Developing Task-Specific RBF Hand Gesture Recognition

A. Jonathan Howell
Kingsley Sage
Hilary Buxton

CSRP 562

April 2003

ISSN 1350–3162

UNIVERSITY OF
SUSSEX
AT BRIGHTON

Cognitive Science Research Papers

Developing Task-Specific RBF Hand Gesture Recognition

A. Jonathan Howell, Kingsley Sage, and Hilary Buxton

School of Cognitive and Computing Sciences,
University of Sussex, Brighton BN1 9QH

{jonh,khs20,hilaryb}@cogs.susx.ac.uk

Abstract. In this paper we develop hand gesture learning and recognition techniques to be used in advanced vision applications, such as the ActIPret system for understanding the activities of expert operators for education and training. Radial Basis Function (RBF) networks have been developed for reactive vision tasks and work well, exhibiting fast learning and classification. Specific extensions of our existing work to allow more general 3-D activity analysis reported here are: 1) action-based representation in a hand frame-of-reference by pre-processing of the trajectory data; 2) adaptation of the time-delay RBF network scheme to use this relative velocity information from the 3-D trajectory information in gesture recognition; and 3) development of multi-task support in the classifications by exploiting prototype similarities extracted from different combinations of direction (target tower) and height (target pod) for the hand trajectory. Finally, a discussion and conclusions for system integration are given.

1 Introduction

Neural network techniques are a powerful, general approach to pattern recognition tasks based on learning, and there are a variety of different methods (for an introduction see [3]). The classical networks do not include a time dimension so they have to be adapted to deal with dynamic scene analysis. Some extended models have internal time like the partially recurrent networks of Elman [6] and Jordan [13]. Others have external time like the time-delay networks described below. Time can be explicitly represented in the architecture at the network level using the connections or can be represented at the neuron level, including the recently developed 'spiking networks'. These model the intrinsic temporal properties of biological neurons, which fire with a pattern of pulses or spikes (for review see [8]). However, these fully dynamic networks have yet to be applied in visual behaviour analysis, although a start has been made [7]. A widely used model is the Time Delay extension of classical Radial Basis Functions (TDRBFs). Networks of this kind have been shown to exhibit rapid training and online processing in tasks such as gesture recognition [10].

Learning in a vision system can be at the level of object models, their movements and actions, and how to control views and processing in the system.

Our work on appearance-based approaches using RBF nets suggests they are very learnable and robust in comparison with structural approaches for general object categorisation on real-world tasks such as face recognition [9,11]. Natural deformable objects are difficult to specify and so are their movements and actions, so adaptive methods are required. At the heart of a visual learning system is the ability to find the relevant mapping from observable or derivable attributes of image(s) onto the visual categories we require for real-world tasks. In the paper, we show how appearance-based techniques can be extended to 3D gesture recognition, based on velocities recovered from hand trajectories, for the ActIPret system.

In the following, the TDRBF model is first described in section 2. Then in sections 3 and 4, the dataset and some results from the generalisation of the generic gesture models and the initial tasks on gesture direction recovery are described. In section 4.2, we consider how robust the recognition is under noise in training and testing trajectory sequences. In section 4.3, the extensions of the task-specific processing are described, together with preliminary results from categorising the target tower and pod to be grasped in the gesture interpretation. Finally, in sections 5 and 6, the implications of the work for task control and system integration are then discussed with conclusions and suggestions for further work.

2 Time Delay RBF Network

The RBF network is a two-layer, hybrid learning network [14,15], 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. Supported by well-developed mathematical theory, the model provides rapid computation and robust generalisation, powerful enough for real-time, real-life tasks [18,19]. The nonlinear decision boundaries of RBF networks make better general function approximations than the hyperplanes created by the multi-layer perceptron (MLP) with sigmoid units [16], 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 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 learned 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 [17]. This underlies successful previous work with RBF networks for face recognition from video sequences [11], which uses an RBF centre for each training example, and rapid pseudo-inverse calculation of weights. An important factor in this approach is the flexibility of the RBF network learning approach, which allows formulation of the training in terms of the specific classes of data to be distinguished. For example, extraction of identity, head pose and expression information can be performed separately on the same face training data to learn a computationally cheap RBF classifier for each separate recognition task [5, 12].

