

Voronoi Based Strategic Positioning for Robot Soccer

S. Kaden, H. Mellmann, M. Scheunemann, and H.-D. Burkhard

Department of Computer Science, Cognitive Robotics Group
Humboldt-Universität zu Berlin, Germany
{kaden,mellmann,scheunem,hdb}@informatik.hu-berlin.de

Abstract. Strategic positioning is a decisive part of the team play within a soccer game. In most solutions the positioning techniques are treated as a constituent of a complete team play strategy. In a comprehensive overview we discuss the team play and positioning methods used within RoboCup and extract the essential requirements for player positioning. In this work, we propose an approach for strategic positioning allowing for flexible formulation of arbitrary strategies. Based on the conditions of a specific strategy, the field is subdivided in regions by a Voronoi tessellation and each region is assigned a weight. Those weights influence the calculation of the optimal robot position as well as the path. A team play strategy can be expressed by the choice of the tessellation as well as the choice of the weights. This provides a powerful abstraction layer simplifying the design of the actual play strategy. We also present an implementation of an example strategy based on this approach and analyze the performance of our approach in simulation.

1 Introduction

With the advancement of RoboCup, team play turns into a key feature of success. While in simulation leagues a strong team play is a decisive aspect since the very beginning, in hardware leagues more basic aspects like motion, perception and modeling usually yield more overall improvement. A central part of team play is the *strategic positioning* of players on the field, i.e., the placement of robots not in possession of the ball. Hereby, two core issues have to be addressed: where does the player have to go to and how does he get there. The answer, depends among others on numerous factors, involving player positions on the field, ball position, distribution of team-mates and opponent players and so on. In this paper we present an approach which allows to formulate all the aspects necessary for the positioning and mechanisms to derive the target position and the path to it.

To achieve that, the whole field is separated into weighted regions. To obtain a practical separation, Voronoi tessellation is employed due to its convex structure and embedded neighboring relation. Each region is evaluated according to some criteria. The weights encode a variety of aspects, e.g. obstacles, free space, the probability that the ball is contained or the dominant regions of players. Based

on those weights, an optimal region can be determined, as well as a feasible path to this region.

For illustration and testing purposes we utilize a supporter example strategy to prove the functionality of our approach. Beside a simple heuristic strategy for positioning, a scalar field is exploited for strategy description and weight calculation for each region.

The residual paper is divided into four main parts. The next section offers a comprehensive overview of strategic positioning approaches proposed within the RoboCup. The description of the VBSM algorithm is given in section 3. In section 4 we discuss the experiments and their results. At the end we present a conclusion and provide an outlook for future work.

2 Related Work

The problem of the strategic positioning within the RoboCup context has been widely investigated in the past years. The ideas vary a lot and there seems to be no *standard approach*. Some major differences are surely reflections of the characteristics exposed by the different leagues. In general, the solutions within the simulation leagues are more elaborated, assume a much more rich and reliable world model and are more computationally complex in comparison to the solutions of the hardware leagues. In most cases the positioning issue is not solved separately, it is mostly part of an entire team play solution. In the following overview our main focus is rather on the positioning ideas than on the high-level parts necessary to organize a team like role assignment.

One of the most direct ways to position a robot is to calculate its target position as a geometric relation to its world model, which results in a reactive behavior. Although very simple, those techniques proved to be quite effective and robust to noisy data. An example for such positioning can be found in [1] by Carlos E. Agüero et al. which targeted the issue of role switching and role positions within the *Four Legged League* (4LL). Here the defender should occupy a point on the line between the goal and the ball to prevent the opponent attacker most effectively from scoring a goal. A supporter chooses the center of the rectangle stretched by the ball and the most distant opponent corner. Another example from 4LL is given by Phillips and Veloso, in [15]. They present reactive supporter strategies. Here the field is subdivided in rough fixed regions, such like offensive, defensive etc., which allows to choose different strategies depending on which region the ball is in. For instance, when the ball is in the offense region, the supporter covers a point left or right in a fixed distance beside the ball. Is the ball situated in the defense region, it is followed by the supporter in one axis which stays in the offense region. In addition, the supporter chooses a corner of the opponent penalty area if the ball is close to the opponent goal to catch the ball if it rebounds. In the *Humanoid League* (HL), K. Petersen, G. Stoll and O. von Stryk [14] developed another reactive behavior for a supporter. Similar to the previous approach, the supporter chooses a position relative to the ball. Thereby the relation may be adjusted depending on the game situation.

