Data-driven Finger Motion Synthesis for Gesturing Characters

Sophie Jörg
Carnegie Mellon University
Clemson University

Jessica Hodgins
Carnegie Mellon University
Disney Research, Pittsburgh

Alla Safonova
Disney Research, Pittsburgh

Abstract

Capturing the body movements of actors to create animations for movies, games, and VR applications has become standard practice, but finger motions are usually added manually as a tedious post-processing step. In this paper, we present a surprisingly simple method to automate this step for gesturing and conversing characters. In a controlled environment, we carefully captured and post-processed finger and body motions from multiple actors. To augment the body motions of virtual characters with plausible and detailed finger movements, our method selects finger motion segments from the resulting database taking into account the similarity of the arm motions and the smoothness of consecutive finger motions. We investigate which parts of the arm motion best discriminate gestures with leave-one-out cross-validation and use the result as a metric to select appropriate finger motions. Our approach provides good results for a number of examples with different gesture types and is validated in a perceptual experiment.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: finger motions, human animation, gesture synthesis, hand motions, virtual characters

Links: DL PDF

1 Introduction

Hand and finger motions are omnipresent in our daily life. We use our hands to interact with objects and with each other. When communicating, we use them to punctuate our speech, point to an object of interest, signal our opinion, and convey emotions. Communication – and therefore hand and finger motions – also plays an increasingly important role in digital applications. Examples of such applications include virtual worlds, such as Second Life, digital games, such as L.A. Noire, or any type of teaching, training, or advice application that uses embodied conversational agents. If we want to create believable and compelling virtual characters, we therefore need to generate convincing hand motions.

Motion capture has become a widely used technique for animating realistic and human-like characters for movies or games. However, the elaborate motions of the fingers with their high number of degrees of freedom and small size are still not easy to capture. As a consequence, hands are rarely captured in their full complexity. In general, two or three markers on fingertips (e.g., on the thumb, index, and pinky) and a few on the back of the hand are used as a reference [Kitagawa and Windsor 2008] and the fingers are keyframed manually, which is a tedious, labor-intensive process.

In this paper, we propose a method to automatically add plausible finger motions to body motions (Figure 1). Our algorithm uses a previously recorded database of body and finger motions to generate finger movements that match an input body motion (Figure 2). We locate suitable finger motions in the database using a metric based on the similarity of the arm movements and the smoothness of the reconstructed finger motions. Our key observation is that plausible finger motions can be inferred from the wrist motion. Our approach preserves the naturalness and subtlety of motion captured movements, without requiring time-consuming manual post-processing. As our method is intended as a post-processing step, it does not impact the motion capture session, saving valuable production time. We apply our algorithm to gesturing and conversing characters. It is not intended for object manipulations, such as grasping, where the fingers must be precisely positioned. We demonstrate our approach with several example databases, consisting of gestures, casual conversations, debates, and giving directions.

Our main contribution is the development of a data-driven method to automatically add realistic and detailed finger motions to the body motions of conversational characters. We explore the validity of several parameters as predictors for the similarity of finger
Figure 2: Our method automatically synthesizes finger movements for an input body motion based on a database of motions.

motions and find that a combination of wrist rotation and wrist position leads to the best results. The rest of the paper is organized as follows. After discussing related work in Section 2, we explain our method in Section 3. In Section 4, we describe how we tested multiple finger motion predictors with a leave-one-out cross-validation applied to a set of controlled gestures. Our results, including examples with different gesture types and a perceptual study, are presented in Section 5. We conclude by discussing the advantages and limitations of our approach and considering future work.

2 Related Work

Several approaches have been suggested to simplify the animation of finger motions for virtual characters. They can be divided into methods to capture finger movements and algorithms to compute them. The latter consist primarily of data- and physics-based approaches.