To extend this research to support *visual interaction*, generic gesture models are developed here for the control of attention in gesture recognition. In previous work a time-delay variant of the Radial Basis Function (TDRBF) network recognised pointing and waving hand gestures in image sequences [10]. This network is created by combining data from a fixed time 'window' into a single vector as input. Characteristic visual evidence is automatically selected during the adaptive learning phase, depending on the task demands. A set of interaction-relevant gestures were modelled and exploited for reactive on-line visual control. These were then interpreted as user intentions for live control of an active camera with adaptive view direction and attentional focus. For ActIPret, some of the ideas for zooming in on activities can still be exploited. Also the gesture recognition is an excellent predictive cue for many of the actions and activities in our ActIPret scenarios. At the earlier levels of processing, but particularly in the gesture recognition, reactive behaviour is important for both camera movement and invoking further 'attentional' processing. The scheme is adapted here to accept 3-D hand trajectories for predictive gesture recognition. The gesture recognition uses tri-phasic gesture detectors as in our previous work on predictive control [12].

3 Gesture Data

The gesture data used for the experiments in this paper was the *Terminal Hand Orientation and Effort Reach Study* Database created by Human Motion Simulation at the Center for Ergonomics, University of Michigan, USA. 3-D hand trajectory data was collected from 22 subjects of varying gender, age, and height. Nineteen of the subjects were right-handed and two were left-handed. 210 target locations and hand orientations were used, giving a total number of 4,410 trials and the 8,820 reach movements.

Fig. 1 shows the target system for the HUMOSIM hand trajectory data. Four towers were used, from 45° left of the subject to 90° right, each of which had three 'pods' as targets. There is further variation in the targets, as each of the pods has five cubes, each of which can use four hand orientations. For the experiments in this paper, we consider only tower/pod combinations (12 in all). Each trial produced a file of 3-D coordinates for two points (six values each time step) on the subject's hand. For each trial, data was collected at 25Hz for a sequence consisting of five distinct phases:

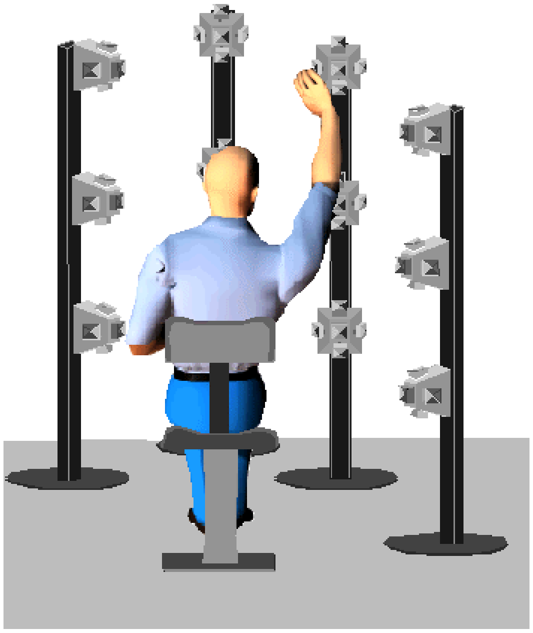

Fig. 1. The target system for the HUMOSIM hand trajectory data.

- Start with a static hand placed at a 'home location' on the subject's leg, followed by:
- A movement toward the target, which we term *get*;
- A static phase while the hand is at the target;
- A second movement, away from the target, which we term *return*;
- A final static phase at the home location.

Each resulting datafile contained 80–135 timesteps. The 3-D location data was pre-processed by differencing it from one time step to the next (relative motion or velocity data).

4 Method

To train the TDRBF network, we used a fixed time delay length of six time steps, and segmented the training data automatically according to the level of relative motion within successive time delay segments. Based on the definition of the trial data above, we assume two distinct gesture movements are contained in each hand trajectory data file, with static periods in between. We impose three phases within each of these movements: a *pre-phase*, at the start of movement a *mid-phase*, at the midpoint between start and end of movement and a *post-phase*, at the end of movement. Adding an extra class for stasis, or no movement, gives seven classes in all:

