Planning for Robust Execution of Humanoid Motions using Future Perceptive Capability

Philipp Michel[†], Christian Scheurer[†‡], James Kuffner[†], Nikolaus Vahrenkamp[‡] and Rüdiger Dillmann[‡]

[†]The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
{pmichel,chrsch,kuffner}@cs.cmu.edu

[‡]Institute of Computer Science and Engineering
University of Karlsruhe
Haid-und-Neu Str. 7
76131 Karlsruhe, Germany
{scheurer,vahrenkamp,dillmann}@ira.uka.de

***Abstract*— We present an approach to motion planning for highly articulated systems that aims to ensure robust execution by augmenting the planning process to reason about the robot's ability to successfully perceive its environment during operation. By simulating the robot's perception system during search, our planner generates a metric, the so-called perceptive capability, that quantifies the 'sensability' of the environment in each state given the task to be accomplished. We have applied our method to the problem of planning robust autonomous manipulations as performed by a humanoid robot in a kitchen environment. Our results indicate that reasoning about the future perceptive capability has the potential to greatly facilitate any task requiring visual feedback during control of the robot manipulator and can thus ensure higher task success rates than perception-unaware planning.**

I. Introduction

As complex robotic systems expand their application domain from traditional, highly constrained manufacturing settings to increasingly free-form office and home environments, the task of autonomous manipulation becomes much more challenging. Unlike a factory, a typical home is not easily purpose-built or augmented to facilitate specific robot tasks. For the case of a humanoid robot manipulating in a kitchen environment, for example, it is unreasonable to assume knowledge of the precise positioning of all objects of interest to the robot manipulator. Instead, sensing is used to localize objects to be manipulated and to guide the positioning of the robot manipulator to ensure a successful grasp of the object.

Traditionally, perceptual input has been used in performing grasping manipulations to initially localize the object to be grasped and to recover its position, with no further visual information being taken into account during the subsequent planning and execution stages. The successful execution of the resulting open-loop motions tends to be highly susceptible to errors in both the initial localization and the control of the manipulator. A typically more successful approach is to plan first a coarse reaching motion that positions the end effector close to the object. Then, a stage of fine-grained, closed-loop control is entered that incorporates visual feedback to servo the end-effector to a precise position from which the object can be grasped [1].

However, the initial planning stage is almost always oblivious to the need for subsequent visual feedback to succeed in performing the grasping task. It is not uncommon for the planner to generate solutions that preclude the object from being successfully perceived when the visual servo stage is entered due to environmental or self-occlusions along parts or at the end of the reach motion. For simple robotic systems, it may be possible to circumvent this problem by analytically calculating visual constraints that the resulting motion plan must satisfy [2]. For complex, high-DOF systems such as humanoids, this quickly becomes unwieldy.

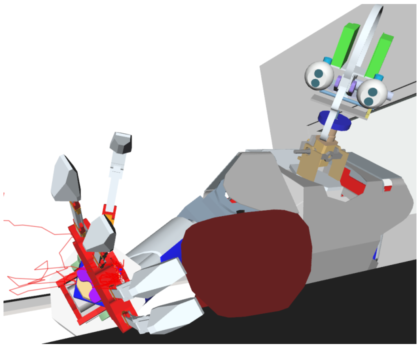


Fig. 1. The ARMAR-III humanoid executing a reach motion trajectory that maximizes the robot's perceptive capability.

This paper presents an argument for extending the planning stage to incorporate reasoning about the future ability to successfully perceive the robot environment during execution in accordance with the task to be achieved. We present the problem of robust autonomous manipulation planning and related work in Section II. Section III provides the algorithmic foundations of our adapted planner. In Section IV, we detail our implementation on the ARMAR-III humanoid robot platform and present simulation results for a series of grasping experiments in increasingly complex scenarios. We summarize in Section V, discuss current limitations and give an outlook of future work.

