Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Systems Integration for Real–World Manipulation Tasks

L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic and H. I. Christensen
Centre for Autonomous Systems, Royal Institute of Technology
SE–100 44 Stockholm, Sweden
{larsp,patric,dennis,mortens,danik,hic}@nada.kth.se [1]

Abstract

A system developed to demonstrate integration of a number of key research areas such as localization, recognition, visual tracking, visual servoing and grasping is presented together with the underlying methodology adopted to facilitate the integration.

Through sequencing of basic skills, provided by the above mentioned competencies, the system has the potential to carry out flexible grasping for fetch and carry in realistic environments. Through careful fusion of reactive and deliberative control and use of multiple sensory modalities a significant flexibility is achieved. Experimental verification of the integrated system is presented.

1 Introduction

The potential of service robotics is by now well established. The use of robotics to carry out domestic tasks offers significant economical and sociological opportunities for robotics. There is, however, a number of problems remaining before a fully operational systems can actually operate robustly in everyday environments. Most research on mobile and service robotics has been reductionistic in the sense that the overall problem has been divided into manageable sub-problems that are being addressed in a number of different programmes. One overall problem remains largely unsolved: How does one integrate these methods into systems that can operate reliably in everyday settings.

To study this problem, a long term effort has been initiated on mobile manipulation in real world scenarios such as a living room. In this paper the problem of integration of real manipulation is discussed. Two major problems are considered: i) *robust* perception-action integration for manipulation and ii) *generic* models for integration of systems. The problem of mobile manipulation was chosen as a test case, as the problem requires addressing of all issues related to: a) navigation, b) object recognition, c) robust segmentation of objects, d) servoing to a position that allows grasping, e) grasp planning, f) control generation and g) integration of the above into a unified framework. None of the above problems can be neglected or simplified if the system is to operate robustly in a real-world setting. Most of the methods reported in the literature are well suited for a particular setting, but few, if any, are suitable for operation over a wide range of operating parameters. To achieve robustness, two different approaches can be adopted: i) a number of complementary approaches can be integrated over the variability of the domain or ii) a sequence of different techniques can be engineered to particular tasks. The first approach requires context identification and methods for self-diagnostics to facilitate robustness. The latter approach, on the other hand, is the easiest to design given a particular problem but its generality is still to be demonstrated. For both approaches a key component is integration of the different techniques into an operational system.

In this paper it is described how careful consideration of the problem characteristics (mobile manipulation) can be used for derivation of a general methodology for interaction with a wide range of different objects. The general problem considered is: "go to the room X and pickup the object Y from the table". The problem involves general *navigation* to a poorly defined position (the table), *recognition* of the specified object, *servoing* to the vicinity of the object, *alignment* of manipulator with object, *grasp selection*, *grasping* of object, and finally delivery to the user. The paper is organized accordingly with sections related to each of the above problems. In addition, a section on a methodology for behavior selection and integration together with an empirical evaluation is included to demonstrate the integrated system operating in a real living room. Finally, a summary and directions for future research are provided. The emphasis throughout the presentation is on the selection of general methods for the specific sub-tasks and the integration of these into an operational control framework.

2 System Overview

A general model of the abstract architecture is shown in Figure 1.

The depicted architecture is quite general with a planner that theoretically can plan any action with skills in sequence or parallel. Included in the set of sensors are virtual sensors, providing for example the absolute position of the robot.

In the introduction it was mentioned that the general problem considered is: "go to the room X and pickup the object Y from

[1]This research has been sponsored by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems. The funding is gratefully acknowledged.

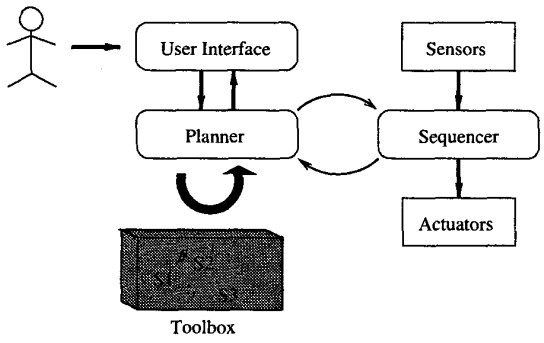

Figure 1: Architectural overview of the system. The toolbox contains skills and means for combining them in different ways using a process algebra.

the table". The general methodology we use throughout this paper is to have the necessary level of accuracy, with as robust algorithms as possible, in each step of the way. By defining a lowest level of accuracy in each step and choose the most robust algorithm that meets the requirements, the overall robustness is greatly increased.

