Finger Motion Estimation and Synthesis for Gesturing Characters

Christos Mousas* Christos-Nikolaos Anagnostopoulos† Paul Newbury‡

Intelligent Multimedia and Virtual Reality Lab
Dept. of Cultural Technology and Communication
University of the Aegean, Greece

Interactive System Lab
School of Engineering and Informatics
University of Sussex, UK

Abstract

In this paper, a novel methodology for estimating a virtual character's finger motion based on parameters provided by the character's wrist is presented. For the motion estimation process, firstly the motion sequences are classified into active and passive finger gesture phases with the active finger gestures classified according to different gesture types. Based on both classifications the system first searches for the gesture phase and then for the most appropriate gesture type. Having found the gesture type, or having determined that the input motion segment belongs to the passive phase, by using a metric, it retrieves the closest motion segment. Such a method can be beneficial in the finger motion estimation process, since both wrong estimations and the computational time of the searching process are reduced. Finally, in addition to the motion estimation process, an optimisation of the motion graphs methodology for searching optimal transitions between two consecutive motions is introduced.

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

Keywords: finger motion, motion estimation, motion features, motion synthesis

1 Introduction

An important need of the computer animation community is to find a method for synthesising the desired motions of such a virtual character that the required naturalness of motion is maintained. These methods should be able to provide the required results automatically, avoiding the use of the traditional key-frame technique that results in a time-consuming process. To achieve the automatic estimation and synthesis of the character's motion a collection of motion data is required. The required motion data can be retrieved by using various solutions for capturing real humans who perform the desired actions. After the capturing process, it is possible to identify the computations that will result in the synthesis of the required motion. Finally, the newly synthesised motion is retargeted on to a character and then is placed in a three-dimensional environment.

However, even if human motion can be edited or synthesised such as retaining the desired naturalness and fulfilling user specified constraints, other factors influence the perception of naturalness of human motion. Among these factors, one of the most important ones is the required detail that the character's motion contains. Upon extensive examination of [Courgeon et al. 2009][Tinwell et al. 2011][Hyde et al. 2013][Clavel et al. 2009][Kendon 2004][Jörg et al. 2010][Samadani et al. 2011] we see that the details of the face and the hands affect the perception of the naturalness of the character's motion. Additionally, as many of these perceptual studies have shown, the key meanings of the character's postures might not be clearly understood if either the finger or the facial motions are omitted (see Figure 1).

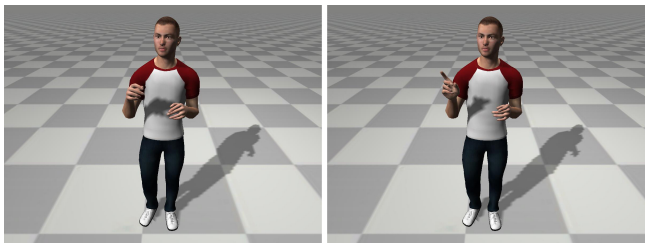

Figure 1: A character that does not include the finger gesture may change the meaning of the pose. A character without applying finger gesture (left), and the same character with finger gesture (right).

This paper presents a novel finger motion estimation and synthesis process. The proposed methodology does not piece wise search the segmented motion data that is contained in the database to find the most valid motion of the character, as it has been proposed in previous solutions, such as in [Jörg et al. 2012] and [Jin and Hahn 2005]. Instead, the proposed solution presents the ability to search for the most appropriate gesture type in a first step; then, having found the gesture type, the system searches through the motion collection to which the gesture belongs and then retrieves the most valid motion. Figure 2 illustrates the searching process of the presented methodology. The advantage of such a search process is the increment of the estimation rate as the evaluations have shown. Specifically, a 10% increment of correct estimation is achieved when compared to [Jörg et al. 2012].

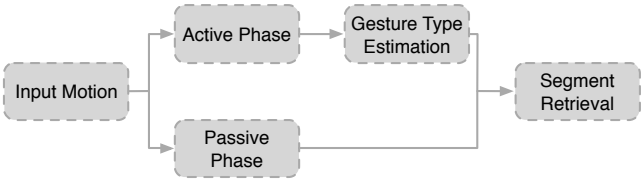

Figure 2: The search process of the presented methodology.

