DELIVERABLE D3.3 (v0.1)

Method of incorporating deictic relations and report of evaluation from scenario placing CD in player

Draft Version

20 May 2004

Authors: Wolfgang Ponweiser, Minu Ayromlou, Michael Zillich, Markus Vincze, Gerald Umgeher

Project acronym: ACTIPRET
Project full title: Interpreting and Understanding Activities of Expert Operators for Teaching and Education

Action Line IV.2.1: Real Time Distributed Systems (Cognitive Vision)
Contract Number: IST-2001-32184

Contents

1 Introduction

Deliverable D3.3 summarises the use of context-dependent information to direct processing in the framework of activity interpretation. In particular, we will shortly outline, how high-level information is used to trigger the selection of lower level components of the ActIPret Framework. At a wider scope the central idea of ActIPret to merge bottom up and top down information flows in one framework have been already covered in several Deliverables (in particular D1.2 and D3.1, for details will be given in the next paragraphs) and is not repeated here.

When Referring to the ActIPret Framework (see Figure below), the report will focus on the Object Relationship Generator (ORG). The ORG acts as coordinator of image processes such as hand and object tracking, detection and recognition. It operates in 3D space and is view independent. This means, that it considers relationships independent of specific viewing directions. The viewing directions will be taken into account in the lower levels, i.e., hand and object tracking, detection and recognition.