Potential fields are another very popular tool for positioning. A potential field is a function which assigns a direction to each point on the field and is usually formulated as the negative gradient of a scalar field. To navigate the robot can simply follow the direction of the field at its current position. This approach may represent the world state, e.g., obstacles are represented as maxima (*repeller*), as well as strategy, e.g., the target position is formulated as a minimum (*attractor*). Potential fields are prone to local minima, which is a minor issue within dynamic situations. They can adapt very smoothly in dynamic scenarios and can be resistant to noise. These properties make them a very tempting tool for dynamic planning scenarios with noisy data. A good example is presented by the team B-Human in [17](SPL). Here, the team play is organized in three fixed *formations* which are activated based on the state of the game, e.g., goal score. A formation defines a role for each player. A specific role defines the actual player position on the field, e.g., the target supporter position is calculated based on the position of the ball and the striker. This target position as well as the current game situation is formulated by a potential field, which is used for navigation. Thereby, the target position is formulated as an attractor whereas the other players, the own penalty area as well as the line between the ball and the opponent goal are formulated as repellers. This way the robot avoids obstacles and forbidden areas on its way to the target position. A very similar approach is presented by Work, Chown, Hermans, Butterfield and McGranaghan in [18](4LL). The formations are additionally organized in strategies and each role splits in a number of sub roles. A sub role is activated depending on the ball position on the field and defines the target position of the player. The positioning itself is also based on the potential fields and is formulated in a similar way to the Team B-Human approach.

In their work [13] (SPL) Nieuwenhuisen, Steffens, and Behnke consider the positioning of a robot as a path planning problem. Hereby, the task of approaching the ball while avoiding obstacles is directly addressed. A multi resolution occupancy grid in egocentric Cartesian and Log-Polar coordinates is used to model the obstacles. The resolution of the grid is higher in the proximity of the robot and the path is determined by the A* search. B-Human use in [17](SPL) an *extended bidirectional Rapidly-Exploring Random Tree* to estimate the path to the ball, which doesn't require discretization of the space.

Another way is to formulate the positioning task as an optimization problem where the world state and the strategy are encoded as conditions to be satisfied. In their work Kyrlov, Razykov and Hou [7,8](S2D) subdivide the field in a grid. Each cell is then rated according to certain criteria. For instance, the player should be open for a direct pass, the distance to opponent players should be maximized and the distance to the reference position defined by the strategy should be minimal. These criteria, e.g., the reference position, are defined by the formation and the roles of the players. The target position is then determined as a Pareto-Optimum based on those criteria. Both publications deal with different scenarios (offensive, defensive) and discuss different criteria for the optimization.

Voronoi diagram and its dual *Delaunay triangulation* state another popular tool used for the player positioning. Hidehisa Akiyama and Itsuki Noda [2](S2D) use in their work a Delaunay triangulation of the field to encode the positioning of the robots depending on the ball position. To construct such triangulation, a representative set of possible ball positions and the corresponding positions of the players are predefined. Those ball positions define the nodes of the triangulation. During the game the actual positions of the players are determined as an interpolation between the positions provided by the nodes of the triangle in which the ball is located. Another way to use Voronoi diagrams for positioning is presented by Hesam Addin Dashti et al. in [4](S2D). Thereby, the actual positions of the robots define the Voronoi cells, i.e., are the Voronoi sites. The repulsing and attracting properties of the objects on the playing field, e.g., ball, opponents, goal, etc., are modeled as forces which affect the agent and are represented by vectors. The vector to the center of gravity of the agent's Voronoi cell provides an additional alignment vector. Thus the player try to relax the Voronoi diagram and to keep as much distance between each other as possible, which results in a good field coverage. In [3](S2D) the technique is extended by including the opponent players as additional cells into the diagram.

