The Acquisition and Use of Interaction Behaviour Models

Neil Johnson, Aphrodite Galata and David Hogg
School of Computer Studies
The University of Leeds, Leeds LS2 9JT, UK
{neilj, afro, dch}@scs.leeds.ac.uk http://www.scs.leeds.ac.uk/vislib/

Abstract

Providing a machine with the ability to learn and use models of natural interaction is a challenging and largely unaddressed problem. A framework is developed enabling both the acquisition of interaction behaviours from the observation of humans, and the use of the acquired behaviour models to simulate a plausible partner during interaction. Statistically based interaction behaviour models are acquired automatically from the observation of interacting humans. Interaction with a virtual human is achieved using the model together with a stochastic tracking algorithm. Experimental results demonstrate the generation and use of the model for a simple human interaction.

1. Introduction

In recent years many researchers have become interested in the development of techniques to allow a more natural form of interface between the user and the machine, utilising interactive spaces equipped with cameras and microphones where such techniques can be developed and tested (see, for example, [11]). In achieving this goal, it is essential that the machine is able to detect and recognise a wide range of human movements and gestures, and this has been a principal avenue of research (see, for example, [2, 3, 4, 5, 8, 10, 13]). We wish to investigate the provision of natural user-machine interaction from a different standpoint, allowing the machine to acquire models of behaviour from the extended observation of interactions between humans, and using these acquired models, to equip a virtual human with the ability to interact in a natural way. This paper describes a novel approach to interaction modelling, using a relatively simple interaction for our experiments - that of shaking hands.

Training data is acquired by automatically locating and tracking individuals within a video corpus of typical interactions. Interactions are modelled by means of a previously developed, statistically based, modelling scheme which allows behaviours to be learnt from the extended observation of image sequences [7]. Interaction is represented as the joint behaviour of object silhouettes just as Kakusho *et al.* consider joint behaviour in their recognition of social dancing [8]. The model is enhanced to enable the extrapolation of realistic future behaviours.

Having learnt a generative interaction model from the observation of image sequences containing individuals performing simple interactions, interaction with a virtual human is investigated. The model provides information about how an interaction may proceed in the form of a Markov chain, and interaction with a virtual human is achieved by following a path through this chain such that, as the interaction proceeds, the real human's silhouette continually matches half of the joint silhouette represented within the model. In a Bayesian approach to interaction tracking, multiple interaction hypotheses are stochastically propagated through the model by a method based on Isard and Blake's CONDENSATION algorithm [6].

2. Acquiring training data

The first stage of the behaviour modelling process involves the acquisition of ordered sets of training data representing instances of the human interactions to be learnt. Each of these sets describes the *state* of interacting individuals at regular intervals throughout an interaction. Each state consists of the instantaneous spatial configuration together with its first derivative. It is assumed that individuals are viewed such that their configuration can be described by left-hand and right-hand shapes, together with their separation and relative size. In our experiments, training data is acquired by tracking interacting individuals in an uncluttered scene, viewed with a static camera (see Figure 1).

Tracking is accomplished using an extension of a silhouette extraction method used by Baumberg and Hogg [1] to collect training data. Using background image subtraction to locate moving objects, this system provides a basic tracker requiring careful use. Object shape is represented by the n control points of a closed uniform B-spline approx-

Figure 1. Image from a simple interaction.

imating the silhouette boundary:

$$\mathbf{S} = (x_1, y_1, x_2, y_2, \ldots, x_{n-1}, y_{n-1}, x_n, y_n)\,, \qquad (1)$$

where control points are evenly spaced around the silhouette and are ordered relative to a consistent point of reference which also defines the object's position $(X = x_1, Y = y_1)$. The method for the location of this reference point has been enhanced from [1] to allow the top of an individual's head to be more accurately located. This enhancement involves local adjustment of the reference point such that it coincides with the locally highest part of the silhouette. Figure 2 illustrates the shape representation, showing control points as circles with reference points shown filled. The tracker provides frame by frame updates to the shape $\mathbf{S}$ and height h (both in image plane coordinates) of uniquely labelled objects.

