

# Interval-Based Motion Blending for Hand Grasping

Matt Brisbin and Bedřich Beneš

Purdue University  
{mbrisbin|bdenes}@purdue.edu

---

## Abstract

*For motion to appear realistic and believable proper motion blending methods must be used in respect to the goal or task at hand. We present a method that extends the theory of move trees [MBC01] by tagging (attaching) information to each clip within a database at intervals and finding the shortest distance per tag while pruning the tree using convergence priority. Our goal is to retain the physical characteristics of motion capture data while using non-destructive blending in a goal-based scenario. With the intrinsically high dimensionality of a human hand our method also is concerned with intelligent pruning of the move tree. By constructing a move tree for hand grasping scenarios that is sampled per interval within clips and adheres to a convergence priority; we plan to develop a method that will autonomously conform a hand to the object being grasped.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Motion Blending

---

## 1. Introduction

Many new advancements in character animation extrapolate information from motion capture data for the purpose of controllability. Often times this extraction and reparameterization fails to account for physically plausible and believable motions which are needed in film, games, virtual environments, and crowds. Since motion capture data is, by nature, physically correct, any imposed constraint can potentially lead to an incorrect or unrealistic result. The use of constraints imposed on a motion capture sequence can be seen in the case of "footstake cleanup" [KGS02] where a characters feet slide across the ground plane and constraints are imposed for correction.

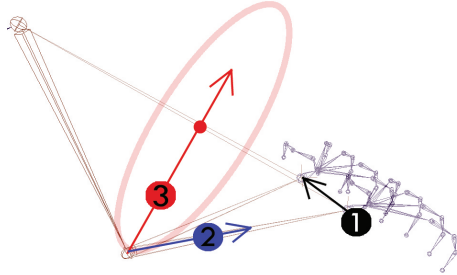
Much attention has been focused on guiding motion capture data by means of inverse kinematics which results in a goal-based animation approach. Although this approach is a viable means of goal-based animation it is very destructive to the integrity of the initial motion capture data, in many cases, rendering the physically correct information unrealistic. Other methods such as artificial intelligence, neural networks, bayesian methods, physics, parameterized models, and neuro-motor models are becoming more prominent in gaming as hardware becomes faster and able to handle the complexity.

This work contributes to motion blending in respect to hand grasping scenarios. In recent years autonomous animation has focused greatly on locomotion [EAPL06], posture [EAPL06], and collision detection [Kal05], [KAAT03] but has given little insight into small articulations such as hand grasping [MKCA03] that add a true sense of believability to an animation. Blend trees [MBC01] or hierarchical structures of motion clips [KGP02] have been the primary method for dynamically creating motion in real-time applications such as games and crowds. This method is the basis for nearly all real-time animation scenarios that require dynamic blending of animation sequences. The main factor that contributes to the speed of blend trees is the high dimensionality of the space in which the character exists. Optimization methods are put in place to confine this large space into a much smaller space for computation but many times techniques such as inverse kinematics are used alleviate complex blending.

Our method extends the theory of blend trees by tagging (attaching) information to each clip within a database at intervals and finding the shortest distance per tag while pruning the tree using convergence priority. A vector-based description of the arm, Fig. 1 is created as a preprocess to allow for temporal pruning of the tree and fast comparison across branches; creating the appropriate walk for the given

space-time configuration. This information as well as a convergence factor will be stored in a tag and associated with the designated frame. Input source for this method are derived from three values;

1. The end effector vector of the wrist(direction of wrist movement).
2. The pole vector of the arm(orientation of the elbow).
3. The elbow to wrist vector(direction of forearm).



**Figure 1:** Key points of the arm stored for comparison. 1.)Direction of moving end effector across frames, 2.)Direction of forearm, 3.)Pole-vector (orientation) of arm.

To apply our method to hand grasping scenarios methods from motion capture and artificial intelligence will be used. Motion capture will provide our system with physically correct actor-based motion that is precise and believable in the raw form. We will then implement move trees [MBC01], or decision trees, to compute and prune unwanted data for the given space-time configuration. Our system will be developed using the Open Reality SDK by Autodesk [Aut] and C++.

This paper is structured as follows. Section 2 describes the previous work and current approaches being taken within the area of motion blending. Section 3 discusses the methodology will use implement the motion blending of hand grasping scenarios. Lastly, Section 4 will conclude with an overview of our method.

## 2. Previous Work

Motion blending has been an area of study ever since digital animation begun. It covers a wide range of disciplines; from games and film to robotics, ergonomics, and simulation. With the advent of motion capture as a viable means for producing animation comes the need to concatenate, process, and handle the raw data.