Considering the time which is needed to reach a point instead of the Euclidean distance to it provides another kind of regions which are called *dominant regions* (DR). Figuratively speaking, a DR is defined as the area on the field which can be reached by a particular player before the others. In general, calculation of dominant regions requires a good motion model of the players which may be especially a problem for opponent robots. In [12](SSL) Nakanishi, Murakami and Naruse introduce the notion of a *dominant polygon* which essentially is an approximation of the area which can be reached by the robot within a fixed given time limit. Thereby a quadratic motion model for the players is used. Those polygons are used on one hand to estimate the dominant regions for all the player on the field and on the other hand to determine a good position for a pass. To achieve a good passing position the robot essentially tries to leave the dominant polygons of the opponents which may interfere. Another example is presented in [11](SSL) by Nakanishi, Bruce, Murakami, Naruse and Veloso where the dominant regions are used to plan pass combination. E.g., if a pass would lead through an opponent region, a third player is moved in between to close the gap and the pass is performed indirectly (1-2-3 shoot). The border between the dominant regions of a pair of robots is calculated analytically assuming a simple quadratic motion model. Calculating those borders pairwise for all the players on the field leads to a full DR-diagram.

Colin McMillen and Manuela Veloso present in [10] (4LL) a different approach based on plans. A *Play* is a plan which assigns roles for each of the players. A role consists of a predefined *responsibility region* and a behavior strategy for each of the three cases: the ball is outside or inside of the region, or the ball is not seen. A particular play is applied when the game situation satisfies some predefined requirements. When several plays are applicable, the one with the higher weight is selected. The corresponding requirements are based on the game state like

remaining game time, count of players and the actual score. Another scenario based approach is *Scenario-Based Team working* (SBT) [16] (S2D) by Ali Ajdari Rad, Navid Qaragozlou and Maryam Zaheri. A scenario contains a sequence of sub-plans consisting of actions for each robot. Furthermore, it is equipped with a *goal*, e.g., shooting a goal or conquer the ball, triggering conditions, expected costs etc.. During the execution each agent tries to satisfy its iterative sub-plan, which defines the agent's actions. To position the players the field is subdivided in regions. The scenarios are structured in a directed graph, where the nodes are particular scenarios and edges are weighted with a probability for the next scenario to be executed. This graph is created before the game by connecting the scenarios depending on their goals and triggers. The weights of the edges are adjusted during the game depending on the success of a scenario.

The *Situation Based Strategic Positioning* (SBSP) described by Luís Paulo Reis, Nuno Lau and Eugénio Costa Oliveira adapts concepts of human team strategies implemented in the simulated domain by *FC Portugal* [6] and partly used by the team *CAMBADA* within the Middle Size League (MSL) [9]. The team strategy in SBSP consists mainly of a set of tactics, tactic activation rules and roles. A tactic itself consists of predefined plans and team formations. The formations describe the positioning of a player inside a formation. The positioning consists of a reference position, a predefined fixed region, as well as behavior for the cases when the ball is inside or outside of this region. To position the robot the reference position is adjusted with respect to the ball.

3 Voronoi Based Situation Map

Based on the analysis of the related work, the most common approach to define a team strategy is defined by three layers: formations assigning each player a role; roles defining the robots positioning on the field and its behavior; and at last the positioning method which determines the actual movements of the robot. In most cases, the positioning itself (the lowest layer) is implemented directly as a kind of a geometric relation to dynamic and static objects on the field, e.g., relative to the ball, a free position based on DR etc; or is formulated as forces applied to alter the position, e.g., potential fields, alignment towards the Voronoi centroids. Few approaches discretize the field in cells and try to determine the position in a more elaborated way, e.g., path planning, Pareto-optimum. In general, a simple positioning layer leads to more complexity in the higher layers to generate sophisticated behavior. The following approach strives to provide a generalized and flexible tool for the formulation of the robot positioning and simplify the upper layers defining the overall strategy.