II. Robust Autonomous Manipulation Planning

A. Overview

For humanoid and mobile robots to operate in increasingly unconstrained and dynamic environments in a truly useful manner, robot tasks such as navigation and manipulation need to be accomplished autonomously, obviating preprogrammed trajectories and necessitating general approaches

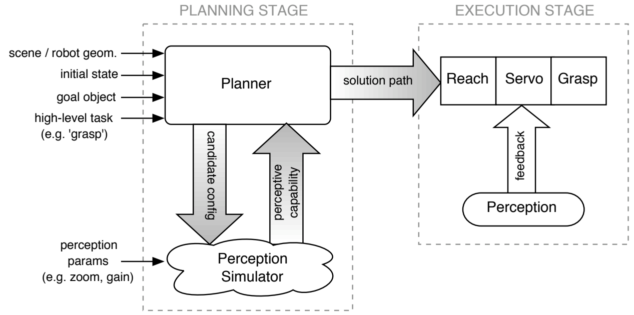


Fig. 2. Reasoning about perception in the planning stage. A simulation of the perception process is used to compute a metric assessing perceptive capability for each configuration examined during planning.

to planning motions [3]–[5]. Perhaps more importantly, tasks need to be accomplished robustly in the face of sensing uncertainty, modeling or execution error and world-induced task failures. Perception has played a key role in reducing errors, especially in the execution stages, by providing validation against the real environment.

We posit that for autonomous manipulation planning to become truly robust, it is necessary to reason about perception additionally as an integral part of the *planning* stage, concurrently with planning the actual motions to be performed. For the purposes of this paper, we focus specifically on the *reaching* subtask, computing a manipulator trajectory from an initial configuration to a goal configuration in which the end-effector is positioned to grasp an object under visual servo control. We augment a sampling-based planner with a simulation of the robot's perception process that, during planning, assesses each candidate robot configuration regarding the 'sensability' of the object to be manipulated, as Figure 2 conceptualizes. Our hypothesis is that the resulting plans maximize the likelihood of successful execution by ensuring that sensor information can be gathered at the crucial stages during execution.

B. Problem Placement

Our approach relates to the issues of sensor planning [6] and the next-best-view problem [7], which are often considered independently from the problem of planning the actual robot motions themselves.

We also take inspiration from the SLAM community, where explicit reasoning about perception during navigation (e.g. in the form of information gain [8]) has resulted in significant progress in robot autonomy.

The mobile and multi-robot community has studied the problem of planning with constraints on visibility in a variety of settings, e.g. to maintain line-of-sight communication constraints between a team of exploring robots [9] or to ensure visibility of a moving target [10]. However, scenarios considered are often amenable to purely geometric reasoning about visibility and make application to high-DOF systems challenging.

Most related to our approach is work that explicitly considers visual servo performance during the planning process by augmenting the robot configuration with image-features to arrive at the so-called perceptual control manifold [11]. A static, external camera is assumed, as is the ability to analytically calculate the presence of image features in a particular configuration using forward kinematics.

Work on visual-servo control [12], [13] suggests that errors in the early sensing stages (due to occlusions, calibration error, low resolution, distance from the object of interest, etc.) have a significant effect on the overall performance of robot motions controlled by visual feedback and that sensing should hence be reasoned about throughout the planning process as well.

C. Problem Formulation

The problem of planning reaching motions is formulated as follows. We seek to find a collision-free motion given by a time-indexed path of the manipulator

$$\tau : t \in [0, T] \mapsto \tau(t) \in \mathcal{C}_{free}$$

from an initial robot configuration $\tau(0) = q_{init}$ to one of a set of valid goal states $\tau(T) = q_{goal} \in G$ from which an appropriate grasp motion can then be executed. Typically, a wide range of solutions $\tau \in \Phi$ to this underconstrained problem exists, with the best solution τ_{best} usually selected to optimize a specific metric $\Gamma(\tau)$ (e.g. shortest distance, manipulability, etc.) over the path as $\tau_{best} = \mathrm{argmin}_{\tau}\, \Gamma(\tau), \tau \in \Phi$.

This basic formulation can be extended to also optimize sensor placement along the trajectory by ensuring that each state $q = \tau(t)$ satisfies some geometrically defined binary visibility constraint $q \in \mathcal{C}_{vis}$ (or perhaps only $q_{goal} \in \mathcal{C}_{vis}$ in case visual servoing only takes place in the grasp phase), ensuring a clear line-of-sight between sensor and object. Several issues arise: 1) For a high-DOF system in a cluttered environment and able to occlude itself, determining analytically whether the visual constraint is satisfied might not be feasible. 2) Optimizing sensor placement in a complex system such as a humanoid may force the use of costly full-body motion planning rather than planning for the arm motion alone. 3) How well the perception system can operate in each state may not be adequately captured by a simple line-of-sight constraint (consider e.g. the need for visible texture in a stereo system).

