?M4V7?%m%\%C%H$NJ*BNA':n$N$?$a$N%Q%9%W%i%s%K%s%0K!

Automating Object Manipulation Tasks for Humanoid Robots

!{ %+%U%J!<!&%8%'!<%'%: (El5~Bg3X) @5 0pMU 2m9, (El5~Bg3X)
@5 0f>e Gn0t (El5~Bg3X)

James KUFFNER. The Univ. of Tokyo. 7-3-1, Hongo, Bunkyo-ku, Tokyo.
Masayuki INABA. Dept. of Mechano-Informatics, The Univ. of Tokyo.
Hirochika INOUE. Dept. of Mechano-Informatics, The Univ. of Tokyo.

We present a manipulation path planner under development for providing high-level software control for humanoid robots. Using a graphical simulation environment, the software is designed to facilitate the automation of object grasping and manipulation tasks. Given a robot's internal model of the environment and a desired posture, we use a randomized path planner to search directly the configuration space of the robot for a collision-free manipulation path. The simulation environment can be used as a graphical user interface for directly controlling or interacting with a robot operating in the real world. We view this as a potential useful tool for the visualization and development of complex robotic systems, as well as an interactive, task-level programming interface.

Key Words : Humanoid Robot, Object Manipulation, Path Planning, Randomized Algorithm

1 Introduction

Research involving humanoid robots has increased during recent years. Advances in computing hardware and software have enabled the implementation of sophisticated motion control strategies. In particular, dynamic simulation software has assisted in the realization of dynamic walking in several humanoid robots. As the technology and algorithms for real-time 3D vision and tactile sensing improve, humanoid robots will be able to perform tasks that involve complex interactions with the environment (e.g. grasping and manipulating objects). The enabling software for such tasks includes motion planning for obstacle avoidance, and integrating planning with visual and tactile sensing data.

To facilitate the deployment of such software, we are currently developing a graphical simulation environment for the H5 dynamic humanoid robot. The project builds upon a software framework that was originally developed for the high-level control of computer animated characters in 3D virtual environments[1]. The software automatically computes object grasping and manipulation trajectories through a combination of inverse kinematics and randomized path planning. Feasible manipulation trajectories can be computed at interactive rates. We hope that through the use of such interactive simulation software, the current and future capabilities of humanoid and other complex robotic systems can be improved.

2 Robot Model and Assumptions

We have based our simulation experiments on an approximate model of the H5 dynamic humanoid robot, including the kinematics and link geometry (see Figure 1). Although we have previously used a model of the link dynamics to generate stepping motions [2], the current manipulation planner computes a strictly kinematic path. We also make the following assumptions:

1. We assume that the robot has acquired a 3D model of the surrounding environment using stereo vision, and has localized itself. The model need not be exact, provided that a reasonable model of sensing error is taken into account by the planner, and that it is possible to obtain visual or tactile feedback while executing the manipulation motion in order to adjust for errors.
2. The robot is given a high-level task command such as *"move the book on the table"*. The task command is provided explicitly by a human operator (e.g. voice commands), or issued by top-level behavior software.
3. For these experiments, we have focused on tasks that can be accomplished by a single arm, though the planner could potentially be extended to handle tasks which require both arms.
4. We assume that an inverse kinematics algorithm is available for computing body configurations from world space frames. Our current inverse kinematics algorithm uses only the six arm joints, thus the shoulder remains fixed during the motion, but algorithms which consider the entire body of the humanoid can also be used with the planner.

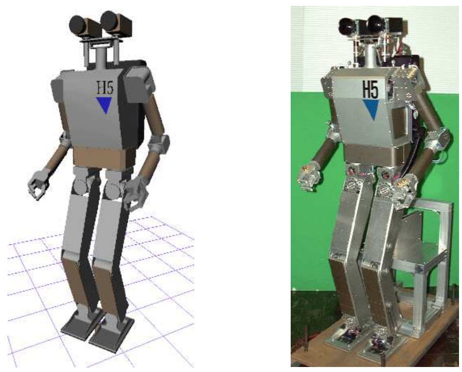

Fig. 1: The virtual and real dynamic humanoid 'H5'.

3 Manipulation Planning

3.1 Task Specification The manipulation task is specified by identifying a target object to be moved and its new location. The motion planning software will then attempt to compute three trajectories:

- *Reach*: Move the robot in position to grasp the object.

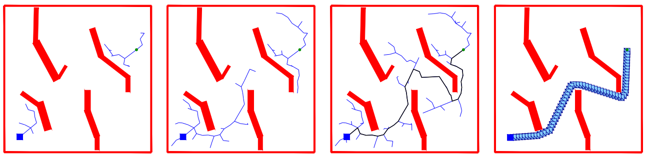

Fig. 2: RRT-Connect builds a connection between two trees.

- *Transfer*: After grasping, move the object to the target location
- *Return*: Once the object has been placed at the target location, release it and return the robot to its rest position.

If a path at each phase is successfully returned by the path planner, the robot executes the motion, possibly guided by visual or force feedback.