Witkin and Popovic [WP95] introduced Motion Warping as a means to concatenate clips of motion to form one fluid sequence of animation. Kovar [KGP02], introduced motion graphs as a method to process and intelligently construct a directed graph that allows for dynamically creating new motion by traversing the graph for new walks. Using this

method generates a general purpose approach to clip concatenation since each raw motion segment is sampled per edge of clip rather than using pre-defined methods that could limit the possible connections.

With artificial intelligence becoming more advanced in games decision trees, or move trees, are becoming more popular and complex. Move trees, [MBC01], form a method for deciding what state a system should be in and what possible actions could occur next. A concrete implementation of move trees can be found in the game of chess and are the basis for many of today's advanced games.

Goal-based animation is yet another area of study that has contributed greatly to motion blending in many respects. Inverse Kinematics (IK), is a mathematical solution for a flexible chain that solves a defined pose. Arm reach [Kal05] scenarios benefit greatly from this method since they are, usually, very goal-driven, i.e. the grasping of an object. Kallmann extended traditional IK methods by introducing collision free motion as well as environment annotations that allowed the developed system to solve complex tasks such as object grasping very quickly. Other extensions of IK have been focused on modifying existing motion capture sequences to fit the needs of a user [MM05] as well as a physically plausible pre-defined model. Meredith uses IK to parameterize motion capture data for the elimination of three common errors that occur frequently; foot sliding, limb penetration, stylized motion. His method is also capable of re-targeting motion to other skeleton structures.

Although motion capture techniques contain physically correct information it becomes difficult for a character to react to their scene due to the motion being pre-defined. Emerging technologies such as, NaturalMotion's, Morpheme application [Ltd05] provide a move tree solution that incorporates; artistic input, physics, motion capture, and hierarchical state machines for human motion synthesis in games [Ltd05]. Physics-based techniques have contributed a great deal to the level of realism in dynamic environments [SHP04] as well as physically correct behavior [ZMCF05]. Physics-based methods allow for reactive motions to occur but also introduce unrealistic behavior when motion capture data is manipulated, spliced, or re-configured [ZMCF05]. Responses in a dynamic environment can be difficult, [ZMCF05], therefore, using a specialized search procedures to compare a physical simulation to the existing motion capture driven sequence provides a method for plausible outcomes to occur in an unpredictable environment. Jordan's work determines what initial pose of a motion capture clip would best fit the exiting pose of the physical simulation and would then select it for blending, this allowed for a seamless re-entry into motion capture data with physically correct motion still intact [ZMCF05]. Pollard, developed a method for preserving plausible motion while blending motion clips using active and passive controllers for contact as well as gravitational forces. This method was applied to ex-

amples of hand grasps captured by means of motion capture [PZ05]. The controllers developed for this simulation extract torque data from the motion capture data to form the physics controllers as well as inverse dynamic models. While this method is flexible it does require pre-processing time, therefore, it is an off-line process. Extending existing motion capture data using physics controllers provides a great deal of realism for hand grasping models. Limitations of this research are based on the breadth of motion capture examples within the database as well as the extent to which the controllers can extract information from each motion clip [PZ05]. Other research has focused on segmenting clips without any prior knowledge and building a plausible outcome using variable length Markov models [GJH01]. This method was used to synthesize complex structures such as sign-language, dance, and aerobics for long-term predictions of movement.

### 3. Methodology

This section outlines the process that will be used to develop our method for motion blending that is directed towards autonomous hand grasping scenarios. First we collect the data via motion capture. Secondly, we construct a blend tree of motion clips as well as the necessary interval defined tags. Next, our system compares the arm motion of the current animation with the stored arm motion and applies the appropriate handshape with respect to a convergence priority. After the smallest distance between arm motions has been found, the solution, the corresponding handshape is then applied and displayed.

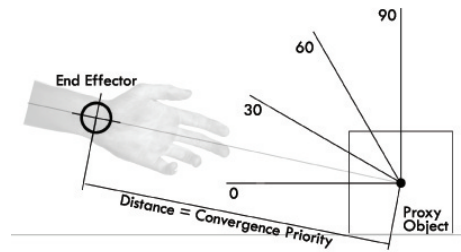
We begin with the data collection process by means of motion capture. Next we formulated the necessary data structures required for off-line and real-time processes. We then discuss the implementation of our method by discerning what is off-line and what is computed in real-time.

#### 3.1. Data Acquisition

A gypsy motion capture suit [gyp] along with a cyber-Glove [cyb] will be used to gather initial data which will form our primary database of motion samples. Hand-shapes as well as arm motion will be captured that reflect the grasping of a primitive-shaped object at defined angles  $\{0^\circ, 30^\circ, 60^\circ, 90^\circ\}$  with respect to the Z-axis of the object. See Fig. 2 This process will be repeated per angle for variability as well as supply the motion clip database with multiple perceptions of object grasping per individual. The database will consist of 10 grasping attempts per angle per individual, providing 40 motion clips per individual.