Capturing finger motions is a challenge. The high number of joints in a very small space requires many small markers. Contrary to the face, where the markers are roughly located on a vertical, two-dimensional plane, markers attached to fingers can point in any direction including the floor. The capture space therefore needs to be very well covered. Frequent gaps in the data must be repaired with manual post-processing. Glove-based solutions exist [Wang and Popović 2009; CyberGlove Systems 2012; Measurand 2012]. However, these have drawbacks, such as lower accuracy or time-consuming calibration, and drift [Kahles et al. 2004]. Other techniques, such as combining optical motion capture with depth information from the Kinect [Zhao et al. 2012] or using computer vision algorithms [Athitsos and Sclaroff 2003], are restricted to small, controlled areas and the capture of specific tasks [Wang et al. 2011].

Majkowska et al. [2006] propose a method to motion capture body motions and hand motions separately and automatically assemble them. This approach allows for a smaller capture space when only hand motions are recorded. Nevertheless, it requires the actor to repeat the motions accurately, a difficult task for longer scenes without a specified choreography. Ye and Liu [2012], also observed that detailed finger motion can be created based on the wrist movements and the motion of a handled object. Their approach, which was developed simultaneously to ours, focuses on the challenge of manipulation strategies and therefore complements our approach.

Most research on creating finger motions has focused on animating specific tasks, such as playing a musical instrument [ElKoura and Singh 2003]. Manipulation tasks have been successfully simulated using physics-based approaches [Liu 2008; Liu 2009; Pollard and Zordan 2005], often using sensors to measure parameters or forces [Kry and Pai 2006]. Neff and Seidel [2006] employ a similar method to animate relaxed hand shapes, deriving the parameters needed for the physics-based simulation from video recordings. Another solution for animating hands and finger motions are detailed, anatomically correct hand models [Tsang et al. 2005]. These models are usually time-consuming to animate.

Finger motions for conversational gestures represent an important application area because gestures are considered an integral part of the act of speaking or producing an utterance [Kendon 2004]. Even though gestures do not have a strict grammar, they do follow rules that allow us to understand and interpret each other’s gestures [McNeill 1992]. Cassell et al. [2001] use a rule-based approach derived from linguistic and gesture studies research to synthesize gestures based on written text. Stone et al. [2004] animate the gestures of a conversational character with a database of recorded speech and captured motions, fitting the gestures to its generated voice. Levine et al. [2010] train a probabilistic model to create gestures based on the prosody of the voice and a database of gestures. Neff et al. [2008] focus on creating different styles of speaker motions with video or text as input. In contrast, our method focuses on the detailed motion of the fingers. We do not take into account the meaning of the conversation or the rhythm in the audio track as we assume that those are adequately conveyed by the wrist motion.

The perception of finger animation has been studied by Jörg et al. [2010], who showed that even small errors in the synchronization of body and finger motions can be perceived and can change the interpretation of an animation. Samadani et al. [2011] show that an emotion (anger) can even be recognized on moving hand-like structures. Hoyet et al. [2012] investigate the perception of finger animations generated with different sets of markers and find that finger motions reconstructed from a reduced set of eight markers per hand are perceived to be similar to motions captured with a set of twenty markers for many types of motions but are perceived to be different for some motions.

Our method searches for short segments of finger motions in a database and finds an optimal path through the resulting motion graph. It therefore builds on the motion graph literature [Rose et al. 1998; Kovar et al. 2002; Lee et al. 2002].

3 Method

The goal of our method is to automatically synthesize finger motions for virtual conversational characters. With the term finger motions, we denote the movements of the phalangeal joints of all five digits of both hands and the carpometacarpal joint of the thumb, while body motions designate the motions of the body joints of the virtual character including the wrist positions and orientations but excluding the finger motions. In its strict anatomical definition, the term finger does not include the thumb [Palastanga et al. 2006] but for simplicity we use the term to refer to all five digits.

The inputs to our system are the body motion of a virtual character and a database of gesturing or conversing motions where both the finger and body motions are present. As output, our approach generates the movements of the finger joints of both hands (Figure 2).