This means that the problem statement above can be broken down into the parts below according to this coarse-to-fine strategy:

- Navigate to a position in front of the table. The necessary accuracy here is to have the table in the field of view.
- Recognise whether the object is in the image or not and return approximate image coordinates. The required accuracy is that the tracker in the next step can be initialized.
- Track the object in the image while moving the robot to the vicinity of the object. This step must result in a view of the object that covers most of the field of view.
- Align the manipulator relative to the object. Accuracy is here a paramount issue, as the grasp selection and execution is highly dependent on this.
- Select a proper grasp scheme and carry out the grasp. The accuracy here depends mostly on the manipulator itself which is usually good enough for individual objects. In the case of constrained objects (e.g. a doorhandle) a force/torque sensor is needed to allow for compliant motion.

These basic steps used to carry out the task are described in more detail in the following sections. The integration of the modules was done in the framework of DCA, Distributed Control Architecture [7].

3 Localization

A prerequisite for many of the complex tasks that a mobile platform such as ours should carry out is that the position of robot base is known.

In [2] a Kalman filter based system is presented for tracking the position of a mobile robot using a minimalistic environmental model consisting only of the most dominant walls in each room. As an example, Fig. 2 shows a detailed model of a living room (left) together with a top view (right) where the four walls included in the model has been high-lighted. Notice that all walls are occluded by tables, bookshelves, etc. to some degree. The underlying assumption here is that a majority of the points extracted in a gating region centred at a wall will originate from the wall. This allows for realiable tracking of elongated structures in a cluttered environment. To further stress the need for localization it can be said that the experimental platform (Nomadics XR4000) sometimes drifts in the order of ten degrees over a distance of 2-3 meters making pure odometric information next to useless for longer missions.

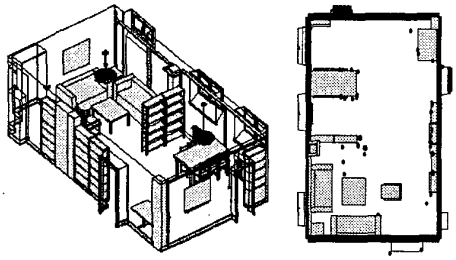

Figure 2: Left: Detailed model of the living room. Right: Simplified model with the four modelled lines marked.

4 Visual Guidance

Vision is commonly used as the underlying sensory modality in cases where steps such as to recognition, tracking or servoing onto the object are involved. Each of the mentioned steps works on different level of accuracy as illustrated in Fig. 3. A general principle for system/task design is to use the simplest possible servoing strategy depending on the current step of the manipulation task, [1]. This is also the idea followed here.

The object to be manipulated is first recognized using the view-based SVM (support vector machine) system presented in [9]. The recognition step delivers the image position and approximate size of the image region occupied by the object. This information is then used by the tracking system to track the part of the image, the *window of attention*, occupied by the object while the robot approaches it. Finally, visual servoing is used for fine-positioning of the gripper.

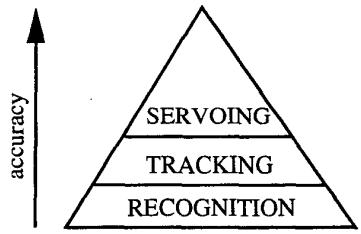

Figure 3: A commonly used hierarchy in terms of visual servo systems.

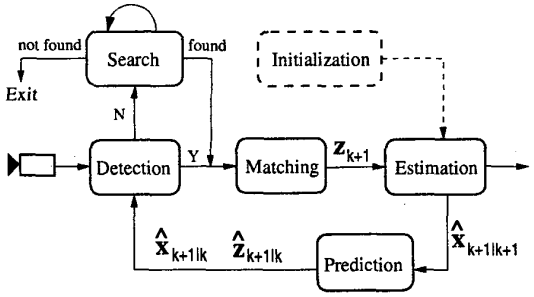

Figure 4: A schematic overview of the tracking system.

