Bridging the Gap between Geometric and Task Level Manipulation Planning

P. Jiménez

Institut de Robòtica i Informàtica Industrial (CSIC - UPC)
Llorens i Artigas 4-6, E-08028 Barcelona, Spain
e-mail: pjimenez@iri.upc.edu

1 Introduction

In manipulation planning works like [Siméon et al., 2000], although two levels have been distinguished, calling "task level" to the higher one, actually everything has been formulated in geometric terms. However, when properly talking about task planning, an abstract symbolic formulation is generally meant. And complex manipulation tasks may require some kind of symbolic structuring in order to bound the huge search space at geometric level.

2 Rule-based approaches

Manipulation planning is viewed in [Gravot et al., 2002] as a geometrical and task (planning) problem. Consequently, a link between the two levels has to be established. This is done, in first instance, through a collection of heuristic rules that constitute "the first level of symbolic control of the purely geometrical PRMs search methods" [Gravot et al., 2002]. In this work, several roadmaps are maintained simultaneously, each corresponding to a particular aspect of the problem (i.e., allowed placements and movements of the object, transit and transfer paths, gripper/object grasp transforms, etc). Basic notions are the "robot composition" (two robots, or a robot and an object, are considered as a single robot, and have an associated roadmap), as well as the aforementioned *reduction property*. The heuristic rules are used to determine which roadmap to expand (that is, where new nodes have to be randomly generated) or to search next.

This approach has evolved towards the *aSyMov* system [Gravot et al., 2005]. It is basically a planner that selects one of the applicable actions at the symbolic level (basic actions include *goto*, *pick*, and *place*) or an roadmap-expansion action, depending on computed costs and heuristics. This selection is made on a probabilistic basis, i.e., not always the best action is selected. Actions are built from predicates stating preconditions and effects of these actions (basic predicates include statements about the composition of two robots or a robot and an object, or the location of a robot at a symbolic position, for example). The connection to the geometrical part of the planning occurs at the *action validation step*, an "incremental instantiation process of the symbolic positions" (where geometrical constraints may back-propagate on the whole plan under construction, not just apply on the action being validated). The different possible instances (nodes in the roadmap) are kept in *accessibility lists*, which are revisited in case of back-propagation due to invalid instances of selected actions.

3 Object-centered approaches

Instead of a set of $\mathcal{C}$-spaces where connected collision-free paths have to be found, the lower planning level may be a battery of motion directives implemented in agent-specific behaviors (the broader term "agent" is used here instead of -and subsuming- "robot", as many contributions come from the Computer Graphics field, specifically from Virtual Human simulation). Nonetheless, such elemental procedures are still too specific as to be efficiently used at task-level. Thus, an intermediate system like the *Object Specific Reasoner* [Levison, 1996, Douville et al., 1996] is necessary for linking the high-level domain-independent task-actions to the completely specified motion directives from the physical (or simulation) world. This system has been developed from the observation that the same task-action, say "pickup", may produce quite different physical actions, graspings in this case, when applied to different objects (or to the same object with different purposes). Objects are grouped into functional categories, which trigger a specific *action outline* for each task-action. All steps of this outline are recursively expanded, until all sub-task-actions are decomposed into motion directives. Possible parameter values are generated, using specific attributes of the agent and the object, and sorted according to the purpose of the action. If the agent's resources are consistent with the object's attributes, the task-action is feasible in the current setting and the instantiated motion directives are transferred to the agent.

The *Smart Objects* in [Kallmann, 2001, Kallmann and Thalmann, 1999, Abaci et al., 2005] provide a similar kind of object-centered manipulation paradigm. The geometrical information describing the shapes of objects is extended with semantic information about their behavior when an interaction with an agent occurs. This information includes whole manipulation sequences, which are not limited to grasping, but also consider reaching the objects, looking at them, changing grasps, etc. Grasps are generated semi-automatically in the design phase of the object (in Robotics they could be learned as well), by allowing the designer to select relevant tubular regions for grasping and to specify parameters like wrist position and orientation, thumb configuration, touch tolerance or finger involved in the grasp, while leaving the low-level computations (collision detection and posture search in the configuration space of the hand) to the system. In [Gonçalves et al., 2001] the system is completed with local perception capabilities that allow the agent to classify the object from its perceived feature set, and to retrieve the actions attached to the corresponding smart object.

