DELIVERABLE D3.1 (v1.1)

Definition of Basic Set of Relationships and the Derived Aggregate Relationships

Final Version

29 April 2002

Authors: Wolfgang Ponweiser, Minu Ayromlou, Michael Zillich, Markus Vincze, Jonathan Howell

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 3
2 Basic relationships 4
2.1 Purposive behaviour trajectory 4
2.2 Distance between two objects 4
2.3 Find objects near to each other 5
3 Aggregate relationships 5
3.1 Object near trajectory of object 5
4 Proposed methods 5
4.1 The RBF network scheme 5
4.2 Trajectory estimation with a Kalman filter 6
5 Relationship to framework 7
5.1 Relationship to other work packages 7
6 References 8

1 Introduction

This deliverable document provides details of the specific object relationships used within the ActIPret Demonstrator (AD). It summarises the work thus far under Task 3.1 (Conception including interface definitions). Section 1 defines basic terminology for these relationships and sections 2 and 3 provide further details on each relationship type. Section 4 gives details of proposed methodologies. Section 5 concludes and describes how this work relates to the overall principles and requirements of the Cognitive Vision (CV) framework.

Relationships are of major importance for the cognitive vision process, since it is through verification of the perceived relationships that the validity of object and task hypotheses can be deduced. This results mainly from the fact that relationships define spatial and temporal relationships between entities and therefore the positive evaluation of an examined relationship adds confidentiality for its correctness. Different relationships will be of use to different levels of processing depending on the application domain.

At a lower level of image processing several relationships between features of an object are used to verify the validity of the features. Surface properties next to edges and the topological relationships given by a wire-frame model of an object are relationships used to enable robust object tracking. The application of these relationships for tracking is described in detail in [13] and the ongoing work of Minu Ayromlou at ACIN.

It should be noted that in this deliverable the focus is on high-level relationships, such as those between objects or the behaviour of objects within the scenario. The focus in this document is on the relationships that are of most utility within the ActIPret project, but could be generalised for other CV frameworks.

The relationships defined in ActIPret can be classified according to 3 properties (as used in Table 1):

- basic/aggregate;
- spatial/spatio-temporal domain, and
- unary/binary.

Table 1 provides an overview of relationships and indicates in the last line details of the realisation within components of the ActIPret Demonstrator.

Relationship	**Basic**	**Aggregate**
Spatial	*Binary Relationships:* • distance between two objects • find objects near to each other	-
Spatio-temporal	*Unary Relationship:* • purposive behaviour trajectory	*Binary Relationships:* • object near trajectory of object
Realisation in ActIPret Demonstrator	Only hand objects considered Implemented in component “Gesture Recognition”	Implemented in component “Object Relationship Generator”

Table 1: Overview of Relationships.

These relationships are implemented in the components Gesture Recogniser (GR) and Object Relationship Generator (ORG). For simplicity the functional descriptions of these components is given with all other functional descriptions in Appendix A of Deliverable D1.1

and the IDL definition of the interfaces of these two components is given together with all other IDL notations in Appendix B of deliverable D1.1.

2 Basic relationships

Basic (as opposed to aggregate) relationships are behaviours either of a single object or relationships between two objects. The basic behaviours that have been identified as relevant thus far for the project are:

- purposive behaviour trajectory;
- distance between two objects; and
- find object near to each other.

2.1 Purposive behaviour trajectory

A single object trajectory is of interest for the synthesis process. If the object (hand) makes a purposive behaviour trajectory then that trajectory is task relevant. An example of this would be a hand moving towards an object. The goal is to detect the purposive trajectory as early as possible to be able to focus predictive processing in the area indicated by the direction of the trajectory.

Two basic purposive trajectories will be detected: the motion of a hand away and towards the body. Of particular interest is the point of change in the direction of the motion and to obtain the actual location and point of time of this instant. This will be used to obtain more information about a grasping activity or of pressing-a-button activity. The location at this point of time is used to focus processing on the area near the transition point to detect/recognise the object grasped and the specific button pressed. Context for these trajectories could be provided either by 3-D hand pose (hand recogniser function) or through reference to a user torso position (e.g., from the hand detector and tracker).