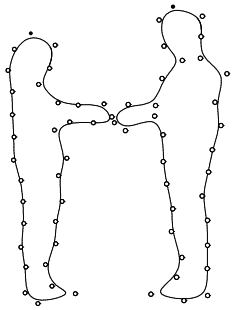


Figure 2. Spline-based shape representation.

Interaction is represented as the joint behaviour of object silhouettes, encoding their shapes, separation and relative scale implicitly. At any particular instant the joint configuration is described by a *combined shape vector* $\mathbf{C} \in \Re^{4n+2}$:

$$\mathbf{C} = \left(\mathbf{S}^L, \mathbf{S}^R, d, s\right), \qquad (2)$$

where $\mathbf{S}^L$ and $\mathbf{S}^R$ are the shape vectors of the left-hand and right-hand individuals. $\mathbf{S}^L$ and $\mathbf{S}^R$ are transformed into actor centred coordinates and scaled by their respective heights (h^L and h^R) such that all components are in the interval $[0,\ 1]$. Normalisation of combined shape vectors in this way enables the integration of data from different sequences. d and s are components describing relative horizontal actor separation and relative actor scale, defined as follows:

$$d = \frac{X^R - X^L}{h^L}, \qquad (3)$$

$$s = \frac{h^R}{h^L}. \qquad (4)$$

The tracker processes frames at a fixed rate, resulting in sequences of combined shape vectors representing a temporally uniform sampling of the interaction. Due to inaccuracies in the tracking process, combined shape vectors will be subject to high frequency noise which is minimised by smoothing vectors over a moving temporal window.

The temporal evolution of an interaction is represented by an ordered set of *state vectors* $\mathbf{F}_t \in \Re^{8n+4}$, consisting of the combined shape vector $\mathbf{C}_t$ and its scaled first derivative $\lambda\dot{\mathbf{C}}_t$, approximated by the difference in combined shape vectors between successive frames:

$$\mathcal{F} = \{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_m\}, \qquad (5)$$

$$\mathbf{F}_t = \left(\mathbf{C}_t, \lambda\dot{\mathbf{C}}_t\right). \qquad (6)$$

In our experiments, the training data consisted of 13 handshake sequences recorded at 25 frames per second. Silhouettes were represented by B-splines with 32 control points resulting in 130-dimensional combined shape vectors. Smoothing was performed using a window of width $w = 5$. Training data sets $\mathcal{F}_j$ of 260-dimensional state vectors were generated using $\lambda = 10$ to scale the differential components.

3. Learning object behaviour models

Having acquired training data sequences, an interaction model representing the range of observed behaviours is learnt. The model is constructed in two stages. Initially, a probabilistic model for the distribution of state vectors is learnt from the training data. This is then used as the basis for a higher level model for the distribution of temporal sequences of state vectors - the behaviour model. The method is described in detail in Johnson and Hogg [7], where the motion behaviour of pedestrians is modelled for event recognition purposes.

The probability density function over the state vector space ($\Re^{8n+4}$) is modelled by the distribution of *prototype vectors* placed by an iterative vector quantisation [9], implemented by a robust competitive learning neural network [9, 12]. Vector quantisation of the state vectors $\mathbf{F}_t$ from the

training data sets $\mathcal{F}_j$ results in a set of u state prototypes $\bar{\alpha}_i$:

$$\mathcal{A} = \{\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_{u-1}, \bar{\alpha}_u\}. \quad (7)$$

State prototypes and a temporal pattern formation strategy are used to encode different length state vector sequences. The resulting *behaviour vectors* thus represent behaviour histories. The temporal pattern formation is achieved by considering the proximity of successive training vectors from $\mathcal{F}$ to the state prototypes $\mathcal{A}$. The proximity p_{it} of a training vector $\mathbf{F}_t$ to a state prototype $\bar{\alpha}_i$ decreases linearly from one to zero as the distance between them increases from zero to the maximum separation within the unit hypercube state space:

$$p_{it} = 1 - \frac{|\mathbf{F}_t - \bar{\alpha}_i|}{\sqrt{8n+4}}. \quad (8)$$

Each component z_i of a behaviour vector $\mathbf{G} \in \Re^u$ corresponds to a state prototype $\bar{\alpha}_i$ and is initially zero ($z_{i0} = 0$). Successive behaviour vectors are calculated by applying a conditional decay operator to each component:

$$\mathbf{G}_t = \left(z_{1t}, z_{2t}, \ldots, z_{(u-1)t}, z_{ut}\right), \quad (9)$$

$$z_{it} = \begin{cases} p_{it} & \text{if } p_{it} > \gamma z_{i(t-1)} \\ \gamma z_{i(t-1)} & \text{otherwise,} \end{cases} \quad (10)$$

where γ is a decay constant in the interval $(0,\ 1)$. This results in the formation of an ordered set of behaviour vectors $\mathbf{G}_t$:

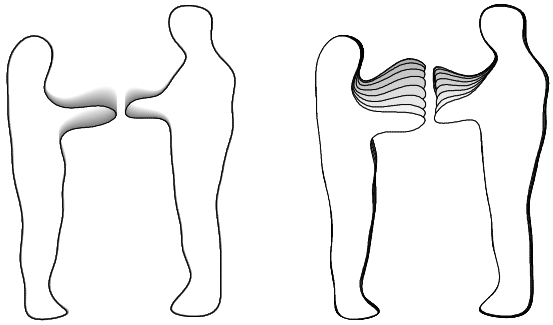
$$\mathcal{G} = \{\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_m\}. \quad (11)$$

The form of Equation 10 allows behaviour vectors to retain a *trace* of the closest proximity between each prototype and previously presented training vectors, thus forming a temporal sequence representation. The *memory* of this representation is governed by the decay constant γ. It is possible to draw parallels between this method and the motion-history image formation of Bobick and Davis [2].

The probability density function over the behaviour vector space ($\Re^u$) is again modelled by the distribution of prototype vectors. Vector quantisation of the behaviour vectors $\mathbf{G}_t$ from sets $\mathcal{G}_j$ results in a set of v behaviour prototypes $\bar{\beta}_i$:

$$\mathcal{B} = \left\{\bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_{v-1}, \bar{\beta}_v\right\}. \quad (12)$$

In our experiments, a set $\mathcal{A}$ of 300 260-dimensional state prototypes were learnt from 1,000,000 iterations of vector quantisation of vectors from the training data sets $\mathcal{F}_j$. The training data sets $\mathcal{F}_j$ and the set $\mathcal{A}$ of 300 state prototypes were used to generate sets $\mathcal{G}_j$ of behaviour vectors using a decay constant of $\gamma = 0.995$. A set $\mathcal{B}$ of 300 300-dimensional behaviour prototypes were learnt from 1,000,000 iterations of vector quantisation of vectors from the behaviour vector sets $\mathcal{G}_j$.



(a) (b)

Figure 3. Behaviour modelling results.

Figure 3(a) illustrates one of the prototypes underlying the state distribution. The differential components are used to generate the brief motion history shown. Figure 3(b) shows the last few frames of an interaction sequence which maps closely to one of the prototypes underlying the behaviour distribution.

4. Behaviour extrapolation

The behaviour models developed are capable of behaviour recognition and typicality assessment (see Johnson and Hogg [7] for more details), but have limited generative capabilities due to the non-invertible nature of the decay operator (Equation 10). Behaviour prototypes cannot be employed to generate the sequences they represent, although the *current* state can be approximated by finding the state prototype corresponding to the highest valued component. Behaviour extrapolation is achieved by the addition of a state based extrapolation scheme, the parameters of which are derived during a further learning phase.

Behaviour extrapolation is performed by traversing a Markov chain which has a single state corresponding to each behaviour prototype. Extrapolations are generated from the current chain state, which can be identified using the behaviour model for recognition, and proceed through the chain until the end state is reached. Traversal of the chain results in the production of the current state prototypes associated with each visited chain state. Since each behaviour prototype represents a behaviour history, the superimposed chain is more strongly Markovian than if the chain were superimposed on the state prototypes, thus forming a more powerful extrapolation model.

The Markov chain is defined by a set of $v+1$ states e_i:

$$\mathcal{E} = \{e_1, e_2, \ldots, e_v, e_{v+1}\}, \quad (13)$$

together with the transition probabilities $P(e_j \text{ at } r+1 \mid e_i \text{ at } r)$ where r denotes the extrapolation step. Each state corresponds to a behaviour prototype ($e_i \mapsto \bar{\beta}_i$) except

e_{v+1} which represents the *end state*. For convenience, the form $\bar{\alpha}(e_i)$ will be used to represent the state prototype corresponding to each state e_i of the Markov chain. At each extrapolation step, a successor state is selected by either sampling from the transition distribution, or identifying the most probable successor.

