Decision-Making and Tactical Behavior With Potential Fields

Jens Meyer, Robert Adolph, Daniel Stephan,
Andreas Daniel, Matthias Seekamp, Volker Weinert, and Ubbo Visser

Department of Mathematics and Computer Science
University of Bremen
Postfach 33 04 40
D-28334 Bremen, Germany
jens@informatik.uni-bremen.de

Abstract. Using potential fields is a technique seldom used in RoboCup scenarios. The existing approaches mainly concentrate on world state representations of single actions such as a kick. In this paper we will show how to apply potential fields to assist fast and precise decisions in an easy and intuitive way. We go beyond the existing approaches by using potential fields to determine all possible player actions, basic and advanced tactics, and also general player behaviors. To ensure fast computing, we mainly use basic mathematical computations for potential field-related calculations. This gives us the advantage of both determining and understanding player actions. Therefore, the integration of future features such as a complex online coach and progressive localization methods will be easier. We implemented the approach in our team **B**remen **U**niversity **G**oal **S**eekers (BUGS) and tested it in numerous games against other simulation league teams. The results show that the CPU-time for decision-making has been decreased significantly. This is a crucial improvement for calculations in time-critical environments.

1 Introduction

The idea to use potential fields is based on retrieving knowledge for the best possible place for an agent to act on. These actions are kick, dribble, and dash, consequently it can easily be adapted to all RoboCup leagues. We are able to represent all possible game situations by taking all necessary information from the already existing world model of CMU-99 and interpreting them as objects in the potential fields. The decision for an action is made by a heuristic based on the determination of the distance to this point. A large distance implies kicking the ball to the point while dribbling would be the action when having a short distance. If we don't have the ball we dash to the target.

There have previously been approaches with regard to potential fields. Similar to electric fields by [Johannson and Saffiotti, 2001] and similar to approaches as described in [Latombe, 1991] we use potential fields to represent world model states. In comparison to the mentioned approaches we focus on the fastest decision-making and general usability possible . This means that we use potential fields to derive any decision that has to be made by an agent. [Nagasaka et al., 2000] use potential fields for actions like a single kick. Our general usability approach goes further. [Johansson, 2001] combines decision-making and navigation in using potential fields. Our approach is similar, however, the difference is the environment: it is real-time, dynamic, and more flexible. Therefore, the processes are more difficult.

2 Using Potential Fields In BUGS

For better understanding of the complex associations discussed later in this paper, we have a closer look towards potential fields and show their flexibility and hidden complexity.

2.1 Basic Use Of Potential Fields

For building a potential field it is necessary to lay a grid upon the soccer field. The grid resolution, although it is customizable, used in the BUGS-client is 60*40, which means $\approx 2m^2$ per grid field. Based on information about all visible moving objects, the game situation and extra knowledge about our own tactic and formation, numeric entries (only integer) in all grid fields are made. The relations between the different aspects are controlled by 15 changeable parameters (which are meant to be online manipulated by the coach, depending on various game statistics).

The point about the speed of our algorithm results from various simplifications in calculations and design of potential fields. One reason is that we don't have functions that will interpolate the resulting potential fields. These interpolations are unnecessary because of the predefined areas of effect of each world object (this operates like *stamps* with integer values). Another reason is using a grid instead of the soccer server coordinates.

Every agent, including the coach, calculates every few cycles (2-8) a potential field based on his own world model. Timing depends on game situation and distribution of CPU-power. Although we have enough CPU power, despite of running all clients on one computer, we tend to keep it well balanced to absolutely guarantee complete decisions for all agents. One starting point only allows the next potential field calculation every other turn, starting with half the agents on an even and the other half on an odd cycle. Situation-based timing is obvious: a ball-leading agent should do calculations every other cycle; a position-holding or adjusting agent, with the ball 60m away, will do so again in about 20 cycles or earlier if the ball comes closer to him.

To decide which action is next, the complete field and some more information (e.g. ball possession and position, own position) are necessary. The best value within the grid always means the best position for the next action. Again, these actions are dashing, kicking, and dribbling. Using only these simple player-actions, the whole space of soccer behavior can be emulated. How far this goes and how it exceeds the obvious will be discussed next.