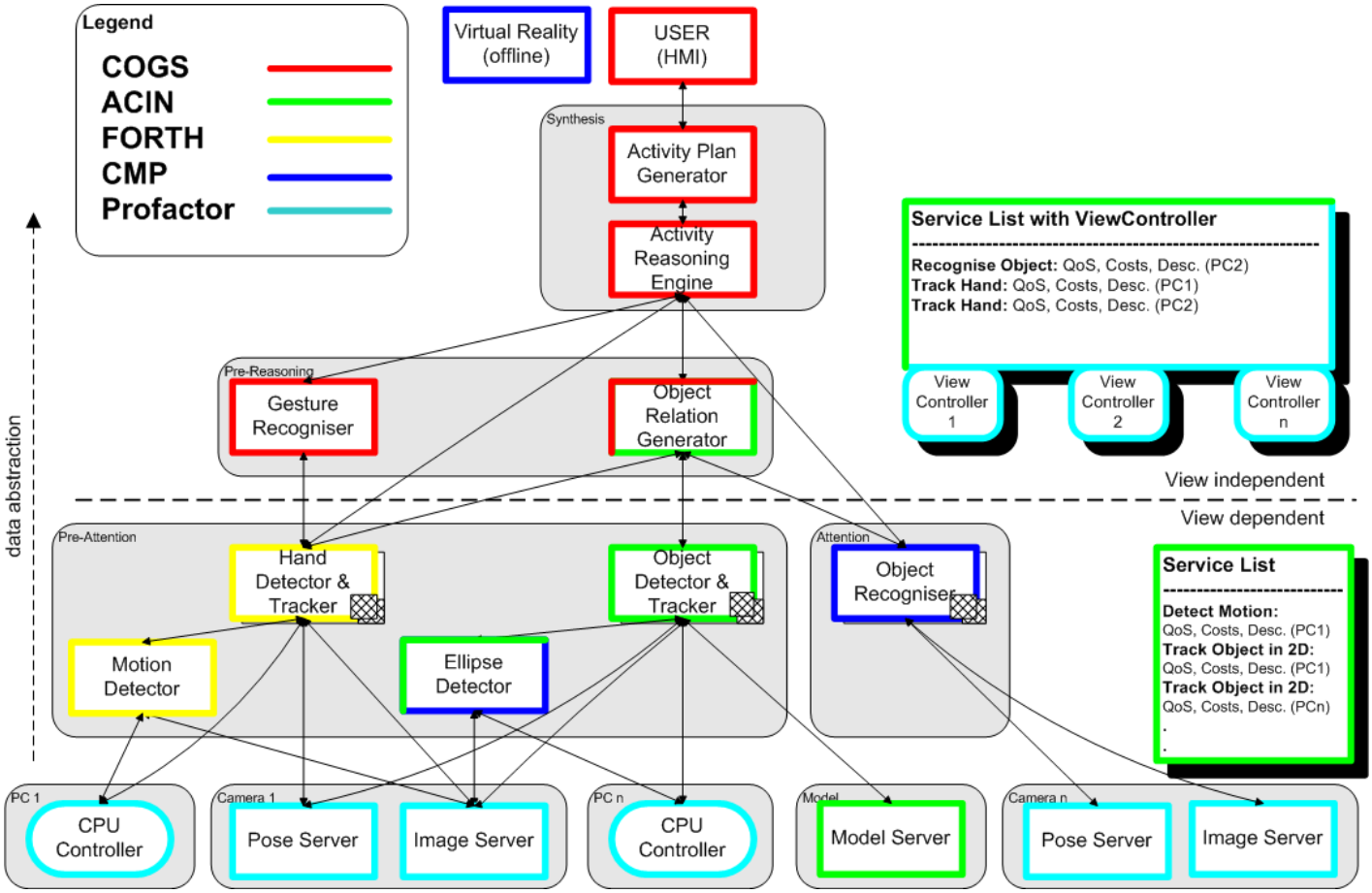

Figure 1: The architecture of the ActIPret demonstrator. It is based on a component-based approach. Each box encapsulates one component. The colours refer the responsible partner(s). The Object Relationship Generator can be seen practically in the centre of this diagram. It communicates with the Activity Reasoning Engine (ARE) to receive high-level directions and reports back the detection or end of relationships. And its coordinates the execution of Hand and object tracking and recognition and handles the reports of these three components to merge it with the other information and report it for final concept synthesis in the ARE.

The basic concepts of generating and maintaining deictic relationships was found to be a key point of the project. Consequently, previous deliverables have already reported most of the relevant information. That is, D1.2 defined the basic task of the ORG component, and D3.1 already explained the basic relationships that will be used. This Deliverable only improves over this report from hindsight and the present implementation.

Furthermore, not included in this Deliverable are methods of using contextual information to execute processing locally, that is, within other components than the ORG. For example, Deliverable 2.2 reported this for the hand tracker. And Deliverable 3.2 outlined the object tracking method which is based on cue integration and the local estimation of the perceived scene complexity to steer cue processing.

The interaction of the contextual processes with the active selection and control of view points is covered in Deliverable 6.3 “Description of methods to provide investigative behaviours and their evaluation”. In particular Section 3 outlines the use of context to share responsibility and select behaviours and Section 4 gives the implementation to coordinated independent components and possibly conflicting requests.

Finally, this Deliverable adds to D1.3 which reported mainly high level relationships and their usage to guide object tracking.

2 Using Object Relationships to Focus Processing

The main function of the Object Relationship Generator (ORG) is the conversion of conceptual terms to data items as used by visual processes and vice versa the conversion of the outcome of visual processes into abstracted relations. The basic concept has been outlined in Deliverable 1.2. In short, the ORG controls the generation and maintenance of spatio-temporal relationships between two objects. It determines relationships between the objects (hand, CD, player, button) in 3-D at one instance and their variation over several cycles.

The ORG’s specific functions are to create, confirm or reject, and to continuously update deictic relationships (see D1.2 for details). The relationships derived by the ORG are task-related operators in two forms (see deliverable 3.1 for more details): *Basic relationships* are generally behaviours of a single object or simple relations between two objects. *Aggregate relationships* are relations between two (or more) objects and in the spatio-temporal domain.

Of particular interest for the ActIPret system is:

- **Purposive behaviour trajectory:** a single object trajectory is of interest for the synthesis process, if the object (hand) makes a purposive behaviour trajectory, that is a trajectory that is task relevant. An example is a hand moving towards an object. The goal is to detect the purposive trajectory as early as possible to be able to focus processing to the area indicated by the direction of the trajectory. An example is grasping, where processing should be focused on the objects at the estimated grasp location. Methods to detect a purposive behaviour trajectory can be based on Kalman filters (as proposed by ACIN) or a time-delay Radial Basis Function (RBF) network (as proposed by COGS).
- **Distance between two objects (mutual proximity):** the exact distance between two objects depends on the representation of the objects used. The simplest representation uses the reference coordinate frames. In this case the distance is the Euclidian distance between the origins of the two object coordinate frames. If the objects are represented with a hull, the distance can be defined as the closest point

between the two hulls. This relation is of specific interest for predictive guidance of the pre-attentive visual processes.