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).

These hand gesture trajectories are used for contextual processing within in the activity reasoning engine. For example, the 'moving hand away from body' gesture can be used to predict two specific actions: 'hand grasping an object' and 'hand putting an object down', depending on whether the hand currently is holding an object.

2.2 Distance between two objects

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. The distance measure is used in other components, e.g. the Activity Reasoning Engine.

As an extension, from distance a measure of 'mutual proximity' could be calculated that either has a non-linear value, e.g. 1/(Euclidean distance^2), which would eliminate the attentional interest value when the two objects were above some distance threshold apart, or some set of qualitative values set of qualitative values, e.g. 'very close', 'close', 'far apart',

etc. The non-linear function or the qualitative descriptors have first to be defined between requesting component and the ORG.

2.3 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.

The relationship “find objects near to each other “ has the task to find objects close to each other among a given set of objects. It uses the distance measure between two objects and a given threshold to evaluate the sub-set of the given objects that fulfil this criteria. It uses the distance calculations outlined in Section 2.2.

3 Aggregate relationships

.Aggregate (as opposed to basic) relationships are behaviours between two (or more) objects and in the spatio-temporal domain. One relevant relationship has been identified thus far for the project:

- object near trajectory of object.

3.1 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.

The object with trajectory can be assumed to be a hand (any other objects that are moving will either be held by a hand, or have been dropped). This service provides either a specific distance of one object from the other's trajectory, or a set of objects, ordered according to their distances from the trajectory. The distance value itself is calculated as outlined above.

4 Proposed methods

We propose to implement these relationships using 2 different methods:

- unary spatio-temporal relationships will use the time-delay RBF model; and
- the binary spatio-temporal relationship is implemented using a Kalman filter approach, which will be used as a competing approach to derive unary spatio-temporal relationships.

Both are outlined in more detail below.

4.1 The RBF network scheme

The Radial Basis Function (RBF) network is a two-layer, hybrid learning network [5, 6], which combines a supervised layer from the hidden to the output units with an unsupervised layer from the input to the hidden units. The network model is characterised by individual radial Gaussian functions for each hidden unit, which simulate the effect of overlapping and locally tuned receptive fields.

The RBF network is characterised by computational simplicity, supported by well-developed mathematical theory, and robust generalisation, powerful enough for real-time real-world tasks [10, 11]. The non-linear decision boundaries of the RBF network make it better in general for function approximation than the hyperplanes created by the multi-layer perceptron (MLP) with sigmoid units [9], and they provide a guaranteed, globally optimal solution via simple, linear optimisation. One advantage of the RBF network, compared to the MLP, is that it gives low false-positive rates in classification problems as it will not extrapolate beyond its learnt example set. This is because its basis functions cover only small localised regions, unlike sigmoidal basis functions which are nonzero over an arbitrarily large region of the input space.

Once training examples have been collected as input-output pairs, with the target class attached to each image, tasks can be learnt directly by the system. This type of supervised learning can be seen in mathematical terms as approximating a multivariate function, so that estimations of function values can be made for previously unseen test data where actual values are not known.

This process can be undertaken by the RBF network using a linear combination of basis functions, one for every training example, because of the smoothness of the manifold formed by the example views of objects in a space of all possible views of that object [8]. This underlies our approach, successful in previous work with RBF networks for face recognition tasks with image sequences [3], which uses an RBF unit for each training example, and single stage pseudo-inverse calculation of weights.

The time-delay RBF model

To construct a dynamic neural network, recurrent connections can be added to standard multi-layer perceptrons which then form a contextual memory for prediction over time [1, 4, 7]. These partially recurrent neural networks can be trained using back-propagation but there may be problems with stability and very long training sequences when using dynamic representations. Instead, we use a simple Time- Delay mechanism [12] in conjunction with an RBF network, which we term a TDRBF network, which we have previously shown can allow fast, robust solutions to difficult real-life problems [2]. Such a network can be created by combining data from a fixed time `window' into a single vector as input. In addition, an integration layer on the TDRBF network can be used to combine results from successive time windows to provide smooth gradations between serial actions.