The key observation behind our approach is that arm and finger motions are highly correlated. When we see an animation of a character without hand motion, we can usually infer what the hand motions must have been. A skilled animator would be able to produce a realistic rendition of the finger motion. To produce realistic finger motions, our method therefore searches the database for body motions that are similar to the body movements of the input motion. We then adapt the associated finger motions to fit the input motion.
We found that a simple approach, similar to the method described by Levine et al. [2009], works well at separating motions into meaningful phases. We split the motion when the speed of the wrist joint crosses a threshold close to zero, thus separating motion phases (high speed) and hold phases (low speed). We restrict segment length to be no less than 0.33 seconds and no more than 2 seconds. If segments are too long, an additional split is added at a suitable local minimum of the speed. These restrictions reduce spurious segmentation due to noise and avoids overly long segments.

Out of our 64 single gestures from the gesture database, 49 are segmented into three phases (see Figure 4), where the first phase lasts from the rest position to the end of the stroke (quick wrist motion), the second phase is the post-stroke hold (slow) and the third one is the retraction (quick). Out of the remaining gestures, ten have short post-stroke holds (our minimum segment length is 0.33s), so the gestures are segmented into only two parts and five gestures are segmented into four parts, e.g. with a long post-stroke hold split in two segments.

### 3.2 Finding Similar Segments

For each segment of the input motion, we search the database for the k most similar segments. The candidate fragment in the database, and \( S_D \) (the segment from the database), we compute the segment cost \( c_S \). First, we adapt \( S_D \) using dynamic time warping, so that it has the same length as \( S_I \). Next, we compute the weighted squared distance of the position and rotation coordinates of the wrist joint in the two segments according to the following formula:

\[
c_S = \frac{1}{n} \left( w_P \sum_{t=1}^{n} (P_I(t) - P_D(t))^2 + w_R \sum_{t=1}^{n} (R_I(t) - R_D(t))^2 \right)
\]

(1)

where \( w_P=1 \) and \( w_R=0.5 \) are weighting terms (see Section 4), \( t = 1, \ldots, n \) is the frame number, \( P_I(t) \) and \( P_D(t) \) are the wrist joint position coordinates of the input motion and the fragment from the database, and \( R_I(t) \) and \( R_D(t) \) are the wrist joint rotation coordinates of the input motion and the fragment from the database, respectively. We use the position and rotation coordinates relative to the root (hip) of the character. We use Euler angles for our computation but the method can be used with any angle representation.
3.3 Transition Cost

Not only do we require a good match to the arm motion segments but we also need the transitions between different finger motion segments to be smooth. We compute the transition cost \( c_T \) for each set of possible consecutive fragments by comparing the orientations and angular velocities of the fingers at the last frame \( A \) of one fragment and the first frame \( B \) of the next fragment. We again use squared differences:

\[
c_T = \frac{1}{m} \left( w_J \sum_{i=1}^{m} (J_A(i) - J_B(i))^2 + w_W \sum_{i=1}^{m} (W_A(i) - W_B(i))^2 \right)
\]

(2)

where \( J_A(i) \) and \( J_B(i) \) are the rotations of the \( i \)-th degree of freedom of the finger joints in frame \( A \) and frame \( B \), respectively, while \( W_A(i) \) and \( W_B(i) \) are the corresponding angular velocities. Weighting terms, \( w_J=1 \) and \( w_W=0.5 \), ensure that angular velocities and rotations have a similar influence. The finger skeleton consists of \( m \) degrees of freedom, for our hand skeleton \( m=25 \).

3.4 Finding the Best Path

To select the final motion, we compute a weighted graph as shown in Figure 3. The start node of this graph is connected to the \( k \) segments from the database that were most similar to the first input motion segment. Then, each segment is connected to the \( k \) segments that were most similar to the second input motion segment. So, each row \( i \) represents the \( k \) best matches for input segment \( i \) and each segment of one row \( i \) is connected to each segment of the following row \( (i+1) \).

