# Interpolation Synthesis for Articulated Figure Motion

Douglas J. Wiley and James K. Hahn
The George Washington University

## Abstract

*Realistic real time articulated figure motion is achieved by reprocessing a stored database of motions. Motions are created to exact specification by interpolation from a set of example motions, effectively forming a parameterized motion model. A pre-processing step involving iterative calculations is used to allow efficient direct computations at run time. An inverse kinematics capability is shown that is based on interpolation. This method preserves the underlying qualities of the data, such as dynamical realism of motion capture, while generating a continuous range of required motions. Relevant applications include networked virtual reality and interactive entertainment.*

## 1: Introduction

Engaging and appealing characters are an important aspect of most conventional media. Empty spaces and buildings would not fare well as television or movie programming. Virtual reality, however, usually offers up such empty spaces. The problem lies in the difficulty of creating engaging interaction and realistic motion of real time computer-generated characters. Video production in the computer and traditional animation industries largely relies on a labor-intensive process known as keyframing, where individual limbs are positioned at successive instants of time. This approach does not translate well to real time characters with intriguingly rich varieties of motion. Motion capture allows more rapid collection of motion, but still suffers from the inherent lack of control of prerecorded motion data. Physical simulation approaches offer promise but suffer from lack of control, difficulty of use, instabilities, and computational cost that usually precludes real time operation.

The predominant approach to the problem of real time motion synthesis is storage of a variety of motions and selection of the most appropriate motion at run time. Computation is restricted to that of displaying the stored motion. Unfortunately, the number of stored motions is limited, resulting in potentially repetitive or imperfectly-matched results.

In this paper, we show that the repertoire of possible motions can be greatly expanded at the cost of additional computation by interpolation synthesis. A set of motions that are similar to the desired motion are combined by interpolation to form a specified motion exactly. A small set of example motions then yields a continuous multidimensional space of possible motions. The example motions can be obtained from keyframing, physical simulations, or motion capture. The subtle qualities of the motion are generally preserved in the interpolation process. Reusable libraries of motion data are thus created from a single step of laborious keyframing, iterative guessing of initial conditions, or correction of errors and dropouts, respectively. This paper demonstrates creation of motion libraries from motion capture, with real time interpolation synthesis of articulated figure motion. This approach is suitable for real time character motion in interactive entertainment, or avatars in multiplayer networked games or virtual worlds. After transmission of a motion database, network traffic would consist merely of a motion specification indicating parameters of the interpolation.

Previous authors have described pairwise interpolation of motion in various representations, as well as the goal of creation of motion from a stored database[1,2,3,5]. Parameterized motion synthesis of gait has also been demonstrated[3]. A technique of combining keyframed data with motion capture data has been shown that allows insertion of specific figure pose into a recorded sequence[1,5]. These authors used interpolation of joint angles as well as frequency-based representations. The necessity of aligning the starting and ending times of motions before interpolation was also described using a non-uniform time scaling[1]. Another work described sequencing different stored motions over time[2].

The stability of the quaternion representation used in this paper allows a greater degree of interpolation than those used previously. Motions of greater dissimilarity can be combined, with subsequent increase in range of interpolation synthesis result. This point is illustrated in this paper by the first reported achievement of inverse kinematic capability by interpolation, where an articulated figure can be made to reach to any specific point in space. This is done by a direct computation, unlike the more common robotics formulations, and without defaulting redundant degrees of freedom arbitrarily.

## 2: Articulated figure representation

An articulated figure is composed of rigid limbs and joints with one, two or three degrees of freedom. Euler





Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)
0-8186-7843-7/97 $10.00 © 1997 IEEE

angles are commonly used to specify the pose of a figure, where each limb's position is specified relative to its attachment point in a hierarchy. Instead, a hybrid position and orientation representation has been used, corresponding to data from magnetic trackers. The center of rotation of the lower limbs (legs, arms), chest and hips is specified in absolute terms, Fig. 1 (trihedral origins). A rigid lower limb, hip or chest segment is attached at the tracker positions (solid lines), while upper limb and torso sections are defined by the line between shoulder or hip points and lower limb origins (dashed lines). The absolute orientation of each segment is specified by a quaternion[4]. While this representation does not ensure constant upper limb or torso length, a step of conversion to fixed limb lengths and orientations is used when accuracy is desired.

