Using Time Encoded Terrain Maps for Cooperation Planning

Tim Edmonds[1*] Antony Rowstron[1] Andy Hopper[1,2]

[1] Laboratory for Communications Engineering
Department of Engineering
University of Cambridge
Trumpington Street
Cambridge, UK

[2] AT&T Research Laboratory Cambridge
24a Trumpington St
Cambridge, UK

Category: full paper

*Contact: tme23@eng.cam.ac.uk

Abstract

This paper describes a novel adaptation and use of Terrain Mapping and Optimal Path Planning to fuse large and diverse time varying data sets into a common structure used for path extraction.

The method described uses a system of map distortion and manipulation to encode anticipated future states of the environment into a single map. By doing so, it effectively encodes time into the map, allowing paths to be planned incorporating short term accuracy and an approximate long term path which accounts for anticipated movement of obstacles in the environment. The path is refined with recalculation as progress is made along the path.

An application of the technique in cooperation planning for multiple physical agents is presented in the context of the RoboCup Robot Football Competition.

1 Introduction

Optimal paths with respect to cost or safety are useful tools for navigation for either people or robots operating in a known environment. Terrain or Cost Value Mapping is a raster based technique in which environmental features such as obstacles and boundaries are presented as geographical features, usually a hill or a depression of some sort, resulting in a value or cost representational array. This leads to a terrain map of a landscape where high ground may be considered as bad and low land may be considered as good. Tracing a path through this terrain which attempts to remain at the lowest cumulative altitude will produce a path that avoids obstacles and favors preferable locations. In this manner of operation, navigation through this terrain is very similar to artificial potential field navigation[1].

Traditional mapping and path planning techniques focus on least cost navigation goals in a primarily static environment. However, path planning in a highly dynamic, adversarial environment, one where obstacles present are mobile and can be expected to actively hinder the planned path, introduces many new considerations[2]. In this case, a pre-calculated least cost path will soon be obsoleted as its execution progresses and obstacles move.

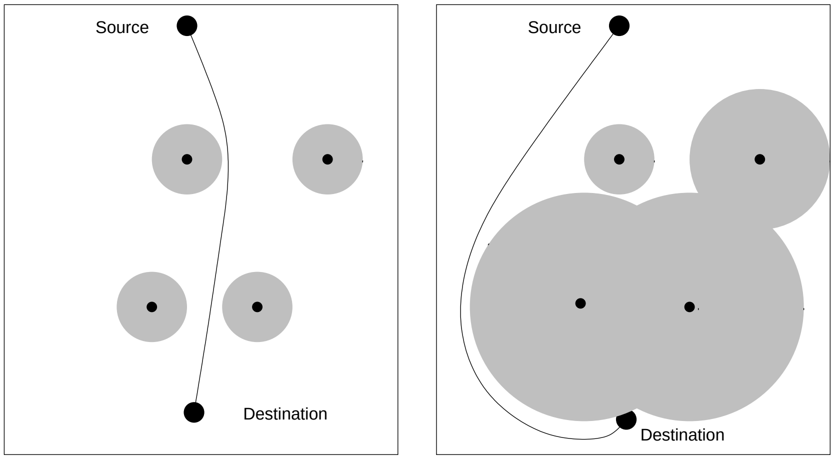
In this paper we present an extension of traditional Terrain Mapping and navigation techniques called Time Encoded Terrain Mapping to better handle situations which are highly changeable and adversarial in nature. Time Encoded Terrain Maps provide an environmental representation which is able to include time-varying environmental features and known environmental characteristics in a data structure suitable for use by a traditional path-finding algorithm such as Dijkstra's least-distance path finder[3] or the A* algorithm[4].

1.1 The Case for Time Encoded Terrain Maps

While traditional mapping and path planning techniques readily provide solutions for least-cost navigation in primarily static environments, the maps that they use consist of a single snapshot of the environment and hence can only represent the environment for a given point in time. These techniques are by themselves unsuitable for path planning in environments of a highly dynamic or adversarial nature. In such environments, it is difficult or impossible to predict far in advance as precise knowledge about the environment state is limited to the very short term.