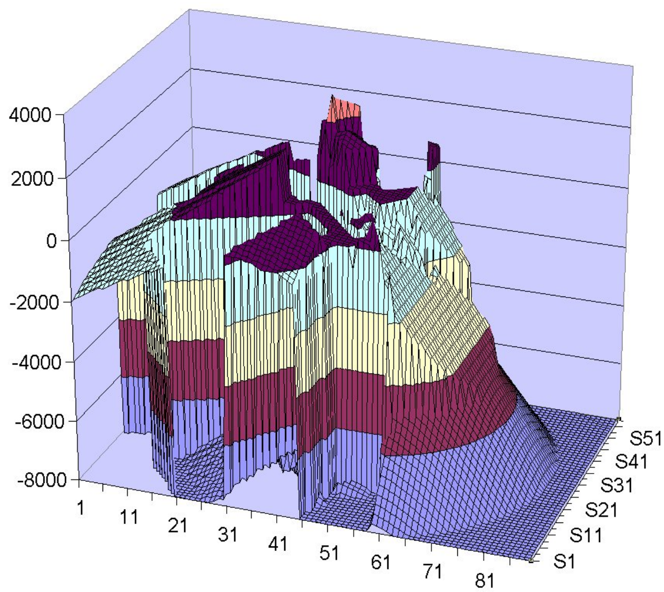

Fig. 1. A typical potential field

Figure 1 shows a typical potential field generated by the ball-leading agent, located on position 32,54. The numerous influences from the other objects on the potential field can be seen (e.g. the cones of the opponents representing the not passable area and also basic potentials indicating the main behaviour). The target point is located on 45,34. Due to the distance between the target and the agents position, the target action is *pass ball*.

2.2 Advanced Use Of Potential Fields

In order to understand the complexity level and the possibilities of potential fields, it is necessary to know their gradual structure. This is the point where a concrete view can be won on later possibilities and implicit conversions of advanced tactics. In fact, the BUGS potential field method includes some tendencies towards planning algorithms. Like a superior plan all clients have a similar basic potential which leads towards the opponent's goal. Each individual action which is decided contains the adherence to these basic guidelines, thus the rough superordinate plan. Whereas following a global intention is not similar to a planning algorithm, viewing all generated potential fields in parallel as one unit means a large step towards a global plan. We need to show the interaction between single potential fields. There are two reasons for the fields to interact with each other. The first reason is rather trivial. Each single field contains its player position such as offense, left mid-field, etc. We get tactical formations owing to tuning these positions and possibly adjusting them to recognized opponent positions (see section 2.4). The second reason seems to be trivial too, but has non-obvious consequences: every potential field is quite similar to the fields generated by neighbor agents, thus based on (nearly) the same inputs which generates similar results. These results are only altered by their own positions and the individual noise transmitted by the soccer server. All agents building potential fields at the same time, each with its own view of the same situation, permanently influence each other with their decisions. While one player holds the ball, the others take position to be passable. This behavior results in building a complete way for the ball into the opponent's goal for most of the time while in ball possession. Due to interceptions, however, most of the time this scenario will not work ; thus, alternatives are created at any time. Similar to planning algorithms we determine sets of action based on the current situation. This might be dangerous because our algorithm has not really a similarity with any planning algorithms from the implementation point of view but in some way the rudimentary behavior is the same, especially for the RoboCup simulation league where world model states and conditions for decision-making change quickly.

2.3 Example for advance use

As we described above, we can assign special values to areas in the grid to gain a special behaviour. The following example shows how it works and gives some views on other tactics, which we can evoke by assigning values to the grid.

Offside A very important tactic in soccer is the use of the offside rule against the other team. Many teams use this tactic to gain free kicks and to interrupt opponents offense easily. Many teams have problems either by setting an offside trap or by recognizing the opponents offside trap. With an potential field we can assign negative potentials to either the own offside area or the opponents offside area. If we assign these potentials to the own offside area we achieve an offside trap. Due to the negative potential in this area, no field player will move into this area on his own. The major exception to this rule is the ball interception after the ball enters this area. Similar happens on the opponents offside area. We assign negative values to this area and achieve that no agent stays in or moves into this area if he don't have the ball or if the ball is already in this area.

