Playing 'Sticky Hands' With A Humanoid Robot

Joshua G. Hale[1 2] and Frank E. Pollick[1 3]

[1] ATR International, ISD, Cyberhuman Project
2-2 Hikaridai, Seika-cho Soraku-gun, Kyoto, 619-0288, Japan
[2] Dept. of Computer Science, University of Glasgow, halej@dcs.gla.ac.uk
[3] Dept. of Psychology, University of Glasgow, frank@psy.gla.ac.uk

Abstract. 'Sticky Hands' is an exercise for two participants requiring them to maintain a gentle hand to hand contact. The intention is to explore graceful and relaxed motion together, develop physical sensitivity and responsiveness, and interact at a comfortably intimate level for physical relaxation and spiritual development. We present a system allowing a person to perform this novel interaction with a humanoid robot. A learning algorithm is presented which observes and predicts the trajectory of the contact point, using a novel internal representation of instantaneously sampled data 'prototypes'. It is capable of generalizing observed trajectories to unseen cases, operating with parameterized time and memory bounds, and coping with the evolving nature of contact trajectories. We thus present a new role for the robot as a playmate for self-development, and hope to facilitate comfortable physical interaction and encourage its perception as a human-like and emotional being. We discuss augmentation of the robot's motion with human-like style and emotional expression, based on perceptual psychology, human motor production, and computer graphics -fields whose relevance for robotics is also discussed.

1 Sticky Hands

The 'Sticky Hands' exercise is drawn from Tai Chi. Some styles of Tai Chi involve practicing physical contact with a partner. By forming a gentle contact with a partner and maintaining it while moving people aim to develop the ability to move in a relaxed and graceful way, and a sensitivity to the forces experienced through the contact. When one partner pushes the other must yield, and vice versa. Prolonged practice allows the development of intuition and response so that the contact may be maintained very lightly throughout a complex and unplanned series of movements. Since the exercise involves becoming comfortable with physical intimacy, and a shared goal of mutual development, as well as being calming and relaxing, some people consider it as a form of spiritual development. A teacher playing the game with a pupil may encourage graceful and rewarding motions by gradually breaking down the muscular tension in the student's movement patterns.

We wanted to develop a system for playing the game with a humanoid robot. We therefore defined a specific version of the exercise using hand to hand contact. Partners stand face to face and each lift one hand to meet their partner's. Slow circling motions are performed to begin the game. The motion may then diverge to an unplanned trajectory that may evolve over time as the partners explore their range of expression.

As a novel application it presents several interesting issues in motion and motor control. The robot must be capable of moving while remaining compliant to contact forces from a human, as well as being able to mimic the motion patterns of humans playing the game. The robot may help a human to explore their range of motion through the mutual development of trajectories that reflect the creativity of the human by learning and imitating their patterns of motion. We wanted to explore the use of a humanoid robot as a playmate facilitating a human's self-development. As such the robot may assume a new social role involving a physically intimate and cooperative interaction. Hopefully, through the interaction people will be encouraged to consider the robot as a humanoid rather than mechanical entity. To further this aim we also consider the production of human-like and emotionally expressive styles of movement.

The larger area of humanoid robotics captures a certain fascination with creating a mechanical entity analogous to our human selves. But as Bergener *et al.*[1] point out motivations stretch further, not least among which is the environment we have constructed. Being highly adapted to human sensory and motor capabilities it begs for artificial agents with analogous capabilities that can usefully coexist in our own living and working spaces. Moreover, anthropomorphic shape and organic motion makes working with such robots aesthetically and emotionally more pleasing. Bergener *et al.* advocate *autonomous* anthropomorphic robots, likening autonomy to a dog that while obeying most commands maintains internal behavioral goals that prevent it from running into walls *etc.* As performed by a humanoid robot the sticky hands interaction embodies human like motion and autonomy within the bounds of the game. The goal of maintaining minimal contact force is tangible while the creative, anticipatory aspect is enhanced by initiative.

As computer science and robotics develop it becomes clear that humans and robots will cooperate with a wide range of tasks. The arising problem is how to communicate and cooperate with people[2]. Based on the principle that physical interaction is a method familiar and reliable for most humans, Sticky hands broaches the area of physical cooperation and the subtle but significant issue of communication through physical movement. This occurs by design in our system mainly through shared attention and eye movement as is discussed in Section 4. The system is also relevant to psychology since implementing imitative behaviors allows us to investigate the minimal competencies for social learning(Scassellati [3]). Full imitation is an extremely complex issue requiring many perceptual, cognitive and motor capabilities. In this work we capitalize on two of the four elements that Scassellati reports as fundamental: interaction through innate social mechanisms, and the exploitation of advantages of the robot's physical embodiment. The other two: developmental progression of skills and integration of multiple sensor modalities are not embodied by this system.

We first present a system overview that describing the relationship between robot control and learning. We then describe robot control, showing how hand placement may be blended with compliance to external force. Following this, we discuss the learning algorithm which observed hand trajectories throughout interaction with a human and predicted the development of a current trajectory. We explain a novel internal representation for trajectory information based on ‘prototypes’ (samples of specific instantaneous data) stored in a voxel array. We explain how this representation allows observed trajectories to be generalized to aid the prediction of previously unobserved trajectories, and how parameterized time and memory bounds may be enforced. We conclude our discussion of the learning algorithm with results drawn from a session with a human player. Following this, we discuss motion augmentation. We intend to augment the motion of the robot to mimic human demeanor and express emotion. We discuss methods drawing on the fields of perceptual psychology, human motor production, and computer graphics. At this point we illustrate the relevance and speculate upon the use of these fields in future humanoid robotics research. We then conclude this paper and discuss its implications for a number of recent assertions regarding humanoid robotics.

1.1 System Overview

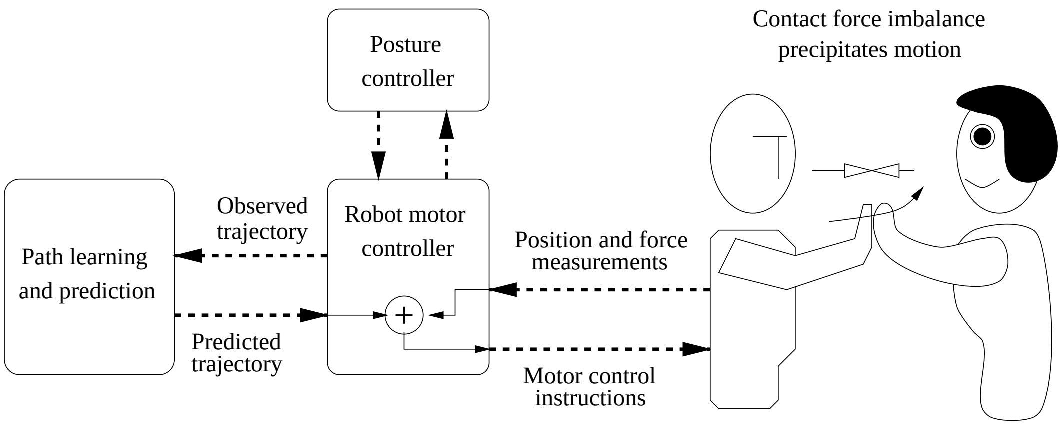


Fig. 1. System breakdown

We divided the task of robot sticky hands interaction into three components. Fig.1 shows a ‘Robot motor controller’ that positioned the hand to follow a trajectory supplied by the learning algorithm. It simultaneously estimated the contact force from the human and responded by adjusting the supplied trajectory. Hand position samples were collected and less noisy averaged samples were returned to the learning algorithm.