The graph is traversed from top to bottom. A weighted sum of the corresponding transition and segment costs is applied to each connection:

\[
\text{cost} = w_S \cdot c_S + w_T \cdot c_T
\]

(3)

The weights \( w_S=1 \) and \( w_T=0.5 \) adjust the segment and transition costs so that they have the same influence on average, with positions measured in cm and rotations in degrees. We find the best path through the weighted graph with Dijkstra’s algorithm.

3.5 Computing the Final Motion

Although the transition cost was taken into account when selecting the fragments, there will be differences between the finger poses of consecutive fragments. Simple blending between each fragment (every 0.33 – 2 seconds) could result in a jittery motion. Therefore, for each transition, we compute the difference between the finger rotations of the first frame of the new fragment and the last frame of the previous fragment and add this difference as an offset to the finger rotations of the new fragment. Over time this offset can increase. We perform a linear blend only when the mean angular difference or the maximum angular difference exceeds a threshold (5° and 20°, respectively).

4 Analyzing the Similarity Function

Our algorithm relies on the hypothesis that it is possible to infer plausible hand motions from body motions, even if there might be multiple solutions with substantially different valid finger motions for a specific body motion. We want to assess the validity of this hypothesis to determine which parts of the body are most appropriate for predicting finger motions. We test different parameters and combinations of parameters, notably the positions and orientations of the shoulders, elbows and wrists.

To assess the performance of these parameters, we created a database with known finger motion solutions. We recorded multiple short samples of eight gesture types: attention, big, big shrug, ok, palm presentation (PP), small, turn, and wipe. These gesture types have characteristic finger motions but can have large variations in their arm motions. For example, an ok-gesture can be performed at different heights and in different directions but the thumb and the index always form a circle and the other fingers are nearly straight (Figure 5). A good similarity function should, when given the body motion corresponding to an ok-gesture as input, find the ok-gestures in a database that includes other gestures.

To decide which gesture types to capture, we annotated several minutes of conversational motions with a method similar to Kipp et al. [2007] and chose gesture types that were present multiple times and had characteristic finger motions. We recorded variations of each gesture type, ensuring that different pacing, hand height, and distances would be represented in each gesture type (Figure 5 and 6).

We used an equal number of samples from each gesture type, leading to 64 gestures in the database (8 repetitions x 8 gesture types). The gestures were performed with one or two hands, depending on what was natural. Each gesture was performed in a controlled and systematic way, starting and ending in a rest position with the arms hanging relaxed at the side of the body. We trimmed the beginning and ending of every gesture to remove rest poses before and after the actual gesture. We then segmented all gestures with the method presented in Section 3.1, typically leading to three segments: preparation and stroke, post-stroke hold, and retraction. The database consisted of 187 segments.

If we take one of the gestures out of our database and reconstruct it using the remaining gestures, we can assume that the motion is plausible if gesture segments of the same type were selected for the reconstruction and incorrect if other gestures were selected. To determine which body segments are good predictors for finger motions, we used leave-one-out cross-validation. For each parameter, we repeat the following. We iterate over all segments in the database and predict its gesture type based on the remaining gesture segments. The prediction itself is done by finding the \( k \) most similar gesture motions in the database based solely on the comparison of the parameter we evaluate, using squared differences and dynamic time warping. We compute the percentage of correctly classified segments amongst the \( k \) closest suggested segments. In our analysis \( k=5 \).
Figure 7: Percentage of correctly selected gesture types for some of the parameters we have considered.

Figure 7 plots the computed probabilities for some of the body segments we tested. When each parameter was considered on its own, the wrist rotation performed best with 75%. We also included weighted combinations of different parameters and concluded that a weighted combination of the position and orientation of the wrist gives the best prediction result. With positions measured in cm and rotations in degrees, the best combination was found to be $P + 0.5 \cdot R$ (80%), which explains the weights used in Equation 1.