To make our idea more clear we introduce a simple example. Imagine a situation with two robots and the ball placed in the opponent part of the field like depicted in the Fig. 1. We define the one closer to the ball to be the *striker* and the other one the *supporter*. Note, the robots will never change their roles in our examples as we are focusing only on the positioning problem itself. In this example neither the striker nor the ball moves. The only acting part is the

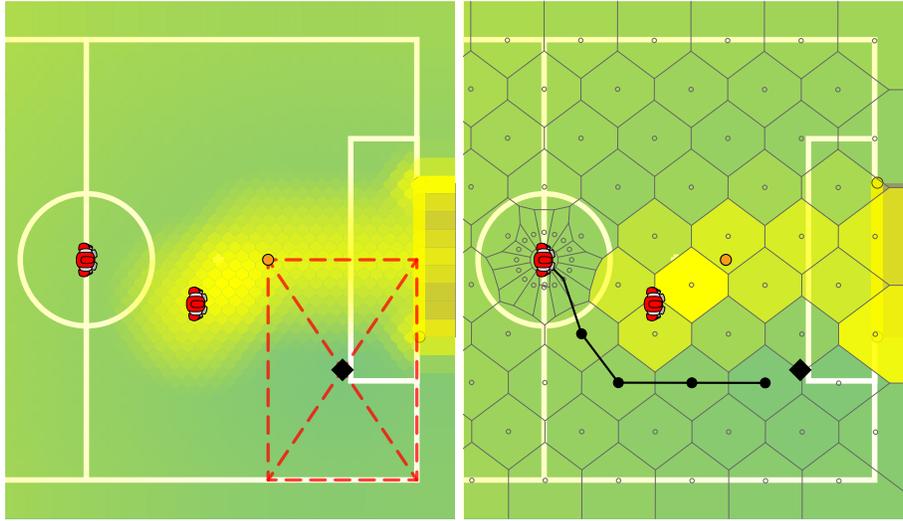


Fig. 1. An example situation: (left) initial positions of the supporter (center) and the attacker (closer to the ball); the center (black diamond) of the red dashed rectangle illustrates the target position for the supporter; the scalar field encoding the strategy is depicted by the intensity of the yellow glow (the global minimum is at the diamond); (right) the Voronoi tessellation with the weights of the regions depicted by the intensity of the yellow color; path calculated by the A*.

supporter. We consider the situation from the point of view of the supporter. Its only task is to move to a position where it could receive a pass or take over the ball in case the striker loses it. For that we assume a simple heuristic strategy for the supporter which is inspired by Carlos E. Agüero et al. in [1]: thereby the supporter's desired position p_0 is defined as the center of the rectangle spanned by the ball and the opposite opponent corner of the field as depicted in the Fig. 1. Now the task is to formulate the problem of the supporter getting to the point p_0 while avoiding collisions with other objects like the striker or the goal posts. Of course one could immediately imagine a simple positioning solution for this case. But the aim of the presented approach is to provide a general solution for a wide range of positioning problems. This example is just fine to illustrate the concept and to explore its basic principles.

3.1 General Idea

The general idea is to separate the field in weighted regions, which are then used to determine the target region as well as the path to this region. The conditions defining the desired position and the path can be formulated in terms of the separation in regions and the choice of weights. With this approach our example strategy could be formulated in a way where the weight of the region

containing the desired position p_0 is minimal and the regions containing obstacles are assigned high weights causing the path finder to avoid them.

There are numerous possibilities to separate the field in regions. At this point we decided to use *Voronoi tessellation*, which is a very powerful and flexible tool. The tessellation is defined by a set of points, called *Voronoi sites*, distributed over the field. Based on those points we use the *Fortune's algorithm* [5] to calculate the tessellation. Apart from a set of regions we also get a graph, called *Delaunay graph*, which is defined by the cells as nodes and the neighborhood as edges. This graph gives us a possibility for efficient search within the tessellation. With this we can easily construct very complex tessellations based on the conditions given by our strategy.

The whole *situation map* is defined by a Voronoi tessellation and positive weights assigned to each cell. Thus, the map consist of the spatial separation of the field in regions and a graph structure over the defining nodes. Basically, we can consider this map as a *weighted undirected graph* where the weights of the nodes are given directly by the definition and the weights for the edges are determined as a combination of the metric distance between the defining points and the weights of the nodes. From another point of view it can be seen as a discretized scalar field. To solve the positioning task we employ the A* algorithm to find the shortest path. Thereby the *start node* is the region containing the position of the robot and the target node defined by the minimal weight.