- **Find objects near to each other**: this service provides a pre-attentive predictive cue, based on the assumption that the closer two objects are to each other, the more likely they are to have some task-relatedness. Especially the event of firstly fulfilling this nearness relation is also used to guide visual routines.
- **Object near trajectory of object:** this service provides a pre-attentive predictive cue, based on the assumption that the closer an object is to a hand's trajectory, the more likely it is to be manipulated by that hand.

2.1 Task driven Operation of the ORG

ORG starts operating when given the task from a higher level component, in the case of the ActIPret demonstrator, the Activity Reasoning Engine (ARE). ARE makes the assumption that activities are hand based (in another scenario, e.g., surveillance, it could be any moving object). As a consequence Hand Tracking (HT) is started. The next function is the main operation of the ORG. It is implemented by the following service.

Service: ObserveRelations

On a service request the following needs to be specified:

- a reference object (usually a hand as the source object of actvities)
- a relationship threshold (in earlier reports this was a relationship type and a relationship threshold; see Section 2.3 for clarification and use).

The result of hand tracking (the hand pose) is used to generate a Space of Interest (SOI, similar to the Region of Interest in an image but in 3D) to focus further image processing, e.g., Recognition, Detection und Tracking. Also the end of these processes, e.g., if a recognized object shall be tracked or not, depends on the SOI respectively the distance to the reference object (= hand). Therewith the nearness relation is used to guide visual processing. This is the control aspect of the ORG.

The aspect of determining and reporting the relationships back to the calling component (or service requester) is described in the next Section.

2.2 Implementation of the Relations as Queries to a Data Base

The actual relationships are calculated from two forms of queries to the ORG data base. Following the concept of a component-based approach and framework, theses queries are formulated in the two following services.

Service 1: GetLostFoundObjects (get lost, found, and persistant objects)

On a service request the following needs to be specified:

- a reference object (usually a hand as the source object of actvities)

- a reference time (which is an indirect specification of the reference pose of this object, because the reference pose the position and orientation of the reference object at the reference time)
- a time interval with StartTime and EndTime, and
- a spatial relationship threshold (in earlier reports this was a relationship type and a relationship threshold; see Section 2.3 for clarification and use).

Using the data bank of past object (hand) poses, the pose at the reference time is retrieved. Then all objects in the data base that have been “near” the reference pose at the StartTime are listed in the StartList. In the same manner all objects in the data base that have been “near” the reference pose at the EndTime are listed in the EndList.

Finally, the report of Lost, Found and Persistent Objects is found by comparison of these two lists:

- Lost object = in Startlist; not in Endlist
- Found object = not in Startliste; in Endlist
- Persistant object = in Startlist; in Endlist

These three lists are then sorted according to „nearness“, listing the nearest object first, and reported back to the service requester.

Service 2: GetAppearedObjects

On a service request the following needs to be specified:

- a reference object (usually a hand as the source object of actvities)
- a time interval with StartTime and EndTime, and
- a relationship threshold (in earlier reports this was a relationship type and a relationship threshold; see Section 2.3 for clarification and use).

For each point of time between StartTime and EndTime, for which a pose of the reference object is stored in the data base, a list of “near” objects is computed. These lists are then merged into one consistent list by deleting multiple entries. Finally, the remaining entries (objects) are again sorted according to distance and reported back.

Finally, it turned out to handle the hand object different from other objects, where for the above calculations of „nearness“ only the object centre and radius is sufficient. If the hand is the reference object, then the above procedure is repeated for the hand-centeroid and all the finger-tips detected. Especially for the activity “press button”, the nearness of the button to a finger-tip is of specific interest. Again, multiple entries are eliminated and objects ordered for each hand point according to distance before reported back to the caller.

2.3 Relationship “Nearness”

In the original proposal (Deliverable 3.1) it was thought that the ORG must be able to handle different types of relationships. As implementation started it became soon evident, that all relationships planned can be broken down into the *nearness* relationship. The threshold is now related to near and gives a maximum distance of interest. In terms of the implementation it specifies the SOI radius.