4 Language specification

The *Task Definition Language* [Vosinakis and Panayiotopoulos, 2003] does also aim at bridging the gap between high-level task specifications and the agent's interaction with the virtual environment. The language allows to combine a number of built-in *primitive actions* -like changing the configuration of entities- in a sequential, parallel, or conditional fashion to generate complex tasks. In particular, regarding interactions with objects, grasping can be implemented as a parallel combination of *inverse kinematic* functions moving the fingers concurrently towards destination spots on the object (or rotating the fingers until collision with the object's surface). The final step would be to have the possibility of giving high level commands in natural language, like the *Parameterized Action Representation* [Badler et al., 1998, Badler et al., 2000]. Although addressed at communicating with virtual humans, i.e., smart avatars in virtual environments, this PAR-based architecture could be inspiring for designing a communication interface with smart embodied agents like robots intended to carry out manipulation tasks. This system is designed to provide a complete description of an action, including the agent that performs it and the list of affected objects, the applicability conditions that must hold, a list of conditional preparatory specifications, the execution steps and the termination conditions.

References

Abaci et al., 2005. Abaci, F., Mortara, M., Patane, G., Spagnulo, M., Vexo, F., and Thalmann, D. (2005). Bridging geometry and semantics for object manipulation and grasping. In *Proceedings of Workshop towards Semantic Virtual Environments (SVE 2005)*, pages 110–119.

Badler et al., 2000. Badler, N., Bindiganavale, R., Allbeck, J., Schuler, W., Zhao, L., Lee, S.-J., Shin, H., and Palmer, M. (2000). Parameterized action representation and natural language instructions for dynamic behavior modification of embodied agents. In *AAAI Spring Symposium*.

Badler et al., 1998. Badler, N., Bindiganavale, R., Bourne, J., Palmer, M., Shi, J., and Schuler, W. (1998). A parameterized action representation for virtual human agents. In *Workshop on Embodied Conversational Characters*.

Douville et al., 1996. Douville, B., Levison, L., and Badler, N. I. (1996). Task-level object grasping for simulated agents. *Presence*, 5(4):416–430.

Gonçalves et al., 2001. Gonçalves, L. M. G., Kallmann, M., and Thalmann, D. (2001). Programming behaviors with local perception and smart objects: An approach to solve autonomous agents tasks. In *Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'01)*, pages 184–191, Washington, DC, USA. IEEE Computer Society.

Gravot et al., 2002. Gravot, F., Alami, R., and Siméon, T. (2002). Playing with several roadmaps to solve manipulation problems. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2311–2316, Natick, MA, USA.

Gravot et al., 2005. Gravot, F., Alami, R., and Siméon, T. (2005). *Robotics Research*, volume 15/2005 of *Springer Tracts in Advanced Robotics*, chapter aSyMov:A Planner That Deals with Intricate Symbolic and Geometric Problems, pages 100–110. Springer Berlin / Heidelberg.

Kallmann, 2001. Kallmann, M. (2001). *Object Interaction in Real-Time Virtual Environments*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland.

Kallmann and Thalmann, 1999. Kallmann, M. and Thalmann, D. (1999). Direct 3d interaction with smart objects. In *ACM Symp. on Virtual Reality, Software and Technology*, pages 124–129.

Levison, 1996. Levison, L. (1996). *Connecting planning and acting via object-specific reasoning*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

Siméon et al., 2000. Siméon, T., Laumond, J.-P., and Nissoux, C. (2000). Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, 14(6):477–493.

Vosinakis and Panayiotopoulos, 2003. Vosinakis, S. and Panayiotopoulos, T. (2003). A task definition language for virtual agents. *Journal of WSCG*, 11(1):512–519.