GEMPA: Graphic Environment for Motion Planning Algorithms

Antonio Benitez and Alejandro Mugarte

Department of Informatics Engineering
Universidad Politécnica de Puebla
Tercer Carril del Ejido "Serrano"
San Mateo Cuanalá s/n. 72820 México
antonio.benitezruiz@gmail.com
http://geocities.com/antonio.benitezruiz

Abstract. This paper describes a graphic tool to simulate virtual environments involved in three dimensional space. GEMPA (Graphic Environment for Motion Planning Algorithms) was developed to be used to simulates the behavior of motion planning algorithms. Therefore, this tool includes the object's representation using different kind of files; the animation of free collision path for a motion planning problem, and graphic user interface to navigate through the virtual environment. Important elements to give more realism to the environments as illumination and textures are considered. Initially, this tool can be used as a didactics software for computer graphics and animation courses showing how transformation on three dimensional space can be used to animate the object movement, but the main application is the simulation of motion planning algorithms.

Key words: 3-D graphic environment, simulation and visualization tool.

1 Introduction

Computer graphics has advanced to a point where generating images of striking realism and complexity has become almost commonplace. However, making objects move convincingly within these pictures remains difficult, particularly as object models grow increasingly complex. The specification and control of motion for computer animation has emerged as one of the principal areas of research within the computer graphics community.

Computer graphics has grown phenomenally in recent decades, progressing from simple 2-D graphics to complex, high-quality, three-dimensional environments. In entertainment, computer graphics is used extensively in movies and computer games. Animated movies are increasingly being made entirely with computers. Even nonanimated movies depend heavily on computer graphics to develop special effects. The capabilities of computer graphics in personal computers and home game consoles have now improved to the extent that low-cost systems are able to display millions of polygons per second.

Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 225-236

Received 07/03/09
Accepted 27/04/09
Final version 05/05/09

There are also significant uses of computer graphics in nonentertainment applications. For example, virtual reality systems are often used in training. Computer graphics is an indispensable tool for scientific visualization and for computer-aided design (CAD).We need good methods for displaying large data sets comprehensibly and for showing the results of large-scale scientific simulations. Over the last few years, many different systems have been developed to represent and simulate scenarios with different kind of objects in virtual environments.

The representation of different environments in such a system is used for a widely researched area, where many different types of problems are addressed, related to animation, interaction, and motion planning algorithms to name a few research topics. Although there is a variety of systems available with many different features, we are still a long way from a completely integrated system that is adaptable for many types of applications.

This motivates us to create and build a visualization tool for planners capable of using physics-based models to generate realistic-looking motions. The main objective is to have a solid platform to create and develop algorithms for motion planning methods that can be launched into a digital environment. The developed of these tools allows to modify or to adapt the visualization tool for different kind of problems.

This paper is organized as follows: First we present the support architecture for GEMPA including different kind of files used to recover information about the objects inside the environment(Section 2). Then we review the graphic user interface (GUI) to navigate trough the virtual graphic environment, Section 3. After we describe how GEMPA is used to simulate motion planning algorithms and to build simple environments in Section 4, we continue in Section 5 to present interesting results for 3-D objects transformations and simulation of motion planning algorithms. We conclude the paper and describe work in progress in Section 6.

2 GEMPA Architecture

GEMPA architecture is supported by necessary elements to represent objects, geometric transformation tools and visualization controls. These elements are integrated to reach initial goals of visualization and animation applied to motion planning problems.

2.1 Related Work

Important graphics tools has been developed to visualize and understand motion plannig problems. As an example, the description of two visualization tools will be described; VIZMO++ and MPK.

VIZMO++. VIZMO++ (A visualization/authoring tool for motion planning) is a 3D visualization/authoring tool for files provided/generated by OBPRM motion planning library [2]. VIZMO++, was developed for visualizing and editing motion planning environments, problem instances, and their solutions. The tool offers a nice and easy to use graphical user interface (GUI) that allows you to display workspace environments, roadmap, path, and start/goal positions. It enables users to interact with and edit the environment. Application where VIZMO++ can be used are: User-Guided Path Planning ; Particle Transport Seismic Ray Tracing and Campus Navigator [1].