4.2 Trajectory estimation with a Kalman filter

When tracking an object there is no way to influence or know the target trajectory. Hence, the task is to estimate the motion a purely stochastic process [20] (as opposed to deterministic processes that could be handled with model predictive methods). A well-known approach for the task of predicting the motion of a target object moving on an unknown trajectory is the Kalman filter [14,15,16]. The Kalman filter presents an optimal estimation of the actual state (e.g., target pose) together with its covariance matrix, which are updated iteratively using the latest measurements.

To achieve optimal settings for a broad range of velocities the gain of Kalman Filter can be adapted using the actual estimation of the acceleration. Details of the design and optimisation of this Adaptive Kalman Filter (AKF) are given in [17]. With this approach a good prediction and damping quality is achieved for smooth motion with a wide range of velocities and accelerations. Unfortunately, at a discontinuity such as a ramp-like motion, the AKF shows oscillatory behaviour with large overshoot. Experimental studies in [18] have shown that the use of a fixed Kalman gain help with discontinuities in motion.

A hybrid approach would be to combine the advantages of these two concepts: whilst object motion is smooth the AKF will perform the prediction, and where discontinuities in the motion

are detected prediction would be carried out by a Kalman Filter with fixed gain. To achieve this two problems must be solved:

- First, and most critical, is the detection of a discontinuity. The problem here is to be sensitive only to discontinuities avoiding a fast and smooth motion with a high noise level. In such cases switching could degrade the performance of the prediction.
- Second, the selection of an appropriate filter with fixed gain. The most convenient filters are the so-called $\alpha\beta$- and $\alpha\beta\gamma$-filter, which are often denoted as tracking filters. These filters, which will work only during a certain time, will be called transition filters.

A solution to these two problems has been proposed and tested using a new predictive algorithm, the Switching Kalman Filter (SKF) [19], which minimises the tracking error (caused by reliance on the estimation) by introducing a Prediction Monitor (PM). The task of the PM is to supervise the prediction error and to decide whether an error indicates a discontinuity in the target motion or not. Because there is no information of the actual target trajectory, only the measurements of the visual sensor can be used. Hence, the PM must cope with the degradation of the prediction quality due to the typically noisy visual data and react only to the actual discontinuities of the target motion. If a discontinuity is detected by the PM, the predictor is re-initialised using a transition filter and an auxiliary controller sets the control law to prevent a large overshoot. The consequence of this approach is an improved estimate of the trajectory.

Using this trajectory estimate the Euclidean distance of objects from this trajectory can be calculated. The resulting covariance matrix of the SKF trajectory estimate can be further used to adaptively change the threshold for deciding if an object is close to the trajectory. The number of steps for predicting the future trajectory can be adjusted also to target velocity or other contextual information.

From the characteristics described above, the SKF might be also used to detect purposive behaviour trajectories.

5 Relationship to framework

These relationships relate to the Pre-reasoning level of the ActIPret Demonstrator, Gesture Recogniser and Object Relation Generator. The level below (pre-attentive and attentive processes) is concerned with single objects. The Synthesis level above is concerned with an overall view of the scene.

In particular, the Gesture Recogniser component implements the basic spatio-temporal relationship of determining a purposive behaviour trajectory of hands. In principle the same methods could be used to infer these relationships for other types of objects, e.g. a person or car. Within the ActIPret Demonstrator only hands are considered as independently moving objects.

The Object Relationship Generator component implements the pure spatial relationships and the relationships between two objects. The relationship “object near trajectory of object” uses the methods proposed in Section 4 to obtain a potential space of interest (SOI) along the trajectory and to evaluate the distance of objects. Within the ActIPret Demonstrator this is done for hand and other objects.

5.1 Relationship to other work packages

Two work packages are related to this task, WP 4 and WP 6:

- **WP 4:** recognition of objects, is considered related to this WP, because relationships can control processing of object recognition. For example, the aggregate relationship

defines one or more SOIs (Spaces Of Interest) where object recognition is supposed to operate. A similar argument holds for the detection components.