To improve upon this simple reasoning about geometric visual constraints, we introduce a cost metric $\varphi_{PC} : q \in \mathcal{C} \mapsto \varphi_{PC}(q) \in \Re$ that encodes the robot's *perceptive capability* in configuration q. The calculation of φ_{PC} may amount to anything from a simple line-of-sight constraint to ensure proper sensor placement to a full statistical simulation of the sensor's response given the state of the robot and world. $\varphi_{PC}(\tau(t))$ thus effectively encodes how well the perception system is expected to perform at a certain stage along a planned motion trajectory given the sensing requirements of the task at hand. The planner now seeks to find an optimal reach trajectory that also affords the robot maximal future perceptive capability when executed, as given by

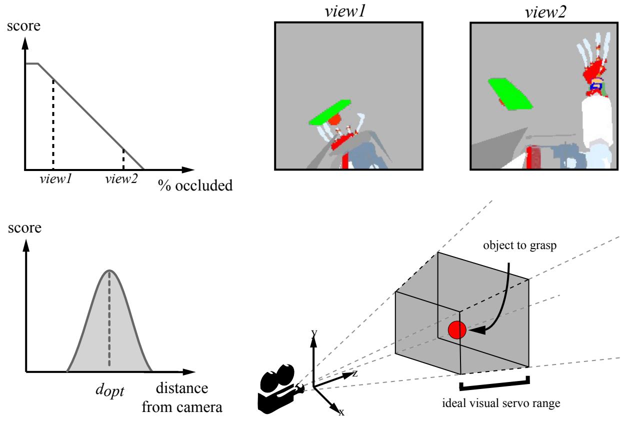


Fig. 3. The perceptive capability of a configuration is determined by the fraction of the object (here: a red sphere) visible (*top*) as well as the location of the object relative to the optimal visual servo range (*bottom*).

$$\tau_{\text{FPC}} = \operatorname*{argmax}_{\tau} \sum_{t=0}^{T} \varphi_{PC}(\tau(t)), \tau \in \Phi$$

III. Planning Reaching Motions

A. Perceptive Capability from Simulation

Due to the high dimensionality and complexity of finding truly optimal reaching trajectories that also maximize the perceptive capability, we employ an efficient sampling-based planner built on the Rapidly-Exploring Random Trees (RRT) [14] framework with guiding heuristics that seek to encode our ultimate goal of increased robustness during full pick-up manipulations (reach-servo-grasp) executed by our physical humanoid robot ARMAR-III. Most of the task fragility and failure likelihood lies in the servoing stage. In accordance with our hypothesis we therefore seek to ensure proper operation of the visual servo system throughout the manipulation to maximize our chance of successful execution. We heuristically define the perceptive capability $h_{PC}(q)$ of a particular configuration to reward an unoccluded view of an object lying inside an optimal operating distance range from the sensor. Figure 3 illustrates.

For every candidate configuration q examined during planning, $h_{PC}(q)$ is evaluated by simulating the robot's perception system. The resulting camera image is then compared to a pre-computed appearance-based template showing the object rendered from the current configuration, but unoccluded. A simple pixel count then allows us to quantify the fraction of the object occluded by scene geometry or the robot itself. More complex methods employing border or texture continuity could also be used to assess occlusion. Further taking into account the optimal servo distance allows us to establish the heuristic perceptive capability $h_{PC} \in [0, 1]$ in configuration q as the weighted sum

$$h_{PC}(q) = w_o p_o(q) + w_{dist} e^{-\frac{1}{2}(d(q)-d_{opt})^2} \qquad \text{with}$$

$$p_o(q) = \frac{\text{\# object pixels visible in } q}{\text{\# template pixels in } q}$$

```
Procedure BUILDRRT($q_{init}$)
Plans a reach motion optimizing the perceptive capability along the trajectory

Data: $T$: search tree
      $p_{bias}$: probability of extending RRT from highest-ranked node
      $q_{init}$: initial state

$T$.init($q_{init}$);
while (goal not reached and search time / space remaining) do
    $q_{rand}$ = SAMPLESTATE();
    if RAND$(0, 1) \leq p_{bias}$ then
        EXTENDHEURISTIC($T$, $q_{rand}$);
    else
        EXTEND($T$, $q_{rand}$);
    end
end
return $T$;
```

```
Procedure EXTEND($T$, $q_{rand}$)
Extends RRT in the direction of configuration $q_{rand}$, checking for collision and minimum perceptive capability

Data: $T$: search tree
      $q_{rand}$: configuration to extend towards
      ranking: priority queue of configurations, sorted by cost $l(q)$

$q_{near} \leftarrow$ NEARESTNEIGHBOR($T, q_{rand}$);
$q_{new} \leftarrow$ NEWCONFIG($q_{rand}, q_{near}$);
if COLLISIONFREE($q_{near}, q_{new}$) and MINPC($q_{near}, q_{new}$) then
    $T$.ADDNODE($q_{new}$);
    ranking.UPDATE($q_{near}, q_{new}$);
    if $q_{new} = q_{rand}$ then
        return REACHED;
    else
        return ADVANCED;
    end
end
return TRAPPED;
```