MPK - Motion Planning Kit. Motion Planning Kit (MPK) is a C++ library and toolkit for developing single- and multi-robot motion planners [3]. It includes SBL, a fast single-query probabilistic roadmap path planner [5] .

MPK can handle arbitrary kinematic tree structures and an arbitrary number of robots and obstacles at the same time. New robots with any combination of prismatic and revolute joints can be defined and added without recompiling. Some importants features of MPK are: C++-library and workbench for motion planning; Allows arbitrary number of robots and obstacles; New robots can be added without recompiling; Arbitrary kinematics for 'hardwired' robots; Efficient and exact dynamic collision checker.

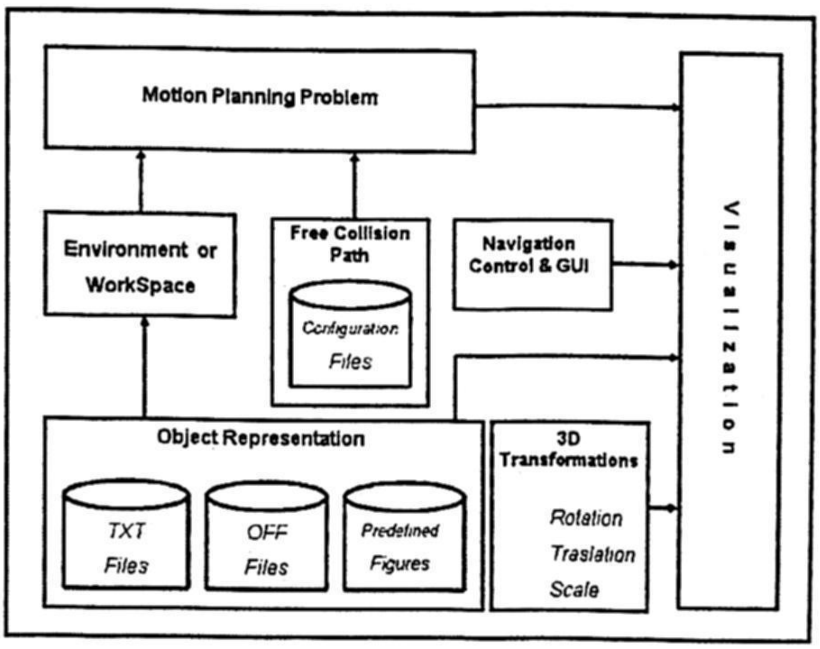

Fig. 1. Several modules conform the initial GEMPA architecture which offer interesting functionalities; visualization 3-D environments as well as animation of motion planning algorithms.

2.2 Recovering Objects Representation

People focus to solve problems using computer graphics, virtual reality and simulation of motion planning techniques used to recover information related to objects inside the environment through file which can storage information about triangle meshes. Hence, several objects can be placed on different positions and orientations to simulate a three-dimensional environment. There exist different formats to represent objects in three-dimensional spaces (3-D), however, two conventions used for many tools to represent triangle meshes are the most popular; objects based on *off - files* and objects based on *txt - files*. In motion planning community there exist benchmarks represented through this kind of files. GEMPA is able to load the triangle meshes used to represent objects from *txt* or *off - files*.

On the other hand, GEMPA allows the user to built news environments using predefined figures as spheres, cones, cubes, etc. This figures are chosen from a menu and the user can place them using translation, rotation and scale transformations. Once the user has built the environment, it can be saved as a *txt - file*. Besides this files can be load from now on.