Further Examples The method described in the last section can be used on all possible tactical areas. To build up an offensive strategy on the field edges we can simply assign positive potentials in these particular regions. If we want an agent to stay in a specific area (e.g. its position in the team), we can assign negative values to areas outside its tactical area or assigning positive values to his tactical area.

We added some additional points of possibilities for assigning values to this section. This is just a small list, which should show the power of assigning values to the grid within the potential fields:

- The own penalty area is an area where the ball shouldn't stay too long. By assigning a negative value to this area we can achieve that the ball is kicked outside this area quickly if an agent has the ball. Because of the negative value in this area, his target point automatically is set outside this area.
- Assigning positive values to the opponents penalty area and goal. The attraction to this points is high enough to let the attacking agents move to and kick to this specific area.

A very important aspect to the value assigning is the online coach which we plan to use (discussed in the next section). With his clear view onto the game he can gain statistics about the game. So he can easily assign basic values to specific areas for all, some and even single agents. We developed a coach language where we can encode data for assigning values to the agents. The coach is able to get information from his statistics which tells him, what areas of the field is mostly used by the opponents. By assigning positive values to this areas, the agents will be able to intercept the ball or the opponents agents earlier.

2.4 Influence of the tactical online coach

We develop a tactical online coach whose purpose is the statistical evaluation of both our own and the opponent's team. In addition, it will log frequency points of position of all moving objects. Both will be used for game evaluation, which is necessary to re-distribute player-resources, change tactics, and re-arrange player formations. Statistic variables are

- ball losses,
- percentile ball possession,
- percentile ball position per team section (defense, mid-field, offense),
- number of wrongly passed balls,
- gaining of ground and some other variables.

These variables show the quality of each team section and in addition its relative efficiency. Based on these values we will modify various player settings, including player type, position, relations between objects in the potential field or tactics for a single agent, and additionally player formation for a team section or the whole team. All of these changes have an influence on the potential fields, changing tactics for example may tilt the whole field (as described above), formations will simply set new orientation points for the agents, which center the agent's preferred area of action. Special attention should be given to the changes of object relations in the potential field because this is the most subtle way to change behavior, although it could have the greatest effects. Here is an example: raising ball priority will probably do nothing because it is already very high, raising team mate priority slightly may result in passing the ball for a little more percentage rather than dribbling with it. A medium change in opponent priority can change the whole game. Raising it will give an evasive play, lowering may result in nearly ignoring (as long it is possible) while in ball position. Sometimes a change in relations has unpredictable consequences, which makes this way of influence as dangerous as it makes it powerful.

3 Evaluation and Results

The adjustment of the priorities for the evaluation algorithm as described above was probably the hardest work. For this we developed a tool which shows the calculated potential fields of all agents. Also, this enables us to identify errors in priority and to change the potential fields in a way that they fulfill the requirements of our original intentions. We are also able to locate errors in priorities and to bring the real potential fields towards our original intentions. Our agents were running on a Pentium II 400 Mhz Processor with 128 MB of RAM located at the computer pool of the Department of Mathematics and Computer Science at our University. The operating system on these machines is RedHat Linux 7.2. The following table 1 shows our performance test based on a tool called *gprof*. This GNU-tool produces an execution profile of C or C++ programs. All values in the table refer to a complete game. The first column describes percentage of the total running time of the program used by this function. The second column describes the number of seconds counted for this function alone. The third column describes the total number of calls. The last column contains the function's names. Both rows are the most evoked functions of our agent. The

Table 1. potential field generation based on time and evocation

% time	**self seconds**	**# calls**	**name**
20.62	0.60	288024	estimate_future_pos(...)
8.59	0.25	1526	getEvaluatedAction(...)