Traditional techniques are able to provide accurate short term paths, but do not guarantee an accurate heading for the longer term. While initial progress may avoid obstacles properly, the progress made may lead it into a future situation where it must *back out* to avoid obstacles which have moved into its path. A pathfinding technique which is able to account for the known potential for change in the environment will be able to provide a path which best avoids possible future path obstruction by dynamic obstacles, ensuring the safest possible complete path to destination. While the unknown movements of obstacles means that such a path cannot be guaranteed to be the optimal path when viewed retrospectively, each iteration of the pathfinder should produce the optimal path with the consideration of the possible obstacle movement or obstacle cost change.

The aim then is to form a representation of the environment in such a way that a path may be extracted which is accurate for the short term, valid for the long term and may be recalculated or refined for the longer term as the path is executed. Because the path is expected to be recalculated or refined frequently, the map building and path planning algorithms should also be efficient and/or tunable to trade off path cost with evaluation or search speed.



(a) A situation using traditional Terrain Mapping.

(b) The situation from (a) but using Time Encoded Terrain Maps.

Figure 1: An example situation where navigation using traditional Terrain Mapping may produce erroneous paths by not accounting for obstacle dynamics while using Time Encoded Terrain Maps anticipates obstacle dynamism and routes around potential impediments.

2 Time Encoded Terrain Maps

Maps serve to represent the environment in such a manner that information, especially navigational and path planning information, can be easily extracted. Terrain Mapping forms a map of a virtual landscape which may map spatially to the environment but with altering of the elevation of the landscape to represent the presence of obstacles or other environmental features. Time Encoded Terrain Maps extends the Terrain Map representation by encoding time and distance variations from the current point of interest. Encoding time into the mapping and pathfinding process provides a simple forward projection of the environment state, allowing fine short term plans and coarse long term plans, reflecting the dynamic nature of the environment. The time encoding is achieved by distorting the map during formation to reflect the influence of each obstacle relative to the current point of interest (the vehicle or robot). In this way, this provides a method to modify the underlying map to encode time variation across and future states of the environment into a single data structure from which the optimal path may be extracted.

Figure 1 illustrates the difficulties of using static environment mapping techniques when obstacles are highly dynamic. In Figure 1(a), traditional mapping and navigation techniques are used to find a path which skirts the current location of obstacles. Although this would be a correct path if the obstacles remained stationary, dynamic adversarial obstacles would soon block the path before it can be followed to completion. Figure 1(b) on the other hand shows the map with the influence of obstacles distorted according to their time-distance from the point of interest, reflecting the potential of those obstacles to move and obstruct the path. In this case, the pathfinder will route around the foreseen potential movement of the far obstacles to guarantee a free path to the destination. This new path is clearly only optimal given the uncertainties of future motion and is not optimal for any arbitrary eventual outcome. Further discussion in Section 2.2 shows how recalculation during path execution can refine and improve the path.

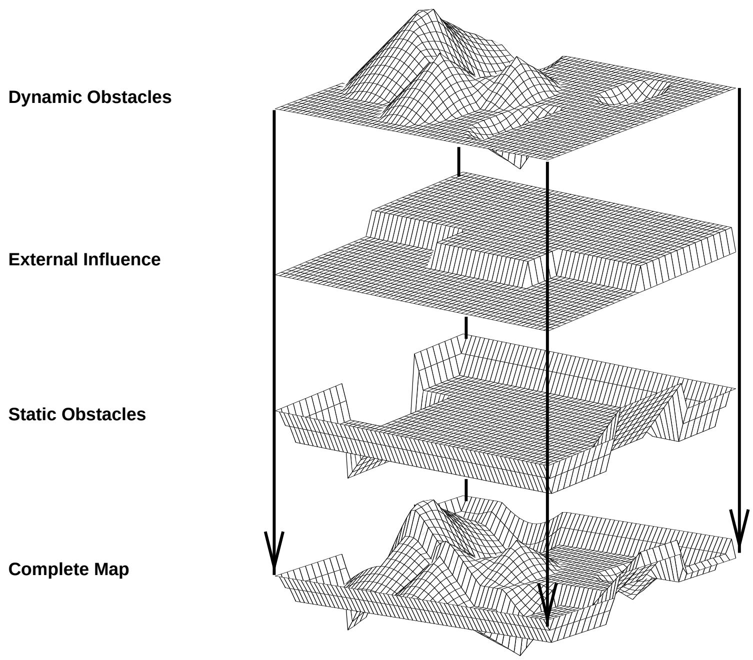


Figure 2: The Terrain Map is formed by the successive superposition of several layers in increasing order of rate of change.