Each module on GEMPA architecture is presented in Figure 1. There, we can see that initially, the main goal is the visualization of 3-D environments and the animation of motion planning algorithms. In the case of visualization of 3-D environments, information is recovered form files and the user can navigate through the envorinment using mouse and keyboard controls. In the second case, the animation of motion planning algorithms, GEMPA needs information about a problem. This problem is described by two elements; the first one is called workspace, where obstacles (objects) and robot representation and configuration (position and orientation) is recovered from files; the second, a set of free collision configuration conform a path, this will be used to animate the robot movement from initial to goal configuration. Both goals are supported on 3-D geometric transformations.

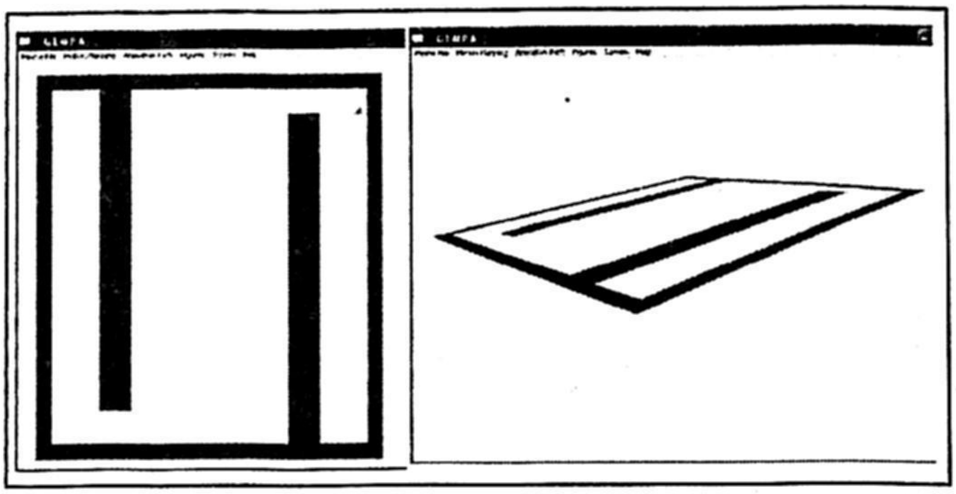

Fig. 2. Two different views of two-dimensional environment since the X-Y plane are painted.

3 GUI and Navigation Tools

Graphic user interface (GUI) is an important element for every computer system. This GUI become essential when application are used to simulate virtual environments, where navigation tools help the user to perceive realism and immersion characteristics. Initially, GEMPA has incorporated two modes to paint an object; wire mode and solid mode. Next, Lambert illumination is implemented to produce more realism, and finally transparency effects are used to visualize the objects.

Along the GUI, camera movements are added to facilitate the navigation inside the environment to display views from different locations.

3.1 2-D and 3-D Visualization

GEMPA was developed using 3-D transformations (translation, rotation and scale), that means that, if we want to display objects in 2-D, the user only have to place zero where z - *coordinate* or *gamma-angle* is required. Two dimension environment are painted on the XY plane. In Figure 2, two different views of the same environment are displayed on the XY plane.

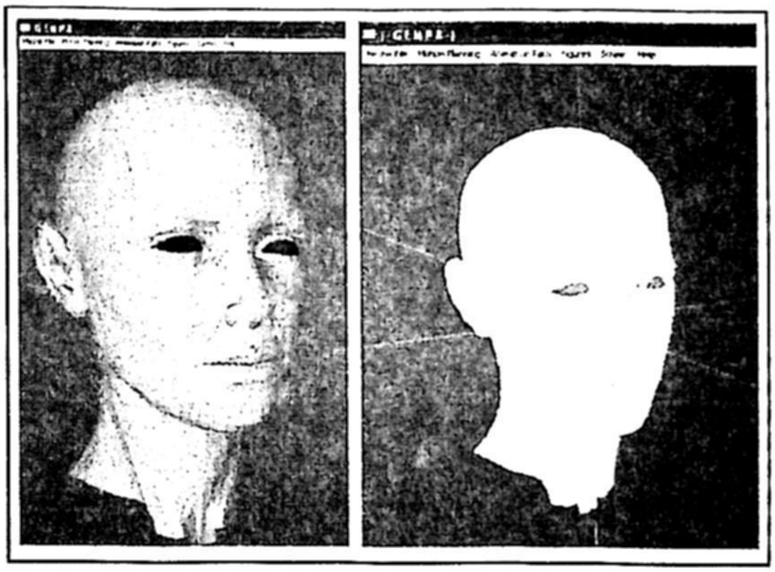

Fig. 3. In the left side, an object is painted using Lambert illumination , in the right side, transparency effect is applied on the object. Both features are used to give more realism the environment.

