Parametric Motion Blending through Wavelet Analysis

Amr Ahmed, Farzin Mokhtarian, and Adrian Hilton

Centre for Computer Vision, Speech, and Signal Processing,

Department of Electronic Engineering,

University of Surrey, Guildford, Surrey GU2 7XH, UK.

Abstract

This paper shows how multiresolution blending can be employed with time-warping for realistic parametric motion generation from pre-stored motion data. The goal is to allow the animator to define the desired motion using its natural parameters such as speed. Generation of a realistic motion is achieved using pre-stored captured animations. Analysis has been carried out to investigate the relationship between the walking speed and blending factor to remove the burden of trial and errors from the animator. As a result, realistic walking motion with the speed specified by the user can be generated. This desired speed should be between the minimum and maximum speeds of the available motion data. Analysis to generalise these results to other motions are in progress. Generating the desired motion for different scaled avatars is also discussed.

1. Introduction

Realistic human motion animation is still a challenging task although human motion appears to us (as humans) to be very easy and natural behaviour. The real human motion has many unique characteristics that identifies it from synthetic ones. The absence of these characteristics (even the very small ones) results in unnatural or robot-like appearance. This unnatural appearance could be easily noticed by humans but most probably it is not easy to identify its source.

With the development of virtual reality, the demand has been increased for virtual humans in a wide variety of fields and applications from games and entertainment to simulation and scientific visualisation. As a result, the need for realistic human motion animation is increasing rapidly. The most realistic animation is that which can preserve the unique human characteristics. In that sense, computer human animation using the motion captured data can produce more natural-looking and realistic animation. As the motion is captured from real people, the generated animation is more realistic and physically correct. The motion captured animation becomes more realistic with the development of more advanced and accurate motion capturing systems and techniques.

The problem appears when the captured animation needs to be modified. Even if the needed modification is very small, most probably the whole capturing procedure should be repeated to satisfy the desired motion. This also happens if the captured animation is to be applied to another human model (with different properties) which is referred to as the retargeting problem.

To benefit from the advantages of the motion captured data in human animation, analysing and editing systems have to be available. These systems should provide an easy and reliable way to edit and/or modify the captured data (within some limits) to produce the desired motion. This may be done by modifying the motion parameters (speed, step frequency/length, .., etc.), mode or emotional status (tired, happy, angry, .., etc.).

The goal of this research is to provide a natural and easy way for the animator to define the desired motion using the natural human motion parameters. The desired motion is generated using the multiresolution blending and time-warping techniques based on existing pre-stored animation data. This results in a parametric motion blending which could be a framework to parametrise the motion captured data.

In the next section, an overview of the previous work in the editing and modification of the human motion animation is shown. Sections 3 and 4 show the use of wavelets as a powerful signal processing tool in motion editing and motion synthesis using the multiresolution blending respectively. The proposed analysis in parametric multiresolution motion blending is presented in section 5. Then, a brief discussion of using it to produce a desired motion for scaled avatars is shown in section 6. As the research is still ongoing and in its early stages, a brief conclusion is introduced in section 7.

2. Previous Work

As motion capture techniques and equipments are rapidly and continuously improving, the interest in developing tools for modifying the motion animation is also increasing. Regardless of how this animation has been created, the editing tools allow us to benefit from any existing animation clips. This section shows the key work in the motion editing and modification.

Ko and Badler [1] used interpolation to generate arbitrary anthropometry walking with arbitrary step length from rotoscoped data. They assumed a linear relationship for different step lengths which is a simplification of the real case.

Wiley and Hahn [2] applied the linear interpolation technique to pre-stored motion data to generate new motions. To synchronise the pre-stored samples, they resampled the data to a uniform time scale. However, this does not guarantee the synchronisation of the key-events of the motions and consequently does not guarantee a realistic motion animation.

Witkin and Kass [3] introduced the spacetime constraints technique. The motion synthesis is considered as a constrained optimisation problem and solved for the whole animation duration instead of individual frames. This results in a high computation complexity and reduces the interactivity with the environment during the animation. Cohen [4] proposed an improved spacetime constraint method called 'spacetime windows' in which the solving is done for sub-periods of the animation to improve interactivity. Gleicher [5] suggests reducing some constraints (physical constraints) in order to improve the performance.

