Coordinating Multiple Virtual Characters with PRM Planning

Julio C. Mora, Abraham Sánchez L., David Nuñez R., Maria A. Osorio

Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570
Puebla, Pue., México
neomora@hotmail.com, asanchez, aosorio@cs.buap.mx, dnunezr@yahoo.com.mx

Abstract. Creating complex animation of virtual characters with simple high-level commands has been a design goal of animation software for a long time. The 2-stages locomotion planning is an effective new approach in this direction aiming to creating realistic animations [4]. Nevertheless, creating coordinated motion between animated characters has always been a great challenge because of the computational complexity of the problem. This paper describes a method to coordinate the motions of several virtual characters, and to animate them. In the method, we combined centralized and decoupled planning methods. The success of this approach is demonstrated through several examples.

1. Introduction

Virtual humans are becoming increasingly popular both in the academic and industrial domains. Applications featuring virtual humans include computer games, virtual storytelling, movie industry, entertainment, military simulations and behavioral modelling.

We know that many dynamic models were designed in the nineties in order to synthesize human figure motion, but it is unlikely that dynamics simulation will solve all animation problem. Researchers have thus turned to other kinds of approaches. The recent progress in motion capture techniques makes it possible to directly use human motion data. For most applications, the captured motion needs to be modified in order to create a variety of specific animations, that may take the synthetic environment into account [1]. In recent years, Mocap system results in amount of realistic human motion data and is widely used in various applications.

The human figure has been frequently represented in computer animation with the articulated mechanisms used in robotics to control manipulators. With this same idea, many techniques have been developed within both fields robotics and computer animation having two different goals in mind. Applications in computer animation have been focused in obtaining realistic-looking motions. On the other hand, the robotic's interest is to generate motions without caring for realism.

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 197-208

Received 06/03/08
Accepted 26/04/08
Final version 04/05/08

Several approaches to automatically plan motions for a human character have been developed in recent years (see for example [2], [3], [4], [5], [6]). In [7], the authors combined techniques used from robotics and computer animation fields accounting for two behaviors: locomotion and manipulation. The main challenge in this research, deals with 3D collision avoidance while preserving the believability of the motions and allowing cooperation between two or more virtual characters. To accomplish the coordination task, a geometric and kinematic decoupling of the system is proposed. The decomposition enables the character's system to plan a collision-free path for a reduced version, then to animate locomotion and grasping behaviors independently, and finally to automatically tune the animation to avoid residual collisions [8].

We present a novel method for animating coordinated virtual characters. First, the geometric path for each character is calculated using a centralized approach and then, the paths are transformed into trajectories; and finally the coordinated motions are calculated using a decoupled approach. The problem of coordinating multiple robots in robotics has studied intensively during the last years [9], [10], [11]. We do not have knowledge of its application in the coordination of multiple virtual characters.

The paper is organized as follows. Section II presents an overview of the multi-robot planning approaches. The proposed procedure to coordinate motions for virtual characters is discussed in Section III. The performance of this proposal is evaluated in Section IV. Finally, we give some concluding remarks and future work in Section V.

2. Multi-robot Planning

The many approaches to multi-robot motion planning are usually compared based on their algorithm's speed, completeness and optimality. For complex problems, it is difficult to meet all of these requirements. Algorithms based on randomized sampling proved to be the only viable algorithmic tool for quickly solving motion planning problems involving many degrees of freedom. Information on the configuration space is acquired by generating samples and finding simple paths among them. Paths and samples are stored in a suitable data structure. Probabilistic Roadmap planners have recently gained popularity because of their speed [11]. However, effective sampling strategies are crucial to achieving successful PRM planning.

We suppose that there are multiple robots that share the same world, $\mathtt{W}$. A path must be computed for each one that avoids collisions with obstacles and with other robots. The i-th robot will be denoted by $\mathtt{A}^i$. Suppose are n robots, $\mathtt{A}^1, \mathtt{A}^2, ..., \mathtt{A}^n$. Each robot, $\mathtt{A}^i$, has its associated configuration space, $\mathtt{C}^i$, and its initial and goal configurations, q^i_{init} and q^i_{goal}. The formulation for multiple-robot motion planning is [12]:

- The world, $\mathtt{W}$ and obstacle region $\mathtt{O}$.
- There are n robots $\mathtt{A}^1, \mathtt{A}^2, ..., \mathtt{A}^n$, which each may consist of one or more moving bodies.