- **WP 6:** attentive control, makes use of the information contained within the relationships, in particular, of relationships between objects and potential occlusions. It was planned that wishes for the resource "view" will come out of the relationships. However, due to the component-based set-up of the framework, the individual vision processing component (e.g. object detection and tracking, object recognition, hand tracker) will directly impose these view requests.

  The rationale behind this set-up is that it is only the 2-D vision components that know about which views are required to solve their tasks. To transfer this knowledge to other components would be dangerous, since the same functionality would be located in two components. On the other hand, to inform a 2-D vision processing component that there is another object that needs to be attended when calculating the view request seems a natural extension to the capability of a vision processing component. Every 2-D vision processing component needs to request views in order to take into account potentially occluding objects.

6 References

[1] J. Elman. "Finding structure in time". Cognitive Science, 14:179-211, 1990.

[2] A. J. Howell and H. Buxton. "Learning gestures for visually mediated interaction". In Proc. BMVC, pp. 508-517, Southampton, UK, 1998.

[3] A. J. Howell and H. Buxton. "Learning identity with radial basis function networks". Neurocomputing, 20:15-34, 1998.

[4] M. I. Jordan. "Serial order: A parallel, distributed processing approach". In J. L. Elman and D. E. Rumelhart, editors, Advances in Connectionist Theory: Speech. Lawrence Erlbaum, Hillsdale, NJ, 1989.

[5] J. Moody and C. Darken. Learning with localized receptive fields". In Proc. 1988 Connectionist Models Summer School, pp. 133-143, Pittsburgh, PA, 1988.

[6] J. Moody and C. Darken. "Fast learning in networks of locally-tuned processing units". Neural Computation, 1:281-294, 1989.

[7] M. C. Mozer. "Neural net architectures for temporal sequence processing". In A. S. Weigend and N. A. Gershenfeld, editors, Time Series Prediction: Predicting the Future and Understanding the Past, pp. 243-264. Addison-Wesley, Redwood City, CA, 1994.

[8] T. Poggio and S. Edelman. "A network that learns to recognize three-dimensional objects". Nature, 343:263-266, 1990.

[9] T. Poggio and F. Girosi. "Regularization algorithms for learning that are equivalent to multilayer networks". Science, 247:978-982, 1990.

[10] D. A. Pomerleau. "ALVINN: An autonomous land vehicle in a neural network". In NIPS vol. 1, pp. 305-313, San Mateo, CA, 1989.

[11] M. Rosenblum and L. S. Davis. "An improved radial basis function network for autonomous road-following". IEEE Trans.Neural Networks, 7:1111-1120, 1996.

[12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. "Phoneme recognition using time-delay neural networks". IEEE Trans. Acoustics, Speech, & Signal Processing, 37:328-339, 1989.

[13] Markus Vincze, Minu Ayromlou, Wolfgang Ponweiser, Michael Zillich: “Edge Projected Integration of Image and Model Cues for Robust Model-Based Object Tracking”; Int. J. of Robotics Research 20(7), pp.533-552, 2001.

[14] R.E. Kalman: “A new approach to linear filtering and prediction problems", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 1960.

[15] E.D. Dickmanns, V. Graefe: “Dynamic Monocular Machine Vision and Applications of Dynamic Monocular Machine Vision”; Machine Vision and Applications, pp.220-261, 1988.

[16] P.I. Corke: “Visual Control of Robots: High Performance Visual Servoing”, Research Studies Press (John Wiley), 1996.

[17] P. Krautgartner, M. Vincze: “Performance Evaluation of Vision-Based Control Tasks," IEEE ICRA, pp. 2315-2320, Leuven, May 12-15, 1998.

[18] S. Chroust, E. Zimmerl, M. Vincze: “Pro and cons of control methods of visual servoing", Proc. 10th Int. Workshop on Robotics in Alpe-Adria-Danube Region, 2001.

[19] S. Chroust, M. Vincze: “Improvement of the Prediction Quality for Visual Servoing with a Switching Kalman Filter”, Int. J. of Robotics Research, submitted.

[20] P.S. Maybeck: ``Stochastic Models, Estimation and Control, volume 2", Academic Press, 1982.