4.1 Visual Tracking

Our tracking algorithm employs the four step *detect–match–update–predict loop*, see Fig. 4. The detection and matching steps are based on integration of multiple visual cues using *voting*, [3]. The visual cues used are motion, color, correlation and intensity variation.

4.2 Visual Servoing

The final positioning if the gripper before grasping occurs is done using visual servoing. The key idea of our servoing system is the use of a reference position for the camera. It is assumed to be known how the object can be grasped from the reference position (see Fig. 5). Using a stored image taken from

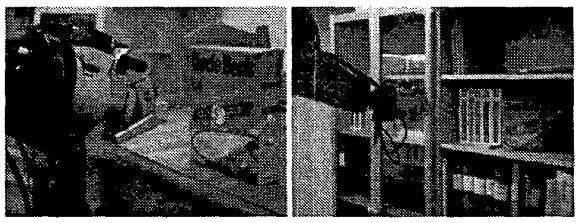

Figure 5: The left image shows the gripper in a reference position while the right image shows the camera in another position and situation. Servoing is then used to move the camera to the reference position relative to the object.

the reference position, the manipulator should be moved in such a way that the current camera view is gradually changed to match the stored reference view. Accomplishing this for general scenes is difficult, but a robust system can be made under the assumption that the objects are piecewise planar [10]. The scheme implemented requires tre major components – initialization, servoing and tracking. Each component exploits the planarity of the tracked object. For initialization, a wide baseline matching algorithm [10] is employed to establish point correspondences between the current and the reference image. The point correspondences enable the computation of a homography H relating the two views. Assuming known internal camera parameters, the homography matrix is used in the "2.5D" servoing algorithm of Malis et al [5]. Finally, as initialization is computationally expensive, matches are established in consecutive images using tracking. This is accomplished by making a prediction of the new homography H′ relating the current and the reference images, given the arm odometry between frames and the homography H relating the reference image and the previous image [10]. H′ is then used in a guided search for new correspondences [8] between the current image and the reference image.

5 Grasping

Grasping an object requires some kind of planning on how and where to place the fingers on the object. An often used, but not so often mentioned, assumption in proposed solutions to this problem is that contacts can be placed anywhere on the object, independent of each other. Thus, by ignoring the kinematic constraints of the hand, one might end up with an 'optimal' grasp that is not reachable by the hand. Ngyen [6] proposed a fast and simple algorithm for finding force-closure grasp on 3D polyhedral objects. In [6] the risk of finding solutions that are not admissible to hand kinematics is greatly reduced by finding independent regions of contact instead of discrete contact points.

For grasping purposes, the robot is equipped with a Barrett Hand, which is a three-fingered gripper with a palm. Each finger has two joints which are coupled with a clever clutch-mecanism. Two of the fingers can also rotate symmetrically around the palm, a feature that gives the hand a wide range of grasping configurations.

A service robot in a domestic environment will perform a lot of fetch-and-carry-tasks, e.g., fetching a bottle of milk from the refrigerator or a box of cereal. Many of these everyday objects can be reduced to generic shapes such as cylinders or boxes. This is the approach taken here: An object is simply seen as box with given dimensions.

When the grasping task begins, the position and orientation of the gripper relative to the object is well known. This is thanks to the high accuracy of visual-servoing algorithm used in the previous step. Thus, for a grasping task, all that has to be known is the grasping configuration and waypoints guiding the gripper to the grasp position. Fig. 6 shows the typical situation when visual servoing is finished: the transformation between the object and the gripper, ${}^{O}_{G_0}T$, is known with good accuracy. Therefore the successive relative transformations ${}^{G_i}_{G_{i+1}}T$ can

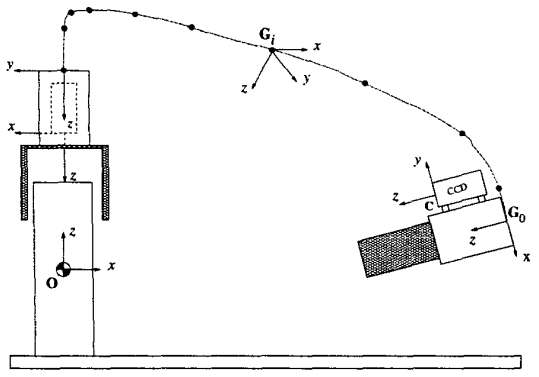

Figure 6: A typical gripper configuration when visual servoing is finished and open loop grasping begins. The black dots are waypoints expressed relative to the object.

be computed and fed to the arm-controller.