The learning algorithm alternately received measured hand position samples and output predicted hand positions. It observed the continual development of the hand trajectory, learning patterns of the hand's motion, generalizing them to predict further developments in the trajectory.

The robot controller communicated with the ‘posture controller’. In our implementation, the posture control component performed a simple inverse kinematics routine to determine joint configurations placing the hand at a desired Cartesian coordinate. Ultimately however, this component is intended to return a posture from a diverse range of possibilities. We may adjust motions to express an internally modeled emotional state, add a human like quality or perform expert-like sticky hands -teaching graceful or rewarding motions to the human player. We discuss the development of this component in detail in the 'motion augmentation' section. The ethos of our motion system may be contrasted with the work of Williamson[4] whose motion controllers were based on positional primitives. A small number of postures were interpolated to produce target joint angles and hence joint torques according to proportional gains. The work advocated the concept of of "behaviors or skills as coarsely parameterized atoms by which more complex tasks can be successfully performed", an approach also proposed for computer animation in works such as the motion verbs and adverbs of Rose *et al.*[5]. Williamson's system is elegant, providing a neatly bounded workspace, but not suitable for our needs since we require a continuous interaction combined with more precise positioning of the robot's hand. Generalizing their approach to arbitrary hand positioning would require a process analogous to inverse kinematics mapping into positional primitive space rather than joint angles. The use of such primitives as a basis also reduces the redundancy of the robot's manipulators.

1.2 Robot Control

Fig. 2. Interacting with the SARCOS dextrous figure

A SARCOS anthropomorphic robot (which may be seen in Fig. 2 performed the sticky hands exercise. It is hydraulically powered, with joint angle and load sensors. The low level controller positions joints by applying gain torques proportional to the angular offset between the measured and target positions. An inverse dynamics algorithm was available to estimate the torques necessary to hold a position, and was used to reduce the magnitude of the oscillations caused by the proportional gains controller. Since the robot is anchored by its pelvis, standing and balancing was not an issue.

We required the robot to balance a force against its hand applied by the human player. A suggested trajectory was performed with possible adjustments to balance the contact force, making the robot hand *actively compliant* to changes of the contact force.

One method of calculating the externally applied force is to measure joint angles, use inverse dynamics to estimate the torques necessary to hold the position and subtract the measured loads. A discrepancy should reveal extra torques generated by the low level controller to oppose a non-gravity external force. Unfortunately inaccuracies in the dynamic model and load sensors required high thresholds so that the contact force had to be large to avoid spurious estimates. Williamson[4] implemented a similar torque based compliant controller for a robot hand that did not attempt to calculate the externally applied torque. The process was described in terms of reflexes that caused joint actuators to yield whenever the sensed torque exceeded a given threshold. While permitting a human to grasp and reposition the robot's hand their thresholds were large enough to exceed the gravity compensating torques the joints had to generate and would be unsuitable for the soft contact required by our system.

Instead we measured the positional offset between target and actual hand positions, assuming any large discrepancy was caused by external non-gravity forces. We were able to threshold by $2\mathrm{cm}$. Thus when the controller followed a suggested trajectory the hand was directed to $5\mathrm{cm}$ further forward from the suggested position. It was assumed that the human would supply a force sufficient to maintain the robot's hand at the suggested position, opposing the $5\mathrm{cm}$ offset. If the human reduced or increased the contact force the robot's hand would move beyond the $2\mathrm{cm}$ threshold and the suggested trajectory consequently adjusted to compensate. By using this indirect kinematic method we facilitated the use of a significantly lighter contact force.

Suggested trajectories were described as piecewise linear splines. We ran the robot controller at $500\mathrm{Hz}$ but the learning algorithm at $10\mathrm{Hz}$, so the prediction points output by the latter could be interpreted as spline end-points. When a change in contact force precipitated an adjustment, we translated the knots accordingly. A smoothing vector was set to translate the suggested point from its newly updated location to its location prior to knot translation. At every cycle the smoothing vector was decayed, tending to zero. This ensured that the hand did not jerk, instead moving smoothly to accommodate knot translations.

We configured the robot's twin camera eyes to point towards its hand, giving the impression that it was concentrating on the game. Whenever knots were translated, indicating that the human had adjusted the robot's trajectory by imbalancing the contact force the eyes moved briefly to focus on a point in front of the robot where the human was assumed to be standing. This was readily noticeable as the robot adjusted its neck angles to gaze at a point, and communicated an indication of how well the human and robot were synchronized, as well as a sense of attention.

Since the robot is capable of sudden and powerful movements we enforced four safety features protecting the human player. The robot was immediately frozen if any of the following constraints were violated: a maximum difference between each joint's target and current angles, a maximum difference between the current and target hand positions, a maximum angular velocity for each joint and a maximum linear velocity of the hand.

2 Learning Algorithm

The learning algorithm was fed with 3D point samples of the space-line describing the robot's hand trajectory. For each sample, a prediction of the progression of the line was output. The algorithm fulfilled the following requirements:

- **Generalize** observed trajectories for prediction of similar new trajectories
- **Extrapolate** properties of a new trajectory for prediction in the absence of similar observed trajectories
- **Fluid internal state** copes with the evolving nature of trajectories
- **Branch points** where similar observed trajectories diverge is not problematic
- **Noise** causing inaccurate position samples is not problematic
- **Parameterizable time bound** ensures real time operation
- **Parameterizable memory bound** facilitates exploitation of host architecture

The algorithm recorded instantaneous properties of the input trajectory in structures we refer to as a 'prototypes'. By comparison to the work of Stokes *et al.*[6] who presented a method for identifying cyclic patterns and their significance in space-line samples, our process focuses on the immediate instant of a trajectory. Salient features are recorded for efficient retrieval but no internal classifications of higher level structures such as cycles are made. Below we define the 'prototype' mathematically, and show how its properties may be applied for *prediction* or *extrapolation*. We then show how the most appropriate prototype for predicting a given trajectory may be selected from a memory bank by means of a distance metric between prototypes and an optimized search procedure. The creation of prototypes from raw position data is explained, followed by a reinforcement memory technique designed to ensure an efficient use of memory. We conclude the section on the learning algorithm by giving sensible ranges for the significant parameters and discussing their effects.