More precisely, the *Voronoi Based Situation Map* (VBSM) is defined by (G, W) where $G := (V, E)$ is the Delaunay graph of the Voronoi sites $V \subset \mathbb{R}^2$. The function $W : V \rightarrow \mathbb{R}$ assigns a weight $w \in \mathbb{R}_+$ to each node of the graph G . Note, each node represents a Voronoi cell and therefore the weights are assigned to each of the Voronoi cells.

3.2 Tessellation of the Field

To achieve a desired tessellation we have to choose the points (the Voronoi sites) appropriately. The resolution of the tessellation, i.e., number of points, has a major impact on the computational cost of the tessellation, the weights as well as the search. To resolve the trade-off between the accuracy and the speed, the field is discretized in two steps. At first points are chosen in a way that their Voronoi cells result in hexagons. These points are used to get a rough base resolution. In the second step the tessellation is refined around the position of the player, e.g., supporter. For that the position itself is added as a Voronoi site as well as 16 Voronoi sites around it, which are equidistant distributed on a circle. As the result, the scalar field will have a higher resolution around the player's position. Thus, the determined path will be more precise in the proximity of the player. Note that the geometry of the tessellation changes over time depending on the position of the player. The path calculated in one frame gives only a rough direction for the movement. The resulting path which emerges through the robot following the given directions will be much smoother as the higher resolution around the robot moves with it. The Fig. 1 (right) illustrates the resulting tessellation.

3.3 Positioning Strategy

As already described in the introductory example, the target position for the supporter is determined as the center of the rectangle, which is defined by the ball's position and the outer opponent corner (from ball's point of view). If the ball is close to the longitudinal axis of symmetry the determined position of the supporter might change the field sides due to noisy ball perception. To avoid this problem a hysteresis is used. We utilize scalar fields to formulate this strategy and to express it in terms of weights of the VBSM. Thereby, the target position is modeled as global minima of a scalar field. The striker, goal posts as well as the line between ball and opponent goal should be avoided and therefore are modeled as maxima of the scalar field. For each of the objects we introduce an id $\mathcal{I} := \{target, striker, line, goalpost, \dots\}$. We assume there is a distance function $d_\iota : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ defined for each of the objects $\iota \in \mathcal{I}$. The distance function d_ι assigns to each point $x \in \mathbb{R}^2$ the distance between x and the object ι . Except for the target position, the objects should have a limited range of influence. To formulate this we define the function $Q : \mathbb{R}_+ \rightarrow [0, 1]$ by

$$Q_{\alpha,\beta}(t) := \begin{cases} e^{\frac{\alpha}{\beta} - \frac{\alpha}{\beta-t}}, & \text{if } \beta > t \\ 0 & , \text{ else} \end{cases} . \quad (1)$$

The function $Q_{\alpha,\beta}$ has a compact support which is bounded by the parameter β and has its maximum equal 1 in $t = 0$. The parameter β describes the radius of the influence and α describes the steepness of the slope of $Q_{\alpha,\beta}$. With this function we now can define scalar fields U_ι for each of the objects in \mathcal{I} :

$$U_{target}(p) := \frac{1}{l} \cdot d_{target}(p) \quad (2)$$

$$U_\iota(p) := Q_{\alpha_\iota, \beta_\iota}(d_\iota(p)) \quad (3)$$

where l is the length of the field diagonal. To model the striker and the line between the ball and the opponent goal we have used in our experiments the values $\alpha_{striker} := \alpha_{line} := 800$, $\beta_{line} := \beta_{striker} := 1000$ and for the goalposts we have used $\alpha_{goalpost} := 800$, $\beta_{goalpost} := 500$. For each Voronoi cell we define the weight as a sum of the scalar fields at the Voronoi site p defining the cell:

$$W(p) := \sum_{\iota \in \mathcal{I}} U_\iota(p) \quad (4)$$

3.4 Path Finding with A*

The graph structure of VBSM makes an efficient application of A* search possible. To define the cost function and the heuristic for the search, the weights of the nodes can be seen as height information. Figuratively, they shape a kind of mountains over the field, where the robot tries to get to the lowest point. With this idea the cost function c can be defined as the Euclidean distance in three dimensions:

$$c(p, q) := \left\| \begin{pmatrix} p_x \\ p_y \\ \alpha \cdot W(p) \end{pmatrix} - \begin{pmatrix} q_x \\ q_y \\ \alpha \cdot W(q) \end{pmatrix} \right\|_2 \quad (5)$$

where p, q are two nodes of the Delaunay graph and the weight W defined in the equation 4. The heuristic for a node p can be defined as the direct cost to the target $h(p) := c(p, p_0)$. The factor $\alpha \in \mathbb{R}^+$ is used to scale the influence of the weight on the cost function and the heuristic. The heuristic function defined this way is consistent which makes the A* search optimal.