B. Planning Algorithm

In addition to generating the heuristic perceptive capability of each node examined, our planner measures progress towards the goal via a heuristic workspace goal function $h_{WS}(q) \mapsto \Re$ assessing the proximity of the end-effector pose to the goal [15]. This includes a term yielding Euclidean distance from the tool center point of the manipulator H to the center of the object G and a term rewarding alignment of the x axis of the end-effector coordinate system with the center of the target object as

$$h_{WS} = w_{dist}||G - H|| + w_{orient}|(G - H) \cdot x_H - 1|$$

The overall cost metric the planner uses to evaluate each node is then given by

$$l(q) = w_{PC}(1 - h_{PC}) + w_{WS}\, h_{WS}$$

and is used to sort nodes into a ranked priority queue during search. Procedures BUILDRRT, EXTEND and EXTENDHEURISTIC outline our RRT-based planner, which generates geometrically valid, collision-free paths that also guarantee a minimum perceptive capability along the trajectory. It has the following key properties:

- No explicit goal configuration is supplied to the planner. Instead, the goal is declared reached if a configuration

Procedure ExtendHeuristic(T, q_{rand})

Extends RRT in the direction of configuration q_{rand} *from the highest ranked node in* T, checking for collision and minimum perceptive capability

```
Data: $T$: search tree
      $q_{rand}$: configuration to extend towards
      ranking: priority queue of configurations, sorted by cost $l(q)$

$q_{best} \leftarrow$ ranking.FRONT();
$q_{new} \leftarrow$ NEWCONFIG($q_{rand}, q_{best}$);
if COLLISIONFREE($q_{best}, q_{new}$) and MINPC($q_{best}, q_{new}$) then
    $T$.ADDNODE($q_{new}$);
    ranking.UPDATE($q_{best}, q_{new}$);
    if $q_{new} = q_{rand}$ then
        return REACHED;
    else
        return ADVANCED;
    end
end
ranking.PENALIZE($q_{best}$);
return TRAPPED;
```

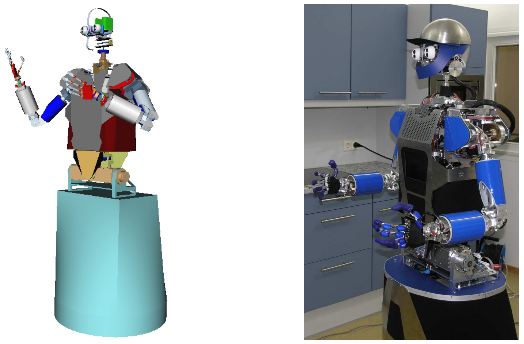
q generated during search satisfies a simple workspace goal proximity criterion on $h_{WS}(q)$.

- We define a lower threshold on the perceptive capability that each candidate node must meet lest it be discarded from the search. We specifically define a percentage of maximum allowable object occlusion that decreases linearly as the manipulator gets closer to the object and hence the servo stage, at which point a maximum 8% occlusion is allowed.
- The procedure ExtendHeuristic is used in a certain fraction of the extension steps during search to bias the search towards configurations closer to the goal region that also exhibit high perceptive capability. This balances the need for uniform space coverage and fast convergence to the goal. For the experiments in this paper, we apply ExtendHeuristic in 50% of extension steps with good convergence results.
- The randomized portion of our algorithm is shared with the original RRT algorithm, which has been shown to be probabilistically complete [14]. This hence also holds true for our algorithm.

