Control of Redundant Attentive and Investigative Behaviors in an Active Cognitive Vision System

Motoki Takagi, Christof Eberst[1], Gerald Umgeher[1]

[1]Profactor Produktionsforschungs GmbH
Wehrgrabengasse 1-5, 4400 Steyr Austria

Introduction

Robotic systems that interact with changing and unknown environments and humans need to possess dependability higher than current vision systems alone can provide. Most autonomous robots therefore fuse vision system with laser-sensors or ultrasonic-sensors.

Within the Cognitive Vision Project ActIPret, we experiment with a pure vision controlled stationary multi-robot system, each robot equipped with a stereo camera system. In order to obtain the required robustness, the active vision systems perform vision tasks redundantly and coordinately. Results from redundant views are merged to increase the robustness and to reject artifacts.

Vision-based behaviors [Ronald C. Arkin, Behavior-Based Robotics (Intelligent Robotics and Autonomous Agents), MIT Press, 1998] and vision-services are dynamically assigned to robots. A de-central view-contract-managing approach assigns vision-behaviors and vision-services to robots - optimized for redundancy. Therefore, each robot serves multiple and potentially conflicting vision behaviors simultaneously. Conflicts are resolved by a de-central approach employing a view-contract-manager and local view-controllers.

The main topic of this paper is the shared responsibility approach: The best robots and best services are selected dynamically, balancing performance optimization by redundant processing with the avoidance of conflicts for the limited resources and the constraint of high reactivity.

Concept

The main idea behind the control concept is a de-central approach: Each involved component, i.e. the service requesting component, the providing component, the view-contract-manager and the local view-controllers all have a limited specific responsibility. By interaction of the components, the final control of the robot's attention considers:

- **Task specific** and **situation specific** needs: considered by the dynamic, not explicitly programmed service selection of the requesting component and the service description of the providing component.
- **Locally** highly reactive and rapid response, by the direct communication between the service requesting component and the view-controller, plus the responsibility of the view-controller for its local operation, cost and quality including rapid local conflict management and fusion of requests.
- **Globally** sub-optimal resource-usage and conflict management, by cooperation of view-contract-manager and view-controllers considering the service descriptions, task request and status of the robots.
- **Feedback from a cognitive "reasoning" component** (externally developed [A. J. Howell and H. Buxton. "Learning gestures for visually mediated interaction". In Proc. BMVC, pp. 508-517, Southampton, UK, 1998.]) have an influence to processing and control.
- **Independence** of HW and SW working with heterogeneous robots and tasks