4 Experimental Analysis

In this section we investigate some of the basic properties of the presented approach in isolated experimental setups. The experiments are performed in the simulator SimSpark. To analyze the properties of the VBSM we utilize the knowledge of the precise positions of the objects within the simulation. Thus, we can assume there is no sensory noise. In particular this allows to observe the actual path or the robot movement. The trajectory of the center of mass is projected onto the playing field as walked path.

The following experiments illustrate the VBSM in an example scenario of supporter positioning. It means, the situation is visualized from its point of view. In particular we consider two different situations. At first we consider the situation described in our introductory example from the section 3. To briefly recall: the ball is placed in the opponent half; a robot (*striker*) is placed next to the ball, while the *supporter* is placed in the center of the field; the only active party is the supporter: its task is to get to its strategic target position as depicted in Fig. 2. In the second situation the striker and the ball are located in the upper opponent field part as shown in Fig. 2 (right). Here, the supporter has to change the side to reach its position. In both experiments the influence of the cell weights on the search costs is varied by changing the parameter α in the cost- and heuristic function 5.

Due to the rough overall resolution of the tessellation, the *estimated path* calculated in one frame approximates the ideal path only roughly. However, as already mentioned, the geometry of the tessellation changes over time depending on the position of the player, which leads to a smooth *final path* which actually emerges through the robot following the estimated path as illustrated in the Fig. 2. The defining conditions are reflected much better as well.

The influence of the parameter α on the final path can clearly be seen in both situations illustrated in the Fig. 2. Figuratively spoken, the height of the mountains defined by the cell weights is scaled by α , which changes the relation between the cost of the actual distance and the costs produced by the weights. Thus, with a small α the way around an obstacle seems to be longer than the way through it and vice versa in the case of a large α . In the Fig. 2 (left) the path is closer to the obstacle for a smaller and farther for a larger α . In the Fig. 2 (right) a too small α causes the robot to ignore the virtual obstacle defined by the line between the ball and the goal. While the path is slightly adjusted in the first scenario, it is changed qualitatively in the second.

The estimated path can also be seen as a *plan* which roughly reflects the situation on the field. The resolution of this plan is defined by the resolution of

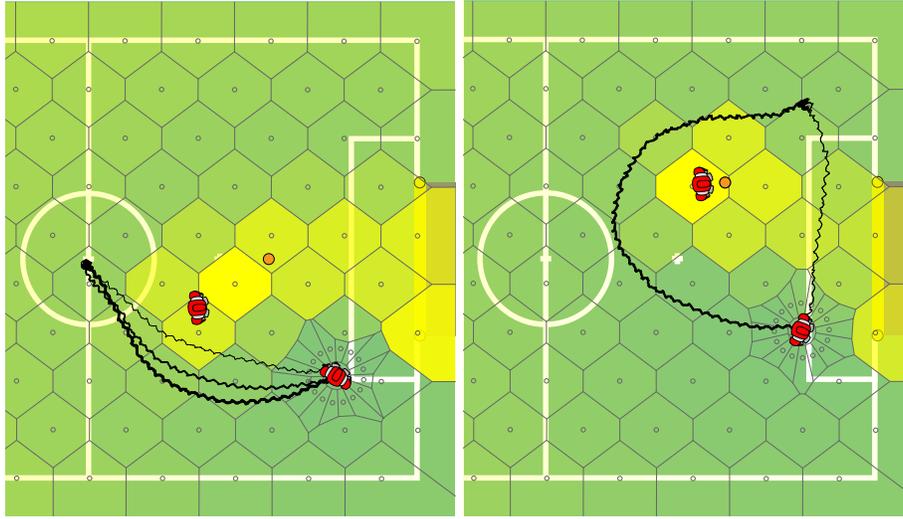


Fig. 2. Walk path of the robot in two different scenarios with different influence α of the cell weights in the search cost: from thin to thick the paths correspond to $\alpha = 600, 1000, 3000$; (left) the supporter walks to its position; (right) the supporter changes the side with parameters $\alpha = 600$ and $\alpha = 3100$;

the VBSM. Thus, the plan is refined as the robot moves, which corresponds to the *least commitment* principle. However, the resolution has to be high enough to reflect crucial qualitative aspects of the situation. For instance, in the case of too rough resolution some maxima might disappear between the cells in analogy to the *Nyquist-Shannon sampling theorem*.