The reader may find it useful to refer to Fig. 2 throughout this prototype learning section.

2.1 Prediction Using Prototypes

Suppose we describe the sequence of input position samples as $\{p_k : k \in \mathbf{N}\}$. The prototype P_i corresponding to p_i is defined using p_{i-1}, p_i and p_{i+1}:

$$P_i = (p_i, v_i, a_i, T_i) \tag{1}$$

$$v_i = p_{i+1} - p_i \in \mathbf{R}^3 \tag{2}$$

$$a_i = \frac{|p_{i+1} - p_i|}{|p_i - p_{i-1}|} \in \mathbf{R} \tag{3}$$

$$T_i = \left(\cos\left(\frac{\theta_i}{2}\right), \sin\left(\frac{\theta_i}{2}\right)(p_{i+1} - p_i) \times (p_i - p_{i-1})\right) \tag{4}$$

$$\theta_i = \cos^{-1}\left(\frac{(p_i - p_{i-1}).(p_{i+1} - p_i)}{|p_i - p_{i-1}||p_{i+1} - p_i|}\right) \in \mathbf{R} \tag{5}$$

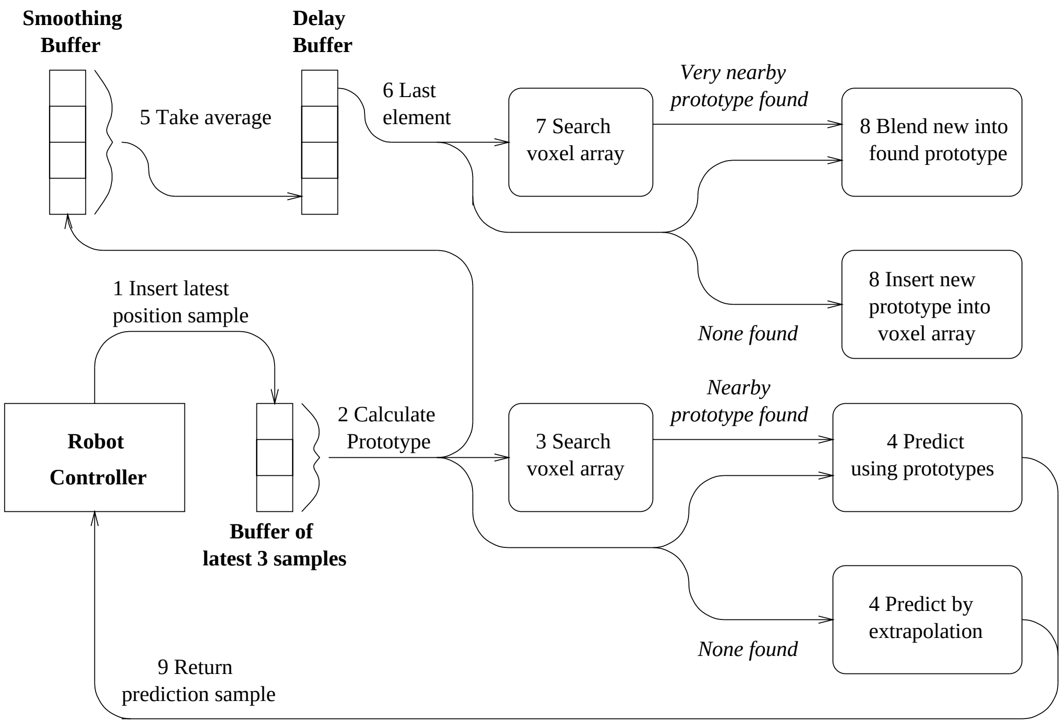


Fig. 3. Learning algorithm

v_i is the velocity out of p_i and a_i is the magnitude of the acceleration. The acceleration *direction* is deducible from T_i, a quaternion describing the change in direction between v_i and v_{i-1} as a rotation through their mutually orthogonal axis.

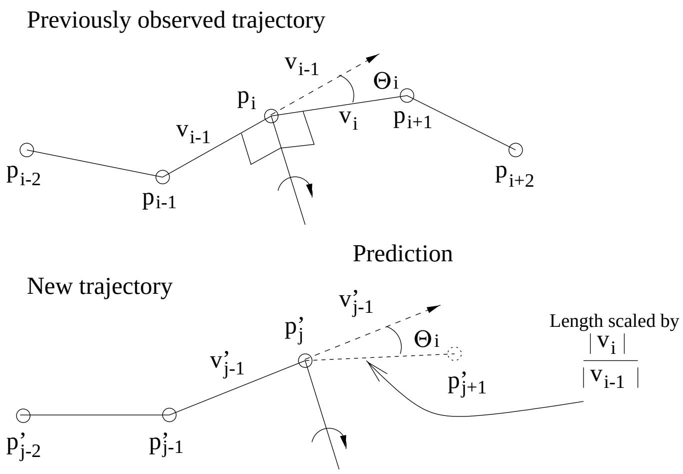


Fig. 4. Trajectory prediction using a prototype

We may predict the progression of a trajectory $\{p'_k : k \in \mathbf{R}\}$ using a prototype. Suppose for now that we know P_i corresponds to p'_j We can estimate p'_{j+1} as $p'_j + a_i T_i(p'_j - p'_{j-1})$. Pre-multiplication of a 3-vector by T_i denotes quaternion rotation in the usual way. We are thus applying the bending and accelerating occurring at p_i to predict the development of p'_j. We also linearly blend the position of p_i into the prediction, and the magnitude of the velocity so that p'_j combines the actual position and velocity of p_i with the prediction duplicating p_i's bending and accelerating characteristics (see Fig. 4):

$$p'_{j+1} = p'_j + s_j T_i . \frac{p'_j - p'_{j-1}}{|p'_j - p'_{j-1}|} + g_{\text{p}} \left(p_i - p'_j \right) \tag{6}$$

$$s_j = (1 - g_{\text{v}}) a_i |p'_j - p'_{j-1}| + g_{\text{v}} |v_i| \tag{7}$$

We can use the blending ratios g_{p} and g_{v} to control the degree to which predictions are entirely general, or repeat previously observed trajectories. *i.e.* how much the robot wants to repeat what it has observed.

In the absence of a corresponding prototype we can calculate P'_{j-1}, and use it to estimate p'_{j+1}, thus extrapolating the current characteristics of the trajectory. Repeated extrapolations lie in a single plane determined by p_{i-2}, p_{i-1} and p_i. This is not problematic because when guided into a new motion pattern we only require the robot to move in a sensible way so that the human may continue to guide it. We must set $g_{\text{p}} = 0$ since positional blending makes no sense when extrapolating, and would cause the trajectory to slow to a halt.