A database labeling the different objects was constructed. Together with each object, its size, the grasping configuration and the necessary waypoints were stored. Due to the limited workspace of the manipulator, these waypoints must be chosen carefully so that the desired trajectory can be executed from a number of starting configurations.

6 Integration

Integrating the modules into one system was accomplished by DCA [7] which is a framework for hierarchical composition of processes that is transparent to location (host) of the processes. The sequencing and fusion of modules is defined through a process algebra that defines the interrelation. The process composition is defined by a set of operators.

The paradigm used is a completely event driven life-cycle of the instantiated processes. A suitable model for this has been developed in [4] where a number of operators are defined. These operators are:

- Concurrent Composition: $T = (P, Q)$.
- Sequential Composition: $T = P; Q$.
- Conditional Composition: $T = P^v : Q_v$.
- Disabling Composition: $T = P\#Q$.
- Synchronous Recurrent Composition: $T = P^v :; Q_v$. This operator is recursively defined as $P^v :; Q_v = P^v : Q_v; P^v :; Q_v$.

T, P and Q are processes and v is a value passed between them.

In this framework a minimalistic, although extensible, architecture was implemented with abstract components according to Fig. 1. Showing the specifics of all the details of the whole integrated system would be out of scope for this paper. However, Fig. 7 gives a view of the system in terms of *processes*.

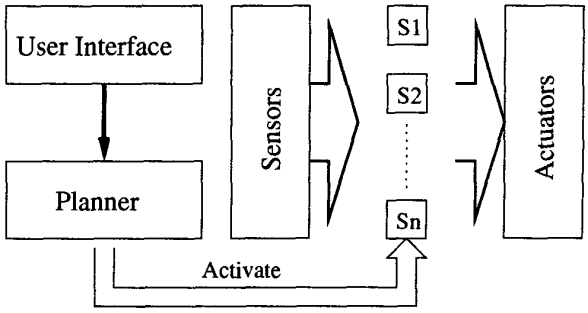

Figure 7: A view of the implemented system in terms of *processes*. S_1 - S_n are "skills" being activated by the planner. Note that in the framework of DCA, a process can also be a collection of several processes grouped and controlled by the syntax and algebra of DCA. The process "Sensors" and "Actuators" are in fact several processes interfacing to the appropriate hardware. Also, the skills themselves can consist of complex structures of processes.

A specification of the depicted system (Fig. 7) using the syntax of DCA looks like the following (internal connections have been omitted due to space limitations):

```
uinterface, sensors, actuators,
   planner:;(
      (activateS1:S1) #
      (activateS2:S2) #
            .
            .
            .
      (activateSn:Sn)
   )
```

In this particular case, the skill that controlled the grasping, ranging from recognition to the actual grasp, was defined as:

```
recognition ; visual_tracking;
visual_servoing ; grasp
```

7 Experiments

The experimental platform is a Nomadic Technologies XR4000 and is equipped with a Puma 560 arm for manipulation (see Fig. 8). The robot has two rings of sonars, a SICK laser scanner, a wrist mounted force/torque sensor (JR3), and a color CCD camera mounted on the gripper (Barrett Hand).

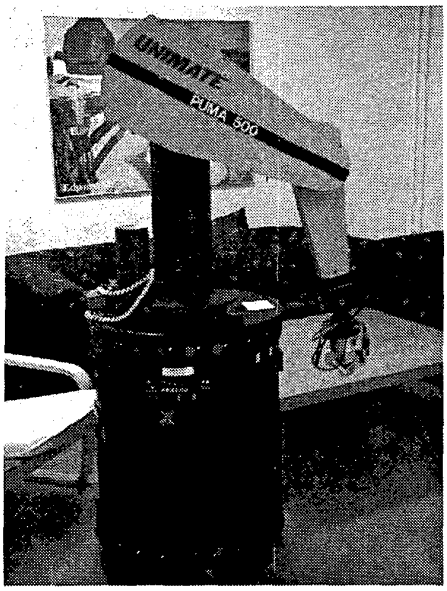

Figure 8: The hardware on which the system was running.

