

Making it Hurt! Pain Constraints on Motion Capture Animations

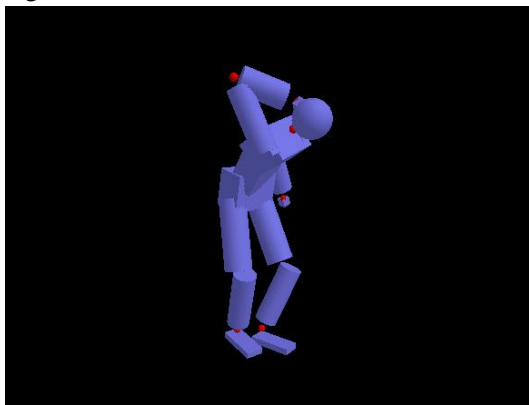
Bilson Jake Libres Campana
Department of Computer Science and Engineering
University of California Riverside
900 University Ave
Riverside, CA, USA, 92521
bcampana@cs.ucr.edu

ABSTRACT

There exists current work that focuses on modifying motion capture animations for transitioning from impact responses. These point specific impact responses are usually physics based in order to add realism to the collision and involve modifying the model's velocities according to the impact parameters. The implication of damage is then inferred by large perturbations such as back stepping or falling. This paper presents an algorithm for inducing subtle motions that reflect a humans reactions to impacts.

1. INTRODUCTION

Upon receiving a collision, a real human should attempt to protect the area. For example one may brace their forearm with their hand, bend toward a torso impact to prevent extension (and pain) of the area, or have restricted movements in joints. Subtle motions such as these may not demand full participation of the body at all times, so the model may continue with its base action while keeping some aspects of the reaction animation(i.e. adding a limp to a walk or holding the forearm while continuing walking).



Also due to this subtleness, realism may be added to animations without large changes to the actor's

Figure 1 - Character is in a walking animation but constraints move the hand and apply the hunch

sequenced path or actions. To accomplish this I apply a set of constraints to a motion capture sequence of a character being stuck and recovering. In section 2, I discuss related work. Section 3 covers the psychology of an impact and the different reactions that may result from different personalities. Section 4 describes the representation of the data. Section 5 contains detail of my plans to modify the motion capture animations. The following four sections discusses the algorithms behind the tools used to create the pain animations. Section 6 cover motion blending, section 7 hand constraining, section 8 motion looping, and 9 decaying constraints. Section 10 contains a discussion and future work. The paper then concludes in section 11 with acknowledgements in section 12.

2. RELATED WORK

In [1] and [2], motion capture sequences are used to react in a dynamic situation where impacts can come from any direct and can vary in force. To accomplish this the authors transition into a dynamic model and then transition into a closely matched reaction motion capture sequence. I build on this work by modifying the synthesized sequence by having the character react to the pain of the impact.

In [2], the character's response to impacts is mostly in the matched reaction animation. Because of this the quality of the reaction is dependent on the richness of the motion capture database. If an impact to the face causes the character to fall back, the algorithm may transition into a "flat on the back" motion or even a "back rolling motion". These motions exhibit the velocity of the impact, but would not have any signs of the impact point unless the actor specifically reacted that location.

In [3], the authors modify motion capture animations by propagating the force of the impact across the body. This technique does react to the location of the

impact but does so by adding large velocities on the body in the direction along the impact. This causes the character to act limp and does not look realistic. My work aims to improve on this idea by allowing the dynamic simulation and motion capture to illustrate the large effects of the impact and then add to this result by applying IK and movement constraints to the character.

3. IMPACTS

3.1 Psychology

When a human receives an impact, our reflex response is to protect ourselves. This could be in the form of a hand to cover the area, a jump away from the impact direction, or a bend around the impact spot to reduce its force. If the pain induced by the impact was large, it could cause restrictions in movement. A hit to the legs may slow locomotion speed and change the character's gait. This can also be caused by psychological effects which do not require an impact to the lower body. The pain caused by a large impact to the arm could distract a human from their original task of walking and therefore slow their speed.

3.2 Personalities

The personality of the individual also affects their reaction. A young kid who stubs their toe on a table may sit down and cry. In the same situation an older man may hobble away cursing. Again, one may excitedly hop around on the other foot. In modeling the character to my own personality, I have it stay stationary to deal with the pain. For the stubbed toe scenario, I would inhale through my teeth while standing stationary on the other foot until most of the pain subsided.