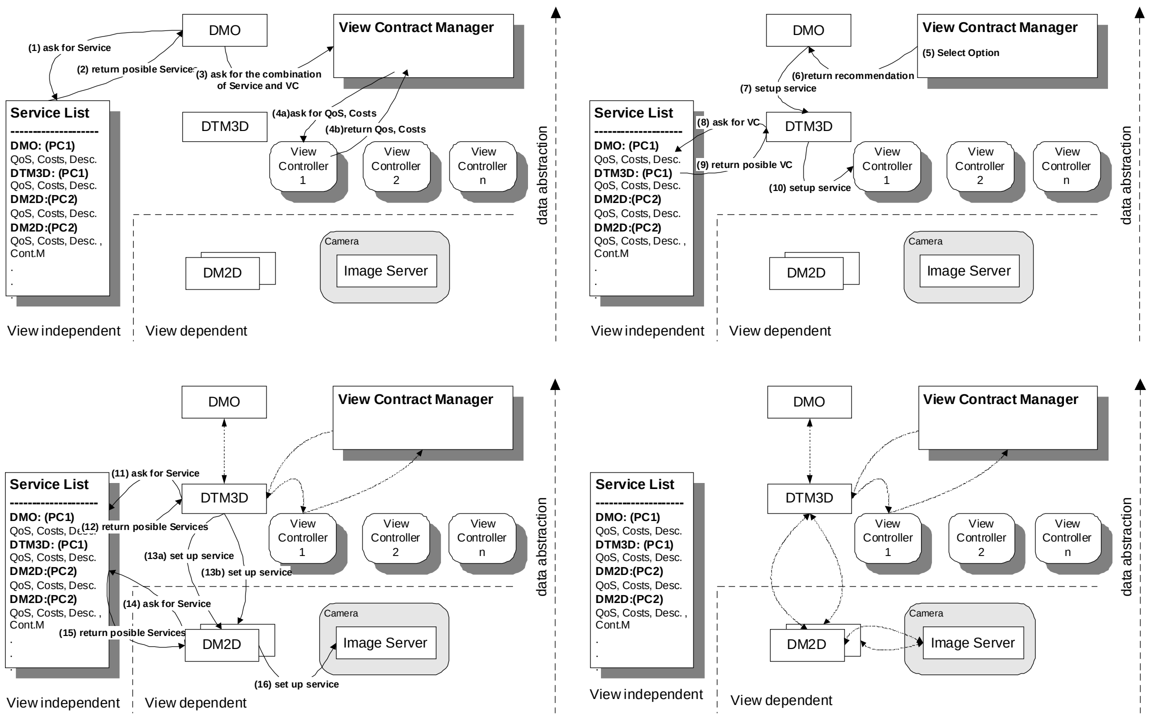

Figure 1: The sequence of attention-control process: Pre-selection of fitting services (upper left), determination of the best fitting service-robot assignments (upper right), setup, start of the service DM2D and establishing the communication to Image Server (lower left), DTM3D is sending view-requests during tracking (lower right). (Dotted lines indicate the already established communications)

The attention-control process is as follows:

Initial service selection and assignment: If a vision service requesting component needs a specific service, it checks a service list (CORBA trading service) for fitting service-components and pre-selects the ones which are best fitting according to the individual service-properties, expected costs and quality (Figure 1 upper left). If the properties of the pre-selected service indicates, that the service requires specific resources like camera views, the requesting component which is independent of the system and has no knowledge about dependencies, sets up and delegates the final decision which services to start on which robot(s) and camera(s) to the view-contract-manager. The view-contract-manager requests a bid from individual view-controllers at what local costs and quality, the service can be processed and which local (resource) conflicts with currently processed services will arise. For services, which require a specific, situation dependent Region of Interest (ROI) at startup, the bid-offer considers this ROI. From the bids of the view-controllers, the view-contract-manager selects the globally (sub-) optimal subset of services and their assignment to robots (Figure 1 upper right). A main feature of this approach is that each view controller's bid include **several alternative options** with their **local** quality, cost and implications on the other services. This enables the view-contract manager to select a **globally** sub-optimal alternative. Thus, the view-contract-manager determines the level of redundancy in the systems, i.e. on how many robots a service is started. The selected services are then started on the assigned robots by the service-requesting component. The selected services connect themselves to the view-controllers and the required services, such as image servers (Figure 1 lower left).

Run-time control: Each providing component offers the service to the requesting components. It is also sending its view request directly to its assigned view-controller, together with the expected quality and cost of its service. The view request is - independent of the robot on which it is performed - specified as region of interest in 3D world coordinates, the optimal orientation from which to observe and the optimal size of the area covered by the image – plus the allowed parameter tolerances.

The view-controller checks feasibility of the request on the given robot, i.e. if it can be reached considering the robot's dexterous workspace and obstacles in its environments. Then it checks for inconsistencies with view requests from the other services, which are active on this specific robot. According to these conditions, the view-controller recalculates local cost and quality for each service combination. In case of larger changes, it sends an update to the view-contract-manager, which can re-assign or stop services (Figure 1 lower right). In case of a server conflict, the view-controller can react rapidly by ignoring the requests of one service and by initiating the termination of the service. For all non-conflicting, active services of its robot, the view-controller merges the requests and recalculates a new reachable, collisions free view trajectory and commands the robot, and changes the focal length of the camera if necessary.

Implementation

View-contract-manager and view-controllers are communicating asynchronously. The view-contract-manager specifies a bid-request for every assignment task, delegated by the service-requesting components and sends the bid-request to all the view-controllers. Each view-controller analyses the bid-request and sends back a bid-offer including several options that specify the local aspects of costs and quality and conflicts.

The selection of services and the assignments to the robots by the view-contract-manager are currently based on fuzzy rules, which include: enable redundant processing, suppress conflicting combinations, keep costs low in order to allow for starting new services without stopping active ones. As an example for a multi-view service, the avoidance of a human body[1], operating in a reachable area, is described:

In order to avoid the moving obstacles, like the human operator, a pair of detectors (Detect-and-Track-Motion-in-2D: DM2D), that have been assigned, recognizes and tracks 2D features in the reachable area of the robots. A pair of DM2Ds is active on each individual robot. Initial assignment of these features is currently user-assisted but shall be learned off-line next. The component Detect-and-Track-Motion-in-3D (DTM3D) reconstructs and tracks the resulting 3D motions of the features and filters out obvious artifacts. It is sending view requests to its assigned view-controller to keep the features in view. The component Detect-Moving-Object (DMO) merges the 3D features from the DTM3D components and tries to map shape and motion of the human operator. DMO sends the estimated position of the human to the obstacle avoidance-service (OA) of each robot. OAs determine the pose that the robot needs to avoid and sends this to the view-controllers, which include these into their path determination. The DMO also feeds back ROIs to its subordinated DTM3D in order to co-ordinate them. The DTM3Ds consider this feedback in their next view-request.