5 Conclusion and Future Work

We presented a new method for strategic positioning in RoboCup based on a VBSM. Thereby the field is subdivided in regions by a Voronoi tessellation and each region is assigned a weight. The region with a minimal weight is chosen as the target region. The path to the target is estimated with A* on the Delaunay graph which is dual to the tessellation. Whereby, the distance between the nodes as well as the assigned weights are represented by a cost function and heuristics. A positioning strategy can be expressed in VBSM by the choice of the tessellation, i.e., the Voronoi points, and the weights of the regions.

An example implementation was illustrated in a scenario of supporter positioning. Thereby, scalar fields have been used to calculate the weights. The tessellation consists of two parts: a rough static tessellation of the whole field and higher resolution around the robots position. This implementation has been tested in simulation. It has been shown to produce a smooth resulting path in static situations despite a rough discretization, which is mainly due to the dynamic tessellation refinement for the direct vicinity of the robot.

This approach utilizes some well studied techniques and yields a flexible and powerful method for a description of positioning strategies. From our experiments the VBSM reveals a lot of capabilities and seems to be a promising approach for further development.

So far only static scenarios have been considered. The main focus of the ongoing research, is on investigating how more complex strategies can be formulated with VBSM as well as its behavior in dynamic situations. Further studies also need to investigate the choice of the tessellation as well as the way to assign the weights. In particular, the refinement of the tessellation seems to be a crucial aspect. Local refinement depending on the weights or along the estimated path could lead to better representation of conditions encoded in the weights. Also relaxing to a centroidal Voronoi tessellation could lead to a higher expressive power of the weights as the defining points would mark the center of a region representing it better. In the current implementation the tessellation is recalculated in every step, adaptive algorithms could reduce the computational costs by adjusting the old tessellation rather than calculating a new one. Similar to the tessellation, the weights could be propagated between the frames. This would provide a possibility for modeling some time dependent properties directly in the VBSM. For instance, the dominant regions could be estimated by propagation of the weights representing an opponent between the cells based on its motion model. Another idea is to lower the weights along the estimated path, which would result in a kind of memory for the path and prevent oscillations.

References

1. Agüero, C.E., Matellán, V., Casas, J.M., Gómez, V.M., Carlos, J.: Switch! dynamic roles exchange among cooperative robots. In: in Proceedings of the 2nd International Workshop on Multi-Agent Robotic Systems - MARS 2006. INSTICC. pp. 99–105. Press (2006)
2. Akiyama, H., Noda, I.: Multi-agent positioning mechanism in the dynamic environment. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007: Robot Soccer World Cup XI, Lecture Notes in Computer Science, vol. 5001, pp. 377–384. Springer Berlin / Heidelberg (2008)
3. Dashti, H.T., Kamali, S., Aghaeepour, N.: Positioning in robots soccer. In: Lima, P. (ed.) Robotic Soccer, pp. 29–44. I-Tech Education and Publishing (2007)
4. Dashti, H., Aghaeepour, N., Asadi, S., Bastani, M., Delafkar, Z., Disfani, F., Ghaderi, S., Kamali, S., Pashami, S., Siahpirani, A.: Dynamic positioning based on voronoi cells (dpvc). In: Bredendfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005: Robot Soccer World Cup IX, Lecture Notes in Computer Science, vol. 4020, pp. 219–229. Springer Berlin / Heidelberg (2006)
5. Fortune, S.: A sweepline algorithm for voronoi diagrams. *Algorithmica* 2, 153–174 (1987)
6. Hannebauer, M., Wendler, J., Pagello, E., Reis, L., Lau, N., Oliveira, E.: Situation based strategic positioning for coordinating a team of homogeneous agents. In: Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Lecture Notes in Computer Science, vol. 2103, pp. 175–197. Springer Berlin / Heidelberg (2001)