In our implementation as well as for the experiments that were conducted in the presented methodology the motion data provided

*e-mail: chr.mousas@aegean.gr
†e-mail: canag@ct.aegean.gr
‡e-mail: p.newbury@sussex.ac.uk

SCCG '15, April 22 – 24, 2015, Smolenice, Slovakia.

DOI: http://dx.doi.org/10.1145/2788539.2788552

in [Jörg 2015] was used. These motion sequences consist of the following parts: a gesture dataset that contains eight repetitions of eight different gestures (namely: attention, big, small, turn, wipe, doubt shrug, palm presentation and OK), which is ideal for a cross validation process, freeform motions instructing directions, conversation-related and debate-related finger gestures, and a large dataset with a variety of finger gestures. Finally, it should be noted that all the aforementioned datasets are provided in conjunction with the full-body motion of a character.

The remainder of this paper is organised in the following sections. In Section 2, related work on finger animation techniques is presented. Section 3 covers the pre-processing step of the proposed solution. Section 4 covers the finger motion estimation process. In Section 5, an optimised search process of the transition between two consecutive motions based on the motion graphs methodology is presented. Section 6 covers the results obtained from the evaluation process of the proposed solution. Finally, conclusions are drawn, with a discussion of potential future work in Section 7.

2 Related Work

The automatic addition of valid motion of the character's finger is a promising research area. Thus, methods that automatically synthesise hand motions have been proposed. In general, the hand-over animation methods rely on the ability to add finger and hand motion capture data to the pre-existing full-body motion capture data. Hence, solutions, such as those proposed by Kang et al. [Kang et al. 2012], Wheatland et al. [Wheatland et al. 2013] and Mousas et al. [Mousas et al. 2014c], which are based on statistical analysis and synthesis techniques that have been used extensively in full-body character control [Mousas et al. 2014b] [Mousas et al. 2014d] [Liu et al. 2011] [Chai and Hodgins 2005], are responsible for adding the required motions of the character's hand even when a reduced number of markers are used for the motion estimation process.

Among the first data-driven methodologies for automatically assembling finger motion to animated characters was proposed by Jin and Hahn [Jin and Hahn 2005]. In this approach, the pose space distance from the character's motion capture data is analysed and a stepwise searching algorithm is used to find the key poses to synthesise the hand motion. Building on the benefits of the previous method, Jörg et al. [Jörg et al. 2012] proposed a simple solution based on a metric for estimating the motion of a character's fingers. This approach is based on the ability to synthesise a character's finger motions by assigning weight factors to the wrist's position and orientation, which is used as the input parameters of the motion estimation process. Another data-driven solution for synthesising a character's finger motion was proposed by Majkowska et al. [Majkowska et al. 2006]. In this methodology, finger and body motion are captured separately in a pre-processing stage. Then, during the composition process, those motion sequences are assembled using spatial and temporal alignment methodologies, and the motion correlation is found.

Generally, it could be stated that by using a whole-body motion it is difficult to predict the exact actions of the hand. Therefore, methodologies for synthesising a character's hand motion that are constrained to specific tasks have also been proposed. Specifically, the solution proposed by Ye and Liu [Ye and Liu 2012] is responsible for synthesising the motion of the fingers based on wrist movements and specified constraints describing manipulation strategies of virtual objects.

Methodologies that automatically animate the motion of a character's fingers while a character interacts with musical instruments have also been proposed. One of the most recent approaches is proposed by Zhu et al. [Zhu et al. 2013]. Specifically, information retrieved from a MIDI input is assigned to specific actions of the fingers, while a character interacts with a piano. Their system generates a valid motion sequence for the virtual character's piano performance. Similarly, the solution proposed by El Koura and Singh [ElKoura and Singh 2003] generates finger motion for specific tasks with a particular application in interaction with musical instruments.