As the captured motion data is a set of time-varying signals, many techniques from signal processing have been applied to the motion editing and modification.

Unuma et.al [6] used the Fourier series expansion of the joint trajectories of the pre-stored data to interpolate and make transitions between motion samples. As the technique is based on the Fourier analysis, it is valid for periodic motions only. They reported also that the transitions are not fully invertable. For example, the transition from walking to running could be achieved while the transition from running to walking does not look natural. Also, there is no guarantee for the resulting motion to be realistic.

Witkin and Popovic [7] show that the motion warping technique could be used for editing captured motion. They reported that the key advantage is the ability of that technique to be integrated with the existing key-framing tools. Their motion warping still has some inherited limitations from the standard key-framing such as the need for additional effort to satisfy the geometric constraints. Moreover, the technique does not incorporate any knowledge about the motion. So, realistic results are not guaranteed.

In [8], Bruderlin and Williams present a simple library of signal processing techniques for motion editing. Pyramid filters were used for multiresolution motion filtering, time warping was used as a useful way of synchronising motions and waveshaping was presented as a simple and effective method of producing some effects on the different degrees of freedom. Motion displacement mapping was introduced as a useful tool for modifying basic motions through a standard key-framing interface. It still needs some interpolation between the modified key-frames.

The most relevant work is [9] in which the wavelet analysis and its multiresolution properties are used to model bipedal locomotion. The decomposed motion curves can be edited or blended on any resolution level independently. Further analysis and extensions of this approach are introduced in this paper to achieve the parametric multiresolution blending to make the animator's task easier for generating realistic animation. In the next sections, using the wavelets as a tool for editing and modifying motion is briefly described and the proposed analysis is presented.

3. Motion Editing Using Wavelet Analysis

Wavelet analysis is one of the signal processing tools that is powerful for analysing signals especially the non-stationary ones [10]. Wavelet analysis is useful in multiresolution curve editing [11, 12]. As the motion properties are hidden within the motion curves, some or all of these properties can be destroyed during editing of these motion curves. Using wavelets multiresolution property, the curve can be decomposed into many resolution levels. The low frequency resolution level of the curve, the coarse level, represents the overall trend or the main pattern of the motion while the high frequency resolution level, fine/details level, represents the style, mode, personality which has been found to be included in the high frequency contents [7, 8, 12].

Editing the curve on its coarse level affects the main pattern of the motion while editing it on its fine level affects the style of the motion. This multiresolution curve editing facilitates a lot of variations of the motion to be produced. Unlike using the Fourier transform in which the variations affect all the motion (as it is local only in frequency domain but not in time domain), the wavelet transform is local in both time and frequency domains. So, editing a specific part of the motion is possible without damaging the other parts.

The next section describes the methodology of synthesising new motion using the multiresolution blending of pre-stored motions.

4. Synthesis by Mutliresolution Blending

The motion blending in general is an operation that employs the interpolation between two (or more) motion data in order to produce a new motion that is related somehow to the blended motions. For two motion curves 'M_1' and 'M_2', the simplest form of the blending operation is using the linear interpolation (or weighted sum) between the two motion curves using a formula like:

$$M_n = X * M_1 + (1-X) * M_2$$

where:

M_n is the new motion curve.

X is called the blending factor.

M_1, M_2 are the original motions curves.

Using the multiresolution property of the wavelets, the blending operation could be applied to each resolution level independently. So, carrying the blending operation on the coarse level results in new motion pattern (related to the original motions). On the other hand, carrying the blending operation on the fine level results in new characteristics (related to the original characteristics). The similarity of the synthesised motion to the original motions is controlled by the value of the blending factor.