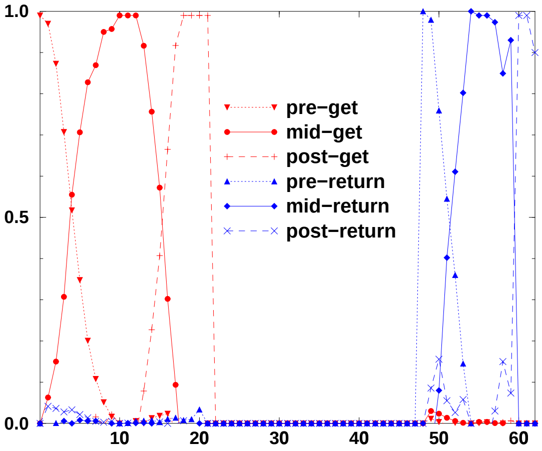

Fig. 2. Gesture phase classification for a TDRBF network trained with data from Tower 0 (45° left of subject) when tested with a single complete hand trajectory also from Tower 0, showing values for output units for each gesture phase class (y-axis) at each time step (x-axis).

- *pre-get, mid-get, post-get*
- *pre-return, mid-return, post-return*
- *stasis*

The three-phase structure for gesture classification is based on previous work [12], where we found that breaking gestures down into smaller parts allowed more reliable recognition as well as supporting prediction. The strategy we developed was to only accept specific plausible sequences of phases as real gestures, eg. the pre-phase needed to be observed before the mid-phase, and confirmed by the post-phase to support appropriate attention frame shifts for visual interaction.

To test the trained TDRBF network, we present successive time-delay vectors over the complete trajectory file, giving a series of outputs representing confidence in each of the six gesture phase classes. Time delay segments with very low levels of relative motion are identified automatically and ignored by the TDRBF network in the test phase, being immediately classified as static.

4.1 Parsing Network Output

Our previous work with the HUMOSIM hand trajectory data [4] was able to show both RBF and HMM methods could learn the individual gesture phases, for example, see Fig. 2 for typical RBF classification. In this example, smooth transitions can be seen between phase classes, and all time steps are correctly classified, even for people and timesteps not included in the training set. We can

Table 1. Generalisation over hand trajectory angle (around y-axis) for TDRBF networks trained with a range of tower data, from Tower 0 (45° left) to Tower 3 (90° right). The '% Correct' values show the proportion of test trajectories where gesture phases were correctly interpreted at every time step of the entire trajectory.

Training Towers		Test Tower, % Correct			
		0	1	2	3
Single	0	100	66	0	0
	1	100	100	0	0
	2	0	0	94	15
	3	0	0	41	94
Consecutive	0 + 1	100	100	0	0
	2 + 3	0	0	100	94
Alternate	0 + 2	83	83	88	15
	1 + 3	91	94	17	94
All		100	94	100	94

go on from these results to use these transitions to accurately signal progress through the gesture phases, and devise a metric for assessing how well the network identifies the overall gestures.

The measure for correct classification we use in this paper is that a complete series of valid gesture phase transitions has to be observed during the test output from a whole trajectory, ie. *pre-get* first, followed by *mid-get*, *post-get*, *pre-return*, *mid-return* and *post-return* (with arbitrary static periods before, middle and after). If this exact sequence is observed, the overall classification is deemed correct. Any other transition, eg. *pre-get* to *post-get*, invalidates the classification of the entire sequence. Although this might seem an unduly harsh measure of success, in practise it is quite hard to 'repair' a classification sequence for a test trajectory once incorrect entries have been entered. One strategy which can help is to use an assumption of temporal continuity, where observed transitions are only accepted after a certain number of consecutive, identical phase classifications, which can exclude transitory mis-classifications.

The advantage of this approach to monitoring the network output is that a complete breakdown of gesture phase start and end positions can be provided for the test trajectory, which is very useful for an online component of a larger vision system.

5 Results

The experiments presented here are in three phases: the first to determine generalisation characteristics over angle of hand trajectory for the TDRBF network, the second to determine how this generalisation is affected by varying levels of random noise, and the third to develop the training of multiple tasks for the network, such as 'which gesture *and* which tower is the hand aiming for?'