Alternatively, physics-based and kinematics-related approaches have been used to generate finger motions, particularly for manipulation and hand-object interaction tasks, such as the solutions proposed by Liu [Liu 2008][Liu 2009], Pollard and Zordan [Pollard and Zordan 2005], Bai and Liu [Bai and Liu 2014], and Andrews and Kry [Andrews and Kry 2013]. Other works use sensors to measure forces [Kry and Pai 2006] to attempt to generate the correct finger motion. Moreover, Neff and Seidel [Neff and Seidel 2006] derived synthesised human hand motions by using relaxed hand shapes and physics based parameters retrieved from video recordings. Another interesting solution for animating detailed and anatomically valid hand and finger motion was proposed by Tsang et al. [Tsang et al. 2005]. In this methodology a skeletal musculotendon model of the human hand and forearm was presented. The aforementioned model permits direct forward dynamics simulation, which accurately predicts hand and finger position given a set of muscle activations. Finally, Zhao et al. [Zhao et al. 2013] proposed a physics-based motion control system for real-time synthesis of human grasping. Specifically, given an object to be grasped, their system automatically computes physics-based motion control that advances the simulation to achieve realistic manipulation of the object.

In this paper a data-driven method for estimating and synthesising the motion of a character's fingers is presented. Comparing with [Jörg et al. 2012], which is the closest solution to the one that is described in this paper, the presented methodology provides a novel way for searching the most valid motion of a character's fingers by splitting the estimation process into sub-problems (gesture phase estimation, gesture type estimation, and motion retrieval). Therefore, the piecewise searching of the most valid motion segment is eliminated, resulting to the decrease of the computational time that is required for retrieving the final motion. Finally, a 10% increment of correct estimation is achieved, which is also an advantage of our method.

3 Pre-Processing Motion Data

Before the finger motion estimation, pre-processing steps are involved. Firstly, the motion data should be segmented, and secondly the required motion features should be found. Both of these steps are presented below.

3.1 Motion Segmentation

A gesture consists of various phases: the preparation, in which the arm moves away from its rest position, the pre-stroke hold, the stroke, the post-stroke hold, and the retraction, where the arm moves back into the resting position, as McNeil [McNeill 1992] and Kendon [Kendon 2004] mentioned. Based on the aforementioned phases of a finger gesture, a simple methodology for segmenting the hand motion has been used by Jörg et al. [Jörg et al. 2012] in their finger motion synthesis process, which had previously been proposed by Kita et al. [Kita et al. 1998]. More specifically, the motion segmentation process evolves while the velocity of the wrist joint crosses a threshold that is close to zero. This approach results in two phases: the motion phase that contains sequences with high speed and the hold phase that contains low-speed motions. In the presented methodology following the analysis performed in [Mousas et al. 2014a], which recognizes the active and passive phases of a finger gesture, the motion data is split while the velocity of the

character's wrist reaches its maximum values. Specifically, the active phase contains the meaningful part of a gesture and the passive phase contains the rest of the gesture that are its preparation and its retraction phases. A simple example is illustrated in Figure 3.

By using the 64 gestures (8 repetitions x 8 gesture types) provided in [Jörg 2015], the motion data is split in 197 segments. More specifically, 59 of the 64 motions are segmented into three phases, while the remaining motions (5 motions) are segmented into 4 phases. In this case, it should be mentioned that there were no problems with the lengths of the segments, as in [Jörg et al. 2012]. This is true due to the fact that the motions with the short post-stroke hold phases can be segmented efficiently while the segmentation evolves during the maximum values of the wrist's velocity. The post-stroke hold could be assumed that is within the active phase of the gesture. Finally, it should be noted that for the motions that are segmented into 4 phases, each segment that is between the maximum wrist velocities is assigned as an active phase.

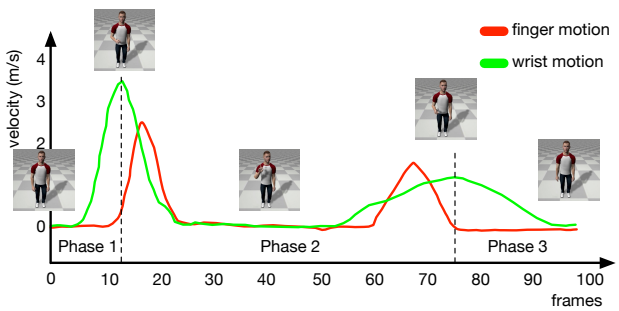

Figure 3: As it can be observed from this graph the active phase of a finger gesture (the phase in which a gesture fades in and out) is between the maximum values of wrist velocity. The green line indicates the velocity of the wrist and the red line the velocity of the fingers.