IV. Implementation & Results

A. The ARMAR-III Humanoid

The ARMAR-III humanoid robot [16], shown in Figure 4, serves as implementation platform for our approach to manipulation planning using future perceptive capability. Its current human-centered operating environment, a full kitchen, serves as prime motivating example for the need for robustness in manipulation planning: almost all envisioned robot activities involve manipulation, the environment is complex and the potential for collisions or occlusions therefore significant. ARMAR has 43 degrees of freedom, including two 7-DOF arms, two 8-DOF hands, a 7-DOF head, a 3-DOF torso and a 3-DOF mobile base, justifying the use of sampling-based planning methods for navigation and manipulation.



(a) (b)

Fig. 4. Rendering of the ARMAR-III humanoid robot *(a)*. The physical ARMAR-III in a kitchen environment *(b)*.

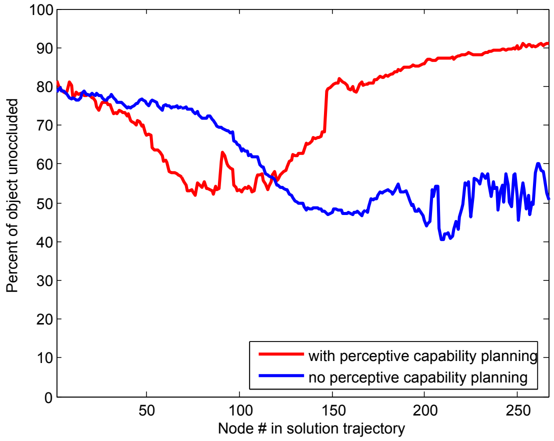


Fig. 5. Plots of evaluated perceptive capability over a reaching trajectory from experiment *CoveredSphere*, when planning explicitly for it (*red*) and when perceptive capability is not considered in the planning stage (*blue*).

All our experiments were carried out in simulation with ARMAR's binocular vision system approximated by two perspective cameras, the views of which are merged into a single camera image for the purposes of evaluating perceptive capability. This raises the issue of how applicable our results from simulation, where the environment is known a priori and perception is noise-free, will be to operation on the real robot. During manipulation on the physical platform, we intend for the planning stage to be preceded by a period of rapid initial object localization and environment reconstruction. This will yield a rough environment model that can then be used for the forward-simulation used during planning to compute the perceptive capability. All of the required perception skills, including the possibility of visual servoing via object tracking during grasp motions, are already present now on ARMAR [17]. As work on an implementation on the physical robot is under way, we hold our insights into perception-aware planning to be transferrable to the real robot domain.

B. Experiments

Our proposed approach was used to solve a series of manipulation problems of increasing difficulty. All experiments were run 30 times on an Intel Pentium 4 CPU clocked

Planned w/ perceptive capability	*Sphere* comp. time (nodes in tree)	*CoveredSphere* comp. time (nodes in tree)	*SphereCupboard* comp. time (nodes in tree)
Yes	5.8 sec (1,250)	12.6 sec (2,597)	95.9 sec (8,720)
No	0.5 sec (693)	0.2 sec (495)	50.2 sec (7,886)

TABLE I

AVERAGED RUNNING TIMES AND NODES IN SOLUTION TREE FOR THE THREE EXPERIMENTS WITH PLANNING FOR PERCEPTIVE CAPABILITY (TOP ROW) AND WITHOUT (BOTTOM ROW).

Planned w/ perceptive capability	*Sphere* occlusion (last 10%)	*CoveredSphere* occlusion (last 10%)	*SphereCupboard* occlusion (last 10%)
Yes	3.4% (2.8%)	22.3% (10.3%)	5.3% (1.1%)
No	22% (13.0%)	36.7% (43.8%)	19.5% (6.7%)

TABLE II

AVERAGED PERCENTAGE OF OBJECT OCCLUSION ALONG THE ENTIRE REACH TRAJECTORY WHEN PLANNING WITH PERCEPTIVE CAPABILITY (TOP ROW) AND WITHOUT (BOTTOM ROW). NUMBERS IN PARENTHESES GIVE AVG. OCCLUSION OVER LAST 10% OF NODES IN SOLUTION.

at 3.20GHz and equipped with 1GB RAM. We use the proximity query package [18] for collision detection and distance computation.

In our first experiment, *Sphere*, the robot had to reach for a sphere floating in an otherwise obstacle-free workspace. In this scenario, only self-occlusions can impact the robot's visibility of the object and hence the perceptive capability. Table I shows the average time taken when planning with and without regard to perceptive capability. We planned for eight of the robot's DOFs. Table II gives the average percentage of the object occluded along the entire solution trajectory. Note the significant reduction in average occlusion from 22% to 3.4% once we plan for future predictive capability.