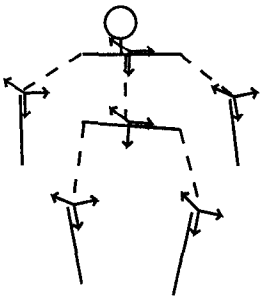

**Fig. 1. Position/quaternion pose representation corresponding to magnetic tracker data.**

## 3: Interpolation synthesis

A set of figure poses or sequences of poses is obtained through motion capture. The challenge then involves efficient selection and recombination of the pose data to form a specific new pose or sequence of poses. This operation can be accomplished efficiently by a stage of pre-processing that interpolates and resamples the data to a new form. This relegates iterative processing and searching computations to an initial step, using direct computations at run time.

Description of the process is best illustrated by the most important demonstration achieved, that of inverse kinematics functionality. A figure stands and reaches out with one arm to a point in space. Given a target point in space, we wish to specify all components of the figure representation that place the hand at the required position. Interpolation synthesis begins by collecting a set of poses that place the hand in a variety of positions in space. The goal is then to find a set of poses that bound the target point in all three dimensions and are the closest possible poses in the set. Due to the inevitable lack of precision in the data, a search can not be made on one dimension at a time. Exhaustive search is one possible solution, but clearly is not the best choice. The answer is to resample the data on one dimension at a time, producing new poses that lie on an orthogonal regular grid of the output space, hand position. Efficient search is then accomplished by merely dividing the target point distances by the grid spacing, and employing the remainder for interpolation coefficient.

Linear interpolation is used for each vector position component of the representation, $a$, $b$ and result $c$,

$$c = (1-u)*a + u*b, \quad u \text{ in } [0,1] \tag{1}$$

and spherical linear interpolation is used for each quaternion orientation component $p$, $q$ and result $r$ [4],

$$r = q*\sin((1-u)W)/\sin(W) + p*\sin(uW)/\sin(W), \quad u \text{ in } [0,1], \; W=q \bullet p. \tag{2}$$

Multiple interpolation in a pyramid or binary-tree progression is used on each dimension successively, $x$, $y$ then $z$ (Fig. 2, data d, intermediate i, final f). This interpolation can also be performed on data that does not lie on a regular grid, after iterative search.

Interpolation coefficient $u$ is obtained by using an assumption of linearity of the inputs with respect to the outputs in each dimension. Given target $t$, lying between grid points $l$, and $h$,

$$u = (t-l)/(h-l), \tag{3}$$

and the same interpolation coefficient is used for the corresponding quaternion interpolations.

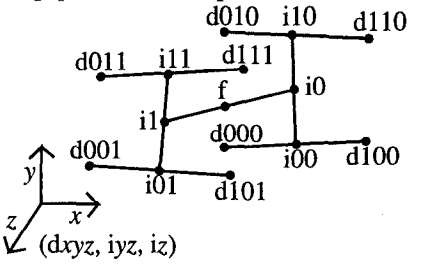


**Fig. 2. Trilinear interpolation pyramid: four interpolations in the *x* direction, two in the *y* direction and a final interpolation in the *z* direction.**

The above direct computation results in a pose that approximates the target goal satisfactorily given sufficiently dense data. Enforcement of limb length can be employed if necessary. Higher accuracy or sparse data can be accommodated using iterative optimizations if desired.

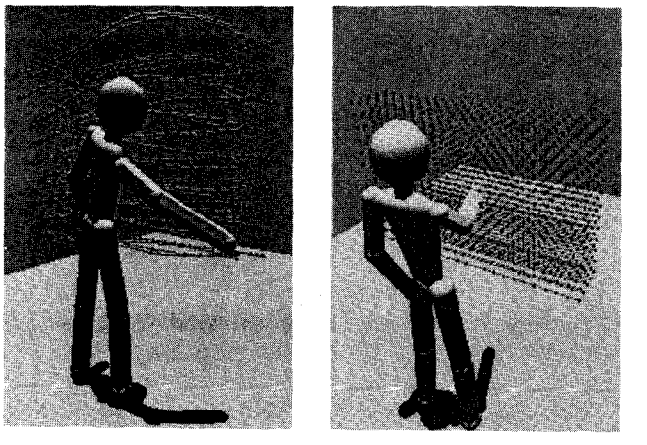

**Fig. 3. Initial, 3a, and resampled data, 3b.**



Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)
0-8186-7843-7/97 $10.00 © 1997 IEEE

