Interactive Manipulation Planning for Animated Characters

James Kuffner, Jr. Jean-Claude Latombe
Department of Computer Science
Stanford University, Stanford, CA 94305 USA
{kuffner, latombe}@cs.stanford.edu
http://robotics.stanford.edu/~kuffner/anim/

Abstract

We present a brief overview of an algorithm for interactively animating object grasping and manipulation tasks for human figures. The technique is designed to efficiently generate feasible single-arm manipulation motions given high-level task commands. For moving an object, the motions necessary for a human arm to reach and grasp the object, reposition it, and return the arm to rest are generated automatically within a few seconds on average.

The method synthesizes motion "on-the-fly" by directly searching the configuration space of the arm. Goal configurations for the arm are computed using an inverse kinematics algorithm that attempts to select a natural posture. A collision-free trajectory connecting the arm initial configuration to the goal configuration is computed using a randomized path planner. A high-level description of the methods is given along with results from some computed examples using a human character model.

1. Introduction

Object manipulation is an important class of motions for interactive animated characters. Manipulation tasks can encompass a virtually unlimited combination of object and obstacle geometries. Thus, it seems unlikely that one would be able to successfully enumerate all possibilities and simply store thousands of pre-recorded motion sequences. Instead, a flexible strategy that can accommodate a wide range of situations is needed.

This paper briefly describes a motion synthesis strategy that combines inverse kinematics and path planning to interactively animate *reaching and object manipulation tasks*. The human arm is modeled as a kinematic chain with seven degrees of freedom, and motion trajectories are computed directly within the configuration space. Because of high-dimensionality, the space cannot be explicitly represented. Instead, the space is sampled using a randomized planner that has been specifically tailored to quickly solving common planning queries involving human arms.

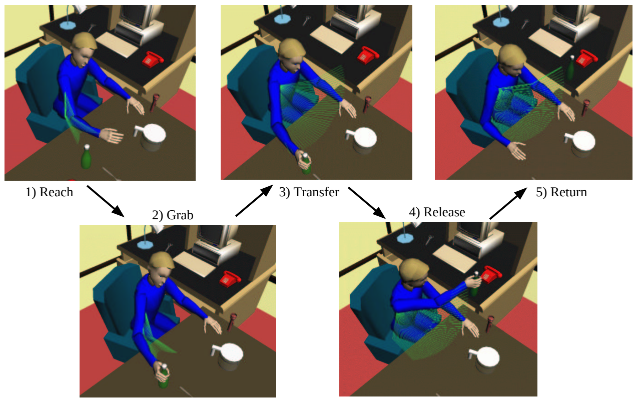


Figure 1. Repositioning a bottle.

For testing and evaluation purposes, we have designed an interactive application involving human characters that can autonomously manipulate objects. The software is able to generate collision-free motions for typical single-arm manipulation tasks at interactive rates. Though many improvements can be made, the generated animation looks fairly realistic.

2. Synthesizing Manipulation Motions

Unlike motion capture editing methods which require an existing motion sequence as input[3, 9], our manipulation planning algorithm synthesizes motion from scratch. We previously presented a path planner suitable for computing dual-arm manipulation motions offline[6]. The single-arm planner described here is fast enough for interactive applications, where manipulation motions for human figures must be computed "on-the-fly" from high-level task commands.

Given a task command to reposition an object in the environment, the planner will attempt to compute three trajectories (Figure 1): 1)*Reach*: Move the arm to grasp the object. 2)*Transfer*: After grasping, transfer the object to the target location. 3)*Return*: Once the object has been placed at the target location, release it and return the arm to its rest position. An inverse kinematics algorithm for the arm is used to compute the goal configurations for the Reach and Transfer tasks. The path planner searches for a collision-free trajectory connecting the initial and the goal configurations.

Inverse Kinematics: We use an inverse kinematics algorithm based on neurophysiological data to resolve the redundancy in a 7DOF model of the human arm[6]. The current implementation uses only the arm joints, but the torso and hip joints could also be used if an appropriate inverse kinematics algorithm is available.

Path Planning: For path planning, we used a monte carlo search algorithm based on rapidly-exploring random trees (RRTs) in the configuration space[7]. This algorithm was selected for its speed in solving single-query path planning problems, particularly in character animation[8]. However, other successful single-query path planning techniques could potentially be used[2, 5].