2.2 Storage and Retrieval

In order to predict p'_{j+1} we would like to find a previously observed trajectory similar to that sampled at p'_j. Assuming a set of recorded prototypes we seek a prototype P_i as used above to predict p'_{j+1}. We can use P'_{j-1} that describes the properties of our current trajectory as a basis for selecting similar observed trajectories described by prototypes. We therefore define a distance metric between prototypes so that we can look for the closest match:

$$d(P_i, P_j) = 1 - \cos(\theta') + \frac{|p_i - p_j|}{M_{\text{p}}} \tag{8}$$

Where,

$$\theta' = \theta \frac{\pi}{2M_{\text{a}}}, \qquad \theta \in [-M_{\text{a}}, M_{\text{a}}] \tag{9}$$

$$= \pi, \qquad \text{otherwise} \tag{10}$$

$$\theta = \cos^{-1} \frac{v_i . v_j}{|v_i||v_j|} \tag{11}$$

M_{a} and M_{p} define the maximum angular and positional differences such that $d(P_i, P_j)$ may be one or less. The metric compares the position of two prototypes, and the *direction* of their velocities. Two prototypes are close if they describe a trajectory traveling in the same direction, in the same place. The absolute velocity, and bending characteristics are not compared. Predictions are therefore general with respect to subsequent developments (so branching points are not problematic) and velocity, so the speed at which an observed trajectory was performed does not affect the way it can be generalized to new trajectories.

When seeking a prototype we might compare *all* recorded prototypes with P'_{j-1} to find the closest. If none exist within a distance of 1 we use P'_{j-1} itself to extrapolate as above. Needless to say we could not compare P'_{j-1} with *all* the recorded prototypes. To optimize the search process we defined a voxel array to store the prototypes. The array encompassed a cuboid enclosing the reachable space of the robot, partitioning it into an array of cuboid voxels indexed by three integer coordinates. New prototypes were placed in a list attached to the voxel containing their positional component p_i. Given P_{j-1} we only needed to consider prototypes stored in voxels within M_{p} of p_{j-1} since prototypes in any others would definitely exceed the maximum distance according to the metric.

We optimized the order in which voxels were considered. As a one time pre-computation we prepared a list $\{(x_0, y_0, z_0, d_0), \ldots, (x_n, y_n, z_n, d_n)\}$ whose elements described the minimum distance d_k between a given voxel and a voxel whose index was offset by (x_k, y_k, z_k). *i.e.* for arbitrary indices a, b and c the minimum distance between the voxels (a, b, c) and $(a + x_k, b + y_k, c + z_k)$ is d_k. The list was sorted into increasing distance order and contained only elements whose distance was at most M_{p}. The first element of the list was thus (0,0,0,0). When searching, given P_{j-1} we first find the voxel indexed (a, b, c) containing p_{j-1}, and consider prototypes stored in the voxel indexed $(a + x_0, b + y_0, c + z_0)$ then $(a + x_1, b + y_1, c + z_1)$ and so on. We maintain the closest matching prototype and its distance according to the above metric, d_{min}. As soon as the list runs out or we reach a list element numbered k such that $d_{\text{min}} < d_k$ the search terminates. This ensures an optimal search of the voxel array since the voxels are considered in an expanding sphere about the voxel containing the original prototype, and the search may terminate as soon as we encounter a voxel that is too far away to contain a prototype with a closer minimum distance than any already found. It also permits the search to be cut short if more time is unavailable. By ensuring that the most likely to match prototypes are considered first, stopping the search prematurely is made optimally likely to return the same result as if it were allowed to proceed to completion. This facilitates the parameterizable time bound since the prototype search is the only major time expense of the learning algorithm. The search time may be reduced in keeping with the capabilities of the host architecture, degrading performance in an optimal way. The inherent assumption is that proximity increases the likelihood of a prototype match, which is not unreasonable given that proximity is part of the distance metric and nearby prototypes are likely to originate from similar trajectories. Regarding the use of a pre-sorted list we acknowledge the likely existence of a function to neatly provide the voxel indices and their minimum distances directly but the list was felt to be straightforward to implement, and its bias towards speed rather than space efficiency insignificant.

2.3 Creation and Maintenance

Prototypes were continually created based on the input position samples describing the observed trajectory. It was possible to create a new prototype for each new sample, which we placed in a cyclic buffer. For each new sample we extracted the average prototype of the buffer to reduce sampling noise. These averaged prototypes were shunted through a delay buffer, before being added to the voxel array. This prevented prototypes describing a current trajectory from being selected to predict its development (extrapolation) when other prototypes were available.

Rather than recording every prototype we limited the total number stored by averaging certain prototypes. This ensures the voxel array does not become clogged up and slow, and reduces the memory requirement. Therefore before inserting the prototype at the end of the delay buffer into the array we first searched it for a similar prototype. If none was found we added the new prototype, otherwise we blended it with the existing one. We therefore associated a count of the number of blends applied to each prototype to facilitate correct averaging with new prototypes. In fact we performed a non-linear averaging that capped the weight of the existing values, allowing the prototypes to tend towards newly evolved motion patterns within a limited number of demonstrations. Suppose P_{a} incorporates n blended prototypes, then a subsequent blending with P_{b} will yield:

$$P_{\mathrm{a'}} = P_{\mathrm{a}} \frac{D(n)-1}{D(n)} + P_{\mathrm{b}} \frac{1}{D(n)} \tag{12}$$

$$D(n) = 1 + A_{\mathrm{M}} - \frac{A_{\mathrm{M}}}{1 + nA_{\mathrm{G}}} \tag{13}$$

$\frac{A_{\mathrm{M}}}{A_{\mathrm{M}}+1}$ defines the maximum weight for the old values, and A_{G} determines how quickly it is reached. A subtle consequence of the averaging process was that prototype's positional component could be altered such that the prototype must be deleted from one voxel and inserted into a neighboring voxel.

We enforced an upper bound on the number of prototypes stored using a deletion indexing strategy. We maintained an integer clock that was incremented for each new sample received. Upon creation each prototype was marked with a deletion index sometime in the future. We maintained a list of all the recorded prototypes sorted by deletion index so that whenever the maximum number of prototypes was reached we removed the first element of the list and deleted the indexed prototype. The list was stored as a *heap*[7] since this data structure permits fast $O(\log(\text{Num Elements}))$ insertion, deletion and repositioning. We manipulated the deletion indices to mirror the reinforcement aspect of human memory. A function $R(n)$ defined the period for which a prototype reinforced n times should be retained. (n is equivalent to the blending count.) Each time a prototype was blended with a new one we calculated the retention period, added the current clock and resorted the prototype index. $R(n)$ increases exponentially up to a maximum asymptote.

$$R(n) = D_{\mathrm{M}} - \frac{D_{\mathrm{M}}}{1 + D_{\mathrm{G}} n^{D_{\mathrm{P}}}} \tag{14}$$

D_{M} gives the maximum asymptote. D_{G} and D_{P} determine the rate of increase.