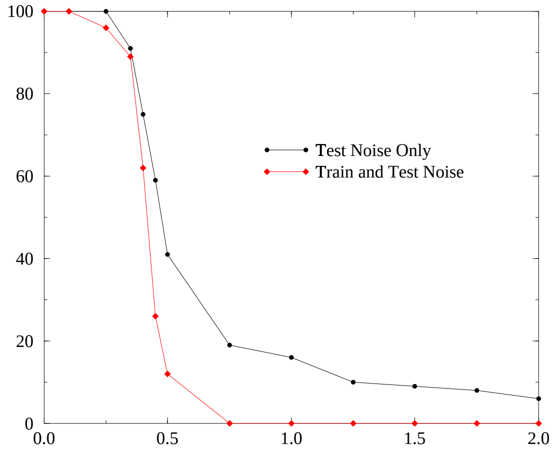

Fig. 3. Classification performance for TDRBF network trained and tested with targets in Tower 0 (45° left), with varying amounts of RMS noise added to the trajectory positions (x-axis, values in cm). The y-axis shows the proportion of test trajectories where gesture phases were correctly interpreted at every time step of the entire trajectory.

5.1 Generalisation over Trajectory Angle

To test generalisation over angle of hand trajectory (around y-axis), we trained and tested the TDRBF network with combinations of data from the four towers (from 45° left to 90° right of the front of the subject), keeping the pod position (x-axis variation) constant. The test set contained trajectories from single towers, but the training data used one of: a single tower, two adjacent towers (eg. 0 and 1), two alternate towers (eg. 0 and 2), or all four towers.

The results for these tests are presented in Table 1. These show that while the networks trained a single tower do not generalise particularly well to other towers, a reasonable performance can be obtained by combining training data from two or more towers.

5.2 Adding Noise

The 'Flock of Birds' magnetic sensor used to record hand coordinates for the HUMOSIM hand trajectory data used in this paper is very highly accurate, giving values to a fraction of a mm. In order to simulate less constrained data, such as might be extracted by visual methods, we apply varying levels of random noise to the coordinate values. This is not to simulate consistent errors, eg. miscalibration, where constant offsets will be observed, but transitory errors, eg. due to uncertainty or occlusion, which will be more common in visual hand tracking.

Noise was added to the 3-D location as random values with a normal distribution with mean zero. The level of noise was determined as an root mean

Table 2. Performance for TDRBF networks trained for multiple tasks: 'Gesture' has six gesture phase classes, 'Tower Position' has four position classes (from 45° left to 90° right) and 'Pod Position' has three (from 45° above to 45° below). The '% Correct' values show the proportion of test trajectories where combinations of gesture phase, tower and pod positions were correctly interpreted at every time step of the entire trajectory.

Trained Tasks	Classes	Test Tower, % Correct			
		0	1	2	3
Gesture + Tower Position	24	100	100	82	84
Tower + Pod Position	12	100	100	100	100
Gesture + Tower + Pod	72	100	88	82	89

square (RMS) value, for example, a noise level of 1.0cm RMS produced random values between about ±1.2cm. To produce a smoother variation of values, each vector of random values had individual values averaged with its neighbour.

Fig. 3 shows how classification performance deteriorates as noise increases. Two test arrangements are shown, each with a separate line on the graph. The first trains the TDRBF network without noise, and tests with varying noise. The second both trains and test with an equal level of noise. The TDRBF network performs slightly better when trained without noise, but overall the limit for useful performance would appear to be around 0.5cm RMS noise (on every axis, every time step).

5.3 Multiple Tasks

In this section, we consider how to learn multiple tasks, such as 'which gesture *and* which tower is the hand aiming for?' In previous work, we have shown that separate RBF networks can learn different tasks (face identity, expression, head pose) from the same training data through altering the training signal [5], and that one TDRBF network could learn both gesture and identity by giving different classes to gestures from different individuals [10].

Three tasks can be learnt from the HUMOSIM hand trajectory data:

- *'Which gesture phase?'*, using six gesture phase classes,
- *'Which tower position is the hand aiming for?*, using four position classes (from 45° left to 90° right),
- *'Which pod position is the hand aiming for?*, using three position classes (from 45° above to 45° below).

As an example of combining these tasks, in order to learn both gesture and tower position, we train a network with individual phase classes for each tower. This uses six phases for each of the four towers, 24 classes in all. The results for networks trained on three combinations of these tasks are shown in Table 2, including one trained with all three tasks, which required 72 classes.

Table 2 shows that minimal reduction in performance is observed, compared the the network trained with all towers in Table 1, whilst useful extra information is provided alongside the gesture output.