- Each robot, A^i, for $1 \leq i \leq n$ has an associated configuration space, C^i.
- The state space, X , is defined as the Cartesian product

$$X = C^1 \times C^2 \times \cdots \times C^n \tag{1}$$

- The obstacle region in X is

$$X_{obs} = \left(\bigcup_{i=1}^{n} X^{i}_{obs} \right) \bigcup \left(\bigcup_{ij, i \neq j} X^{ij}_{obs} \right), \tag{2}$$

  in which X^{i}_{obs} and X^{ij}_{obs} are the robot-obstacle and robot-robot collision states.
- A state x_i X_{free} is designated as the initial state, in which $x_i = (q^1_i, ..., q^n_i)$.
- A state x_g X_{free} is designated as the goal state, in which $x_g = (q^1_g, ..., q^n_g)$.
- The task is to compute a continuous path, $\tau : [0,1]$ X_{free} such that $\tau(0) = x_{init}$ and $\tau(1)$ x_{goal}.

Multi-robot motion planners are usually classified according to whether the planning is centralized or decoupled.

Centralized planning consists of considering all the robots as if they were forming a single multi-robot, by encoding their degrees of freedom (dof) in a single "composite" C and searching that space for a free path between the initial and goal configurations. In principle, any sufficiently general motion planning algorithm can be used to implement centralized planning. In the past, this type of planning has not been considered practical because it usually leads to searching large-dimensional configuration spaces that are beyond the practical capabilities of existing planning techniques [10].

The complexity of the centralized approach led to the emergence of the decoupled approaches to multi-robot motion planning, where completeness is sacrificed in the favor of complexity. These approaches first design motions for the robots while ignoring robot-robot interactions. Once these interactions are considered, the choices available to each robot are already constrained by the designed motions. If a problem arises, these approaches are typically unable to reverse their commitments. Therefore, completeness is lost. Nevertheless, decoupled approaches are quite practical, and in some cases completeness can be recovered [9].

Decoupled planning is a two-phase approach, in the first phase, a collision-free path is generated for each robot by considering only the obstacles in the environment and ignoring the other robots; in the second phase, called the velocity tuning, the relative velocities of the robots along their respective paths are selected to avoid collision among them. Velocity tuning consists of searching a coordination space [10]. Decoupled planning leads to searching lower-dimensional spaces than centralized planning.

A straightforward approach to decoupled planning is to sort the robots by priority, and plan for higher-priority robots first. Lower-priority robots plan by viewing the higher-priority robots as moving obstacles [13].

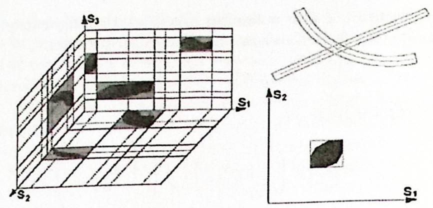


Fig. 1. Coordination space for three and two robots [14].

An interesting study [11] compared the decoupled and the centralized approaches by implementing the SBL planner. The study concluded that decoupled planners are very unreliable in practice due to their inherent incomplete nature. Hence they suggest that a centralized should usually be the preferred choice (especially in cases requiring tight robot coordination), even though they are slower than that a decoupled planner when the latter succeeds.

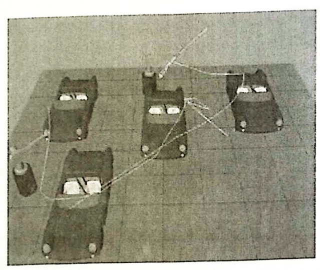


Fig. 2. Two mobile manipulators. The paths were computed by using a centralized approach.