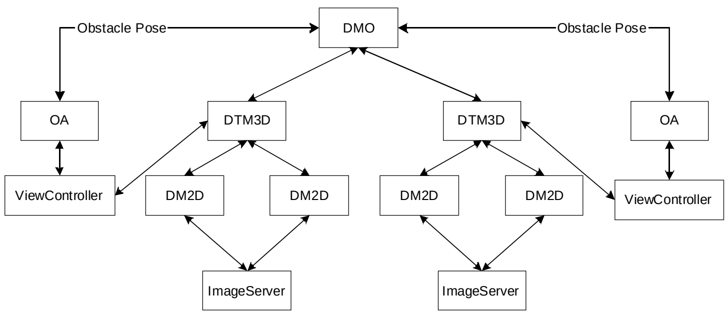

Figure 2: Structure of the multi-view obstacle avoidance.

Experiments

The setup: The multi-robot system consists of four AMTEC robotic systems (each between 6 and 3 DOF) controlled by a PC-based industrial controller. A CORBA-interface / command-set added to the controller allows simple and efficient commanding of the robot directly out of the ActIPret framework.

[1] Tracking of the operator's hands/arms is implemented as parallel running behaviour by other project-partners.

Figure 3: Setup of the demonstrator for experiments.

Experiments have been performed with different heterogeneous services simultaneously, such as reconstruction and tracking. The work is in progress, first experiments (see Figure 3) have been made with all vision components implemented rudimentary.
In the current version, the tracking is based on the CMVision colour-tracker [http://www-2.cs.cmu.edu/~jbruce/cmvision] and the object[2] size is limited by the degrading 3D reconstruction to small sizes. The entire system is operating at 5 Hz on two 866MHz Dual-PIII, including image processing, 3D tracking, view-contract-management and view point selection and control of arm motion. The 2D tracking components will be replaced by a more advance version that works with more significant features.

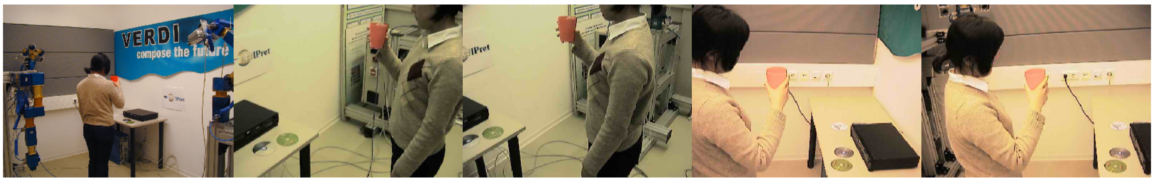

Figure 4: Experiment: 3D tracking of the object pose from multiple views, including avoidance.

Conclusion and Future Work

We have been presenting an approach for the de-central control of attentive and investigative vision behaviours within the framework of the CV Project ActIPret. The work is in progress; main focus is an increase of the performance of the system and the exchange of the DM2D components in order to allow for tracking of larger objects close to the cameras.

Acknowledgement

The work presented was partially supported by the EU IST Project ActIPret, Contract Nr.: IST-2001-32184. Special acknowledgement to Stefan Blum of TU Munich, for his support at the OSCAR system.

References

1. Ronald C. Arkin, Behavior-Based Robotics (Intelligent Robotics and Autonomous Agents), MIT Press, 1998
2. A. J. Howell and H. Buxton. “Learning gestures for visually mediated interaction”. In Proc. BMVC, pp. 508-517, Southampton, UK, 1998.
3. Stefan A. Blum, “A CORBA-based System Architecture for the Exploration of Indoor Environments with an Autonomous Robot”
4. http://actipret.infa.tuwien.ac.at
5. http://www-2.cs.cmu.edu/~jbruce/cmvision
6. http://www.oscar-net.org
7. http://www.amtec-robotics.com/

[2] We currently use marker to track the human operator in 3D, or track objects which are easy to track and reconstruct, e.g. the cup shown in figure 3.