3.2 Motion Features

Human motion can be characterised by its features. Those features are responsible for describing the human motion in either a spatial or a temporal way. Therefore, a variety of motion features can be computed from different body parts for analysing the human motion. In the presented methodology, for each segment s_i contained into our database, the distance (the travelled distance) d_i, the average orientation $\bar{o}_i$, the average velocity $\bar{v}_i$, the average angular velocity $\bar{av}_i$, the average acceleration $\bar{a}_i$, and the average angular acceleration $\bar{aa}_i$ were taken into consideration. Based on these features a segment s_i that is assigned to all of the features is represented as $s_i = \{d_i, \bar{o}_i, \bar{v}_i, \bar{av}_i, \bar{a}_i, \bar{aa}_i\}$. It should be mentioned that a variety of other features could be used, such as the height, the temporal extent, the curvature, etc. However, as the evaluation has shown the collected features provide reasonably improved results.

4 Finger Motion Estimation

In the following subsections, the ability to recognise the phase of input motion segments, the ability to search through the active phases to find the most appropriate gesture type, and finally, a metric for retrieving the closest motion segment are introduced.

4.1 Gesture Phase Estimation

The computation that indicates whether the input segment belongs to an active phase is one of the most important parts of the proposed solution because a wrong estimation directly affects the selection of the most appropriate motion. Therefore, careful consideration is required. As mentioned earlier, the majority of the motion sequences are segmented into three phases, where two of these phases are marked as passive phases, and the third is marked as active phase. In this case, by plotting various features against each standard deviation, as well as by evaluating each of the gestures contained in the database, the active with the passive phases are separated either by using the wrist's orientation or the wrist's distance (see condition in Equation 1). More specifically, on one hand, based on the gesture dataset provided by [Jörg 2015] motion segments attributed to attention, big, small, turn, palm presentation and wipe, gestures can be recognised by using the travelled distance of the character's wrist. On the other hand, motion segments, that are attributed to the doubt shrug and okay gestures, can be recognised by using the wrist's average orientation. The plots that distinguish the active from the passive phases that are used for this process can be found in the supplemental material.

In the proposed solution, the process that distinguishes the active with the passive segments can be easily computed. In this case, given an input segment s_j, the aforementioned distinction is achieved by the following condition:

$$s_j = \begin{cases} active & \text{if } g_d^a \leq g_d^p \text{ or } g_o^a \leq g_o^p \\ passive & \text{otherwise} \end{cases} \tag{1}$$

where g_d^a, and g_d^p represents difference between the distance feature of the active and passive phases respectively, and g_o^a, and g_o^a represents the distance between the orientation feature of the active and the passive phases respectively. The distances between the input motion segment and the reference motion data of the examined features g_d, and g_o are computed according to the following equations:

$$g_d = \frac{(d_j - \bar{D})^2}{\sigma_D^2} \tag{2}$$

$$g_o = \frac{(\bar{o}_j - \bar{O})^2}{\sigma_O^2} \tag{3}$$

where d_j and $\bar{o}_j$ represent the distance and the average orientation of the wrist respectively and both of these are computed from the input motion segment. Moreover, $\bar{D}$ and $\bar{O}$ represent the centroid values of the distance and orientation features where both of these are provided by the motion capture data contained in the database. Additionally, σ_O and σ_D represent the standard deviations computed from the registered reference orientations and travelled distance respectively as provided by the character's wrist. Finally, it should be noted that both g_d and g_o are computed for both the active and passive phases of $\bar{D}$ and $\bar{O}$.