2.1 Map Formation

The basis of map formation for Time Encoded Terrain Maps is the same as that for traditional terrain mapping. The terrain map is built by successive superposition of layers of information (Figure 2), each representing a different level of dynamism.

A map of the static environment is used as a base - being static, it need only be calculated once. On top of this base map, further influencing factors which may be dynamic are added in order of their expected rates of change. Finally, the complete map can be formed by adding the influences of each dynamic obstacle in the environment. By layering the map in order of dynamic characteristics, rebuilding of the map can be limited to the recalculation of changed layers and then recombination with previous lower layers.

Figure 2 shows the layers used to build a map for strategic path planning in the RoboCup Decision Engine presented in Section 3. In this case, the map comprises three layers. First is the base map containing static elements both physical and intangible such as the field boundary walls and the goal area weightings. Secondly, the external influence layer adds the coaching bias as generated by the global strategy analyzer. Finally the layer representing the influences of the dynamic obstacles, the robots, is added.

2.1.1 Influence Shapes

The alterations made to the terrain map by an obstacle's presence is termed that obstacle's *influence*. The influence of an obstacle over a particular area may be defined as the likelihood that that area may be obstructed by the obstacle in a given time. Hence, an obstacle's influence over a point is a measure of the possibility of the obstacle being at that point in the relevant future. As this depends on the amount of time it is given to get there from its current known position, the influence of an obstacle derives from its time from the current point of interest.

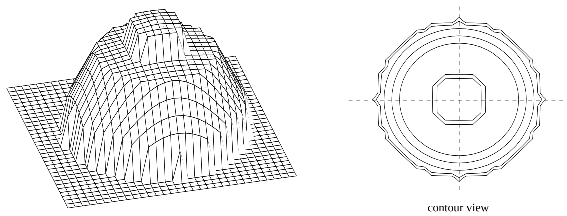


Figure 3: The influence of an omnidirectional or fast turning robot

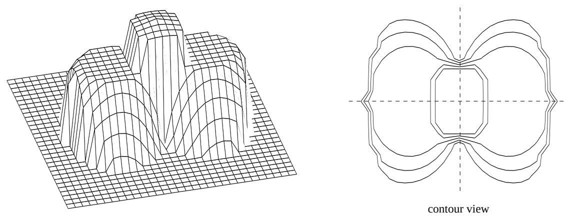


Figure 4: The influence of a slow turning robot

As the key concern is the path of the vehicle, the influence of an obstacle on an area should correctly be interpreted as the influence of the obstacle on the vehicle in that area. Thus, the time allowance that is given an obstacle in which its possible actions are to be considered is proportional to its distance from the vehicle and may be calculated as a conservative estimate of the time of travel of the vehicle from its current location to the obstacle. Hence, an obstacle further from the vehicle will have a far broader influence than an obstacle closer to the vehicle. Although initially this may seem counter-intuitive, consider the following example in the context of RoboCup when planning a path for the ball:

A robot only $10cm$ from a moving ball that is not heading directly towards it has almost zero influence on the ball as the ball will have traveled further away before it can react. However, a robot at the opposite end of the field from the ball will have a broad influence which indicates that by time the ball has traveled some distance in its general direction, it could possibly make its way to any of the points within its influence.

The shape of the obstacle's influence varies according to the orientation of the obstacle and its dynamic characteristics. For example, an omnidirectional obstacle would have a uniform influence in all directions, whereas an obstacle which moved quickly along its axis but was slow to turn would have an influence in a peanut shape representing its difficulty to translate perpendicular to its axis of movement. These examples are illustrated in Figures 3 and 4.

Similarly, more complex influences can be derived to include further information such as velocity or other particular characteristics. For example, to include velocity information, the shape may be translated or elongated in the direction of motion to represent the inertia of the obstacle.

Finally, the shape of an obstacle's influence is scaled and distorted to reflect its performance characteristics. This is the basis for time encoding in the map. Simplisticly, linear scaling of an obstacle's influence with time from the point of interest provides an approximation of the potential coverage of an obstacle.

In summary, to form the map from a set of environmental obstacles,for each obstacle:

1. determine the obstacle's influence pattern according to the obstacle's type and what is known about its performance characteristics (this will usually be a selection from some predefined influence patterns.)
2. rotate the influence to be aligned with the obstacle's current orientation. If the influence

pattern is rotationally symmetric (eg: for the omnidirectional robot) this may be a null operation.