This deletion mechanism facilitates a parameterizable memory bound. Non-linear prototype blending and reinforcement learning provide a fluid internal state. The former allows repeated trajectories that gradually evolve to continually adjust their corresponding prototype based descriptions. The latter ensures that favored trajectories are well learned while one off or spurious trajectories are quickly discarded and obsolete trajectories may be forgotten over a longer period.

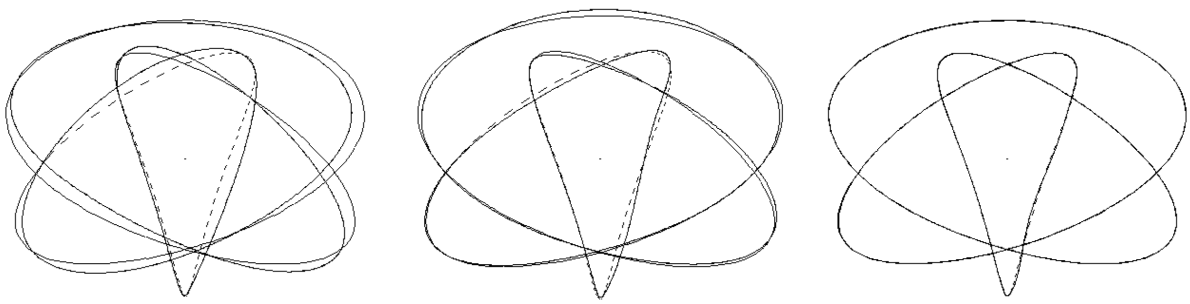
2.4 Tuning

We used an array of $50 \times 50 \times 50$ 4cm cubic voxels encompassing a 2m cubic work space. This generously enclosed the reachable space of the robot's hand. With 4 byte pointers the storage requirement of the empty array was 0.5Mb. We presented samples to the learning algorithm at 10Hz and used a delay buffer of around 50 elements, so new trajectories would not be used for prediction for at least 5 seconds. We set the cyclic smoothing buffer to 5 or 6 elements, which proved sufficient to average out spurious samples.

Table 1. Parameter values

Parameter	Value	Equations
g_p, g_v	both $\in [0.01, 0.001]$	6, 7
M_p, M_a	$15\text{cm}, \pi/4$	9, 10
$M_\text{p}^-, M_\text{a}^-$	$2\text{cm}, \pi/8$	9, 10
A_M, A_G	10, 0.1	12
$D_\text{M}, D_\text{G}, D_\text{P}$	$20000, 0.05, 2$	14

Table 1 gives the values assigned to the parameters included in the formulae above. We chose g_p and g_v by feeding predictions back into the learning algorithm, as if the contact force were always perfectly balanced. We found the smallest values (therefore permitting maximum generality of prediction), that demonstrated the robot performing learned trajectories *ad infinitum* without eventually diverging. Figure 5 shows the effect of various values of g_p. The dotted line represents a single learned trajectory and as a sum of orthogonal sinusoids has non-zero torsion. As g_p is increased so does the tendency of the predictions to gravitate towards the recorded trajectory spatially. The likelihood of synthesizing a trajectory that will continue to match one that has been recorded may thus be controlled at the expense of the system's willingness to pursue trajectories in a new spatial location by generalization of an observed trajectory. g_v has an analogous controlling effect on the velocity of a predicted trajectory with respect to the velocity of a recorded trajectory.



$g_p = 0.001$ $g_p = 0.0025$ $g_p = 0.01$

Fig. 5. The effect of the positional gravity term g_p

M_a and M_p describe the maximum angular and positional discrepancies between a new trajectory and one selected to predict it. These were chosen by hand. They defined the cut-off point at which the algorithm recognizes a trajectory as defining a 'new' motion pattern, thus resorting to extrapolation. M_a^- and $M_\text{p}-$ were used to limit the prototype search prior to blending or inserting a new prototype. The limit was tighter, ensuring that new prototypes would be added to describe a motion pattern before enough were forgotten that the system had to extrapolate. *i.e.* the balance point between maintained prototypes and the deletion of unused prototypes was raised above the threshold necessary to match for prediction.

A_M and A_G were chosen to make the non-linear averaging denominator initially linear like standard averaging, with an upper bound of 11. 30 demonstrations are sufficient to almost completely change the values of a prototype.

$D_{\mathrm{M}}, D_{\mathrm{G}}$ and D_{P} defined a maximum retention period of 33 minutes at $10\mathrm{Hz}$. The first reinforcement extended a prototype's minimum deletion index by about 2 minutes, subsequent reinforcements roughly doubled this up to a maximum of 33 minutes.

2.5 Results

Fig. 6 shows the position of recorded prototypes for the default initial state and a stored state after a human has played with the robot for a few minutes. The two data sets are each viewed from two directions. The coordinates are in millimeters, with the x, y & z axes positive in the robot's left, up and forward directions respectively. The point $(0, 0, 0)$ corresponds to the robot's sacrum. Please observe that the robot icons illustrate orientation only, and not scale. Each point represents a prototype stored in the motion predictor's memory. The trajectory of the hand loosely corresponds to the spacing of prototypes but not exactly because sometimes new prototypes are blended with old prototypes according to the similarities between each's position and velocity vectors.

The initial state was taught to the robot and approximates a circle $10\mathrm{cm}$ in radius and centered in front of the left elbow joint (when the arm is relaxed) in the frontal plane about $30\mathrm{cm}$ in front of the robot. The prototypes correspond to the robot's left hand, which was in contact with the human's right hand. We observe that after following the initial circling motion preferred by the robot the human instigated a progression towards circling motions centered near to the robot's left shoulder, and making a little exploration to the right shoulder. Mostly the changes in the trajectory occur gradually as human and robot slowly develop repeated cycles. Once learned, the robot can switch between any previously performed trajectories.