7. Kyrylov, V., Hou, E.: Pareto-optimal collaborative defensive player positioning in simulated soccer. In: Baltes, J., Lagoudakis, M., Naruse, T., Ghidary, S. (eds.) *RoboCup 2009: Robot Soccer World Cup XIII*, Lecture Notes in Computer Science, vol. 5949, pp. 179–191. Springer Berlin / Heidelberg (2010)
8. Kyrylov, V., Razykov, S.: Pareto-optimal offensive player positioning in simulated soccer. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) *RoboCup 2007: Robot Soccer World Cup XI*, Lecture Notes in Computer Science, vol. 5001, pp. 228–237. Springer Berlin / Heidelberg (2008)
9. Lau, N., Seabra Lopes, L., Filipe, N., Corrente, G.: Roles, positionings and set plays to coordinate a robocup msl team. In: Lopes, L., Lau, N., Mariano, P., Rocha, L. (eds.) *Progress in Artificial Intelligence*, Lecture Notes in Computer Science, vol. 5816, pp. 323–337. Springer Berlin / Heidelberg (2009)
10. McMillen, C., Veloso, M.: Distributed, play-based coordination for robot teams in dynamic environments. In: Lakemeyer, G., Sklar, E., Sorrenti, D., Takahashi, T. (eds.) *RoboCup 2006: Robot Soccer World Cup X*, Lecture Notes in Computer Science, vol. 4434, pp. 483–490. Springer Berlin / Heidelberg (2007)
11. Nakanishi, R., Bruce, J., Murakami, K., Naruse, T., Veloso, M.: Cooperative 3-robot passing and shooting in the robocup small size league. In: Lakemeyer, G., Sklar, E., Sorrenti, D., Takahashi, T. (eds.) *RoboCup 2006: Robot Soccer World Cup X*, Lecture Notes in Computer Science, vol. 4434, pp. 418–425. Springer Berlin / Heidelberg (2007)
12. Nakanishi, R., Murakami, K., Naruse, T.: Dynamic positioning method based on dominant region diagram to realize successful cooperative play. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) *RoboCup 2007: Robot Soccer World Cup XI*, Lecture Notes in Computer Science, vol. 5001, pp. 488–495. Springer Berlin / Heidelberg (2008)
13. Nieuwenhuisen, M., Steffens, R., Behnke, S.: Local multiresolution path planning in soccer games based on projected intentions. In: Röfer, T., Mayer, N., Savage, J., Saranlı, U. (eds.) *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, vol. 7416, pp. 495–506. Springer Berlin Heidelberg (2012)
14. Petersen, K., Stoll, G., von Stryk, O.: A supporter behavior for soccer playing humanoid robots. In: Ruiz-del Solar, J., Chown, E., Plöger, P. (eds.) *RoboCup 2010: Robot Soccer World Cup XIV*, Lecture Notes in Computer Science, vol. 6556, pp. 386–396. Springer Berlin / Heidelberg (2011)
15. Phillips, M., Veloso, M.: Robust supporting role in coordinated two-robot soccer attack. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) *RoboCup 2008: Robot Soccer World Cup XII*, Lecture Notes in Computer Science, vol. 5399, pp. 235–246. Springer Berlin / Heidelberg (2009)
16. Rad, A., Qaragozlu, N., Zaheri, M.: Scenario-based teamworking, how to learn, create, and teach complex plans? In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) *RoboCup 2003: Robot Soccer World Cup VII*, Lecture Notes in Computer Science, vol. 3020, pp. 137–144. Springer Berlin / Heidelberg (2004)
17. Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Humann, A., Honsel, D., Kastner, P., Kastner, T., Könemann, C., Markowsky, B., Riemann, O.J.L., Wenk, F.: B-human team report and code release 2011. Tech. rep. (2011)
18. Work, H., Chown, E., Hermans, T., Butterfield, J., McGranaghan, M.: Player positioning in the four-legged league. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) *RoboCup 2008: Robot Soccer World Cup XII*, Lecture Notes in Computer Science, vol. 5399, pp. 391–402. Springer Berlin / Heidelberg (2009)