6 Summary

In this paper we have shown:

- The TDRBF network can learn individual gesture phases from 3-D hand trajectories collected from a magnetic sensor.
- An efficient method for parsing network output and measuring correct classification over an entire hand trajectory file has been developed.
- The 3-D coordinate representation limits trajectory angle generalisation due to values moving from one axis to another as the angle is varied, but this can be overcome by explicit training for several target positions.
- Although the magnetic sensor hand trajectory data is very constrained, the trained TDRBF network was shown to be tolerant to a fairly high level of instantious random variations in coordinates (around 0.5cm RMS noise on every axis, every time step).
- An efficient method for training the TDRBF network to learn multiple tasks, such as 'which gesture *and* which tower is the hand aiming for?' has been shown.

7 Conclusions

We have developed a task-specific Gesture Recognition component above and shown that this approach yields promising results, using hybrid learning in the TDRBF. Although the first layer of weights learned during training are unsupervised in the TDRBF, the mapping of class prototypes onto the task-relevant classes needs to be supervised and a seven phase structure was imposed. Further task-specific sub-classes to identify towers, pods and grasp were also defined in the extensions given in section 4.3. Performance on the learning and generalisation tasks was simply supported by rapid weight training in the RBF network. This kind of class-based processing [1, 2] has many advantages, including the possibility of learning sufficient information from a single example by exploiting class similarities [20].

The TDRBF used here was coded in C and adapted from previous work in the ISCANIT project [12]. This kind of multi-task model can be generalised to select any systematic variations known to exist in the dataset as the sub-classes, which can then support activity analysis in the full system [4]. It is premature to give full QoS and computational costs for the ActIPret System but these will be established in future work. As in the discussion above, there is great potential for task-specific processing using the TDRBF to supply fast, reactive results.

Acknowledgements

The authors gratefully acknowledge the invaluable help provided by the Laboratory for Human Motion Simulation (HUMOSIM) at the University of Michigan, USA in allowing us access to their 'Terminal Hand Orientation and Effort Reach Study, 2000' hand trajectory database; also framework concepts and funding from the EU ActIPret project.

References

1. R. Basri. Recognition by prototypes. *International Journal of Computer Vision*, 19:147–168, 1996.
2. D. J. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 1996.
3. C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
4. H. Buxton, A. J. Howell, and K. Sage. The role of task control and context in learning to recognise gesture. In *Cognitive Vision Workshop*, Zürich, Switzerland, 2002.
5. S. Duvdevani-Bar, S. Edelman, A. J. Howell, and H. Buxton. A similarity-based method for the generalization of face recognition over pose and expression. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 118–123, Nara, Japan, 1998.
6. J. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
7. J. Feng, Y. L. Sun, and H. Buxton. Training the integrate-and-fire model with the Informax Principle II. *IEEE Transactions on Neural Networks*, 14:accepted, 2003.
8. W. Gerstner. Time structure of the activity in neural networks. *Physical Review*, E 51:738–758, 1995.
9. A. J. Howell and H. Buxton. Invariance in radial basis function networks in human face classification. *Neural Processing Letters*, 2:26–30, 1995.
10. A. J. Howell and H. Buxton. Learning gestures for visually mediated interaction. In *British Machine Vision Conference*, Southampton, UK, 1998.
11. A. J. Howell and H. Buxton. Learning identity with radial basis function networks. *Neurocomputing*, 20:15–34, 1998.
12. A. J. Howell and H. Buxton. Time-delay RBF networks for attentional frames in visually mediated interaction. *Neural Processing Letters*, 15:197–211, 2002.
13. 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, 1989.
14. J. Moody and C. Darken. Learning with localized receptive fields. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of 1988 Connectionist Models Summer School*, pages 133–143, 1988.
15. J. Moody and C. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281–294, 1989.
16. T. Poggio and S. Edelman. A network that learns to recognise three-dimensional objects. *Nature*, 343:263–266, 1990.
17. T. Poggio and F. Girosi. Regularisation algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

18. D. A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 305–313, 1989.
19. M. Rosenblum, Y. Yacoob, and L. D. Davis. Human emotion recognition from motion using a radial basis function network architecture. *IEEE Transactions on Neural Networks*, 7:1121–1138, 1996.
20. T. Vetter and T. Poggio. Image synthesis from a single example image. In *European Conference on Computer Vision, Lecture Notes in Computer Science*, volume 1065, pages 652–659, Cambridge, UK, 1996.