In experiment *CoveredSphere*, our robot ARMAR had to position itself and find a grasping position for a spherical object covered on the top by a plate-like obstacle. Here, in addition to handling self-occlusion, our planner must also actively find an appropriate trajectory for the robot head that positions the cameras, so that our minimum perceptive capability threshold is met. This heightened complexity and the fact we plan for ten DOFs in this experiment accounts for the increased difference in computation time between the two experimental conditions. Figure 5 shows the perceptive capability plotted over a typical reaching trajectory. Note how, when explicitly considered in the planning phase, the measure increases significantly towards the end of the trajectory to meet our increasing minimum acceptable threshold and provides good operating conditions for a subsequent visual servo phase. This does not hold true when not planning for perceptive capability.

Finally, experiment *SphereCupboard* required ARMAR to reach for a sphere tucked away in a cupboard, leading to a higher possibility of collisions and making it harder to generate trajectories exhibiting good object visibility. Note a less pronounced difference in runtime and similar solution sizes when planning with and without perceptive capability, as collision checking dominates the computation in both conditions. The average occlusion in the perceptive capability condition is again significantly lower.

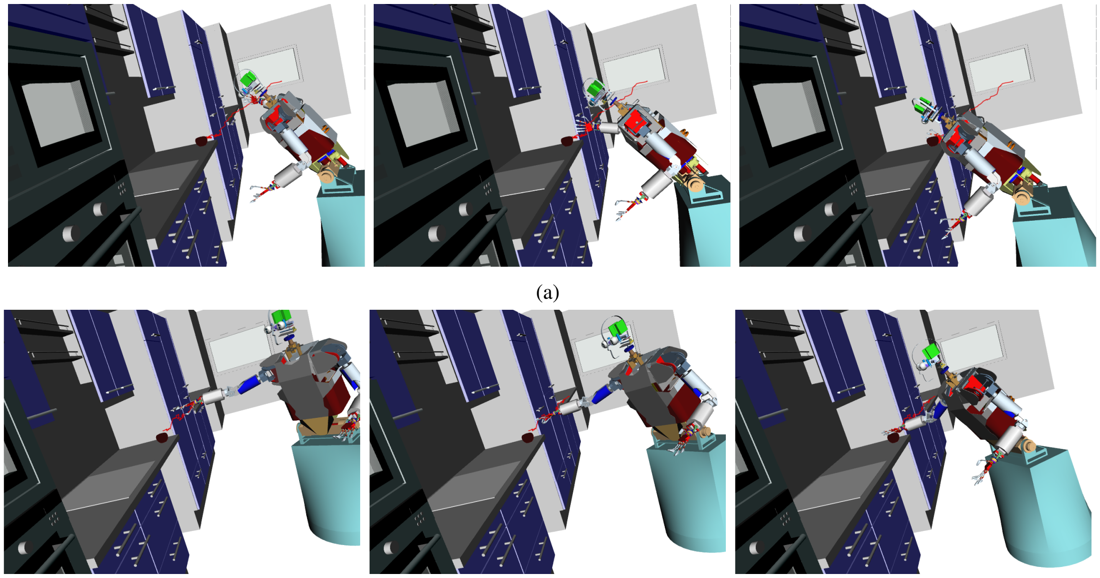
V. DISCUSSION

We have presented an approach to motion planning for highly articulated systems that seeks to reason, in the planning phase, about the robot's ability to successfully perceive its environment during operation. It is our belief that the usual lack of consideration for the environment 'sensability' during planning can often result in motion plans that do not exploit the full potential of the robotic platform to execute motions robustly using visual feedback. Our method calculates a metric assessing the robot's perceptive capability for each configuration examined during planning by simulating the robot's sensing system for the given state. We use the generated metric together with a heuristic workspace metric as part of an RRT-based planner to compute reaching trajectories for a manipulation task on a high-DOF humanoid robot. Our simulation results on the ARMAR-III platform indicate that our method generates collision-free paths that increase the chance that the perception system stays operational along the trajectory. Our proposed approach has several specific advantages over prior work in this area:

- It can operate with an arbitrarily complex model of the perception system to arrive at the perceptive capability of each state.
- By computing the perceptive capability via simulation, we are able to deal with intricate robot and environment geometry that would otherwise preclude calculation of straightforward visual constraints as a metric of sensabilty.
- Our approach does not require the sensor used to remain stationary. By exploiting sampling-based planning, we are able to compute solution trajectories that involve whole-body motion for complex systems with significant kinematic chains between sensor and manipulator.