3. Coordinated Planning for Virtual Characters

The issue of robot motion planning has been studied for a couple of decades and many important contributions to the problem have been made [9]. Most of these algorithms are based on the use of the C. The inherent difficulty with this approach is its high dimensionality. It is well known that the worst-case time bound for any complete motion planning algorithm is exponential in the dimensionality of C. For this reason, attention has focused on randomized or probabilistic motion planning methods. Probabilistic Roadmap Methods (PRM) is a general planning scheme building probabilistic roadmaps by randomly selecting configurations from the free-configuration space (F) and interconnecting certain pairs by simple feasible paths. This process continues until the graph covers the connectedness of the space.

The application of motion planning techniques in computer animation is more recent. Much works has been shown to be effective in facilitating the automatic generation of complex animations for human characters. We have developed two multi-robot planners respectively based on centralized planning and decoupled planning.

Animating very complex model such as virtual humans is usually done by extracting a simpler representation of the model, a "skeleton", that is an articulated figure made of rigid links connected by hinges. We first prepare a skeleton model represented with a hierarchical structure of rotational joints. The number of joints of the model and the degrees of freedom depend largely on the desired reality or quality. In this work, we employ a relatively simple model with 52 degrees of freedom (dofs). The structure of the character is modelled in two levels. Pelvis and legs are used for the locomotion, all the 18 dofs are said to be active dofs. The 34 other ones are said to be reactive dofs, they deal with the control of the arms and the spine. The pelvis is the root of five kinematics chains modelling respectively the arms, the legs and the spine. Root's trajectory problem concerns only $[x, y, x, \theta, \phi, \psi]$ parameter's evolution. The inputs are two collision-free configurations of the character in the C. Bounding the character's geometry by a cylinder allows motion planning for navigation to be reduced to planning collision-free trajectories for this cylinder in 3D.

The centralized approach is exactly the PRM planner running in the composite C of the characters (cylinders). The collision test of a point in C is done by calling the collision checker on every pair consisting of a rigid body of a character and an environment obstacle, and on every pair of rigid bodies belonging to two distinct characters. To handle the requirement for speed, a single-query probabilistic roadmap planner is adapted for this purpose.

The decoupled approach makes $n + 1$ calls to PRM planner, where n is the number of characters: n calls to plan the path of each character (ignoring the other characters); then, another call to plan a collision-free path in the space $X = [0, 1]^n$. The roadmap in X is built by considering the start and goal configurations at $(0, ..., 0)$ and $(1, ..., 1)$. The collision test of a point in X is done by calling the checker on every pair of rigid bodies belonging to two distinct characters.

Our procedure consists of the following steps:

1. **Roadmap construction**: Given a virtual environment, we randomly sample valid configurations of the cylinder bounding the lower part of the characters. The roadmap can be modeled as a directed graph whose nodes represent valid samples of the configuration space. A pair of nodes are connected by and edge if the character can move from one node to the other with a prescribed motion while preserving its lifelikeness.

2. **Roadmap search**: Once the roadmap reflects the connectivity of F , it can be used to answer motion planning queries [15]. Then, a path in the roadmap corresponds to the motion of each character. The first path is smoothed by a classical dichotomy technique.
3. **Motion generation**: Simply computing a collision-free path in the environment is not enough to produce realistic animation. This step transforms the obtained path into a discrete set of time stamped positions for the character along the trajectory, respecting some criteria of velocities and accelerations. Finally, a locomotion control procedure transforms a set of time parameterized positions into a walk sequence, or other sequences.
4. **Coordinated motion**: Let suppose that each character, A^i is constrained to follow a path $\tau_i : [0, 1] \quad C^i_{free}$, which can be computed using any motion planning technique. The n-dimensional state space for n characters is called a coordination space. The schedules of the characters motions will be performed in the coordination space without collision.
   For n characters, the coordination space, X is defined as the n-dimensional unit cube $X = [0, 1]^n$. The i^{th} coordinate of X represents the domain $S_1 = [0, 1]$, of the path τ_i. A state, $x \quad X$, therefore indicates the configuration of every character. For each i, the configuration $q^i \quad C^i$ is given by $q^i = \tau_i(x_i)$. At state $(0, ..., 0) \quad X$, every character is in its initial configuration, $q^i_{init} = \tau_i(0)$, and at state $(1, ..., 1) \quad X$, every character is in its goal configuration $q^i_{goal} = \tau_i(1)$. Any continuous path, $\sigma : [0, 1] \quad X$, for which $\sigma(0) = (0, ..., 0)$ and $\sigma(1) = (1, ..., 1)$ will move the character toward their goal configurations. The path σ does not even need to be monotonic, in contrast to prioritized planning.