It could be demonstrated that relationships such as “VeryNear” or “Near” can be subsumed in one relationship, which simplified the structure for the two service requests given in

Section 2.2. The two relationships differ not in the geometric handling of the near relationship (note that ARE has no capability to reason about geometry, hence cannot request such reasoning that is solely handled and generated by the ORG), but in the temporal aspects. In the case of GetLostFoundObjects objects are compared for nearness for exactly two points in time. In the case of GetAppearedObjects the complete trajectory of one object is used to query the relationship “Nearness”.

The main reason why a distinction of relationships such as “VeryNear” or “Near” is not useful is that both concepts would be specified with thresholds and thresholds are inherently context dependent, tend to be brittle in relation to system performance and usage in another application is limited to tune a parameter.

Unfortunately the threshold could not be eliminated. But the relationship (“Nearness” as used now, enables to specify a secure (large) threshold to make sure that the confidence interval to include the correct near object as needed to interpret the activity is included, e.g., a three sigma threshold. The key point is that listing the “Near” objects according to distance has the effect that in most actual cases for the ActIPret demonstrator, the correct object is pursued in subsequent reasoning at the ARE as one of the first objects/relationships and hence any other pre-filtering at this stage would only eliminate the correct hypothesis which is locally not known (or cannot be verified) in the ORG.

2.4 Example

An example for the process is derived from the ActIPret scenario of placing a CD in the CD player. The idea is that the hand drives the focus of attention and hence processing. Once a consistent hand trajectory is found a SOI is constructed and the ORG triggers the recognition of object in this SOI (using the service ObserveRelations). For example, the CDs should be found that are subsequently grasped. If the Gesture Recogniser detects an event of the hand such as stopping and returning, the service GetAppearedObjects is used to locate objects of interest near this point of return in the hand trajectory. For example, pressing a botton usually produces such a hand gesture and the ORG would then have the task to find the button that has been pressed. Figure 2 gives an example