3. calculate effective distance d_i of the obstacle from the vehicle.
4. calculate the possible radius of movement of the obstacle (r_i) using the expected velocities of the obstacle (v_i) and the vehicle (v_V) $r_i = \frac{d_i \times v_i}{v_v}$.
5. scale the influence pattern so that its radius matches that calculated above.
6. superimpose the influence on the map.

2.2 Time Encoding

The time encoding within the map effectively combines the potential future states of the environment into the map. The scope of the future states encoded for each obstacle (ie: the scale of an obstacle's influence) is determined by each obstacle's immediate relevance to the current position and velocity of the vehicle.

The resulting map features fine definition of obstacles near the vehicle, with definition deteriorating with distance (or travel time) from the vehicle. Map features near to the vehicle's current position will be finely detailed. Further from the vehicle, the distortion and scaling of the obstacles' influences will cause spreading and blurring of the terrain map to reflect the potential for change in the future. Planning a path through this distorted map results in a path capable of giving immediate directions to the vehicle which lead into longer term general directions for the rest of the path.

As the vehicle progresses along the path, the path can be recalculated to determine accurate paths for the new short term. Figure 5 extends the example of Figure 1 to illustrate a scenario where refinement during path execution is possible when some of the obstacles have moved or obstacles' influences are further constrained by the movement of the vehicle. In this case, the vehicle has traveled a distance along the original path in which time one of the obstacles has moved forwards to intercept the vehicle but remains within its initial predicted area of influence. Now that the vehicle is closer to the obstacle and it has moved, its influence is reduced, allowing a new path to destination to be planned which is shorter or faster than the remainder of the original. In this way, the refinement of the path during execution can improve the original path to allow for the actual movement of the obstacles while still allowing for potential future motion. The vehicle avoids the obstacles as planned but is able to take advantages of changes to the environment to improve the path to destination.

3 Application: A Decision Engine for Robot Football

This section describes a Decision Engine[5] implemented for the Cambridge University entry in the 1998 RoboCup Robot Football Competition[6] with a particular emphasis on the use of Time Encoded Terrain Maps to perform strategy generation and cooperation planning. The Cambridge University entry was in the small league for RoboCup'98[7].

The RoboCup small league allows for a global overhead vision system and external field-side control computers. The Cambridge system uses these field-side computers to perform all of the vision, strategy and control, communicating primitive move and turn commands via radio with the team of thin client robots.

3.1 A RoboCup Decision Engine Using Time Encoded Terrain Maps

The basic program structure consists of team behavior and coordination layered over robot representative pseudo-agents which break down the basic skills for the robot into simple commands.

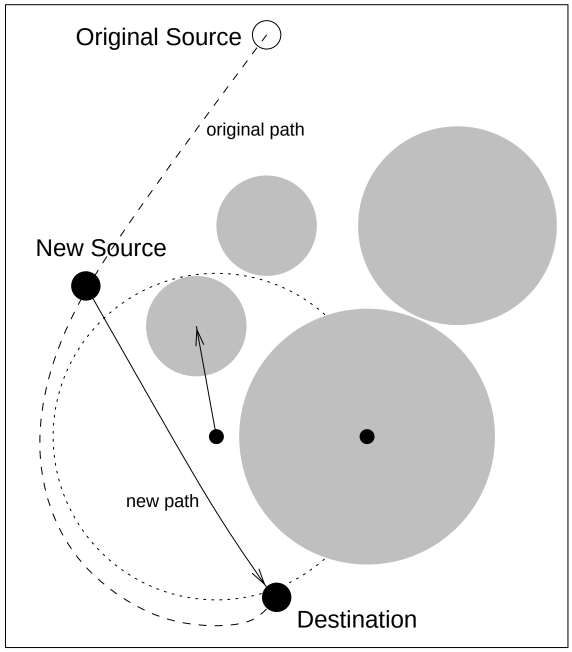


Figure 5: Recalculation of the path during execution allows for refinement to account for actual known motion of the obstacles

This layout is shown in Figure 6.