4. Experimental Results

The RMP3D software, developed at the Computer Science Department - BUAP, has been used as support platform for implementing the method. The motion planning platform RMP3D was implemented on an Intel c Pentium IV processor-based PC running at 2.6 GHz with 1Gb RAM, using Builder C++ Version 6.0 and OpenGL. RMP3D is a multipurpose tool for visualizing and editing motion planning environments, problem instances, and their solutions. The advantages of our platform are many with respect to other proposals, but the most important thing is the level of modelling (environments and characters), since this is possible by using c Inivis AC3D.

Given start and goal positions in a virtual environment for each character, our objective is to find a coordinated motion of the human characters to move them from the start to the goal.

Figures 3 and 4 show different stages of our method. It can be seen the un-smoothed paths obtained with the centralized approach. Due to the probabilistic nature of the algorithm and the absence of cycles, paths built during the roadmap construction are long and irregulars. We used an iterative relaxation-based method that incrementally reduces the path length. Once the path has been smoothed, this path comes near to a composition of Bézier curves respecting C^1 continuity constraint. Then, the resulting geometric path is transformed into a trajectory respecting some criteria of maximal velocities and accelerations. The locomotion controller generates an animation for all previously computed key-frames from a motion capture data set. In order to obtain the coordinated motions of the characters, we applied a decoupled approach. Finally, we can see the animation of coordinated movements.

In last few years, motion capture techniques have been widely used to animate 3D rigid body skeleton. By using magnetic or optical technologies, it is possible to store the positions and orientations of points located on the human body. A further computation provides the correspondence between the synthetic skeleton and the real skeleton, in order to adapt data to the new morphology. For most applications, the captured motion needs to be modified in order to create a variety of specific animations, that may take the synthetic environment into account. All techniques that take well-known trajectories (described by keyframes or motion captured trajectories) and modify them in order to change the motion are called motion warping.

The motion warping is an editing technique introduced in [16]. The Witkin and Popovic's method modifies the original motion curve $x(t)$ by interactively turning the positions of pre-selected x-coordinates and by scaling and shifting the motion curve as a whole. The pattern of the original motion curve is preserved, but the resulting curve satisfies the constraints of new key-frames. A key advantage of motion warping is that it fits well into the familiar key-frame animation paradigm. It slightly modifies the predefined animation on the reactive degrees of freedom. In such a way the realism of the original animation is preserved at the best. In the figure 4, the characters would collide with the sheep and the tree placed in the virtual scene.

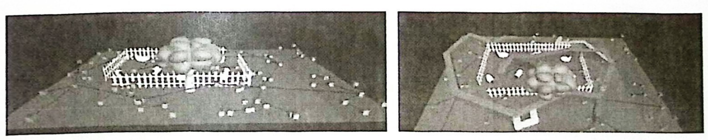


Fig. 3. Roadmap construction and roadmap search.

Obstacle avoidance is taken into account at two distinct levels. The first planned path is guaranteed to be collision-free for the lower part of the each character body. Then by applying the motion controller along that path, all degrees of freedom of the every character are animated. Only the upper parts of the body can be in collision. This obstacle avoidance module is important since the collision-free motion planning is only assured for the active degrees of freedom of the character (i.e., those degrees of freedom bounded by a cylinder). The goal of the warping strategy is to locally modify the animation of the upper