Videos of a human playing the game with the robot or another human may be down-loaded from: `http://www.dcs.gla.ac.uk/ halej/stickyHands.htm`

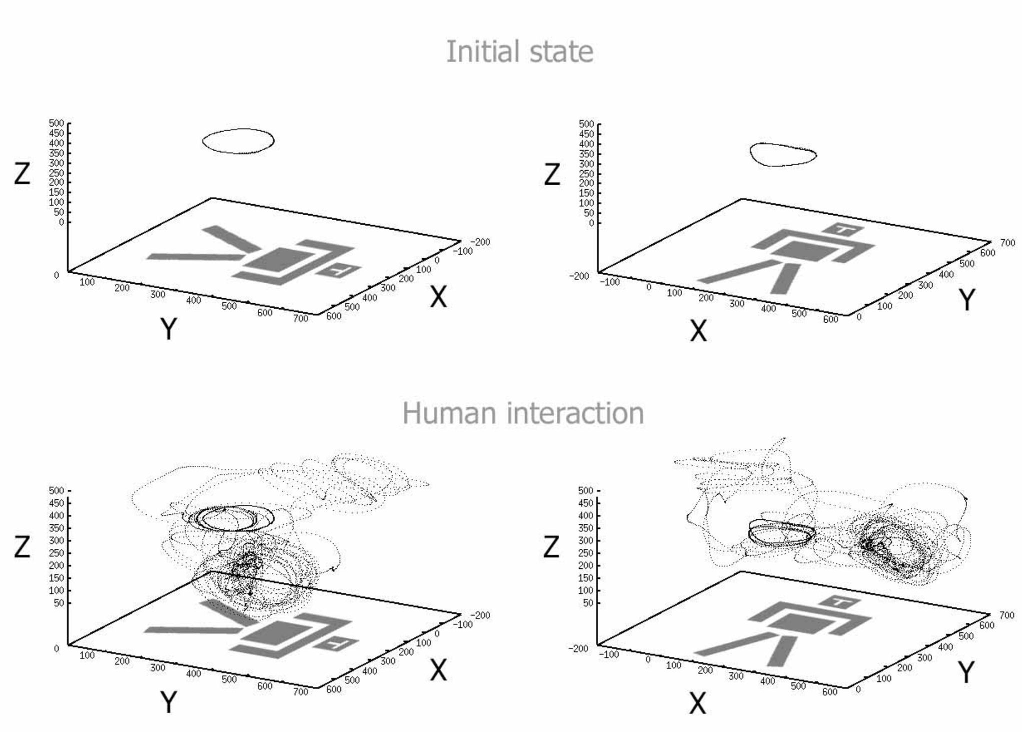


Fig. 6. Prototype state corresponding to a sample interaction

3 Motion Augmentation

We now discuss development of the posture control component shown in Fig. 1. Our goal is to control the robot so that it moves naturally and is able to alter its movement style to mimic different demeanor. We currently view this goal as having two steps, the first is the creation of neutral movements which appear natural to the player, the second is to be able to modify these neutral movements to achieve movements that are perceived to display different affects or demonstrate expert sticky hands technique.

Possible strategies to achieve natural and affective movements have been suggested in the related field of computer animation of human movement. The 'dynamics filter' proposed by Yamane and Nakamura[26] illustrates the duality between realistic human animation and humanoid robotics. Their technique corrects the dynamics of a kinematicly described motion, allowing different environmental or kinematic specifications to be enforced while providing a dynamically consistent result. Hodgins[27] provides a means for scaling dynamical control parameters for models with different characteristics such as mass and geometry, thus facilitating the realistic emulation of humanoids with specific physical characteristics based on an control parameters for an arbitrary figure. In a definitive paper followed up by a number of authors[29–33] Witkin & Kass[28] proposed the use of global optimization techniques for organic motion, observing that minimization of the product of angular velocity and torque produced more realistic motion than smoothness in the differential sense. Recently, optimization methods have been used in conjunction with musculoskeletal models[19] as a means to produce human like motion by minimizing muscle activation. In computer graphics the need for convincing performances by virtual actors precipitated the development of methods for modeling emotional state. Such methods are equally applicable to determining appropriate emotional responses of humanoid robots. Algorithms such as FLAME[8] and AIR[34] mimic the emotional responses of humans by evaluating the consequences of various events with respect to a set of parameters describing mood. Affective motions have been produced by Amaya, Bruderlin & Calvert[9] using emotional transforms that *warp* the kinematics of neutral movements. Likewise, spatiotemporal characteristics[35], and kinematic properties based on effort[36] have also been exaggerated to produce affective styles.

Computer animation provides a rich source of techniques for generating dynamically correct motion, simulating natural or expressive movements and adjusting kinematic or dynamically described motion to display affect or achieve new goals. Techniques for motion production, adjustment, and automatic behavior control are covered in the literature. While some techniques exploit the use of off-line computing time since they are intended for animators who do not expect to author sequences in real-time, analogous real-time techniques may be natural extensions in many cases. Real-time techniques are also a part of the literature, particularly in relation to simulation.

Beyond consideration of the technical means to obtain natural-looking movements that show affect, skill, etc. it is also useful to examine the production and visual perception of human movement for cues to the design of humanoid motion. The study of human motor control for instance not only provides the potential for better techniques to mimic human movement, but it can be assumed that interactions between humans and humanoids would be facilitated if both had similar representations of movement. For instance, in the current scenario the goal is for the humanoid and the human to achieve a smooth and graceful trajectory. However, there are several objective ways to express smoothness and it can be anticipated that if the humanoid and human shared the same representation of smoothness then the two would more quickly converge on a graceful path. Study of the visual perception of human movement holds the potential to isolate the aspects of movement that are crucial for a correct interpretation. For example, although movement can generally be thought of as a complicated spatiotemporal pattern, recognition of particular styles of movement might rely on a limited spatial or temporal aspect of the movement. Knowledge of human motor control and the visual perception of human movement could greatly facilitate the design of humanoid movements. The following two paragraphs give a brief review of relevant results.

There are several results from human motor control and motor psychophysics which contribute to our understanding of natural human movements. It is generally thought that the smoothness of human arm movements is a result not only of the low-pass filter characteristics of the musculoskeletal system, but also the use of smoothness criteria in central movement planning. Criteria which minimize the jerk[10], torque change[11], motor-command change[12], or signal dependent error[13] have all been suggested as possibilities. Besides smoothness, further regularities to human arm movements have been reported. These include that the endpoint trajectory of the hand behaves like a segmented collection[14] of piecewise planar segments joined together[15] and that velocity of a movement is related to its geometry defined by curvature and torsion. Specifically, it has been reported that for planar segments velocity is inversely proportional to curvature raised to the $\frac{1}{3}$ power, and that for non-planar segments the velocity is inversely proportional to the $\frac{1}{3}$ power of curvature multiplied by $\frac{1}{6}$ power of torsion[16–18, 24].

Recent research into the perception of human movement has investigated the relationship between movement kinematics and the perception of movement style. Results have shown that exaggerating either temporal[22] or spatial[23] differences among sets of movements can be used to enhance the recognition of style. Further evidence for kinematic specification of movement style has been found in the perception of affect from arm movements[20]. These results showed that the psychological structure of perceived affect had two dimensions, the first of which was highly correlated to the average velocity of the movement, the second of which was mildly correlated to smoothness of the movement. The interpretation of this space within a circumplex model of affect[21] suggests that high velocity signals the energy of a movement along a energy/lethargy dimension and that high smoothness signals the positivity of affect along a positivity/negativity dimension. These results on how movement kinematics specify movement style not only help to constrain the design of humanoid motion, but also open the opportunity to use the humanoid robot to explore the possibility that meaningful differences in kinematics rely on the control of movement dynamics.