function named *estimate_future_pos(...)* is a CMU-function mostly used by the world model itself. The function in the second row is the function which is used to generate potential fields. The result shows that our complete potential field generation uses less than 9% of the time. Until now these figures are difficult to compare in RoboCup scenarios. A comparison of our evaluation algorithm with similar decision algorithms of other teams is difficult because we can't isolate their decision module. The only thing we can compare is the used CPU-time and the amount of memory. The used memory is of lesser interest because there is enough of it available in a tournament. In order to extract these results we simply used **top** (Unix-command) while playing a normal game. Both teams and the soccer server were each running on different computers (the type mentioned above). We repeated each game 15 times and took average values. Karlsruhe-Brainstormers and Mainz-Rolling-Brains ran with the old soccer server v. 7.x, our team and FC-Portugal on soccer server v. 8.x. The use of different soccer servers should not make any difference to the results. The BUGS-team appears twice in the table because of two different grid resolutions to show the relation between resolution and performance. We chose FC-Portugal because it is also based on the CMU-99 sources. Karlsruhe Brainstormers01 was chosen because of its good performance in Seattle, and Mainz-Rolling-Brains completes the list of reference teams. Results are given in the following table 2: We can see that

Table 2. Best performance test based on a time evaluation relation for the algorithm

Team	**Max CPU**	**Min CPU**	**Min Memory**	**Max Memory**
FC Portugal 00	12.0%	0.3%	0.5%	1.0
BUGS(90*60)	7.6%	$< 0.1\%$	0.7%	0.8%
BUGS(60*40)	4.6%	$< 0.1\%$	0.6%	0.7%
K. Brainstormers 00	9.8%	0.1%	2.0%	2.1%
Mainz Rolling Brains 00	5.1%	1.5%	1.1%	1.1%

our team BUGS has the best performance with regard to the maximum CPU time used with a grid-resolution of 60*40. It uses only between 40 - 64% of the time that FC Portugal needs and is twice as fast as the Brainstormers, again, with a grid-resolution of 60*40. Similar relations can be seen in the column 'minimum used CPU' where the BUGS team uses less than 0.1%. Here, the team from Mainz has the highest values with 1.5%. As far as memory is concerned, we can note that the Brainstormers always use the same amount of memory. This is probably due to the fact that they are completely based on artificial neural networks. The same relation between maximum and minimum memory used also holds for the BUGS team.It remains constant at a low rate. Only the team from FC Portugal shows a difference in the memory. This indicates that they use various techniques for decision-making. Although we used more than twice the original field size, we were still performing well .

4 Conclusion

We proposed a new idea using potential fields to represent all game situations. In addition to similar approaches such as [Nagasaka et al., 2000] we employ potential fields for *all* possible actions, not only for a kick. They are also used to decide which action to take and to judge the current situation. This method is both intuitive and fast. The main advantage is that we are able to use a single algorithm to determine the agent's action ("One algorithm to fit them all"). Another advantage is the waiving of complex rules and algorithms.

Potential field can be employed to find a teammate to pass the ball as well as to find a position a teammate will pass to. Using an online coach in the near future would make the decision even better. With a coach we are able to give simple advises to the playing agents. Additionally, we can pass messages to single agents indicating specific positions, which makes the potential field even exacter. We use the potential field approach in our own team in the simulation league scenario. At present, we can't make a significant statement about the quality of this decision. However, we have shown that the decision we determined is done due to an easy and especially fast algorithm. Both the CPU-time and the memory used by an agent is very low.

Acknowledgement

We thank the Carnegie Mellon team for letting us use their basic client sources.

References

[Johannson and Saffiotti, 2001] Johannson, S. J. and Saffiotti, A. (2001). Using the electric field approach in the robocup domain. In Birk, A., Coradeschi, S., and Tadokoro, S., editors, *The RoboCup 2001 International Symposium*, Lecture Notes in Artificial Intelligence, Seattle, WA. Springer. in print.

[Johansson, 2001] Johansson, S. J. (2001). An electric field approach - a strategy for sony four-legged robot. M.sc.thesis mse-2001-03, Blekinge Institute of Technology, Sweden.

[Latombe, 1991] Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston, USA.

[Nagasaka et al., 2000] Nagasaka, Y., Murakami, K., Naruse, T., Takahashi, T., and Mori, Y. (2000). Potential field approach to short term action planning in robocup f180 league. In Stone, P., Balch, T., and Kraetschmar, G., editors, *RoboCup Symposium 2000, Robot Soccer World Cup IV*, volume 2019 of *Lecture Notes in Artificial Intelligence*, pages 345–350, Melbourne, Australia. Springer.