In normal play, the cooperation planner is responsible for the formation of general strategy for the team and then uses this information to assign roles and activities to each of the robots. The general strategy and plan of action for the team is formulated via the use of Time Encoded Terrain Mapping and is represented as a series of action stages. Different execution stages are then assigned to one or more robots, allowing team play to fulfill the plan.

Each role has a specific task which may be to kick the ball, receive the ball, block the ball, or to just return to home positions. The task of the Robot Representative Pseudo-Agent is to use its knowledge of the exact capabilities of the robot it represents to translate the action descriptions given into individual commands for the robot.

3.1.1 Strategy Through Terrain Mapping

Key to any such team controllers are strategy generation and cooperation planning. Strategy generation is the process of creating a plan of action to execute in order to best achieve the goals of the game. Cooperation planning is the process of organizing the use of the resources available, the robots, in the best manner to execute the strategy generated.

For the Cambridge University Decision Engine, the Cooperation Planner uses a Time Encoded Terrain Map approach to strategy and path planning. The use of Time Encoded Terrain Maps to perform these critical functions allows the full incorporation of any knowledge about robot performance characteristics and position and velocity data into the decision making process. This has a long term perspective advantage over methods used by other teams in the competition, many of which avoid these issues by discounting the future states of the field as *too dynamic* to model and restrict their plans to the short term[8].

Strategy generation using Time Encoded Terrain Maps is based on path planning for the

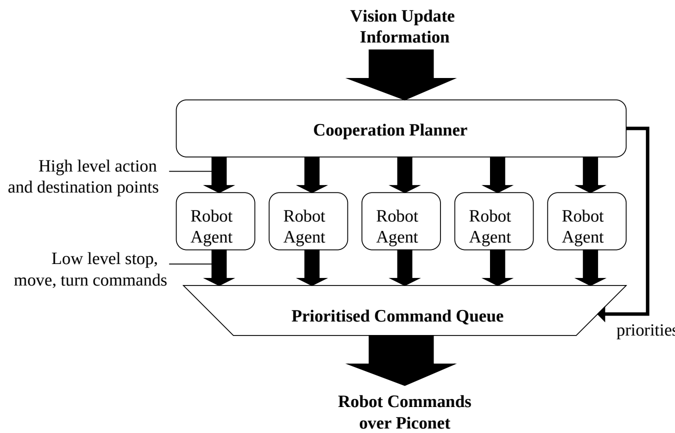


Figure 6: Decision Engine program structure

intended route of the ball. From the ball's current, or estimated future position, an optimal path is calculated to the provided destination point, usually the opposition's goal. With the appropriate translations from robots to influences, the properties of this path will be such that the path will avoid opposition robots and bad areas and favor locations near our own robots or open areas while gravitating towards the oppositions goal.

The calculation of the path derives directly from the information contained in the map. This path may be considered as a strategy for the team in the sense that if the robots are used to force the ball to follow this path, a goal will be scored, yet the time-encoding ensures that the plan is flexible and easily adapted to movements of the opposition. Hence, it can be seen that it is not necessary to be concerned with the exact positioning of the robots during path extraction, but rather their respective influences on the formation of the Time Encoded Terrain Map are sufficient to direct the strategy to best use the resources available. Once the map has been formed and the path extracted, the path is interpreted as a strategy and appropriate roles and actions are allocated to each robot to fulfill it.

Terraforming — The terrain map is a discretized representation of the field broken to a granularity of $80mm \times 80mm$ squares. The granularity may be varied with a smaller granularity giving greater accuracy while a larger granularity has significant speed benefits.

For the Decision Engine, The terrain map is formed as in Figure 7 by the superposition of several layers of information derived from various sources. In this case, the base map is fixed at start up time and has high terrain (badness) around such areas as the walls and our goal as it is best if these areas are avoided. Similarly, it has low areas around their goal as the general tendency should be to bring the ball to this area. Above this, the coach map adds biasing to reflect the favored areas calculated by the game analyzer from previous game situations.

Finally, the influence shapes of the robots, determined according to their performance characteristics and relative positions, are superimposed on the map. The performance of our own robots is known by performing calibration trials during training while that of the opposition robots may be estimated beforehand and then refined during play by the game analyzer through observation of their motion.