As we have different resolution levels and we can blend each level independently, we actually have more than one blending factor. There are 'n' blending factors in the case of 'n' resolution levels. Generally, given the two original motions (S_1 and S_2) in the wavelet transform representation (as a list of coarse 'C' and details 'D' coefficients)as follows:

$$S_1 = C_1^N, D_1^{n-1}, D_1^{n-2}, ..., D_1^N$$

$$S_2 = C_2^N, D_2^{n-1}, D_2^{n-2}, ..., D_2^N$$

Where:

N= *the coarse level.*

n= *the highest resolution version of the motion (as it is originally given).*

C^N = *the coarse coefficients at the level 'N'.*

D^i = *the details/fine coefficients at the level 'i'.*

The new motion could be found by the multiresolution interpolation as follows:

$$S_3 = [X_C\ C_1^N + (1 - X_C) C_2^N], [X_{n-1}\ D_1^{n-1} + (1 - X_{n-1}) D_2^{n-1}], [X_{n-2}\ D_1^{n-2} + (1 - X_{n-2}) D_2^{n-2}], ..., [X_N\ D_1^N + (1 - X_N)\ D_2^N]$$

where:

X_c is the blending factor for the coarse coefficients.

X_i is the blending factor for the details coefficients at the i^{th} resolution level.

5. Parametric Multiresolution Motion Blending

In this section, the details of the current status of the research are presented. We would like to mention that this is still ongoing research and in its early stages.

Using the normal blending operation for generating new motion (or variations of the motion), the blending factor is the controlling parameter which is not a representative parameter of the motion. Moreover, the animator has to carry out many trial and error sessions to find the best value of the blending factor that generates the required motion (or the nearest one to it). The current goal of this research is to provide the animator with a parametric motion blending tool in which the desired motion could be defined by its natural parameter not by the blending factor. This also can be called the inverse motion blending as the blending factor is generated according to the desired motion not the opposite way. Incorporating wavelet analysis allows the multiresolution editing and blending. So, combining the natural way of defining the desired motion, the multiresolution properties of wavelet analysis, with the pre-stored captured motion is found to be a promising technique for generating realistic motion. Moreover, it could be used for motion from other sources as well.

At this stage, some experiments have been carried out on the human walking motion as an example of a complex human motion. These experiments are carried out to investigate the relationship between the blending factor and walking speed as the most natural parameter of the walking motion (Investigating other motions is in progress).

Two categories of experiments have been done. The first category is based on synthetic data and the second category is based on captured data samples of human walking. In the synthetic data experiments, the original motions used are human walking motions with different speeds. These motions have been synthesised using the 'Walking Generator' module of the 'LifeForms' package.

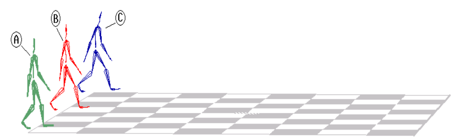
A few assumptions have been made while carrying these experiments as follows:

1. Using the animation data of one walking cycle is found to be enough as the human walking motion is approximately periodic. The main advantage of this assumption is the simplicity and reduction of computing time. On the other hand, extracting the walking motion cycle should be done first (manually at this stage) as well as duplicating the walking cycle after the calculation to make presentation of the result.
2. The used walking cycles consists of the normal steps, which means that they don't include any special cases like start and/or stop steps. This is only for simplicity of the experiments and to provide an easy way to show the results. As the small changes in the walking motion may not be easy to be noticed during short distances (i.e. small number of walking cycles), choosing the distance to be long enough is important to show the results. This is why the normal steps are selected as it could be easily duplicated for any arbitrary number to achieve the desired distance. This helps to present the resulting motion simultaneously with the original motions (on the same stage and view) for any distance or selected number of steps (see Figure 1).

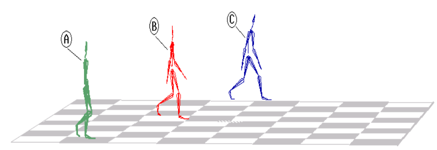
   It is expected that the start and/or stop steps could be included at any further experiment as we don't have any other reason that may prevent applying the same procedure to them as well.

An overview of the procedure used to carry out the experiments can be summarised in the following steps:

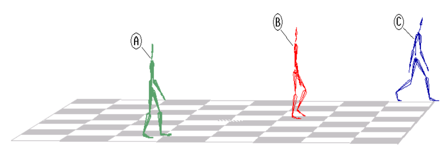
1. From each sample, one normal walking cycle is extracted.
2. For each extracted cycle, the timing of the key events is manually determined. The selected events to guide the motion synchronisation are the heel-strike and mid-swing of both legs.
3. The multiresolution blending module is applied to the extracted cycles of the two samples with the aid of the determined key events timing.
4. The resulting blended motion cycle is duplicated by the required number of cycles for the presentation. Then, it is viewed simultaneously with the original motions.