3. Experiments

Figure 2 shows some example manipulation tasks solved by the planner. All motions were computed in 1 to 3 seconds on average on a 270 MHz SGI O2 running Irix 6.5 (see table below). The human arm is modeled as a 7-DOF kinematic chain, and each scene contains over 10,000 triangle primitives. The 3D collision checking software used was the RAPID library based on OBB-Trees developed by the University of North Carolina[4].

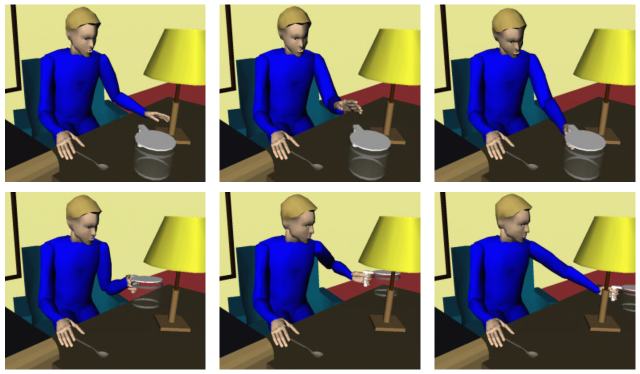
Task Description	Computation Time (seconds)			
	min	max	avg	stdev
Reposition coffeepot	0.44	3.12	1.43	0.77
Move chess piece	0.17	1.84	0.78	0.48
Reach and move hammer	0.92	6.88	2.79	1.74

4. Summary

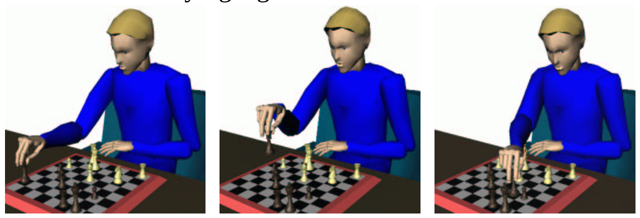
For planning single-arm manipulation tasks for human characters, we have proposed a motion generation strategy that relies primarily on inverse kinematics and path planning software. Our algorithm synthesizes from scratch realistic-looking, relatively complex manipulation motions for human figures at interactive rates. The planner has the desirable characteristics of requiring no pre-processing, no “magic numbers” to tweak, and has been shown to converge towards a uniform exploration of the configuration space[7]. Future work includes extending the planner to handle dual-arm manipulation tasks, and incorporating additional degrees of freedom.

Acknowledgments: This research was supported in part by a National Science Foundation Graduate Fellowship in Engineering, and MURI grant DAAH04-96-1-007 (Army). We also thank Steve LaValle and David Hsu for many useful discussions regarding path planning.

Moving a coffee pot around a lamp obstacle.



Playing a game of virtual chess.



Withdrawing a hammer from within a box.

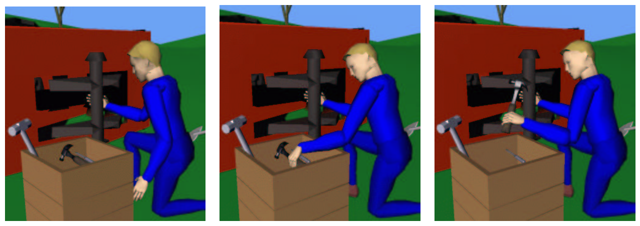


Figure 2. Some computed examples.

References

[1] S. Bandi. *Discrete Object Space Methods for Computer Animation*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1998.

[2] R. Bohlin and L. Kavraki. Path planning using lazy PRM. In *In Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA’2000)*, Apr. 2000.

[3] M. Gleicher. Retargetting motion to new characters. In *Proc. SIGGRAPH ’98*, 1998.

[4] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTREE: A hierarchical structure for rapid interference detection. In *Proc. SIGGRAPH ’96*, 1996.

[5] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. of Computational Geometry and Applications*, 9(4-5):495–512, 1997.

[6] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe. Planning motions with intentions. In *Proc. SIGGRAPH ’94*, 1994.

[7] J. Kuffner and S. LaValle. RRT-Connect: An efficient approach to single-query path planning. In *In Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA’2000)*, Apr. 2000.

[8] J. Kuffner Jr. *Autonomous Agents for Real-time Animation*. PhD thesis, Stanford University, 1999.

[9] A. Witkin and Z. Popovic. Motion warping. In *Proc. SIGGRAPH ’95*, 1995.