A series of possible improvements have been identified and several are actively being pursued:

- We are working on an implementation suitable for the physical ARMAR-III robot using an environment model gathered in a fast, prior sensing phase.
- We would like to use a more realistic sensor representation, incorporating models for noise and dynamic range, among others.
- Information from all available on-board sensors should be fused when computing the perceptive capability of each state.
- While all our rendering operations currently make use of hardware acceleration, there is a potential for further significant speedup by explicitly executing components of the perceptive capability computation on the GPU.
- A more efficient approximate nearest neighbor algorithm should further reduce the computation time



(a)

(b)

Fig. 6. Key frames from reach trajectories of ARMAR attempting to pick up an apple in a kitchen environment. When planning for future perceptive capability (*a*), the apple is grasped from the side and the head-mounted cameras are appropriately positioned. Without regard to perception (*b*), the robot does not fixate the object and occludes it with the manipulator.

needed for RRT construction.

- Due to the randomized nature of the RRT algorithm, our resulting trajectories can appear jagged. An appropriate smoothing method should help.

Exploring these implementation concerns and performing further analysis forms the basis of future work that will extend the proof-of-concept experiments presented here into a full-featured robotic implementation.

VI. Acknowledgements

The work described in this paper was partially conducted within the the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft). It is also based upon work supported in part by the National Science Foundation under grant EEC-0540865. The authors would like to thank Tamim Asfour for his comments and suggestions. We also thank the InterACT program [19] for enabling us to carry out this collaborative research.

References

[1] L. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal on Robotics and Automation*, vol. RA-3, no. 5, October 1987.

[2] H. H. González-Baños, L. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, R. Motwani, and C. Tomasi, "Motion planning with visibility constraints: Building autonomous observers," in *Proc. Eighth Int'l Symp. on Robotics Research*, Y. Shirai and S. Hirose, Eds. Berlin: Springer-Verlag, 1998, pp. 95–101.

[3] T. Lozano-Perez, "Spatial planning: a configuration space approach," *IEEE Trans. Comput.*, pp. 108–120, 1983.

[4] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.

[5] S. M. LaValle, *Planning Algorithms*. Cambridge University Press (also available at http://msl.cs.uiuc.edu/planning/), 2006.

[6] M. K. Reed and P. K. Allen, "Constraint-based sensor planning for scene modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1460–1467, 2000.

[7] R. Pito, "A sensor based solution to the next best view problem," in *IAPR Int. Conf. Pattern Recognition, ICPR'96*, Vienna, Austria, Aug. 1996, pp. 941–945.

[8] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information-based adaptive robotic exploration," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland, 2002, pp. 540–545.

[9] N. Kalra, D. Ferguson, and A. Stentz, "Constrained exploration for studies in multirobot coordination," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'06)*, May 2006, pp. 4300–4302.

[10] S. LaValle, H. H. González-Baños, C. Becker, and J. Latombe, "Motion strategies for maintaining visibility of a moving target," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'97)*, 1997, pp. 731–736.

[11] H. Sutanto and R. Sharma, "Practical motion planning with visual constraints," in *Proc. IEEE Int'l Symp. on Assembly and Task Planning*, Marina del Rey, CA, August 1997, pp. 237–242.

[12] S. A. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct 1996.

[13] V. Kyrki, D. Kragic, and H. I. Christensen, "Measurement errors in visual servoing," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'04)*, New Orleans, LA, April 2004, pp. 1861–1867.

[14] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[15] D. Bertram, J. J. Kuffner, R. Dillmann, and T. Asfour, "An integrated approach to inverse kinematics and path planning for redundant manipulators," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'06)*, May 2006, pp. 1874–1879.

[16] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Proc. of the IEEE-RAS/RSJ Int. Conf. on Humanoid Robots (Humanoids'06)*, Genova, Italy, December 2006, pp. 169–175.

[17] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann, "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'06)*, Beijing, China, October 2006.

[18] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," Department of Computer Science, University of N. Carolina, Chapel Hill, Tech. Rep., 1999.

[19] "interACT center," http://www.is.cs.cmu.edu/.