At the start (Frame 1)



In the middle (Frame 54)



At the end (Frame 106)

From Left to right; the lowest speed sample ('A') , the blended sample ('B'), and the highest speed sample ('C')

Figure 1: Screenshots of the resulting animation

The implemented multiresolution blending module works as follows:

1. Synchronise the key events on both motions using time-warping.
2. Apply the discrete wavelet transform (DWT) to each motion curve to obtain its different resolution levels.
3. Apply the blending operation on each resolution level independently with arbitrary values for the blending factor of each resolution level. Keeping in mind that the blending factor of the coarse level has main effect on the main pattern of the motion.

4. Apply the inverse discrete wavelet transform (IDWT) on the resulting blended motion to reconstruct its curves from their different resolution levels.
5. As the resulting motion has its key event timing still following the original reference motion, another time-warping is required to map it to its own timing which is different from both of the original motions. Estimating the key events timing of the new motion is carried out by interpolating the key events times of the original motions.

As the main difference between the original motions is the walking speed, the generated motion has a different speed somewhere between the two original speeds depending on the blending factor. Figure 1 shows snapshots of the resulting animation (as well as the original ones for comparison) at three different times.

The resulting animation looks as realistic as the original motions. Also, the feet don't slip, don't penetrate the ground and the whole cycle looks normal.

To investigate the relationship between the blending factor and the resulting speed, the previous procedure is repeated many times for different values of the blending factor as well as for different walking samples with different speeds. The main observation of these experiments is that the generated motion speed is not linearly related to the blending factor. This means that using a blending factor of '0.5' does not produce a motion with speed exactly in between the original speeds.

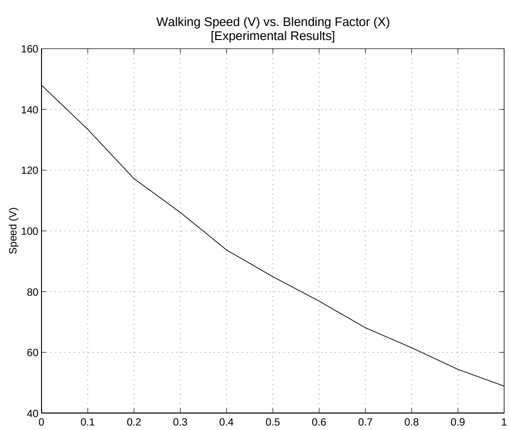


Figure 2: Relation between the speed and the blending factor [Experimental Results]

As shown in Figure 2, the generated speed is usually below its linear values. Using the curve fitting, a 2nd order polynomial is found to suitably represent this relation as follows:

$$V(x) = 54.99 * x^2 - 152.44 * x + 147.25$$

With sum of errors = -1.4e-14.

This relation gives us the speed related to certain blending factor. In practice, we need to generate a certain speed. The blending factor that generates that speed is required. To benefit from that relation in generating a certain speed, the blending factor (X) is needed to be expressed as a function of the speed (V) [i.e. X(v) instead V(x)]. Again from the experimental results, a 2nd order polynomial is found to represent the relation as follows (see Figure 3):

$$X(v) = 0.0001 * v^2 - 0.0209 * v + 1.8753$$

So, given the original two samples, a new realistic motion with specific speed could be generated using this relation.

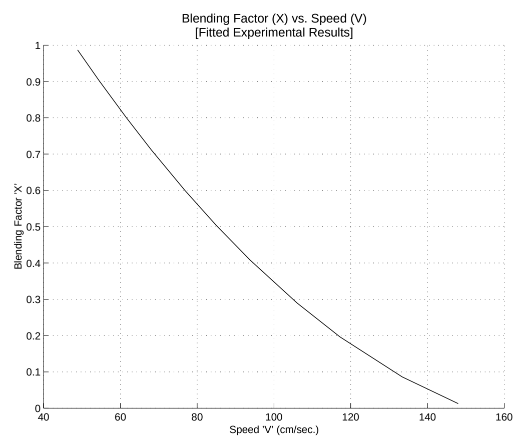


Figure 3: The blending factor as a function of the required speed

5.1 Constraints

The main drawback of most of the motion editing techniques is that there is no guarantee for the modified or edited motion to be realistic. This is because the motion data implicitly preserves some constraints and properties of the motion that is hidden inside the raw data. So, the edited motion may violate some or all of its constraints which results in unrealistic motion.