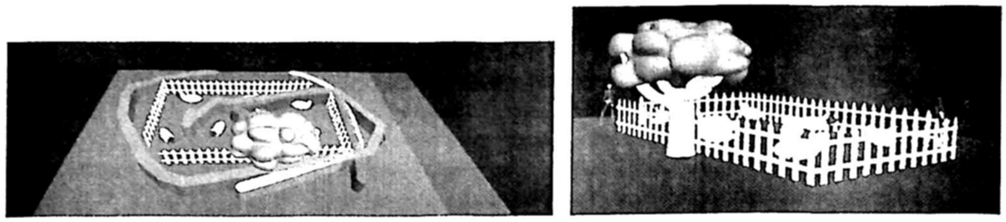


Fig. 4. Each path is smoothed with a dichotomy technique and a snapshot where one can see the execution of the coordinated motions.

bodies of the character (the arms and spine) when a collision occurs in the produced animation. The animation is a sequence of key-frames that specifies all the degrees of freedom of the character. In each key-frame of the sequence is performed a collision test.

The motion library is provided by the CMU Graphics Lab Motion Capture Database. In our experiments presented in this paper, we only showed examples of simple movements like walking or running, although with our approach it is possible to use any other movement. Figures 5 and 6 illustrate examples of coordinated motions. In both cases the characters must avoid collisions with the obstacles while they execute their motions (this concerns the reactive degrees belonging to the arms or the head). In the chess environment (figure 6 at the left), one of the characters collides with a black chess piece (the character with clothes in white), the realism of the motion is preserved after the application of the warping module. In the forest environment, both characters collide with the branches of the trees (figure 6 at the right).

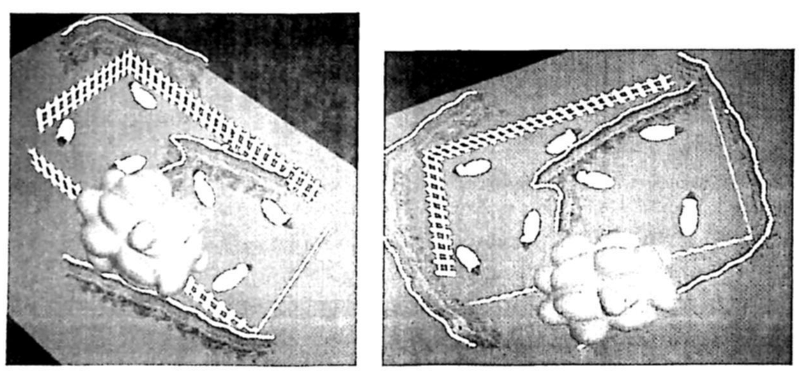


Fig. 5. Two different views of the execution of coordinated motions in an environment that contains some sheep as obstacles.

The automated synthesis of motion for characters in unstructured environments is difficult because it requires solving a planning problem subject to multiple constraints. Obstacles in the environment constrain the motion in an obvious fashion, as typified by narrow passages. Other types of constraints include a character's joint limits, the requirements for balance, the character's natural disposition for particular postures and motion, and so on.

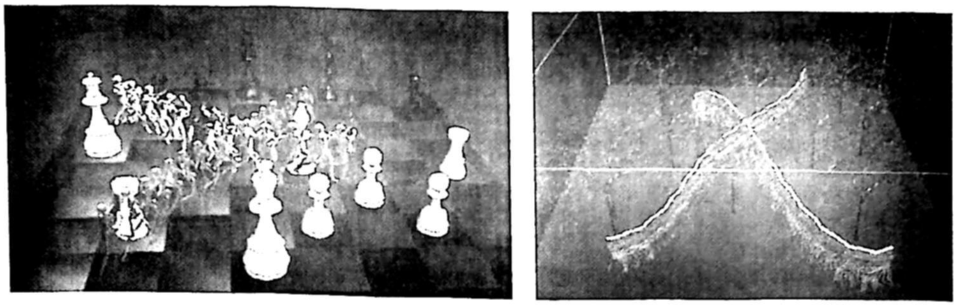


Fig. 6. Two different scenes that show the coordinated motions of two characters.

The increase in the running times when the number of characters grows is caused by the following observations: i) the quadratic increase in the number of pairs of bodies that must be proved by collision and ii) the great difficulty of the problems due to the constraints imposed by the additional characters upon the motions of the others. After executing many tests in different scenes with different number of characters, we can mention that in the coordination module, the planner fails in certain canonical problems (see for example, figures 7 and 8).