The state transition probabilities are estimated from the relative frequency of transitions between behaviour prototypes observed in the training data, taking the closest behaviour prototype at each time instant. Only transitions causing state change are considered.

Traversing such a chain to perform behaviour extrapolation results in an ordered set of state vectors $\bar{\alpha}(e_{i_r})$ associated with visited chain states:

$$\mathcal{Q} = \left\{ \bar{\alpha}(e_{i_0}), \bar{\alpha}(e_{i_1}), \ldots, \bar{\alpha}(e_{i_k}) \right\}, \qquad (14)$$

where the time interval between successive state vectors is initially unspecified, and e_{i_0}, the initial chain state, is identified using the behaviour model. To ensure a smooth join between previous behaviour and the extrapolation, $\bar{\alpha}(e_{i_0})$ is replaced by the current interaction state $\mathbf{F}_t$.

To generate an output state vector sequence at video frame rate, an interpolant of $\mathcal{Q}$ must be sampled. Since change in combined shape may be non-linear, a (cubic) Hermite interpolation is used. Assuming zero acceleration, the time interval T_r between successive state vectors can be approximated by the mean speed of combined shape vectors from $\bar{\alpha}(e_{i_r})$ and $\bar{\alpha}(e_{i_{r+1}})$:

$$T_r = 2\frac{|\mathbf{C}_{r+1} - \mathbf{C}_r|}{|\dot{\mathbf{C}}_r| + |\dot{\mathbf{C}}_{r+1}|}. \qquad (15)$$

The Hermite interpolant is defined by the endpoints $\mathbf{C}_r$ and $\mathbf{C}_{r+1}$ and tangent vectors $\dot{\mathbf{C}}_r$ and $\dot{\mathbf{C}}_{r+1}$ (scaled by T_r). Using the approximate time intervals and interpolants between successive vectors, an ordered set of state vectors can be produced by sampling at data frame rate:

$$\mathcal{P} = \{\mathbf{F}_{t+1}, \mathbf{F}_{t+2}, \ldots, \mathbf{F}_{t+l}\}. \qquad (16)$$

In the absence of a fragment of behaviour from which to extrapolate, entirely hypothetical sequences can be generated using an *initial state distribution* to select the initial state e_{i_0}:

$$\mathcal{S} = \{\pi_1, \pi_2, \ldots, \pi_{v-1}, \pi_v\}, \quad \pi_i = P(e_i \text{ at } r = 0). \qquad (17)$$

The selection of a state from $\mathcal{S}$ can be based on sampling from the distribution, or identifying the most probable start state. $\mathcal{S}$ is approximated from the relative frequency of starting at particular behaviour prototypes in the training data.

In our experiments, a 301-state Markov chain was associated with the 300 behaviour prototypes $\mathcal{B}$, and trained during a further learning phase. Figure 4 illustrates the combined shape components of the first few frames of two extrapolations generated using this chain.

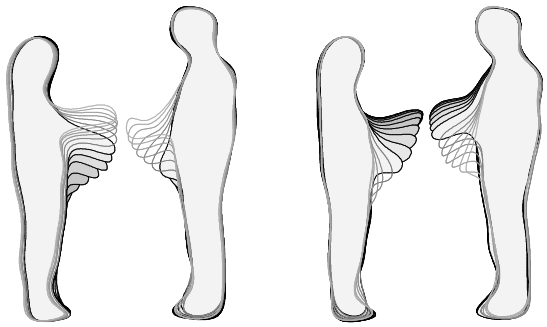


Figure 4. Behaviour extrapolation results.

5. Performing virtual interaction

A modelling framework has been developed which enables the analysis and generation of interaction behaviour from the observation of characteristic behaviours. The same models may also be used to simulate the evolving shape of a plausible partner during an interaction with a person. Two approaches to this problem are presented which correspond to subtly different objectives.