In general, the constraints are two categories; Time constraints (or temporal constraints) and Geometric constraints (or spatial constraint). In the normal editing process, most probably, an additional effort is needed to preserve or recover both types of constraints. However, in the synthesis using blending operation, the modification is guided by the original motions. This guidance provides some sort of implicit knowledge about the motion which preserve many constraints.

The time constraints are very important to be satisfied for a successful blending process. Since the original motions usually don't have the same time duration, most probably the key events of each motion happen at a different timing from the corresponding key events of the other motion. The simple stretching and/or compression (in time) of one motion to unify the time duration does not guarantee the synchronisation of the key events. So, synchronising the original motions has to be done before carrying the blending operation. This synchronisation is commonly done by the time-warping algorithm [8].

Firstly, times of the key events in both motions should be determined. So, we will have $[t_i^1]$ and $[t_i^2]$ as a list of key events times of the first and second motion respectively. It should be noted that the key events and their times depend on the type of the motion we are working with. So, some level of knowledge about the motion should be available to provide a base for selecting the key events. For example, based on knowledge about the walking motion, the heel-strike and/or toe-off events are some of the candidate key events. They help in preserving both geometric and timing constraints during the blending process.

Once the key event times are determined for both motions, one of the motions could be selected as the reference motion. The key events times of the selected reference motion are considered the reference timing to which the key events times of the other motion should be warped (mapped) to synchronise the two motions. There is more than one formula for warping. Witkin and Popvic [7] have used the cardinal spline and Sun [9] has used a linear warping function and reported that it has no problems in practice.

Using the determined key events times and assuming that the first motion is selected as the reference motion, the time warping could be done as follows:

For each $t^1 \in [t_i^1, t_{i+1}^1]$, the warped time Wt^2 of the second motion could be obtained using the formula :

$$Wt^2 = t_i^2 + (t^1 - t_i^1) * \frac{(t_{i+1}^2 - t_i^2)}{(t_{i+1}^1 - t_i^1)}$$

After synchronising the two motions, the multiresolution blending can be applied.

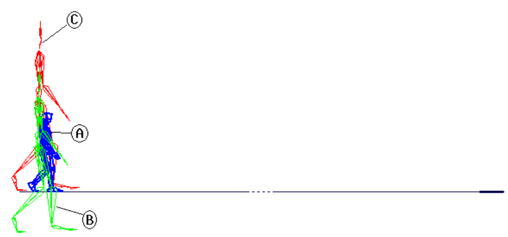
6. Scaled Avatars

Using the captured motion data for animating different avatars rather than the avatar which the data was captured for is an interesting issue. It is known as the motion retargeting problem as a form of the captured motion reusing. Although this is not the main target of this research, it is useful to show how combining the proposed technique with the motion retargeting could be helpful. For simplicity, the scaled avatars only is tested at this stage with the required changes needed to generate animation with specific speed for different scaled avatars are studied.

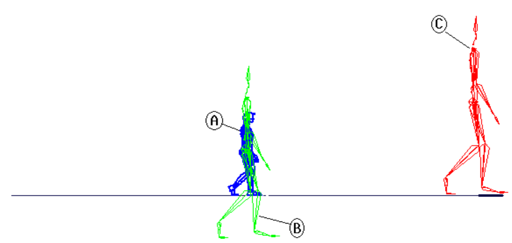
If the pre-stored data is applied to another scaled avatar, the resulting speed could be calculated from the original speed using simple geometry calculations. The formula looks like:

Speed(Scaled) = Scale * Speed(Original)

However, to make the animation of the scaled avatar realistic using the captured data, two major parameters should be updated.



At the Start (all together)



At the End (different position due to the speed)

The Original avatar ('A'), the wrong scaled avatar ('B'), and the correct scaled avatar('C').