#### 3.2. Data Structures

Our system is composed of three main data structures which are linked per interval of frames which is set by the user. A



**Figure 2:** Angles used for Motion Capture acquisition. Distance to the object will result in priority of sampled clips within the database

shorter interval (per frame) will result in more computational time but with greater accuracy and a larger interval results in a fast computation but with less accuracy. First, the shoulder, elbow, and wrist are used to store the following calculations.

- The end effector vector of the wrist(direction of wrist movement).
- The pole vector of the arm(orientation of the elbow).
- The elbow to wrist vector(direction of forearm).

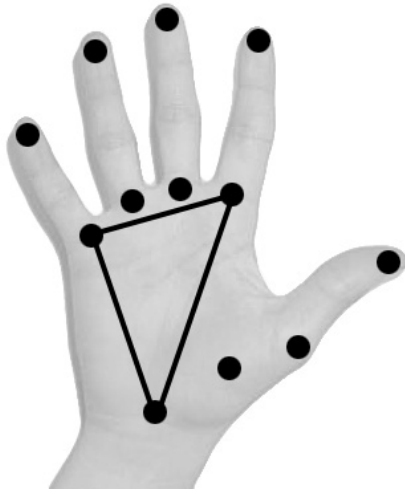
These three components, Fig. 1 make up the arm definition which will be used for real-time comparison later on. Secondly, a convergence factor is stored per interval which serves as a determining factor as to what the state-of-the-hand is over a duration of time. See Fig. 4. Lastly, our system parameterizes a set of handshapes, Fig. 3 by extracting the main point of the hand for efficient comparison later. End points of the fingers are chosen to calculate minimum and maximum flex to arrive at a collision free solution. Three points that define the palm allow orientation will be calculated in world space, and combined they allow for calculation of a convergence factor which will be discussed later. Fig. 4.

#### 3.3. Off-Line process

From the arm data structure we will calculate three components that will be used later for real-time comparison. They are as follows;

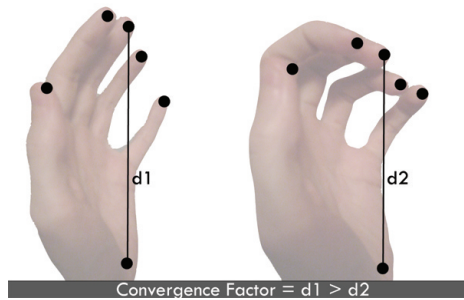
- The end effector vector of the wrist(direction of wrist movement). This vector will be calculated by storing the position of the wrist at the next sequential frame and subtracting it from the position of the wrist at the current frame. See Figure. 1
- The pole vector of the arm(orientation of the elbow). Will be found by finding the orientation of the elbow with respect to the shoulder and wrist joint.
- The elbow to wrist vector(direction of forearm). Will be calculated by subtracting the position of the wrist from the elbow. See Figure. 1.

These components will be calculated at a determined interval per clip (i.e. every 10 frames)and stored along with



**Figure 3:** Key points of the hand used for parametrization

the parameterized 12 degree-of-freedom handshape it represents, Fig. 3. Next, we assume that the closer the end joints of the fingers are to the palm the more closed the hand is. From the stored parameterized model of the hand we calculated the average distance that all fingers move towards the palm at the specified interval, Fig. 4. A short distance defines a closed handshape and a large distance defines the open or flat handshape. This distance is added to the existing data structure of handshapes for use in determining the state-of-the-handshape over time. By creating various walks through this structure we can extrapolate new motions that were not present in the original data gathered by the motion capture system. The final sequence of hand-poses gathered from this structure lends itself to the final motion being comprised of many perceptions that an individual has when grasping an object.

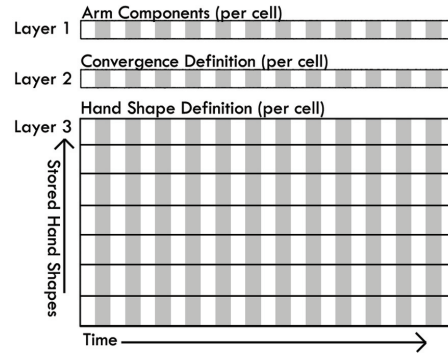


**Figure 4:** Determination of Convergence Factor. We assume that a smaller average distance between all finger tips and the palm represents a closed handshape. This average is used to determine the degree to which the hand is closed/open throughout the duration of a motion clip.

### 3.4. Real-Time process

To apply the correct handshape during runtime our method completes the following actions:

1. Computes the virtual characters arm components. Fig. 1.
2. Computes distance to the object being grasped
3. Searches the top level of the database for similar values in stored arm components. Fig. 5 Chooses stored data that is within a defined threshold.
4. Prunes the selection by eliminating all handshapes that contain convergence priorities not equal to the current distance.
5. Applies the handshape that has the corresponding smallest distance between stored arm components.