A class confusion matrix can be seen in Figure 8. The gesture type attention was the hardest to recognize. However, even for this gesture type, 55% of the suggestions were correct. In 18% of the cases a segment from an ok gesture and in 14% a segment from a small gesture was suggested instead. Both are gestures with relatively similar arm motions to attention. In the full algorithm, the transitions help to avoid segments with finger motions that are very different from those of the surrounding segments. Therefore, the success rate when a complete motion is reconstructed is likely to be higher.

5 Results

To demonstrate the flexibility of our approach, we motion captured an actress and two actors with optical Vicon motion capture systems consisting of 13 to 18 cameras. In each capture, fifty-six markers were attached to the actor’s body and 22 markers to each hand. The actors were asked to perform a series of gestures and conversations. The resulting data was post-processed and a skeleton was fitted to it. For each hand and for the body, we recorded a series of movements rotating each joint across its full range of motion. We then computed three skeletons (one for each hand and one for the body) separately to increase their accuracy. The skeletons were then assembled into one. In the following sections, we present a series of examples computed with different gesture databases. The resulting animations can be seen in the accompanying video.

5.1 Short Gestures

We synthesized the finger motions of several gestures from the large gesture database presented in Section 4. We chose one of the 64 gestures as an input motion and deleted its finger movements. The remaining 63 gestures formed the database that we used to synthesize new finger motions. In the video, we show a comparison between the ground truth motion capture and our reconstructed finger motion for an ok and a doubt-shrug gesture. We also show an attention gesture as a failure case. In this example, the finger motion of the gesture’s stroke phase was selected from an ok gesture instead of an attention gesture.

We also synthesize the finger motions of the attention gesture from the small gesture database using a different actor and consisting of seven types of motions, each captured twice in a very similar manner. Motion types were attention, count, no, ok, point, snap, and thumbs-up. We use the body motion of one of the attention gestures as input motion and reconstruct its finger motion using the remaining 13 gestures as a database. The finger motion is synthesized correctly using the attention gesture that remained in the database.

We also synthesize the finger motions of the attention gesture from the small gesture database using the large gesture database. To account for differences in size between the actors, we scale the position data based on the height of each character. The motion was reconstructed appropriately as can be seen in the video.

5.2 Conversations, Debates, and Directions

We generated three databases consisting of longer scenes: conversations, debates, and directions. The databases were recorded with different actors and the topics were chosen to lead to different gesture types.

For the conversation database, we recorded more than 13 minutes of motions of an actor conversing about such subjects as his latest holiday trip, a project he was working on, or a movie that he recently watched. To increase the naturalness of the motions, a second person stood behind the cameras acting as a conversational partner. For motion synthesis, we used sequences of 2-4 minutes as input motions, leaving a database of about 10 minutes. The synthesized finger motions look realistic and extracts can be seen in the video.

Our second database, debates, consists of 9 minutes of debates from a different actor. We chose debating because we hoped that it would lead to more energetic gestures. We listed a dozen of popular topics from debating clubs and let the actor choose the subjects that he preferred. Our five takes were each between 1.5 and 2 minutes in length. To create examples, we excluded a take from the database, and synthesized its finger motions from the remaining database.

Third, for the directions database, we asked an actress to describe how to get to different places, such as the airport or her home. Giving directions involves very specific gestures, for instance, for turning or stopping. Four takes, each 2-3 minutes long, resulted in over 9 minutes of data. We generated examples in the same way as for the previous databases.
Figure 9: Means of the rated realism of short animated clips with different finger motions. Animations without finger motions were rated significantly less realistic than animations with motion captured finger motions or with finger motions synthesized with our method. Error bars represent one standard error of the mean.

5.3 Perceptual Experiment

We tested the realism of our results in a perceptual experiment. Participants were shown short clips of animations with three types of finger motions: "motion captured", "our method", and "no finger motions". The total stimuli consisted of 8 (segments) x 3 (motion types) x 3 (repetitions) = 72 clips. Five segments were chosen from the debates database and three from the large gesture database, including the incorrect reconstruction of the attention gesture shown in the video. We chose a camera perspective focusing on the characters hand space (thigh to neck) for most clips, because we did not want to show the face as it is not animated.