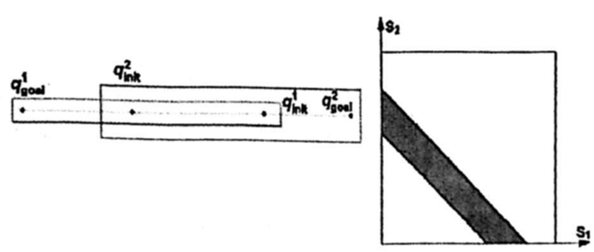


Fig. 7. The crossing in the same uniform route.

The experimental results show examples of up to 3 characters, because the time to coordinate and to generate the movements of them grows exponentially.

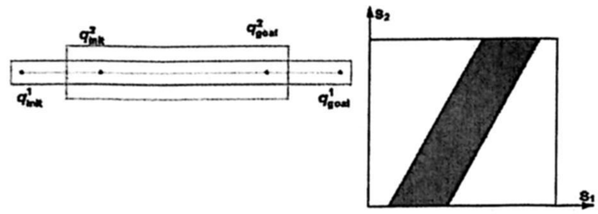


Fig. 8. Surpassing in the same route.

5. Conclusion and Future Work

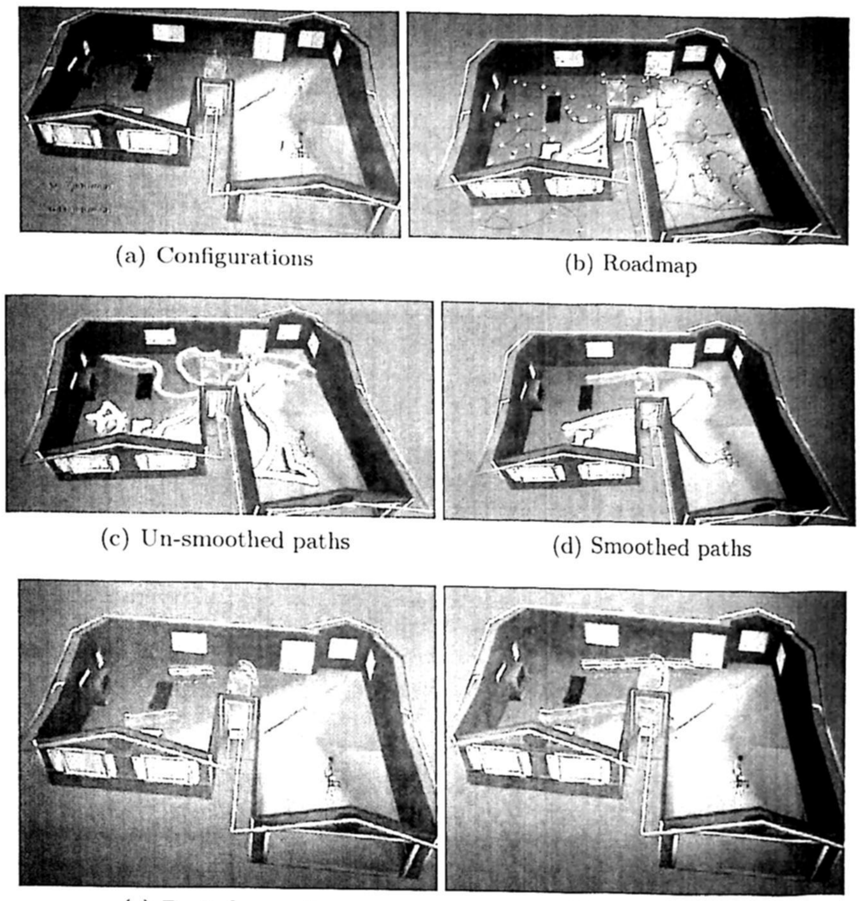
This paper takes advantage of both view points computer graphics and robotics to address virtual human coordination. It focuses in the following problem: how to automatically compute a coordinated motion of several characters in a virtual environment while guaranteeing 3D obstacle avoidance?