3.2 Path Planning Path planning problems involve searching a system configuration space for a collision-free path connecting a given start and goal configuration[3]. For problems in low dimensions, the configuration space can often be explicitly represented and exact algorithms can be used effectively. For high-dimensional configuration spaces however (such as 6 DOF robot arms), it is typically impractical to explicitly represent the configuration space. Instead, the space is *sampled* in order to discover free configurations (for example, see [4]). Here, the fundamental challenge lies in devising a practical and efficient sampling strategy.

We have built our simulations upon a randomized path planner with a sampling heuristic, known as *RRT-Connect*, that has been specifically designed for computing collision-free manipulation motions [5]. The sampling heuristic is based on Rapidly-Exploring Random Trees (RRTs)[6], and is optimized to quickly handle single-query path planning problems without any preprocessing of the configuration space. The method exhibits rapid convergence for simple spaces and relative immunity to pathological cases. The planner incrementally build two trees of free configurations in a way such that the expansion of the trees is heavily biased towards the unexplored regions of the space. Due to the way that RRTs are constructed, the distribution of samples has been shown to ultimately converge toward a uniform distribution over the free configuration space [5]. Snapshots of the RRT-Connect planner solving a simple 2D problem is illustrated in Figure 2.

4 Experiments

We have implemented a prototype manipulation planner that runs within a graphical simulation environment. By using an efficient inverse kinematics algorithm, the software facilitates an interactive task-level control mechanism for planning manipulation motions. Through a graphical user interface, an operator can click and drag an object to a target location and issue a *move* command. The software will then automatically compute the motions necessary to complete the task.

Figure 3 shows the humanoid robot H5 playing chess. Each of the motions to reach, grasp, and reposition a game piece on the virtual chessboard were generated using the planner in less than one second on a 270MHz SGI O2 workstation. Each arm consists of a 6-DOF kinematic chain, and the entire scene contains over 10,000 triangle primitives. The 3D collision checking software used for these experiments was the RAPID library based on OBB-Trees developed by the University of North Carolina[7]. Figure 4 shows the planner being used in combination with a model of the robot's visual sensing.

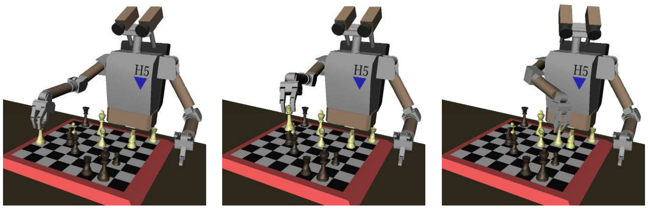

Fig. 3: The H5 humanoid robot playing chess.

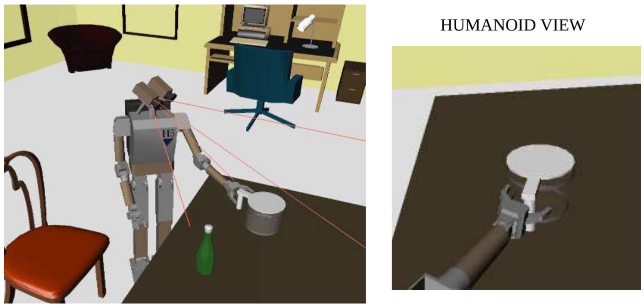

Fig. 4: Reaching and grasping a coffee pot.

5 Discussion

This paper presents an experimental interactive path planner for automatically generating manipulation trajectories for humanoid robots. There are many potential uses for such software, with the primary one being a high-level control interface for solving complex object manipulation tasks. By using a graphical simulation environment, sophisticated motion generation algorithms can be efficiently developed and debugged, reducing the costs and safety risks involved in testing software for humanoid robots.

Future work includes extending the planner to handle multi-arm tasks, taking into account the arm and object dynamics, and integrating visual and tactile feedback into the planning process.

Acknowledgments

This research is supported in part by a Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Foreign Scholars in Science and Engineering.

References

1) J. J. Kuffner Jr. *Autonomous Agents for Real-time Animation.* PhD thesis, Stanford University, 1999.

2) K. Nagasaka, M. Inaba, and H. Inoue. Walking pattern generation for a humanoid robot based on optimal gradient method. In *In Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

3) J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

4) L. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics and Automation*, Vol. 12, No. 4, pp. 566–580, 1996.

5) J.J. Kuffner and S.M. LaValle. RRT-Connect: An efficient approach to single-query path planning. In *To appear in Proc. of the IEEE International Conf. on Robotics and Automation (ICRA'2000)*, San Francisco, CA, April 2000.

6) S.M. LaValle and J.J Kuffner. Randomized kinodynamic planning. In *Proc. of the IEEE International Conf. on Robotics and Automation (ICRA'99)*, Detroit, MI, May 1999.

7) S. Gottschalk, M. C. Lin, and D. Manocha. OBBTREE: A hierarchical structure for rapid interference detection. In *SIGGRAPH '96 Proc.*, 1996.