The system is controlled by three different on-board computers. One computer, running QNX, is used for controlling the manipulator arm. The other two are running Linux. One of these is dedicated for vision and the other one is used for the remaining tasks.

7.1 Experimental Results

The demonstrated task was to go to a table cluttered with several objects and instruct the robot to pick up e.g. a package of rice. The object of interest could have been placed anywhere on the table so to solve the task there was clearly a need for adaptive algorithms using sensory feed-back. Also, the complete task was of such a complex nature that it to a large extent exploited and well demonstrated the features of the Distributed Control Architecture.

The sequence in Figure 9 shows an experiment where the robot started at the far end of the room, moving towards a point where it turns around to get a good view of the table. The robot is then instructed to pick up the package of rice which it then recognises and locates in the scene. Further on, the robot moves closer to the table while tracking the package. As indicated in the monitor, the manipulator then aligns itself with the object. Finally, the package is grasped, lifted and handed over to the user.

8 Discussion

In this paper the problem of real manipulation was discussed and in particular two major challenges was considered, namely i) robust perception-action integration for manipulation and ii) a generic model for integration of systems. Mobile manipulation was chosen as a test case since it addresses many of the core issues in robotics, i.e. navigation, object recognition, segmentation of objects, servoing, grasping, control and integration.

The article demonstrates how a highly complex task like this can be robustly implemented by breaking down the given task according to a coarse-to-fine strategy where as robust algorithms as possible, with the necessary level of accuracy, is used in each step of the way. This scheme allowed a straight forward implementation of a robotic system that autonomously could perform manipulation of objects in a real-world environment. Integration of components was performed in a modular framework for hierarchical composition of processes whose execution and evolution over time were controlled by a process algebra.

Key aspects of the running system was to successfully navigate to certain locations and from a poorly defined position perform manipulation by the use of recognition, visual tracking and visual servoing.

A limiting factor of this particular system was the small workspace due to singularities which is a known problem of the Puma560 arm. This can e.g. be resolved by using manipulators with more degrees of freedom which emphasizes the need to address the problem of redundancy resolution schemes. A future improvement of the described system will therefore be to include the mobile platform as a part of the manipulator during visual servoing and grasping, adding two degrees of freedom. This way one can avoid the arm ending up in a singular position, thus giving a more robust and flexible system.

Acknowledgement

We would like to thank former members of the lab, David Austin, Danny Roobaert and Michael Zillich, for the use of some of their software in parts of the described system.

References

[1] Z. Dodds, M. Jägersand, G. Hager, and K. Toyama. A hierarchical vision architecture for robotic manipulation tasks. In *Proceedings of the International Conference on Computer Vision Systems, ICVS'99*, pages 312–331, 1999.

[2] P. Jensfelt. *Approaches to Mobile Robot Localization in Indoor Environments*. PhD thesis, Signal, Sensors and Systems (S3), Royal Institute of Technology, SE-100 44 Stockholm, Sweden, 2001.

[3] D. Kragic. *Visual Servoing for Manipulation: Robustness and Integration Issues*. PhD thesis, Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, 2001.

[4] D. Lyons and M. Arbib. A formal model of computation for sensory-based robotics. *IEEE Trans. Robotics and Automation*, (3):280–293, 1989.

[5] E. Malis, F. Chaumette, and S. Boudet. Positioning a coarse-calibrated camera with respect to an unknown object by 2-1/2-d visual servoing. In *ICRA*, pages 1352–1359, 1998.

Figure 9: Sequency showing an experiment where the robot moves up to the table, recognises the rice box, approaches it, picks it up and hands it over to the user.

[6] V.-D. Nguyen. Constructing force-closure grasps. *The Int. J. of Robotics Research*, 7(3):3–16, 1988.

[7] L. Petersson, D. Austin, and H.I. Christensen. DCA: A Distributed Control Architecture for Robotics. In *IROS*, pages 2361–2368, Maui, October 2001. IEEE.

[8] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV*, pages 863–869, 1998.

[9] D. Roobaert. *Pedagogical Support Vector Learning: A Pure Learning Appr oach to Object Recognition.* PhD thesis, Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, May 2001.

[10] D. Tell. *Wide baseline matching with applications to visual servoing*. PhD thesis, Computational Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, 2002.