As well as allowing behaviour extrapolation, the Markov chain represents the space of learnt behaviour sequences. If such chains are learnt from a fair sample of the interaction population then any natural interaction will follow one of the possible paths through the chain. The virtual human's behaviour can therefore be entirely defined by the Markov chain. Natural interaction with a human is achieved by providing responses such that the resulting sequence of state vectors forms a valid path through the chain. Since no behaviour recognition is required, the chain is used in isolation from the behaviour models which it enhanced.

5.1. Single hypothesis propagation

One approach to performing virtual interaction is to propagate a single *interaction state hypothesis* $\mathcal{H}_t$ through the Markov chain, using the evolving shape of the real human to choose the start state and the transitions when required. $\mathcal{H}_t$ is a pair $(\mathbf{F}_t, f_t)$ where $\mathbf{F}_t$ is a state vector and f_t identifies the current chain state. At each time instant t, the scaled shape vector $\mathbf{S}_t^H$, position (X_t^H, Y_t^H) and height h_t^H of the human are extracted from the current image as described in §2. The extent to which a hypothesis $\mathcal{H}_t = (\mathbf{F}_t, f_t)$ is consistent with the current shape of the real actor $\mathbf{S}_t^H$ is given by the Euclidean distance between shape vectors:

$$E(\mathbf{F}_t, \mathbf{S}_t^H) = \min\left\{ |\mathbf{S}_t^L - \mathbf{S}_t^H|, |\mathbf{S}_t^R - \mathbf{S}_t^H| \right\}, \qquad (18)$$

where $\mathbf{S}_t^L$ and $\mathbf{S}_t^R$ are derived from $\mathbf{F}_t$, and the minimisation also identifies the human's position within the interaction.

Interaction with a virtual human is achieved with the following algorithm:

1. Select the initial hypothesis $\mathcal{H}_0$ from the set $\mathcal{X}_0$ of all potential initial hypotheses such that the *error* $E(\mathbf{F}_0, \mathbf{S}_0^H)$ is minimised. The potential hypotheses $\mathcal{X}_0$ are taken from *valid* initial chain states where $\pi_j \neq 0$.
2. Produce the virtual human's response $\mathbf{S}_t^V$ from $\mathcal{H}_t$.
3. Select the future hypothesis $\mathcal{H}_{t+1}$ from the set $\mathcal{X}_{t+1}$ of all potential future hypotheses such that the *error* $E(\mathbf{F}_{t+1}, \mathbf{S}_{t+1}^H)$ is minimised. The potential hypotheses $\mathcal{X}_{t+1}$ are extrapolations at time $t+1$ from $\mathcal{H}_t$.
4. Repeat steps 2–3 until the end state is reached.

The virtual human's response $\mathbf{S}_t^V$ is produced from hypothesis $\mathcal{H}_t$ by scaling and translating the shape vector which gave rise to the *maximum* error in Equation 18. This transformation is achieved by re-arranging Equations 3 and 4 and inserting relevant values (d_t, s_t, X_t^H and h_t^H). It is interesting to note that set $\mathcal{X}_{t+1}$ will only contain *multiple* potential hypotheses when a decision point in the Markov chain is reached before time $t+1$.

When propagating a single state hypothesis, the selection of the start state and each successor state fixes the range of possible future behaviours. This has two important consequences. Firstly, if noisy data or model inaccuracies result in an undesirable selection, recovery may not be possible. Secondly, at points where the set of potential hypotheses contains multiple (approximately) equally minimal hypotheses, the selection becomes arbitrary and may result in distinctly different future behaviours. This can be viewed as empowering the virtual human with the task of decision-making in such situations.

5.2. Multiple hypothesis propagation

A more robust form of interaction in which the human fully determines the progress of the interaction can be achieved from the stochastic propagation of *multiple* state hypotheses $\mathcal{H}_t^i$. This forms a Bayesian approach to tracking the interaction, propagating a conditional density representation:

$$P(\mathbf{F}_t|\mathbf{S}_t^H, \ldots, \mathbf{S}_0^H) \propto P(\mathbf{S}_t^H|\mathbf{F}_t)P(\mathbf{F}_t|\mathbf{S}_{t-1}^H, \ldots, \mathbf{S}_0^H), \quad (19)$$

where $P(\mathbf{F}_t|\mathbf{S}_t^H, \ldots, \mathbf{S}_0^H)$ is the conditional distribution of interaction state given an observation history, $P(\mathbf{S}_t^H|\mathbf{F}_t)$ measures the *likelihood* of a state $\mathbf{F}_t$ giving rise to observation $\mathbf{S}_t^H$, and $P(\mathbf{F}_t|\mathbf{S}_{t-1}^H, \ldots, \mathbf{S}_0^H)$ is the *prior* distribution representing predictions from $P(\mathbf{F}_{t-1}|\mathbf{S}_{t-1}^H, \ldots, \mathbf{S}_0^H)$, the *posterior* distribution from the previous time step.