4. REPRESENTATION

4.1 Character Model

The human character model is composed of n body parts. Figure 2 illustrates our model of 18 body parts. Parts of the model correspond to (in numbered order): head, right clavicle, left clavicle, right upper arm, left upper arm, right forearm, left forearm, right hand, left hand, chest, stomach, pelvis, right thigh, left thigh, right shank, left shank, right foot, and left foot. Between each body part there also exists a ball joint, graphically represented by red spheres, which exhibits three degrees of freedom and no constraints in motion. These joints are (from the top down): neck, collars, shoulders, spine, elbows, waist, hips, wrists,

knees, and ankles. These parts and joints are defined in ODE by their dimensions, world relative position, and mass. A root body part is also defined, in my application it is the pelvis. After the positions of the model are defined, the coordinates of all body parts are modified to be relative to the root location.

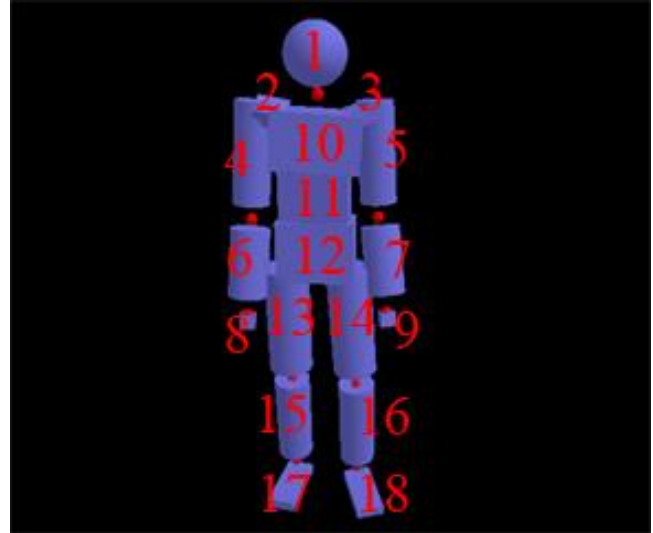


Figure 2 Model Representation - Body parts labeled and described in text

4.2 Coordinate Space Representation

Motion capture sequences are stored as a list of vectors for position and quaternions for orientation. For every time instance, a vector / quaternion pair is described for each body part. The vectors and quaternions are kept relative to world space.

5. ANIMATING PAIN

In the control animation, generated by [2], the character is in a walking animation and is then struck in the back of the head. This induces a physics based fall and then transitions into a motion capture recovery step. The character can then transition back into the original walking animation. To add realism to this animation, I will

- 1) Bring the characters right hand to the impact point.
- 2) Have the character stay in one spot until some pain subsides to reflect the implied personality.
- 3) Slowly start walking while holding its head.
- 4) Have the character release the head and fully transition back into the walk animation.

6. MOTION BLENDING

Blending is required to transition between two motion capture clips. Though preferably we blend motions that are similar, there will likely be differences in body part positions, orientation, velocity, and acceleration. A blend between two motions has the parameters: initial blend a of motion A , initial blend frame b of motion B , blend length t_b , and an interpolation function.

6.1 Alignment

Before we can blend we must preprocess the motion data by aligning their frames. In the following text we will assume that we are blending from a motion A to another motion B . Because motion capture data is kept in a global coordinate frame, two identical animations recorded in different areas in the world space will highly differ in their position vectors. Blending of these two raw motions will result in the same motion, but translated to their median distance. To correct for this we must translate motion B to align in the same space as motion A .

First, calculate the offset vector by taking the root position in frame a and subtracting the root position from frame b .

$$positionOffset = a_{root} - b_{root}$$

Then, for every body part in motion B , in frames b to frame $b+t-1$, subtract $positionOffset$ from its position vector.

$$BPartPosition_i = BPartPosition_i - positionOffset$$

These operations effectively translate the relevant frames of B to the origin and then to the position of A , aligning the frames relative to their roots in b and a .

6.2 Interpolation Curves

For every blend iteration $\{f \in \mathbb{Z} \mid 0 \leq f \leq t_b - 1\}$, a ratio r is calculated which controls the interpolation strength of each motion and is used to control the interpolation strengths in a blending function

$$posture_{interpolated} = posture_B \times r + posture_A \times (1-r).$$

At $f = 0$, $r = 0$ and the interpolated position is $posture_a$. At $f = 1$, $r = 1$ and we are at $posture_b$.