A version of the direct computation is depicted in Fig. 3. Four vertical planes of data consisting of nine lines were traced out by the motion capture actor. Fig. 3a shows initial data, Fig. 3b, grid resampled data. The data was first resampled over each line in the figure's left to right direction on a regular grid, then the results were connected vertically and resampled at a regular grid in height, yielding four planes of data on a regular grid in width and height but not depth. Finally, lines connecting grid points in each plane were formed and resampled at a regular grid in depth. Real time figure positioning was then demonstrated by trilinear interpolation with interactive goal point input. Convincing subtleties such as knee bending, back bending and motion of the other arm occur during operation, increasing the realism of the figure's motion. No apparent inaccuracy in positioning is observable in this demonstration due to the use of approximately 100 data points in each plane. The result motion looks convincingly like direct motion capture data due to the motion capture origin of the data. Synthesis using expressive and exaggerated motion from keyframing would conceivably retain those qualities as well.

## 4: Motion interpolation synthesis

The previous example synthesized static pose instantaneously from a specification that was varied in real time. The interpolation synthesis technique also extends to synthesis of entire motions of finite duration or finite period. This will be illustrated by a generalization of the above to synthesis of a reaching motion. The figure stands at rest with arms at its sides, then reaches out to a target point, then the arm returns to the original position. Given a target point, we wish to generate a sequence of poses that produce the appropriate reach.

The first step is to obtain a set of reach motions from rest to various points in space. Interpolation synthesis is controlled by a single pose at the instant of reaching to the point in space. The data is organized by this pose, and synthesis is carried out on the basis of this pose much as for static pose synthesis. All other poses of the motion are interpolated identically. However, an additional step of re-sampling in time is necessary.

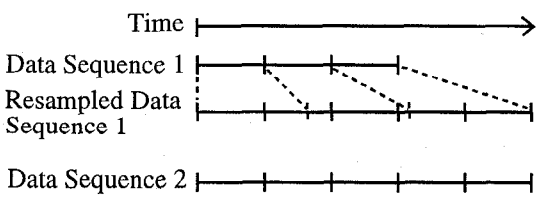


**Fig. 4. Resampling an interpolated sequence to rescale to a uniform number of samples (longest of set is sequence 2).**

Each reach sequence is resampled to a uniform (longest of set) number of samples so that interpolation can be applied at each time point between data sequences, Fig. 4. A uniform time scaling is used to prevent alteration of the natural qualities of the data. The durations of the original data are stored and multiple interpolation is performed to determine the required duration of the final result. A final time scaling then produces the result from the uniform duration multiple interpolation.

## 5: Results of interpolation synthesis

Real time reach synthesis was demonstrated with interactive goal point positioning. Reach motion data for two planes of nine goal points was recorded, Fig. 5a. Uniform duration reach sequences were stored as the data for multiple interpolation. No spatial re-sampling was used, since only four possible interpolation windows existed. Reach motion was generated using limb length enforcement and iterative accuracy optimization, Fig 5b. A further demonstration was made using the reach point for inverse kinematics. A helical path was specified with a reach to the initial position and return to rest from the end position. The helix was deformed to a "D" shape by limb length enforcement in Fig. 5c. All demonstrations ran in real time, with negligible delay to calculate an interpolated result motion, on a 100 MHz R4400 processor workstation.

An example of parameterized gait synthesis was also shown. A motion capture actor was recorded walking up and down five different actual slopes. The data was blended at the ends of the walk sequence to form a seamless loop, and resampled to a uniform number of samples. Interpolation synthesis then provided a walk cycle at any intermediate slope value by interpolating the data sequence just above and below the target slope. Slope was altered interactively, and both gait and gait cycle duration changed continuously in real time, Fig. 6. The result has a realism that would be difficult to generate by keyframing or conventional robotic inverse kinematic techniques.

Chalkboard writing was shown in a variation of static pose generation. The entire alphabet was written in a single plane of space, and only the hand position was recorded. The alphabet hand trajectory was used to drive the figure pose, producing any sequence of letters starting at any position in space upon interactive keyboard entry, Fig. 7.