3.2 Navigation Controls and Realism Characteristics

This controls allow the user navigate through the environment. GEMPA support this navigation using keys and mouse movements. Hence, the tool include:

Camera movement on X-axis, Y-axis and Z-axis (zoom), and rotation around X,Y and Z axis. Both kind of controls are available since keyboard and mouse.

On the other hand, to give more realism the environment displayed by GEMPA, Lambert illumination has bee included along with transparency effect on the objects; colors for every predefined figure can be chosen. Besides, the tool is able to paint the axis to give the user information about the position where the observed is placed.

In Figure 3 we can see two samples where Lambert illumination and transparency effect are used to paint each object respectively.

4 Environments Construction and Motion Planning Algorithms

Even though GEMPA is able to read information about complex 3-D environments used to simulate motion planning algorithms [4, 7, 6], this tool besides allows the user to create or built his own environments.

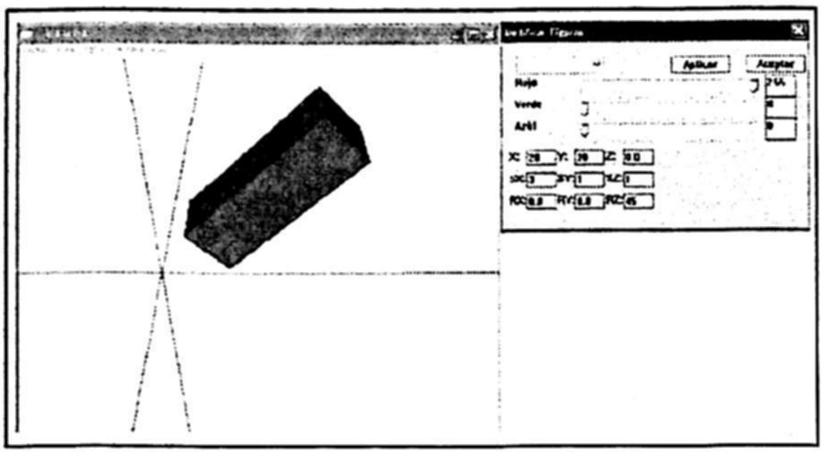

Fig. 4. The GUI includes a dialog box where position, orientation and color can be specified to create a figure.

4.1 Building Environments

GEMPA include the capability to incorporate objects with predefined figures as cubes, cones, spheres and torus. These figures are placed at the origin of a reference frame and the tool contemplate functionalities to apply geometric transformations to scale, move, or rotate the object, this can be seen in Figure 4, where a dialog box is provided to allow the user to change values for the position and orientation where every object will be placed inside the workspace. Therefore, an environment can be conformed by different objects distributes through the workspace, an example of such distribution is presented in Figure

5. It is important to say that, although figures used to build the environment seems simples, the user can construct more complex elements using repeatedly these objects.

Once an environment has been built, a function to save the environment as a *txt - files* is available. The file generated contains information about each object into the workspace and its position and orientation (*configuration*). This way, the user can reload the environment to be reused.