The experiment was fast paced, with each clip lasting between 3 and 5 seconds, as a complete lack of finger motions is usually obvious in longer sequences. The 72 clips were shown in random order. Between each clip participants had 5 seconds to respond.

The participants were asked to pay attention to the finger motions and to rate the realism of each animation on a scale from 1 (very unrealistic) to 5 (very realistic). The 14 participants were students and postdocs (7m, 7f, aged 18-35) who were recruited through posters on campus and announcements in classes. The experiment took about 20 minutes and participants were rewarded with $5. Participants took part in five groups of two to three.

To analyze the results we averaged the three repetitions of each segment for each motion type and participant. A repeated measures ANOVA (within effects motion type and segment) with a Huynh-Feldt correction determined a significant effect of motion type: F(2,26)=25.784, p<0.01. A Newman-Keuls post-hoc analysis showed that the animations with the motion types “motion capture” and “our method” were rated significantly more realistic than segments with “no finger motions” (see Figure 9). The differences between the motion captured animations and our method were not significant.

6 Discussion and Future Work

In this paper, we present a novel and surprisingly simple approach to generate detailed, lifelike finger motions for gesturing characters. Our method segments the arm movements and, for each resulting motion fragment, searches a database for the k fragments with the most similar wrist motion. It then takes into account the smoothness of the transitions between consecutive segments to compute the final motions.

Finger movements generated with our approach contain realistic details. However, the meaning of the conversation is not taken into account. Could the generated motions therefore be confusing or convey a different connotation than desired? Most gestures in an average everyday conversation support the emphasis and the rhythm of a conversation and do not contain specific semantic meaning. We believe that one reason our approach generates plausible results for many examples is that the timing, intonation, and even to some extent the meaning of the speech are also represented in the motions and the rhythm of the body, and more specifically of the wrist. Nevertheless, if a particular, semantically meaningful gesture is required our algorithm might produce incorrect results, especially if the wrist motion of that gesture is similar to the wrist motion of other gestures.

We showed that a weighted combination of the wrist position and rotation provides the best results in predicting finger motions amongst the options tested. But there might be more powerful ways to exploit the correlations between body and finger motions. Many other similarity functions are conceivable and some of them might provide a better performance. Bigger databases that include a broader range of gestures would be necessary to investigate the full extent of the correlations between body and finger motions.

As with all data-driven approaches, our results highly depend on the available data. A finger motion that is not present in the database can not be synthesized. Our method allows the use of a database from one character to augment the body motions of a different character. We believe that our approach translates well across characters because the gestures were in free-space mitigating the retargeting problem. However, we do not take the individual style or cultural backgrounds of those characters into account.

Our method is intended as a post-processing step, which has the advantage that the motion capture session is significantly simplified because only the body needs to be captured. Applying our algorithm in post-production also enables the user to actively influence the results for example by selecting amongst the top candidates for the most appropriate expressive gesture.

In our algorithm, right and left hand motions are synthesized independently to enlarge the variability of our databases. However, it would also be possible to compute both hands together to guarantee synchronization in gestures where the finger motions of the hands are coordinated.

We applied our approach to several databases of motion captured movements. It can also handle databases created by other means as long as the wrist motion is available. For example, keyframed animations or stylized motions could also be used. Our largest database consists of 13 minutes of conversational motions. Further tests and databases are necessary to explore the scalability, the potentials and limits of our approach. Finger and body motion databases are still labor-intensive to create. We hope to contribute to future research on finger motions and gestures by making our databases publicly available: http://graphics.cs.cmu.edu/projects/fingerMotions/

Acknowledgments

Thanks to Justin Macey for post-processing motion capture data, Moshe Mahler for modeling the characters, Valeria Reznitskaya for video editing, and Brooke Kelly for labeling gestures. This research has been supported by the ONR DR-IRIS MURI project grant N00014-09-1-1052 and the NSF grants IIS-1018486 and CCF-0811450.
References