Virtual environments are widely used in computer entertainment, where multiple agents are moving around in the virtual environment and their motions have to be coordinated. Motion planning and path finding have been studied extensively, both in the game community, and in the robotics community. The standard approaches for motion planning of multiple agents in the robotics community can be divided into two major categories: centralized planning and decoupled planning. The former considers all agents as one robotic system with many degrees of freedom, and its time complexity is exponential in the dimension of the composite configuration space.

We have presented the application of both approaches. The obtained results were interesting and the running times are acceptable (see Figure 9). We believe that this approach could be easily adapted to the case of multiple virtual characters in dynamic environments.

The core planners employed in both centralized planning and decoupled planning require either an expensive pre-processing phase (for multi-query planners) or more expensive query times (for single-query planners), particularly for a large number of agents. To plan motions of multiple agents in virtual environments and games, a planner may not spend more than a few seconds in the pre-processing phase, and it must be able to perform path query in real-time (when the game is being played). The runtime of such a planner must also scale well with the number of agents in virtual environments. For difficult motion planning problems, centralized planning may be necessary in order to find feasible paths; however, virtual environments in computer games are often structured. Often these environments can be characterized as open spaces connected by multiple narrow passages.

An interesting idea would be to consider human behaviors, especially in the context of interaction with physical objects. To generate complete motion sequences of one or more virtual characters transporting a bulky object in cluttered environments, can be future research.



(a) Configurations (b) Roadmap

(c) Un-smoothed paths (d) Smoothed paths

(e) Partial execution (f) Coordination

Fig. 9. Snapshots showing the complete approach.

References

1. Multon F., France L., Cani M. P. and Debunne G. "Computer animation of human walking: A survey", *The Journal of Visualization and Computer Animation*, Vol. 10, (1999) 39-54
2. Koga Y., Kondo K., Kuffner J. J. and Latombe J. C. "Planning motions with intentions", *ACM SIGGRAPH Computer Graphics*, Vol. 28, (1994) 395-408
3. Kuffner J. J. "Goal-directed navigation for animated characters using real-time path planning and control", *CAPTECH*, (1998) 171-186
4. Pettre J., Laumond J. P. and Simeon T. "A 2-stages locomotion planning for digital actors", *Eurographics/SIGGRAPH Symposium on Computer Animation*, (2003) 258-264
5. Sánchez L. A., Sánchez T. J. and Zapata R. "Planning motions for animated characters", *Journal Research in Computing Science*, Vo. 17, (2005) 265-274
6. Choi M., Lee J. and Shin S. "Planning biped locomotion using motion capture data and probabilistic roadmaps", *ACM Transactions on Graphics*, Vol. 22, No. 2, (2003) 182-203
7. Esteves C., Arechavaleta G. and Laumond J. P. "Planning cooperative motions for animated characters", *International Symposium on Robotics and Automation (ISRA)*, (2004)
8. Esteves C., Arechavaleta G., Pettre J. and Laumond J. P. "Animation planning for virtual characters cooperation", *ACM Transactions on Graphics*, Vol. 25, No. 2, (2006) 319-339
9. Latombe J. C. "Robot motion planning", *Kluwer Academic Publishers*, (1991)
10. Sánchez G. and Latombe J. C. "On delaying colllision checking in PRM planning: Application to multi-robot coordination", The International Journal of Robotics Research, Vol. 21, No. 1, (2002), 5-26
11. Sánchez G. and Latombe J. C. "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems", *Proc. of the IEEE Robotics and Automation Conference*, (2002) 2112-2119
12. LaValle Steven M. "Planning algorithms", *Cambridge University Press*, (2006)
13. Erdmann M. and Lozano-Perez T. "On multiple moving objects", *Proc. of the IEEE Robotics and Automation Conference*, (1986) 1419-1424
14. Leroy S., Laumond J. C. and Siméon T. "Multiple path coordination for mobile robots: A geometric algorithm", *Proc. of International Joint Conference on Artificial Intelligence*, (1999) 1118-1123
15. Kavraki L., Švetska P., Latombe J. C. and Overmars M. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996) 566-580
16. Witkin A. and Popovic Z. "Motion warping", *Proc. of SIGGRAPH* (105-108) 1995