The dominant cost in rebuilding the map is the addition of the robot influences to the combined base and coach maps. The influence of each of M robots must be rotated and scaled and superimposed on the map. The performance of these functions is proportional to the magnitude of the influence in terms of the number of nodes it covers. If each influence i

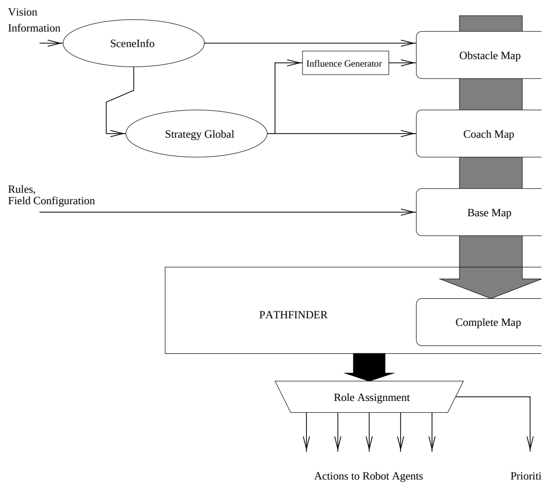


Figure 7: Cooperation Planner structure illustrating the core use of Time Encoded Terrain Maps to fuse multiple data sources and extract a unified strategy.

covers N_i nodes then scaling by factor k_i, rotation and adding it to the base map involves kN_i operations. So the total cost is $M \times k_i N_i$ giving an operational time complexity of $O(M \times N)$. In RoboCup, the number of obstacles is fixed and $N \propto d^2$ where d is a measure of the field granularity (nodes per meter or similar), hence the complexity becomes $O(d^2)$. For systems where the number of obstacles is not fixed, the complexity is linear with the number of obstacles.

Path Finding — Pathfinding in the terrain map is performed using the A* algorithm, a variant of Dijkstra's least distance path finding algorithm which uses a guiding heuristic to speed the process. Dijkstra's algorithm, using a Fibonacci Heap as its sorting mechanism, performs in $O(N \log N)$ time where N is the number of nodes which the field is broken into. In terms of field dimensions d, the time complexity becomes $O(d^2 \log d)$ [1]. The time complexity using the A* algorithm is the same although with a lower constant. When compared to the complexity of the map formation ($O(d^2)$), it is clear that the path finding is the dominant factor in determining overall complexity.

The use of the guiding heuristic can additionally be used as a parameter to trade off search time against path quality. If the heuristic uses a conservative estimate of the remaining penalty to destination, then the optimal path is assured. However as the search will close faster with a stronger guiding heuristic, setting the estimate to be greater than the theoretical minimum penalty to destination will produce a faster search, but trading off path optimality for that speed.

3.1.2 Path Interpretation

The final path produced by the pathfinder represents the strategy to be implemented. It is presented in the form of a list of way-points and stages which designate a piecewise linear route for the ball. The decision engine can then use this information to assign a robot to take the ball and follow this plan.

The segments are grouped into stages and assigned to different robots by examining the path for kick sections and turning points and using these as passing points. Once the path has been broken down, a robot is assigned to handle each stage. This becomes this robot's role. If there are more stages than can easily be handled by the number of robots available, only the first stages will be handled. Robots that are not assigned an active role are assigned a default *home position* role.

The strategy generated by the Cooperation Planner is a multi-stage multi-role strategy which provides a complete plan from the current point to goal divided into clearly defined intermediary goals and stages. This advanced level of planning allows the Decision Engine to perform robot actions and movement in advance of requirement for stages which are planned to occur in the near future. This allows the controller to move secondary robots into positions to receive the ball or to support the primary robot with the ball before the primary robot has even kicked the ball. As the robots move into positions for subsequent stages of the strategy their influences on the Time Encoded Terrain Map vary accordingly so that on successive iterations, the plan is effectively emphasized to encourage its continued adoption.

3.2 Operational Results

Verification or validation of the operation of the Decision Engine is difficult to perform in a realistic manner given the number of parameters involved and the dynamic environment in which it is designed to operate. Additionally, scientific formulation of a football strategy is not available and so the strategies generated by the Decision Engine can only be subject to human validation as being *sensible* or *how we might play the same situation.* This is reflected in the many parameters and weights which are tuned by hand to reflect the authors' views on the relative merits of various tactics (e.g.: passing versus dribbling or offense versus defense). Thus, testing of the system reduces to testing the system response to fixed situations and then comparing the strategy generated to what a human might decide to ensure that it is a sensible and credible plan.