The brief literature review on human motor control and visual perception of human movement above provides a starting point for the design of interaction with humanoid robots. These results focus mainly on the motion of the robot and try to deal in a bottom up fashion, assuming that basic relations can be found that would indicate whether or not a movement appears natural or with the proper affect. However, it has yet to be seen that cognitive factors, in the form of expectancies and top down influences might not dominate interactions between humans and humanoids. The possibility exists that the humanoid could produce a natural movement with affect, but would be misinterpreted due to the fact that it is expected that the robot would not move naturally or display affect. While this might seem unexpected, preliminary results from our lab indicate that even the perception of naturalness appears to vary markedly under the influence of cognitive factors. Our current work is aimed at studying human-humanoid interactions at the sticky-hands game with the bottom-up approach to the generation of natural and affective movements. However, as this work develops we anticipate an increasing role for cognitive factors in the design of humanoid interaction.

We have recorded 3D data describing the arm movements of people playing sticky hands, attempting to move in a graceful way, and performing partial Tai Chi forms. We plan to record and analyze people's motions, including Tai Chi experts and novices, to compare the mathematical characteristics described above. We consider that the partner interactive aspect of the sticky hands game may invalidate some of the motor control theories since they all pertain to solitary motions. The most significant characteristics of human-like and expert-like motions, and characteristics encapsulating affect, will be used to modify the joint configurations output by the ‘posture controller’ of Fig. 1.

Such modifications may be applied prior to, during or after the inverse kinematics (IK) algorithm. Trajectory based constraints may be applied by adjusting the target coordinate before engaging IK. Constraints on the joint configurations may be incorporated into the IK's search for a solution or enforced by transforming output postures. Spatiotemporal characteristics may be varied[25] by digitally filtering the joint angles to extract the necessary data. Some postural adjustments may cause the hand position to change. As long as changes do not occur discontinuously this is desirable since it incorporates a change in the actual paths performed by the robot according to its mood or style.

4 Discussion

We proposed the ‘sticky hands’ game as a novel interaction between human and robot. We implemented the game using a robot controller process and learning algorithm with a novel internal representation. Since our approach was non-pattern based, branching trajectories were implicitly handled without the need for segmentation analysis. Arbitrary speed and space limits could also be set by restricting the number of prototypes examined or stored. We thus facilitated physically intimate interactions with the humanoid robot, allowing it to take on the role of playmate and partner aiding a human's self-development. The system required minimal sensor input. Only torque and joint position sensors were used by the low level motor controller for the robot. We endorse that observation that making the best use of minimum sensor capability maintains minimum system load and is partially justified through comparison with the human prototype that capitalizes on limited inaccurate sensors[1]. Our system is limited to the context of the sticky hands interaction but we have demonstrated that subtle and responsive behaviors can be synthesized without recourse to alternate sensor modalities. Our work may be considered as a novel communication mechanism that accords with the idea that an autonomous humanoid robot should accept command input and maintain behavioral goals at the same level as sensory input[1]. However, regarding the issue of human instruction we have demonstrated that the blending of internal goals with sensed input can yield complex behaviors that demonstrate a degree of initiative. This places us in opposition to the assertion that without human instruction the design of reinforcement functions or progress estimators is a difficult problem often leading to learning brittle behaviors[3].

Our intention was, in part to facilitate comfortable interaction with a humanoid robot, and encourage its perception as a humanoid entity. We therefore discussed our intended development of the system: to augment the robot's motion with human-like style and emotionally expressive qualities. In so doing we illustrated the relevance and speculated on the future uses of the fields of perceptual psychology, human motor production and computer graphics for humanoid robotics. One issue that we have not yet discussed is mutual attention. The focal interest of the human's and robot's hands precipitated a predictable locus of attention on the part of the human and likewise an expectation for the anthropomorphic robot's attention to be the same. Driscoll *et al.*[37] identified that mutual attention is important for interaction and communication between human and robot, suggesting that an active vision head was of particular importance. Our system simulated the robot's attention using the simple technique of maintaining the robot's gaze (distinguishable by the orientation of its two eye cameras) at the tip of its own hand. The gaze point saccaded to the human's face at significant instants perceived as mistakes. *i.e.* when the trajectory prediction algorithm was corrected following a contact force imbalance. The location of the human's face was estimated based on their expected stance in front of the robot. Although not ratified by experiment we assert that this simple mechanism engendered a significant sense of attention and responsiveness on the part of the robot. It has been remarked that mutual attention is a critical precursor to social behaviors and that performance may be evaluated through emotive, postural and attentive states of a partner[3]. In our system it is certainly true that a human is more aware of their mistakes regarding the contact force when the robot appears anxious due to repeated saccades to the human. We are inclined to disagree by example with the assertion in the same work that even the simplest of joint attention behaviors require the coordination of a large number of perceptual, sensory, motor, attentional and cognitive processes. However, we also consider that our gaze point system is very simplistic given the multiple functions of gaze such as determining the important features of a task, indicating attention and mood state, qualifying comprehension and querying a partner[3]. A detailed examination of saccadic eye motions in humans reveals that the relationship between attention and fixation point is complex and not yet fully established[38]. Most models relate saccadic eye motions directly to spatial attention although the pre-motor theory also incorporates body motions. We made a simplistic choice of setting the gaze point by sharing the angles of head and eyes by a fixed proportion.

Finally, as a future goal we would also like to investigate the development of a long term relationship through sticky hands, allowing user-specific profiles to be constructed. In this way a user's motion style may be developed and analyzed. The robot may respond to the human's motion in a style complementing the perceived emotional state, perhaps acting therapeutically. *i.e.* by yielding to rapid irritable motions or encouraging development when the human is perceived to be calm and relaxed.

5 Acknowledgments

We would like to thank Marcia Riley for much useful assistance during the development of the system, Chris Atkeson for much helpful advice, Shin'ya Kotosaka for his kind and patient help with the robot control system, Tomohiro Shibata for mechanical assistance with the robot, Ales Ude for his assistance with a video presentation, and Stefan Schaal for developing the simulation environment that facilitated the completion of this project within the time constraints.