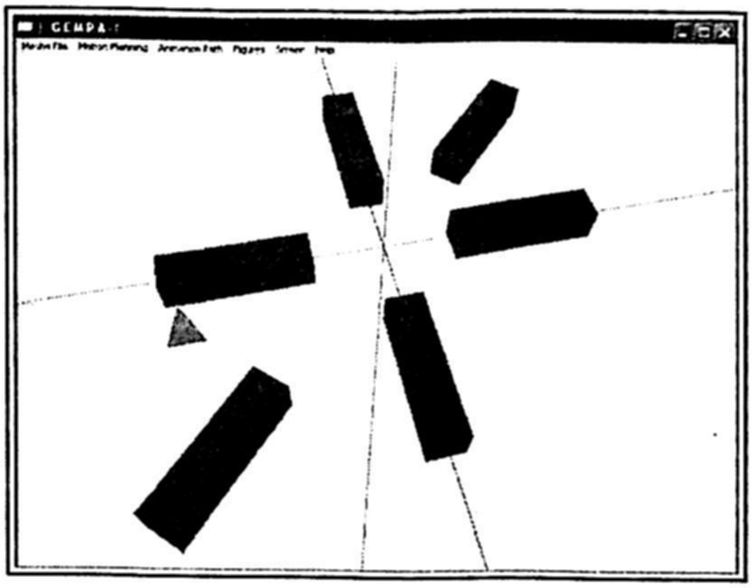

Fig. 5. The environment is built applying successive 3-D transformations on every figure to distribute them on the workspace.

4.2 Simulation of Motion Planning Problems

Motion planning methods are used in robotics to solve motion problems. Motion planning algorithms called probabilistic roadmap methods (PRM) are used to find a free collision path between a initial configuration and a goal configuration. There are many applications of motion planning algorithms. For this work, only PRM for free flying objects are considered as an initial application of GEMPA. Taking into account this assumption, the workspace is conformed by a set of obstacles (objects) distributed on the environment, these objects has movement restrictions, that mean that, the obstacles can not change their position inside the environment.

In addition, an object that can move through the workspace is added to the environment and is called robot. The robot can move through the workspace using the free collision path to move from the initial configuration to the goal configuration.

For PRM for free flying objects, only a robot can be defined and the workspace can include any obstacles as the problem need.

4.3 Simulating Free Collision Paths

PRMs generate as result a free collision path between init and goal configurations (if and only if the algorithm is able to find it). This path is a set of n configuration, where each configuration is represented as a six-tuple of parameters, which the three first parameters represent the position an the three last values represent the orientation. This free collision path is saved as a *configuration - file*, and the number of configurations will determinate how acute will be the animation. On the other hand, GEMPA include the capability to recover from

Environment File	Configuration File
Robot robot_angular.txt 0.0 0.0 0.0 0.0 0.0 0.0 1.0 20.0 0.0 0.7 1.2 0.8 Obstacle #1 bench_lamina_angosta_grande.txt 3.0 10 8.0 0.0 0.0 0.0 Obstacle #2 bench_lamina_angosta_grande.txt 3.0 10 -8.0 0.0 0.0 0.0 End	163 5.0492 -0.7257 -3.2830 2.2233 6.1148 4.0886 4.3394 -0.7601 -3.3466 2.1348 6.0969 4.1308 3.6292 -0.7945 -3.4102 2.0463 6.0791 4.1730 2.9189 -0.8289 -3.4737 1.9578 6.0613 4.2153 2.2086 -0.8633 -3.5373 1.8692 6.0434 4.2575 1.4983 -0.8977 -3.6008 1.7807 6.0256 4.2997 0.7881 -0.9322 -3.6644 1.6922 6.0078 4.3420 0.0778 -0.9666 -3.7280 1.6037 5.9900 4.3842 -0.6324 -1.0010 -3.7915 1.5152 5.9721 4.4264 -1.3427 -1.0354 -3.8551 1.4267 5.9543 4.4687 -2.0529 -1.0698 -3.9186 1.3382 5.9365 4.5109 . . . .