However, real world application of the strategy is not as straight-forward. Such coordinated efforts require a high degree of precision on behalf of the robots and the robot controllers. Indeed, at the RoboCup'98 event, an informal survey of team capabilities showed that only one other team even attempted to pass the ball (CMUnited[8]), and even then it was limited to their two forward players only. The generality of the Decision Engine enabled this multi-stage approach to be applicable in further situations and other examples of passing and team coordination were observed during match play. Although the robots used by Cambridge in the matches were relatively slow and imprecise compared to those of many teams present, the Cambridge Robot Football Team achieved a creditable fourth placing out of the eleven teams present not via a specific rule base but through generic strategy and cooperation determination.

4 Conclusions

In this paper we have presented a Time Encoded Terrain Maps as a novel technique for incorporating temporal data into a spatial representation of an environment. This allow the extraction of path plans and decisions without explicit knowledge of the time varying intricacies involved. This technique provides path finding functionality for environments which are highly dynamic or adversarial which cannot be obtained using traditional mapping and path finding techniques. The maps encode time through the distortion and manipulation of obstacle representations in the map relative to the current path source point to reflect their potential motion over the course of execution of the path. This results in a path which is characterized by short term accuracy and long term validity. Recalculation of the path during execution gives scope for path refinement for the new short term.

The technique is computationally intensive ($O(d^2 \log d$ for field of dimension d) but remains suitable for use in real time applications where computation time is limited as the actual execution time coefficient is small and the mapping resolution and path finding performance can be traded against execution speed. Scope exists for further extension of this work to increase its suitability for real time or physically situated applications. The use of algorithms such as the D* path finding algorithm[9] for the actual path finding can decrease path finding time by reusing path information from previous calculations when refining a path. Similarly, multi-resolution or resolution independent grid-based methods[10] may be applied to the mapping procedure for speed and accuracy benefits.

The application of this technique to implement a strategy generator and cooperation planner for RoboCup Robot Football has shown that the techniques as it stands is suitable for actual real time situations which fit the above criteria and further demonstrates the flexibility of the technique to handle the fusion of various data sources.

The ability of the Cooperation Planner to incorporate potential future states of the environment through Time Encoded Terrain Maps gave a clear advantage in play by being able to plan several steps in advance. This was clearly illustrated in the abilities of the robots on field to move into suitable positions to receive ball passes even before the ball had been kicked.

Acknowledgments

The authors would like to thank the Olivetti and Oracle Research Laboratory (now AT&T Laboratories Cambridge) for funding this work and also other members of the Cambridge University RoboCup team, namely: Ben Bradshaw, Dave Crosby, Steve Hodges, Steve Lloyd, Jian Wang and Stuart Wray.

References

[1] F. M. Jönsson. An optimal pathfinder for vehicles in real world digital terrain maps. Master's thesis, Royal Institute of Science, Stockholm, Sweden, 1997.

[2] S. M. LaValle and R. Sharma. On motion planning in changing, partially-predictable environments. *International Journal of Robotics Research*, 16(6):775–805, 1997.

[3] R. Sedgewick. *Algorithms*, volume 2nd edition. Addison-Wesley, 1988.

[4] S. C. Shapiro. *Encyclopedia of Artificial Intelligence, 2nd edition.* Wiley, New York, 1992.

[5] T. Edmonds. A decision engine for robot football. Technical report, Laboratory for Communications Engineering, Engineering Department, University of Cambridge, Trumpington St, Cambridge, UK, CB2 1PZ, July 1998.

[6] H. Kitano, M. Asada, I. Noda, E. Osawa, and H. Matsubara. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, 1997.

[7] A. Rowstron, B. Bradshaw, D. Crosby, T. Edmonds, S. Hodges, A. Hopper, S. Lloyd, J. Wang, and S. Wray. CURF: Cambridge university robot football team. In Minoru Asada, editor, *RoboCup-98: Proceedings of the second RoboCup Workshop*, pages 503–509, July 1998.

[8] M. Veloso, P. Stone, K. Han, and S. Achim. CMUnited: A team of robotic soccer agents collaberating in an adversarial environment. Carnegie Mellon University, 1997.

[9] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings,1994 International Conference on Robotics and Automation*, volume 4, pages 3310–3317, 1994.

[10] G. Krishnaswamy and A. Stentz. Resolution independent grid-based path planning. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1995.