References

1. Bergener, T.; Bruckhoff, C.; Dahm, P.; Janben, H.; Joublin, F. & Menzner, R. (1997): *Arnold: An Anthropomorphic Autonomous Robot for Human Environments*, Proc. Selbstorganisation von Adaptivem Verhalten (SOAVE 97), 23-24 Sept., Technische Universitt Ilmenau
2. Takeda, H.; Kobayashi, N.; Matsubara, Y. & Nishida, T. (1997): *Towards Ubiquitous Human-Robot Interaction*, Proc. of IJCAI Workshop on Intelligent Multimodal Systems, Nagoya Congress Centre, Nagoya, Japan Aug. 24
3. Scassellati, B. (1999): *Knowing What to Imitate and Knowing When You Succeed*, Proc. of AISB Symposium on Imitation in Animals and Artifacts, Edinburgh, Scotland, 7-9th April
4. Williamson, M.M. (1996): *Postural primitives: Interactive Behavior for a Humanoid Robot Arm*, Proc. of SAB 96, Cape Cod, MA, USA
5. Rose, C.; Bodenheimer, B. & Cohen, M.F. (1998): *Verbs and Adverbs: Multidimensional Motion Interpolation*, IEEE Computer Graphics & Applications 18(5)
6. Stokes, V.P.; Lanshammar, H. & Thorstensson, A. (1999): *Dominant Pattern Extraction from 3-D Kinematic Data*, IEEE Transactions on Biomedical Engineering 46(1)
7. Cormen, T.H.; Leiserson, C.E. & Rivest, R.L.: *Introduction To Algorithms*, McGraw-Hill, ISBN 0-07-013143-0
8. El-Nasr, M.S.; Ioerger, T.R.; Yen, J.; House, D.H. & Parke, F.I. (1999): *Emotionally Expressive Agents*, Proc. Computer Animation '99, 48-57. Los Alamitos, California: IEEE Computer Society Press
9. Amaya, K.; Bruderlin, A.; & Calvert, T. (1996): *Emotion from Motion*, In Graphics Interface '99, Davis, W.A. & Bartels, R. (eds), 222-229
10. Flash,T. & Hogan,N. (1985): *The coordination of arm movements: An experimentally confirmed mathematical model*, Journal of Neuroscience 5:1688-1703

11. Uno, Y.; Kawato, M. & Suzuki, R. (1989): *Formation and control of optimal trajectory in human multijoint arm movement*, Biological Cybernetics 61:89-101
12. Kawato, M. (1992): *Optimization and learning in neural networks for formation and control of coordinated movement*, In Meyer, D. & Kornblum, S. (eds), Attention and performance XIV:821-849. Cambridge, MA: MIT Press
13. Harris, C.M. & Wolpert, D.M. (1998): *Signal-dependent noise determines motor planning. Nature 394:780-784*
14. Soechting, J.F. & Terzuolo, C.A. (1987): *Organization of arm movements. Motion is segmented*, Neuroscience 23:39-51
15. Soechting, J.F. & Terzuolo, C.A. (1987): *Organization of arm movements in three-dimensional space. Wrist motion is piecewise planar*, Neuroscience 23:53-61
16. Lacquaniti, F.; Terzuolo, C.A & Viviani, P. (1983): *The law relating the kinematic and figural aspects of drawing movements*, Acta Psychologica 54:115-130
17. Pollick, F.E. & Sapiro, G. (1996): *Constant affine velocity predicts the 1/3 power law of planar motion perception and generation*, Vision Research 37:347-353
18. Pollick, F.E.; Flash, T.; Giblin, P.J. & Sapiro, G. (1997): *Three-dimensional movements at constant affine velocity*, Society for Neuroscience Abstracts 23(2):2237
19. Komura, T. & Shinagawa, Y. (1997): *A Muscle-based Feed-forward controller for the Human Body*, Computer Graphics forum 16(3):165-176
20. Paterson, H.; Pollick, F.E.; Calder, A.J. & Sanford, A.J. (1999): *The perception of affect from point-light displays of simple arm movements*, European Conference on Visual Perception, Perception 28:S53
21. Russell, J.A. (1980): *A circumplex model of affect*, Journal of personality and social psychology 6:1161-1178
22. Hill, H. & Pollick, F.E. (2000): *Exaggerating temporal differences enhances recognition of individuals from point light displays*, Psychological Science, Psychological Science 11(3):223-228
23. Fidopiastis, C.M. & Pollick, F.E.: *Recognizing the style of spatially exaggerated tennis serves*, Perception, *in press*
24. Viviani, P. & Stucchi, N. (1992) *Biological movements look uniform: Evidence of motor-perceptual interactions*, Journal of Experimental Psychology: Human Perception and Performance 18:602-623
25. Bruderlin, A. & Williams, L. (1995): *Motion Signal Processing*, Proc. SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series, 97-104
26. Yamane, K. & Nakamura, Y. (2000): *Dynamics Filter: Towards Real-Time and Interactive Motion Generator for Human Figures*, Proc. WIRE 2000, 27-34, Carnegie Mellon University, Pittsburgh, PA
27. Hidgins, J.K. & Pollard, N.S. (1997), Proc. SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, 153-162
28. Witkin, A. & Kass, M. (1988): *Spacetime Constraints*, Proc. SIGGRAPH 88, Computer Graphics Proceedings, Annual Conference Series, 159-168
29. Cohen, M.F. (1992): *Interactive Spacetime Control for Animation*, In Proc. SIGGRAPH 92, Computer Graphics Proceedings, Annual Conference Series, 293-302
30. Ngo, J.T. & Marks, J. (1993): *Spacetime Constraints Revisited*, In Proc. SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series, 343-350
31. Li, Z.; Gortler, S.J. & Cohen, M.F. (1994): *Hierarchical Spacetime Control*, In Proc. SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series, 35-42
32. Rose, C.; Guenter, B,; Bodenheimer, B. & Cohen, M.F. (1996): *Efficient Generation of Motion Transitions using Spacetime Constraints*, In Proc. SIGGRAPH 96, Computer Graphics Proceedings, Annual Conference Series, 147-154
33. Gleicher, M. (1997): *Motion Editing with Spacetime Constraints*, In Proc. 1997 Symposium on Interactive 3D Graphics
34. Sato, J. & Miyasato, T. (1997): *Autonomous Behavior Control of Virtual Actors Based on AIR Model*, Computer Animation 97, 113-118
35. Unuma, M., Anjyo, K. & Takeuchi, R. (1995): *Fourier Principles for Emotion-based Human Figure Animation*, Proc SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series, 91-96
36. Badler, N.I. & Chi D.M. (1999): *Virtual Human Animation Based on Movement Observation and Cognitive Behavior Models*, Computer Animation 99, 128-137
37. Driscoll, J.A.; Peters II, R.A. & Cave, K.R. (1998): *A Visual Attention Network for a Humanoid Robot*, Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Innovations in Theory, Practice and Applications, IEEE, New York, NY, USA, 3:1968-74
38. Clark, J.J. (1999): *Spatial Attention and Latencies of Saccadic Eye Movements*, Vision Research 39(3):585-602