In a method based on the CONDENSATION tracking algorithm of Isard and Blake [6], the posterior density is represented by a set of sample hypotheses, generated using the likelihood to weight sampling from the prior - *factored sampling*. A Gaussian likelihood function is used, based on the hypothesis' error $E(\mathbf{F}_t^i, \mathbf{S}_t^H)$:

$$P(\mathbf{S}_t^H|\mathbf{F}_t^i) = \exp\left(-\frac{E(\mathbf{F}_t^i, \mathbf{S}_t^H)^2}{2\sigma^2}\right). \quad (20)$$

Interaction with a virtual user is achieved with the following multiple hypothesis propagation algorithm:

1. Generate a set $\mathcal{X}_0$ of N hypotheses to represent the initial prior, where $\mathcal{X}_0$ is obtained under sampling with replacement from the initial state distribution $\mathcal{S}$.
2. For each $\mathcal{H}_t^i \in \mathcal{X}_t$, use the error $E(\mathbf{F}_t^i, \mathbf{S}_t^H)$ to calculate the likelihood of the hypothesis, using Equation 20.
3. Use relative likelihood values to weight sampling from $\mathcal{X}_t$, the prior, resulting in a set $\mathcal{Y}_t$ of N hypotheses representing the posterior distribution.
4. Produce the virtual human's response $\mathbf{S}_t^V$ from the hypothesis $\mathcal{H}_t^i \in \mathcal{Y}_t$ with maximum likelihood.
5. Generate a new set $\mathcal{X}_{t+1}$ of N hypotheses to represent the new prior, where each $\mathcal{H}_{t+1}^i \in \mathcal{X}_{t+1}$ is a stochastic extrapolation at time $t+1$ from $\mathcal{H}_t^i \in \mathcal{Y}_t$.
6. Repeat steps 2–5 until the interaction is complete.

Response generation is unchanged from the single hypothesis approach. When generating stochastic extrapolations, noise is introduced to the time interval approximation (Equation 15). This allows model uncertainty to be represented, resulting in a more reasonable prior. Noise is sampled from a uniform distribution over $[-\frac{3}{4}T, \frac{1}{4}T]$ and added to the time interval. This distribution is biased towards decreasing the approximate time interval to compensate for the model's tendency to overestimate the time interval between state vectors representing static shapes.

The propagation of multiple hypotheses representing a conditional density forms a robust approach to tracking an interaction. The algorithm described does not fully realise this potential in one respect - the virtual human's response is generated from the hypothesis with *maximum likelihood*, and not that with *maximum a posteriori* probability. Since the posterior is represented by the $\mathcal{H}_t^i \in \mathcal{Y}_t$, the maxima could be located by calculating the number of hypotheses that fall within a hypersphere of radius δ, centred on each hypothesis:

$$\max_i\{|\{\mathbf{F}_f^j \ : \ |\mathbf{F}_f^j - \mathbf{F}_f^i| < \delta, \ \ j \neq i\}|\}, \quad (21)$$

where the value of δ could be determined experimentally.

In our experiments, the 301-state Markov chain and the multiple hypothesis propagation algorithm were used to enable interaction with a virtual human. A value of $\sigma = 0.05$ was used in the likelihood function and 200 hypotheses were propagated.

Due to the computational requirements of the algorithm described, our initial experiments have been performed offline. We first attempted to generate test data by capturing sequences of a single person performing a 'blind' handshake. It was, however, soon discovered that the behaviour exhibited was markedly different to that exhibited in real interactions. To compensate for this inability to behave naturally in the absence of an interacting partner, test sequences involving two individuals were captured and one of the individuals was masked before object tracking was performed.