## 6: Analysis and discussion

The biggest problem with interpolation synthesis is the limitation to small numbers of parameters, on the order of ten. The examples presented here involved three and one parameter. The amount of data required at least doubles for every additional parameter, since the entire previous data set is repeated for at least two values of the additional parameter. In general, for $p$ parameters, $2^p - 1$ interpolations are performed with a window of $2^p$ data samples. The interpolations are performed for every time



Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)
0-8186-7843-7/97 $10.00 © 1997 IEEE

step of the motion, for all degrees of freedom of the character, 6*(3+4) for the representation used here. Motion segment durations are also interpolated $2^p$-1 times. A final time rescaling occurs for each degree of freedom of the result. The interpolation window and blending coefficients are found only once in synthesis of an entire motion, as the same window and coefficients are used at each time step. Pre-processing steps on the data (performed only once) consist of re-sampling to uniform length, and to a regular grid in the synthesis parameters.

Storage, interpolation requirements, and final time re-scaling calculations scale linearly with both number of degrees of freedom of the character, and time sample rate of the data. This is important for extension of the work to more detailed representations.

A tradeoff exists between density of the data in the parameter space and storage requirements. Higher data density leads to greater accuracy, stability and continuity between adjacent parameter set values. The data can be resampled at varying grid spacings as well, again trading off accuracy for storage.

The drawbacks of the two-stage process are outweighed by the benefits. Motion capture data that requires extensive manual correction of errors and dropouts can be corrected just once, with subsequent interpolation synthesis based on the corrected data. The trial and error involved in keyframing could also be reused by interpolation synthesis from keyframed data. The process also greatly decreases the accuracy requirements of motion capture recording sessions or simulation runs. A representative sample set is needed, as opposed to the exact motion that will be used.