The choice of interpolation function controls the velocity and acceleration we have while transitioning

from $posture_a$ to $posture_b$. Figure 3 illustrates the characteristics

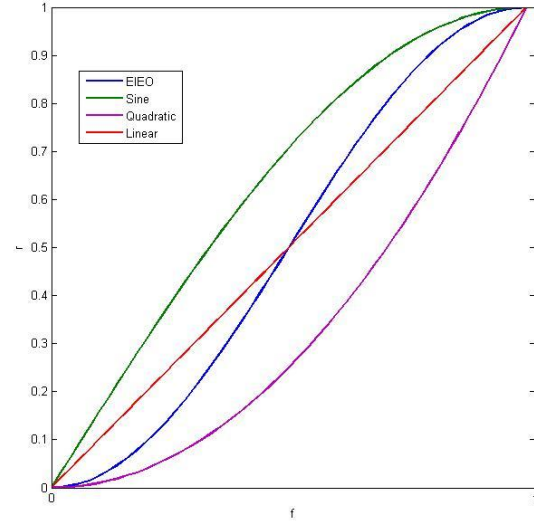


Figure 3 Interpolation Curves - Interpolations curves EIEO, sine, quadratic, and linear with frame iteration f vs. ratio r

of four possible interpolation functions. A *linear* function in the form

$$r_f = \frac{f}{t_b - 1}$$

gives an unbiased transition velocity from motion A to motion B . A *sine* function in the form

$$r_f = \sin\left(\frac{\pi}{2 \times t_b - 2} \times f\right)$$

provides a faster initial transition velocity to motion B that decreases as $f \rightarrow t - 1$. A *parabolic* function in the form

$$r_f = \frac{f^2}{(t_b - 1)^2}$$

produces an slow starting transition velocity from A which increases. An *ease in ease out (EIEO)* function in the form

$$r_f = \frac{\left(\sin\left(\left(\frac{\pi}{2 \times t_b - 2} \times 2f\right) \times -\frac{\pi}{2}\right) + 1\right)}{2}$$

yields a combination of the quadratic and sine functions with slow initial and final transitions out of A and into B and a fast transition in between.

6.3 Partial Blending

In the previous blending method, there was a single r_f used for each body part at blend interval f . Another blending method used in this work, for decaying constraints (discussed in section 9), involves assigning individual blend ratios r_{fj} by calculating a second blend ratio p_{fj} for each body part j in a posture at blend interval f and then multiplying it to r_f .

$$r_{fj} = p_{fj} \times r_f$$

The ratios differ to allow a non uniform blend across the body between motions. With a non uniform blend we can add select parts of motion B and parts of motion A to blend. For example, consider a walking sequence for motion A and a waving sequence for motion B . By setting $r_{f\text{arm}} = 1$ and $r_{f\text{others}} = 0$, a walking animation with a waving arm is created. r_{fj} can vary with time (as described in section 9) and can be any value $0 \leq r_{fj} \leq 1$.

7. HAND CONSTRAINTS

7.1 Inverse Kinematics

To help make the character seem more aware of the point specific impact, I have him move his right hand to the impact point. To accomplish this I utilize the IKAN inverse kinematics algorithms described in [4] and implemented by [5]. This implementation applies an offset vector towards a goal position to a body part. For my application I apply an offset vector v_{offset} to the right and in order to bring it to the back of the head. Similar to aligning frame posture roots, the hand position vector v_{hand} is translated to the root and then translated to the head position v_{head} . This gives the offset vector

$$v_{\text{offset}} = [v_{\text{offset}x}, v_{\text{offset}y}, v_{\text{offset}z}] = v_{\text{head}} - v_{\text{hand}}$$

This offset places the hand at the center of the head, but we wish to have it on the rear surface. We cannot simply subtract the head radius r_{head} on the z axis because the head may be rotated and/or tilted. To

match the heads orientation we rotate the head surface vector

$$v_{\text{headsurface}} = [0, 0, -r_{\text{head}}]$$

by the angle specified by the head orientation vector. We can then use this rotated vector to move the hand one final step

$$v_{\text{offset}} = v_{\text{offset}} + \text{rotated}V_{\text{headsurface}}$$

7.2 Animating Hand Constraint

To animate the hand grasping the back of the head v_{offset} is interpolated, from $[0,0,0]$, in the same manner use in blending. The speed of the hand placement can be controlled by selecting an appropriate interpolation curve and hand transition time t_{hand} .