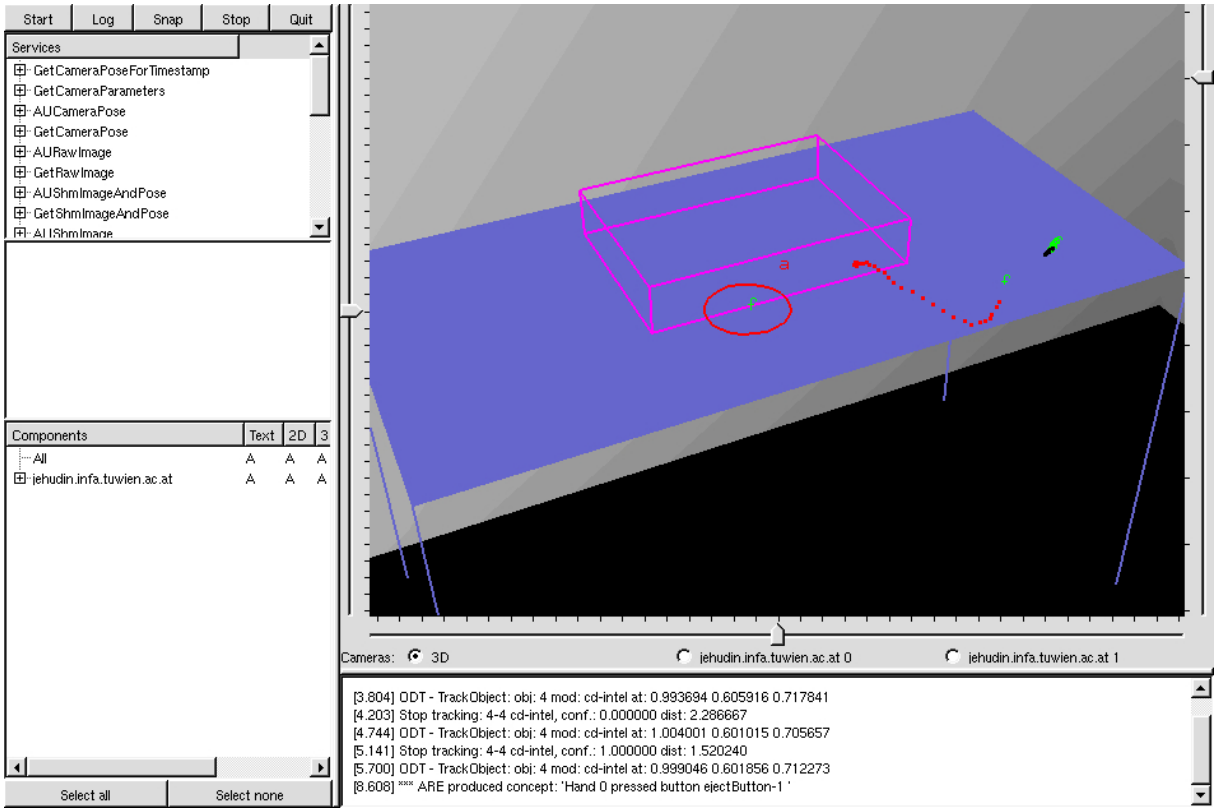

Figure 2: GUI of the ActIPret demonstrator displaying online 3D information about GetLostFoundObjects (denoted by a small "f") and GetAppearedObjects (denoted by a small "a"). At the beginning of the hand trajectory (to the right) the "f" indicates an object found. Close to the end of the hand trajectory the "a" indicates the appeared object along the trajectory. In this case the open button of the CD player. Finally, the "f" in the ellipse centre indicates that the open drawer has been found already, since it is "near" the hand after pressing the button.

3 Evaluation

The ORG has been tested as stand alone component, where the communication with the ARE has been simulated in the ORGTest component. For these tests the thanks go to Gerald Umgeher of Profactor, who conducted tests for all subsystems and the complete ActIPret demonstrator as part of WP8. The table below summarises the test results for the ORG on three of the test sequences. With the new sequences from ECCV 2004 new tests will be conducted.

Components Tested				
HTTest_FORTH HT_FORTH IS_Profactor CDS_Profactor	ORTest_CMP OR_CMP IS_Profactor CDS_Profactor	GRTest_COGS GR_COGS HT_FORTH IS_Profactor CDS_Profactor	ORGTest_ACIN ORG_ACIN ODT_ACIN_V4R FD_Ellipse_ACIN OR_CMP IS_Profactor CDS_Profactor	Test Sequence / Timing Factor
100%	96%	100%	98%	P0020 / VirtualTime 8
100%	96%	100%	98%	P0023 / VirtualTime 8
100%	96%	100%	99%	P0024 / VirtualTime 8

Table 1: summary of test results for the ORG and the components in the levels below.The percentage gives the number of correct responses over all possible correct responses.

4 Conclusion

This Deliverable reported on the implementation of deictic selection of processing in the top-down branch of the cognitive vision approach developed in ActIPret. A single spatil relationship finally implements all the relationships between two or more objects as foreseen in Deliverable D3.1, the “nearness” relationship. The reduction also helped to make an efficient data base implementation.

Several related ideas have been treated in other Deliverables: the deictic response of the Hand tracker in D2.2, of the Object Tracker in D3.2, and the interaction with proceeses to control attention in D6.3.

Although tests of the ORG demonstrate good performance, this is local performance and does not include the generation of high-level concepts. Tests including the generation of the semantic concepts show that accuracy is low such that the correct hypothesis is often not the first but in the first three. For example, the CD player has many buttons close to each other. Even for the human eye it is difficult to detect which button has been pressed from visual evidence alone. Consequently other evidence, such as the drawing opening needs to be used to verify one of several hypotheses.

Finally, Table 1 also indicates the remaining problem of heavy computational load of the lower level image processing operations, in particular, object recognition. The last Column “Virtual Time 8” refers to the fact that best performance is presently achieved at a frame rate of 50/8 Hz. This means that either several processors are used to run the system or faster PCs must be used (reference is a 2.4 MHz PC). Although the reviewers stressed the point that real-time should not be the main point of interest, to demonstrate the system this aspect will be considered for the final review. While speeding up the process is one option, it is also considered to investigate how processes can take processing time into account.