Fig. 6. In the left side, an example of environment file is showed, in the right side an example of configuration file is presented.

an *environment - file* information about the position and orientation for each object inside a workspace including the robot configuration. Hence, GEMPA can draw each element to simulate the workspace associated. Therefore, initially GEMPA can recover information about the workspace, an example of this file can be see in Figure 6 (left side), *environment - file* include initial and goal configuration for the robot, beside to include x,y,z parameter for position and α, β, γ parameters for orientation for every objects inside the wokspace. Along whit this *environment file*, a *configuration - file* can also be loaded to generate the corresponding animation of the free collision path. This *configuration - file* has the form presented in Figure 6 (right side). This file is conformed by n six-tuples $(x, x, y, \alpha, \beta, \gamma)$ to represent each configuration included in the free collision path.

Once GEMPA has recovered information about workspace and collision free path, the tool allows the user to display the animation on three different modes.

Mode 1: Animation painting all configurations.
This modality paint every configuration from the free collision path. Each configuration is painted without clear the environment, that is, although visualization seems confused, all configurations are painted at the same time. An example of

this mode is presented in Figure 7 (left-side).

Mode 2: Animation painting configurations using a step control.
In these option, each configuration is painted after than previous configuration has been deleted from the environment, avoiding two configuration can be overlapped during the animation. Besides, under this molality, each configuration is painted manually when de user request it.

Mode 3: Animation using automatic step.
In this case, an automatic animation is displayed, showing one configuration at time and using an automatic step. This mode is the most used when the user need to generate a video of the simulation. An example of this mode can be seen in Figure 7 (right-side).

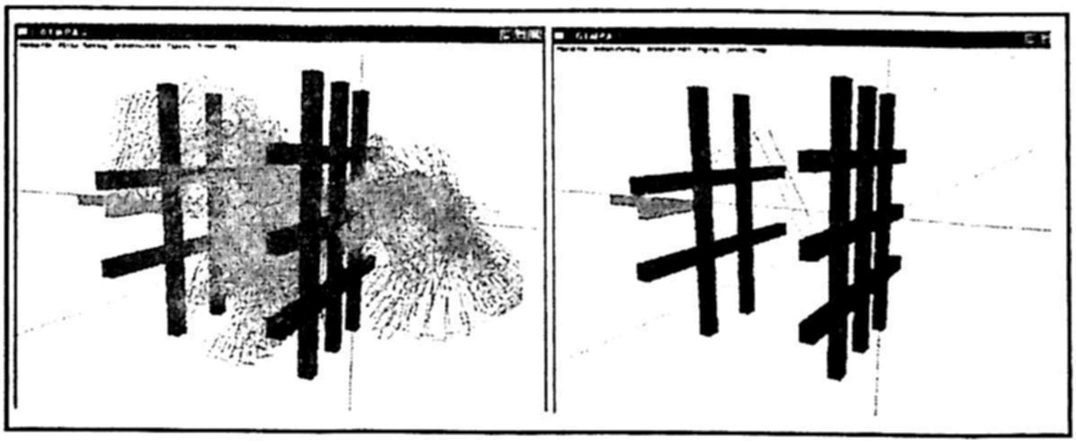

Fig. 7. Animation painting all configurations (left side), and animation using automatic step (right side) are displayed.

5 Applications

The proposed tool can be used to simulate and/or to implement later practical systems in different areas of computer science such as graphics, computational geometry and robotics. Four cases will be described as initial GEMPA applications.

Simple Figures and Geometric Transformations.
Initially, GEMPA has been an effective one in classroom teaching. It not only cuts down, significantly, on the instructor's time and effort but also motivates senior/graduate students to pursue work in this specific area of research. Important results has been reach to explain computer graphics topics, where geometric transformation are used to move, rotate or scale an object into a tree dimen-

sional environment. An example of this application can bee seen in Figure 4.