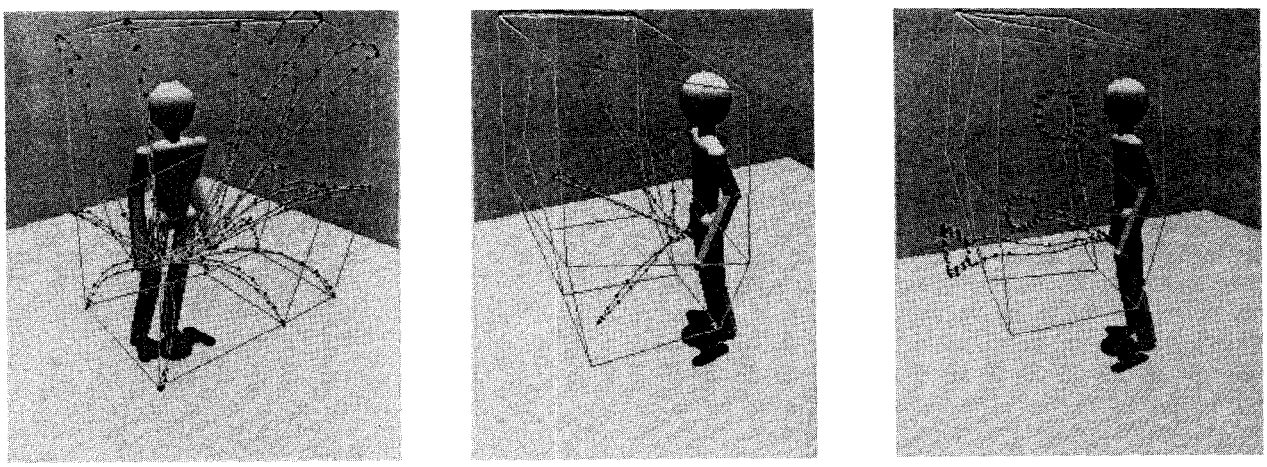

**Fig. 5. Initial reach data, 5a, reach to specific points in space, 5b, and helical path insertion, 5c.**

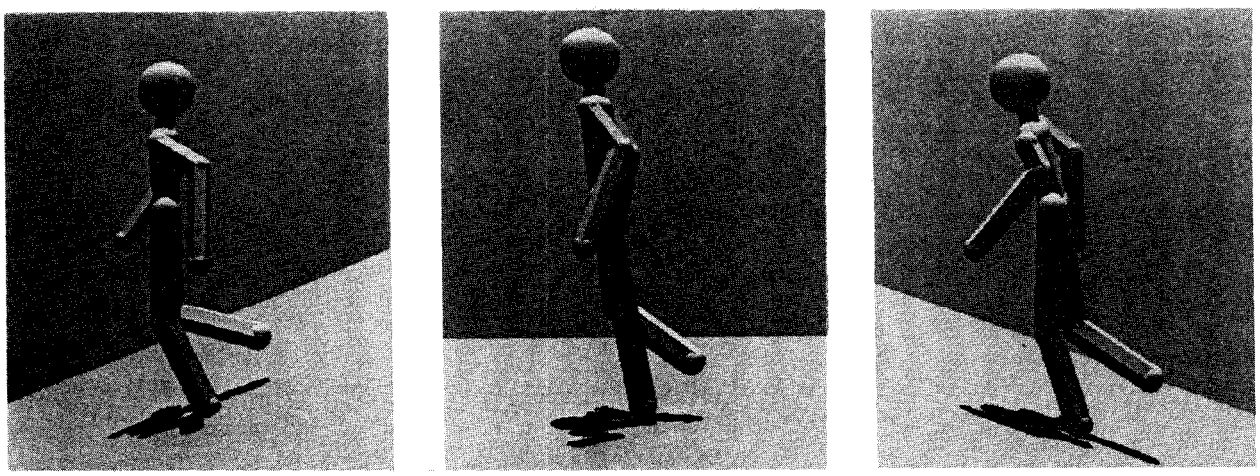

**Fig. 6. Variable slope walk cycle downhill, level and uphill.**



Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)
0-8186-7843-7/97 $10.00 © 1997 IEEE

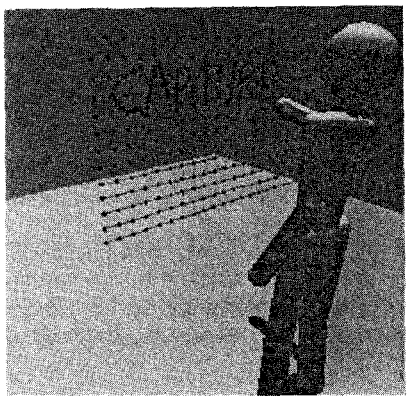

**Fig. 7. Chalkboard writing using interpolation inverse kinematics.**

## 7: Conclusions

The interpolation synthesis process using position/quaternion representation yields stable multiple interpolation with surprisingly sparse data, as the reach example shows. Limb length enforcement can be applied if necessary, and iterative optimization can improve accuracy as needed. Sufficiently dense data with direct computation is adequate for many interactive entertainment applications. The interpolation process thus allows trading off increased (client) interpolation calculation for decreased storage (or networked transmission) requirements. The spatial ordering and resampling techniques are useful even if motions are used exactly as stored, providing a direct search and choice of sampling density for stored motion. The emergence of low cost, powerful processors and three dimensional graphics hardware should make interpolation calculations less of a penalty in future consumer graphics and networked virtual reality, however.

This approach offers a unique combination of realism and controllability for real time motion synthesis. While limited to a small number of parameters, the technique provides a far richer range of motions than the use of pre-stored motions directly. The combination of motion capture and interpolation synthesis has the potential for enabling the characters needed in currently empty virtual reality environments, and adding a rich variety of motion to avatars in networked virtual worlds.

## Acknowledgements

Support for this work was provided by the Naval Research Laboratory in Washington D.C., under contract N00014-94-2-C002 and grant N00014-94-K-2009 to George Washington University.

## References

[1] Bruderlin, A., and Williams, L., "Motion Signal Processing," *Computer Graphics* (Proc. SIGGRAPH '95) Vol. 29, No. 4, August 1995, pp. 97-104.

[2] Perlin, K, "Real Time Responsive Animation with Personality," IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 1, March 1995, pp. 5-15.

[3] Unuma, M., Anjyo, K., and Takeuchi, R., "Fourier Principles for Emotion-Based Human Figure Animation," *Computer Graphics* (Proc. SIGGRAPH '95) Vol. 29, No. 4, August 1995, pp. 91-96.

[4] Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques*, Addison-Wesley, Wokingham, England, 1993.

[5] Witkin, A., and Popovic, Z., "Motion Warping," *Computer Graphics* (Proc. SIGGRAPH '95) Vol. 29, No. 4, August 1995, pp. 105-108.



Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)
0-8186-7843-7/97 $10.00 © 1997 IEEE