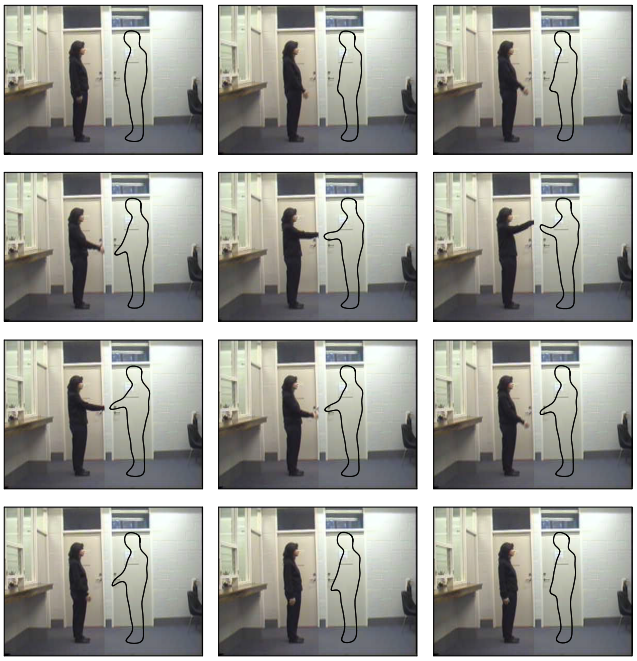


Figure 5. Virtual interaction results.

Figure 5 shows a selection of frames from a virtual interaction test sequence. In each frame, the virtual human is displayed as a black silhouette. Observation of the entire set of hypotheses during the interaction suggests a distribution with a mode at the maximum likelihood state and further transient modes describing alternative paths at decision points in the chain. This distribution rapidly tails off along past and future paths. Modes describing alternative paths tend to diminish rapidly once the current shape of the real actor becomes inconsistent with the hypotheses.

6. Conclusions

A statistically based interaction behaviour model has been presented which enables both the learning of behaviours from the observation of interacting humans, and the use of the acquired models to provide realistic responses during interaction. Interaction with a virtual human has been achieved via a robust multiple hypothesis propagation algorithm. Experiments based on a simple human interaction show encouraging results. In the future we envisage the use of more detailed models of individuals and their behaviours, capable of richer kinds interaction (a kind of *Virtual Immortality* ?).

References

[1] A. Baumberg and D. Hogg. Learning Flexible Models from Image Sequences. In *Proc. 3rd European Conference on Computer Vision*, volume 1, pages 299–308, May 1994.

[2] A. F. Bobick and J. W. Davis. Action Recognition Using Temporal Templates. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 125–146. Kluwer Academic Publishers, 1997.

[3] A. F. Bobick and A. D. Wilson. A State-based Technique for the Summarization and Recognition of Gesture. In *Proc. 5th International Conference on Computer Vision*, pages 382–388, June 1995.

[4] L. W. Campbell and A. F. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *Proc. 5th International Conference on Computer Vision*, pages 624–630, June 1995.

[5] D. M. Gavrila and L. S. Davis. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 272–277, 1995.

[6] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proc. 4th European Conference on Computer Vision*, volume 1, pages 343–356, Apr. 1996.

[7] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, Aug. 1996.

[8] K. Kakusho, N. Babaguchi, and T. Kitahashi. Recognition of Social Dancing from Auditory and Visual Information. In *Proc. 2nd International Conference on Automatic Face and Gesture Recognition*, pages 289–294, Oct. 1996.

[9] T. Kohonen. The Self-Organizing Map. *Proceedings of the IEEE*, 78(9):1464–1480, Sept. 1990.

[10] S. Nagaya, S. Seki, and R. Oka. A Theoretical Consideration of Pattern Space Trajectory for Gesture Spotting Recognition. In *Proc. 2nd International Conference on Automatic Face and Gesture Recognition*, pages 72–77, Oct. 1996.

[11] A. Pentland. Machine Understanding of Human Motion. Technical Report 350, MIT Media Laboratory Perceptual Computing Section, Sept. 1995.

[12] D. E. Rumelhart and D. Zipser. Feature Discovery by Competitive Learning. *Cognitive Science*, 9:75–112, 1985.

[13] T. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.