Simulation OFF Environments.
Important benchmark in motion planning are defined using *off - files*, hence the importance of managing and include this kind of files for objects representation. It is important to say that *off - files* not only allow represent a triangle meshes, this files also can use any polygon to represent an object. GEMPA provides the capability of building environments using these different polygons.
Figure 8 shows a combination of TXT and OFF files objects to build a three dimensional environment. **Simulation Motion Planning Algorithms.**

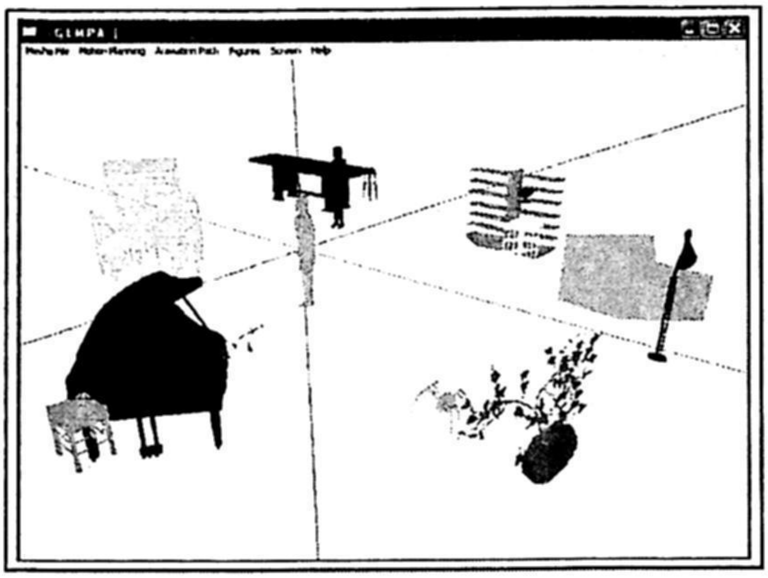

Fig. 8. Graphic environment built using OFF and TXT - Meshes Files.

A basic problem of PRM were described in section 4.2. Actually, GEMPA only can display animation for free flying objects problems defined on two-dimensional and three-dimensional spaces. Nevertheless, we are working to integrate different PRMs to GEMPA. In Figure 9 the free collision path to solve the problem is painted using the animation painting all configurations mode. In similar way, a 3-D sample of motion plannig problem is presented in Figure 10.

6 Conclusions and Work in Progress

GEMPA is a 3-D visualization tool for simulation environments and problems of PRMs (Probabilistic Roadmap Methods). This initial version of GEMPA was developed for visualizing and editing motion planning environments and animate collision free paths. GEMPA offers a nice and easy to use graphical user interface (GUI) that allows you to display workspace environments, path, and

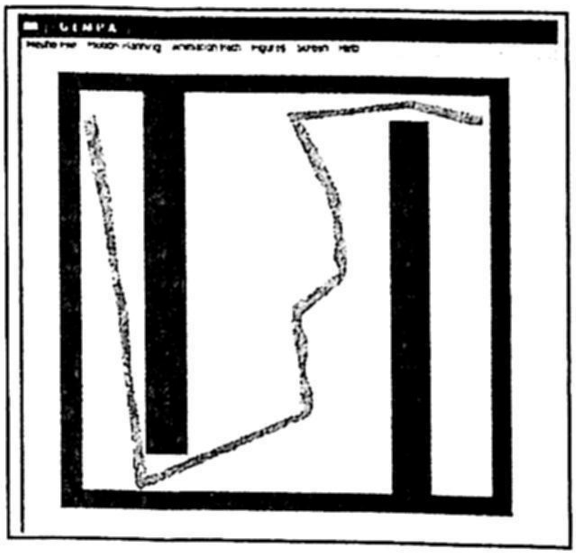

Fig. 9. A complete free collision path is painted as an example of two-dimensional motion planning problem.