7.3 Pathfinding

Because the model exhibits no constraints on joint angles, the character may self collide or make an inhuman motion (i.e. shoulder rotations that are not physically possible). In the case of our arm movement, for each frame f_{hand} of the hand transition a second offset modifies v_{offset} .

$$v_{\text{offset}x} = v_{\text{offset}x} + \lambda_x \times \sin\left(\pi \times \frac{f_{\text{hand}}}{t_{\text{hand}}}\right)$$

$$v_{\text{offset}y} = v_{\text{offset}y} - \lambda_y \times \sin\left(\pi \times \frac{f_{\text{hand}}}{t_{\text{hand}}}\right)$$

$$v_{\text{offset}z} = v_{\text{offset}z} + \lambda_z \times \sin\left(\pi \times \frac{f_{\text{hand}}}{t_{\text{hand}}}\right)$$

The signs of λ in the three equations correspond to motions away from the body for the right hand. Motions forward, right, and up respectively. The magnitude of λ signifies the strength of the perturbations.

8. MOTION LOOPING

An impact inducing a large amount of pain could require that the character remain stationary for an indefinite amount of time $t_{\text{stationary}}$. Freezing the character would make it seem robotic, as if it were turned off. Slowing the character down would keep it in motion, but large values of $t_{\text{stationary}}$ would make

the it move inhumanly slow. Continuously looping the animation would keep it in motion, but again large values of $t_{stationary}$ would make the animation inhuman as the repetition becomes more apparent. I chose a process of continuous loops with added noise to reduce repetitious motion. The noise is generated by traversing a motion graph of the looped animation. This method is applied to the recovery animation synthesized by [2].

8.1 Selective Looping

To produce a smooth continuous loop, non stationary portions of the recovery animation are trimmed (i.e. steps taken in the a recovery motion). After trimming, the first posture in the recovery animation is used as a reference to remove postures with a distance from the reference posture greater than δ . The distance between two postures is defined as the sum of Euclidean distances of their corresponding parts (which are root relative). δ can be modified to adjust the loop animation. Large values of δ can create erratic animations while low values can diminish all movement in the character.

8.2 Loop Scripting

The pair wise distances of postures in the resulting loop is calculated and a motion graph is created using transitions under a distance threshold $d_{threshold}$. Large values of $d_{threshold}$ may synthesize erratic loops while small values may contain little motion at all.

9. DECAYING CONSTRAINTS

As time progresses after the impact, the induced pain diminishes. When the amount of pain is large, the character is kept stationary. As the amount of pain drops below a threshold the constraints are relaxed starting at the body parts furthest from the impacted area. In my impact scenario, these is the lower body. The character is allowed to slowly transitions into a walk. To accomplish this partial blending is utilized with the stationary scripted loop to the original walking animation. By setting the r_{f_j} values of the pelvis and body parts below to 1 and others to 0, the lower body of the character interpolates into the walking animation while the upper body stays constrained in the hurt phase. As the pain reduces below a second threshold, the r_{f_j} values of the upper body interpolates to 1. This allows the character to smoothly transition into the entire walking animation.

10. DISCUSSION AND FUTURE WORK

Professor Zordan brought up an interesting idea of replacing the scripted looping by stopping the character and switching to a balance controller. Minor perturbations can then be added for noise and under dampening the controller could make these subtle motions seem further natural. This work involves a large set of parameters, though most of these can be calculated directly from the impact conditions (velocity, force, etc.).

11. CONCLUSION

By applying constraints to motion capture reaction animations, realism is added and the effects of a strike are accounted for by more than large perturbations of the character. The ability to tune these constraints allows for a wide variety of effects including transitions to and from stationary states and restriction of movements to simulate the physical and physiological effects of pain. The processes to apply these constraints also allow for synthesis of a variety of animations through blending. A waving motion capture can be partially blended with a walking motion to create a new waving and walking motion. Using these techniques, the affects of impacts can be felt through novel motion capture sequences.

12. ACKNOWLEDGEMENTS

I would like to thank Chun-Chih James Wu [5] for providing code and giving an introduction to motion capture animation and Professor Victor B. Zordan [5] for guidance on the project.

13. REFERENCES

- [1] Zordan, V., Macchietto, A., Medin, J., Soriano, M., Wu, C., Metoyer, R., and Rose, R. 2007. Anticipation from example. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*
- [2] Zordan, V. B., Majkowska, A., Chiu, B., Fast, M., Dynamic Response for Motion Capture Animation , *Transactions on Graphics/ACM SIGGRAPH* 2005.
- [3] Arikan, O., Forsyth, D. A., and O'Brien, J. F. 2005. Pushing people around. In *Proceedings of the 2005 ACM Siggraph/Eurographics Symposium on Computer Animation*
- [4] Deepak Tolani and Ambarish Goswami and Norman I. Badler. *Real-Time Inverse Kinematics*

Techniques for Anthropomorphic Limbs.
Graphical Models. 2000.

[5] Zordan, V. et al. Riverside Graphics Laboratory.
<http://www.cs.ucr.edu/~vbz/>