Figure 4: Screenshot for Scaled Avatars

As the main translation of the avatar's skeleton comes from its root translation, the root translation in the forward walking direction should be updated according to the following formula:

Z(Scaled) = Scale*Slope * (Frame_No – 1) + Z_start(original)

Where:

Scale=Ratio between the two avatars

Slope=(Z_end-Z_start)/(Last_Frame_No-Start_Frame_No)

The other parameter is the vertical position of the root which is affected by the height of the avatar. The vertical position of the avatar should be updated by the formula:

$$Y(Scaled) = Shift + Y(Original)$$

Where:

Shift=(Scale-1)*Y_start(Original)

Y_start(Original)= Starting position of the root (at frame 1)

Figure 4 shows two screenshots for original and two scaled avatars (200%). The original animation data is applied to both of the scaled avatars but the mentioned updates are incorporated in only one of them (‘C’).

The suggested procedure for generating motion with certain speed for scaled avatar can be summarised as follows:

1. Given the scale of the scaled avatar relative to the original avatar, the proposed parametric motion blending is used to generate a motion for the original avatar but with a speed equal to the desired speed divided by the given scale.
2. The described parameters are updated with the mentioned formulas.
3. Finally, the generated motion data for the original avatar (with the updated parameters) is applied to the scaled avatar. This results in the animation of the desired scaled avatar with the defined speed.

The next step to improve is the retargeting process for avatars with different segment lengths.

7. Conclusion

It has been shown that the multiresolution blending based on wavelet analysis could be applied with the time-warping to generate new motions from pre-stored animation data. The key of the proposed research is that the desired motion could be defined by its natural parameter (such as speed) instead of the blending factor and then a realistic animation is generated from pre-stored animation data. This releases the animator from looping through trial and error sessions of finding the suitable blending factor for the desired motion. Generating the desired motion for different scaled avatars is also discussed as an example of using the proposed analysis.

As this research is still in its early stages, only a brief conclusion about the current status has been given. Currently, the human walking motion is used as an example of a complex human motion with the walking speed as the main parameter. Further analysis is in progress to generalise these results and apply it for other motions.

References

1. H. Ko and N.I. Badler, “Straight Line Walking Animation Based on Kinematic Generalisation that Preserves the Original Characteristics”, In Proceedings of Graphics Interface, pp 9-16, 1993.
2. D.J. Wiley and J.K. Hahn, “Interpolation synthesis of articulated figure motion”, IEEE Computer Graphics and Application, 17(6), november 1997.
3. A. Witkin and M. Kass, “Spacetime constraints”, In Proceedings of ACM SIGGRAPH, pp. 159-168, August 1988.
4. M. Cohen, “ Interactive Spacetime Control for Animation”, In Proceedings of ACM SIGGRAPH, pp 293-302, July 1992.
5. M. Gleicher, “Motion editing with spacetime constraints”, In Proceedings 1997 symposium on Interactive 3D Graphics, pp 139-148, April 1997.
6. M. Unuma, K. Anjyo, and R. Takeuchi, ”Fourier principles for emotion-based Human Figure Animation”, In Proceedings of ACM SIGGRAPH, pp 91-96, August 1995.
7. Andrew Witkin and Zoroan Popovic, “Motion Warping”, In Proceedings of ACM SIGGRAPH, pp 105-108, August 1995.
8. Armin Bruderlin and Lance Williams, “Motion Signal Processing”, In Proceedings of ACM SIGGRAPH, pp 97-104, August 1995.
9. W. Sun, “Modelling Bipedal Locomotion using Wavelets for Figure Animation”, PhD thesis, Department of Computer and Information Sciences, De Montfort University, UK, May 2000.
10. Alexander D. Pou.and Boca Raton, Fla., “The Transforms and applications handbook”, CRC Press in cooperation with IEEE Press, 1996.
11. A. Finkelstein and D. H. Salesin, “Multiresolution Curves”, In proceedings of ACM SIGGRAPH, pp 261-268, 1994.
12. E.J. Stollnitz, T. De Rose and D.H. Salesin, “Wavelets for Computer Graphics: Theory and Application”, Morgan Kaufmann Publishers Inc. 1996.