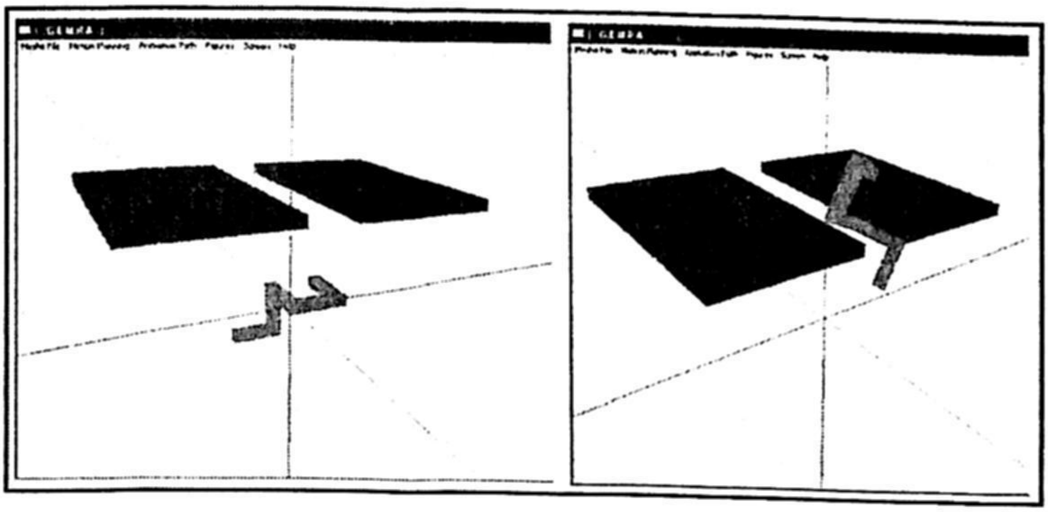

Fig. 10. 3-D Motion planning problem is displayed, where the collision free path is painted as an animation mode (one configuration at time).

start/goal positions. It enables users to interact with the environment. For example, it allows users manipulate objects configurations (or obstacles) and robot configurations, save new environments to be able to work on them later.

There many interesting a useful functionalities to be added to GEMPA. Now, this project is working on developed the follows goals.

Integration of PRMs to GEMPA. This will allows the user to generate solutions for motion planning problems selecting between at least two different techniques. Besides, GEMPA will be able to display the roadmap generated by the selected method. This functionality will enable users to run new queries.

Hence our tool will provides a convenient interface to select planners and set their parameters.

Simulation of Collision Detection Algorithms. Collision detection is a fundamental problem in robotics, computer animation, physically-based modeling, molecular modeling and computer-simulated environments. A realistic simulation system, which couples geometric modeling and physical prototyping, can provide a useful toolset for applications in robotics, CAD/CAM design, molecular modeling, manufacturing design simulations, etc. Due that motion planning algorithms used collision detection routines to solve the problems, GEMPA is developing a module capable to integrate a show the performance of collision detection algorithms.

Visualization and control for Kinematic Chains. Important application of robotics are related with articulated robots, these can be controlled using kinematic chains, applying forward and inverse kinematic to calculate them movements. Therefore, further GEMPA version is going to contemplate functionalities to manage this kind of robots, particularly forward and inverse kinematics for humanoids representation.

References

1. Jyh-Ming Lien Aime Vargas E. and Nancy M. Amato. Vizmo++: a visualization, authoring, and educational tool for motion planning. In *RIn Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 727–732, 2005.
2. L. Dale C. Jones and N. Amato, B. Bayazit and D. Vallejo. Obprm: An obstacle-based prm for 3d workspaces. In L. Kavraki In P. Agarwal and M. Mason, editors, *Robotics: The Algorithmic Perspective*, pages 156–168. AK Peters, 1997.
3. M. Saha F. Schwarzer and J.-C. Latombe. Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. In *IEEE Tr. on Robotics and Automation*, 2005.
4. L. Kavraki and J.C. Latombe. Randomized preprocessing of configuration space for path planning. In *IEEE Int. Conf. Robot. and Autom*, pages 2138–2139. xxx, 1994.
5. J.C. Latombe. *Robot Motion Planning*. Kluwer,Boston, MA, 1991.
6. Steven M. LaValle and James J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *International Workshop on Algorithmic Foundations of Robotics (WAFR).*, page xxxx, 2000.
7. R. Motwani L.E. Kavraki, J.C. Latombe and P. Raghavan. Randomized preprocessing of configuration space for path planning. In *Proc. ACM Symp. on Theory of Computing.*, pages 353–362, 1995.