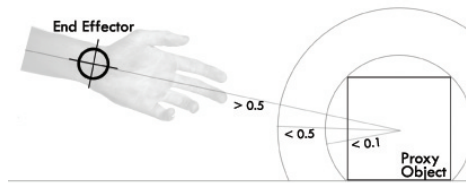


**Figure 5:** Pruning of Data

First the scene is searched for the arm components, Fig. 1 and they are compared to layer 1 of our data structure. Fig. 5. Once the shortest distance between the characters arm components and the stored arm components is found the tree is pruned and the distance from the end-effector to the object is calculated. If the distance is greater than  $1.5 \cdot$  the size of the bounding box of the object all handshapes that contain convergence factors greater than 0.5 can be used for blending. Otherwise, if the hand is within this range all handshapes which have a convergence factor less than 0.5 but greater than 0.2 are used. When the distance is inside the original bounding box of the object any handshape that has a convergence factor less than 0.1 can be used. See Fig. 6. Once these conditions are met the handshape is chosen and mapped to the character.

### 3.5. Comparison and Fitting

Our system creates a walk across the tree by comparisons that are made between the virtual actor's motion and the stored information gained via motion capture. To eliminate the high-dimensionality of this search we have imposed a hierarchy of comparisons which are made in real-time. First the virtual actors arm components are compared to layer 1 in the database for shortest distance between values. Fig. 5.



**Figure 6:** Selection of HandShape based on level of convergence

Values that lie outside the given threshold are discarded as well as their children branches. Secondly, we calculated the distance to the object and compare it to the corresponding convergence factor stored within Layer 2 of the database. Fig. 5. This allows us to only sample handshapes that correspond to the given distance to the object, simulating the actual perceptions of grasping that was recorded via motion capture. At this point we can associate a handshape to the virtual actors hand. If our system finds many handshapes that could be a potential fit it then searches for the closest distance between arm components and applies it.

#### 4. Conclusions

Retaining physically correct motion through the use of non-destructive blending methods is our main goal of this research. Many mathematical solutions to goal-based animation exist such as inverse kinematics that do not result in a believable or realistic outcome. Other physics-based methods aim to retarget existing data to a model that is known to be physically correct. By constructing a hierarchical move tree for hand grasping scenarios that is sampled per interval within clips and adheres to a convergence priority we plan to develop a method that will autonomously conform a hand to the object being grasped.

#### References

- [Aut] Autodesk. <http://usa.autodesk.com>.
- [cyb] Immersion. <http://www.immersion.com>.
- [EAPL06] ESTEVES C., ARECHAVALTA G., PETTRÉ J., LAUMOND J.: Animation planning for virtual characters cooperation. *ACM Transactions on Graphics (TOG)* 25, 2 (2006), 319–339.
- [GJH01] GALATA A., JOHNSON N., HOGG D.: Learning Variable-Length Markov Models of Behavior. *Computer Vision and Image Understanding* 81, 3 (2001), 398–413.
- [gyp] Animazoo. <http://www.animazoo.com>.
- [KAAT03] KALLMANN M., AUBEL A., ABACI T.,

THALMANN D.: Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping. *Computer Graphics Forum* 22, 3 (2003), 313–322.

- [Kal05] KALLMANN M.: Scalable Solutions for Interactive Virtual Humans that can Manipulate Objects. *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, Marina del Rey, CA (2005).
- [KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 473–482.
- [KGS02] KOVAR L., GLEICHER M., SCHREINER J.: Footstake cleanup for motion capture. *Proceedings of ACM SIGGRAPH Symposium on Computer Animation (SCAS02)* (2002).
- [Ltd05] LTD. N.: *Dynamic Motion Synthesis*. Tech. rep., Oxford, UK, March 2005.
- [MBC01] MIZUGUCHI M., BUCHANAN J., CALVERT T.: Data driven motion transitions for interactive games. *Eurographics 2001 Short Presentations* 2, 3 (2001), 6.
- [MKCA03] MILLER A., KNOOP S., CHRISTENSEN H., ALLEN P.: Automatic grasp planning using shape primitives. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* 2 (2003).
- [MM05] MEREDITH M., MADDOCK S.: Adapting motion capture data using weighted real-time inverse kinematics. *Computers in Entertainment (CIE)* 3, 1 (2005), 5–5.
- [PZ05] POLLARD N., ZORDAN V.: Physically based grasping control from example. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), 311–318.
- [SHP04] SAFONOVA A., HODGINS J., POLLARD N.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 514–521.
- [WP95] WITKIN A., POPOVIC Z.: Motion warping. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), 105–108.
- [ZMCF05] ZORDAN V., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *Proceedings of ACM SIGGRAPH 2005* 24, 3 (2005), 697–701.