoning*, pages 181–187.

[15] Jære, M. D., Aamodt, A., and Skalle, P. (2002). *Advances in Case-Based Reasoning: 6th ECCBR 2002 Aberdeen, Scotland, UK, September 4–7, 2002 Proceedings*, chapter Representing Temporal Knowledge for Case-Based Prediction, pages 174–188. Springer Berlin Heidelberg, Berlin, Heidelberg.

[16] Karol, A., Nebel, B., Stanton, C., and Williams, M. (2003). Case based game play in the robocup four-legged league part i the theoretical model. In Polani, D., Browning, B., Bonarini, A., and Yoshida, K., editors, *RoboCup*, volume 3020 of *LNCS*, pages 739–747. Springer.

[17] Kolodner, J. (1993). *Case-based Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[18] Ligozat, G. (2013). *Qualitative Spatial and Temporal Reasoning*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

[19] Lin, Y., Liu, A., and Chen, K. (2002). A hybrid architecture of case-based reasoning and fuzzy behavioral control applied to robot soccer. In *Workshop on Artificial Intelligence (ICS2002)*, Hualien, Taiwan. National Dong Hwa University.

[20] Liu, Z., Fu, L., and Zhou, Y. (2014). Case-Based Reasoning Algorithm Based on Qualitative Causality. In *2014 Seventh International Joint Conference on CSO*, pages 519–523. Ieee.

[21] Marling, C., Tomko, M., Gillen, M., Alex, D., and Chelberg, D. (2003). Case-based reasoning for planning and world modeling in the robocup small sized league. In *Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World Modeling, Planning, Learning, and Communicating (IJCAI)*.

[22] Martin, A., Santos, P. E., and Safi-Samghabad, M. (2017). Consistency check in a multiple viewpoint system reasoning about occlusion. In *The 30th International Conference on Industrial, Engineering, Other Applications of Applied Intelligent Systems*. to appear.

[23] Moratz, R. (2006). Representing relative direction as a binary relation of oriented points. In Brewka, G., Coradeschi, S., and Perini, A., editors, *17th ECAI*, pages 407–411. IOS Press Amsterdam.

[24] Moratz, R. and Wallgrün, J. O. (2012). Spatial reasoning with augmented points: Extending cardinal directions with local distances. *Journal of Spatial Information Science*, 5(5):1–30.

[25] Mossakowski, T. and Moratz, R. (2012). Qualitative reasoning about relative direction of oriented points. *Artif. Intell.*, 180-181:34–45.

[26] Nehmzow, U. (2006). *Scientific Methods in Mobile Robotics: Quantitative Analysis of Agent Behaviour*. Springer-Verlag London.

[27] Perico, D. H., Bianchi, R. A. C., Santos, P. E., and de Mántaras, R. L. (2016a). Collaborative communication of qualitative spatial perceptions for multi-robot systems. In *Proc. 29th International Workshop on Qualitative Reasoning (IJCAI)*, pages 77–84, New York, NY, USA.

[28] Perico, D. H., Homem, T. P., Almeida, A. C., Silva, I. J., Vilao, C. O., Ferreira, V. N., and Bianchi, R. A. (2016b). A Robot Simulator Based on the Cross Architecture for the Development of Cognitive Robotics. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pages 317–322. IEEE.

[29] Perico, D. H., Silva, I. J., Homem, T. P. D., Destro, R. C., and Bianchi, R. A. C. (2014). Hardware and software aspects of the design and assembly of a new humanoid robot for robocup soccer. In *2014 Joint Conference on Robotics: SBR-LARS*, pages 73 – 78. IEEE.

[30] Randell, D., Witkowski, M., and Shanahan, M. (2001). From images to bodies: Modelling and exploiting spatial occlusion and motion parallax. *IJCAI International Joint Conference on Artificial Intelligence*, pages 57–63.

[31] Randell, D. A., Cui, Z., and Cohn, A. G. (1992). A Spatial Logic based on Regions and Connection. In *3rd. Int. Conf. on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann.

[32] Randell, D. A. and Witkowski, M. (2004). Tracking Regions using Conceptual Neighbourhoods. *Proceedings of the Workshop on Spatial and Temporal Reasoning ECAI - 2004*, pages 63–71.

[33] Renz, J. and Mitra, D. (2004). Qualitative Direction Calculi with Arbitrary Granularity. In *PRICAI 2004: Trends in Artificial Intelligence: 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004. Proceedings*, pages 65–74, Berlin, Heidelberg. Springer Berlin Heidelberg.

[34] RoboCup (2015). 2015 robocup soccer humanoid league rules and setup. *http://www.robocuphumanoid.org*.

[35] Ros, R., Arcos, J. L., de Mantaras, R. L., and Veloso, M. (2009). A case-based approach for coordinated action selection in robot soccer. *Artificial Intelligence*, 173(9-10):1014–1039.

[36] Santos, P. E., Ligozat, G., and Safi-Samghabad, M. (2015). An occlusion calculus based on an interval algebra. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 128–133.

[37] Southey, T. and Little, J. J. (2007). Learning Qualitative Spatial Relations for Object Classification. In *IROS 2007 Workshop: From Sensors to Human Spatial Concepts*.

[38] Stone, P., Sutton, R. S., and Kuhlmann, G. (2005). Reinforcement Learning for RoboCup Soccer Keepaway. *Adaptive Behavior*, 13(3):165–188.

[39] Wolter, D. and Wallgrün, J. (2012). Qualitative spatial reasoning for applications: New challenges and the SparQ toolbox. *Qualitative spatio-temporal representation and reasoning: Trends and future directions*, pages 336–362.

[40] Young, J. and Hawes, N. (2013). Predicting Situated Behaviour Using Sequences of Abstract Spatial Relations. *AAAI 2013 Proceedings, Fall Symposium Series*.

[41] Young, J. and Hawes, N. (2015). Learning by Observation Using Qualitative Spatial Relations. In *Proceedings of the 2015 AAMAS*, pages 745–751, Richland, SC.