4.2 Gesture Type Estimation

While knowing in advance that the input segmented motion belongs to the active phase, a method is required now for estimating the gesture type. A simple solution could be the direct estimation of the most probable gesture type by simply summing the distances between the features that were computed from the input motion segment and the centroid values of the analysed features contained in the database. However, each different feature counted in such a function has different metrics, which means that optimisation weights should be computed in order to provide such a metric for the required results. However, in the presented methodology,

taking in consideration the time consuming process for estimating the optimal weights manually as in [Jörg et al. 2012], a methodology that solves the direct estimation automatically is introduced.

Specifically, firstly we assumed a "ghost" feature, f_{ghost} that follows a normal distribution i.e., $\mu_{ghost} = 0$ and $\sigma_{ghost} = 1$, which is represented by $\mathcal{N}(0,1)$. Based on f_{ghost}, it is possible to align both the centroid values and the distributions for each of the motion features to f_{ghost} by shifting and scaling its values. For achieving this transformation process, firstly the mean and the standard deviation of each feature are computed as:

$$\mu_f = \sum_{i=1}^{N} \frac{f_i}{N} \tag{4}$$

$$\sigma_f = \sqrt{\sum_{i=1}^{N} \frac{(f_i)^2}{N}} \tag{5}$$

where μ_f and σ_f are the mean and the standard deviation of each of the features f, f_i represents a feature of the $i-th$ motion segment and N the number of motion segments that belong to a gesture type. Having computed μ_f and σ_f of each feature, and having been given an input motion segment s_j that is required being estimated, it is possible to shift and scale each features of this segment to the normalised space. This is achieved as follows:

$$f_j = f_j - \mu_f \tag{6}$$

$$f_j = f_j \times \frac{\sigma_{ghost}}{\sigma_f} + \mu_{ghost} \tag{7}$$

The aforementioned two steps are responsible for normalising each motion feature according to the ghost feature. It should be noted that μ_{ghost} could also be removed from Equation 7 since $\mathcal{N}(0,1)$. However, Equation 7 illustrates the generalised representation of the scaling process of each feature.

Having an input motion segment s_j as well as knowing the normalised position of a feature associated with s_j it is now possible to compute the distance between f_j and its centroid value $\bar{F}$ by:

$$g_f = \|f_j - \bar{F}\|^2 \tag{8}$$

where f_j could be any feature of s_j and $\bar{F}$ represents the centroid value of the features f. To conclude, considering the six motion features mentioned in Section 3.2 it is now possible to assign the most probable gesture type estimation into the following metric:

$$g_T = g_D + g_O + g_V + g_{AV} + g_A + g_{AA} \tag{9}$$

where g_D, g_O, g_V, g_{AV}, g_A, and g_{AA} denote the difference between the centroid values of the distance, orientation, velocity, angular velocity, acceleration and angular acceleration motion features, and the input motion segment as each one is estimated according to Equation 9. Therefore, based on Equation 9, the most probable gesture type is the type for which the value of g_T provides a minimum value.

4.3 Segment Retrieval

Having found that the input motion belongs to either a gesture type or a passive phase, a method is required to retrieve the closest segment. In this step, this goal is achieved by assigning the exact motion estimation process to a distance metric over the trajectories of the motion sequences instead of the motion features as performed previously.

Generally, it could be stated that having estimated the most probable gesture type, it is not necessary to search again through the motion collection of the estimated gesture. This finding is especially true for the active phases, since knowing the gesture type any of the finger motions assigned to a character it will be marked as a correct decision. However, the most probable motion that is expected to fulfil at least the closest spatial parameters of the input motion must be estimated. This final estimation process does not only ensure the meaning of the synthesised motion but also ensures the required naturalness that the synthesised motion must have.

For that reason, a very simple metric is presented. Specifically, considering an input motion segment s_j and reference motion segments s_D contained into the collection that the system searches, the following equation is computed over the trajectories of the motion segments:

$$D_M = \frac{1}{N} \sum_{t=1}^{N} (o_i(t) - o_D(t))^2 \tag{10}$$

where $o_j(t)$ and $o_D(t)$ denote the orientation of the input motion and each of the motions that are contained in the data base at the $t-th$ frame for $t = 1,...,N$. For the computational process of the metric, a dynamic time warping function is used [Kovar and Gleicher 2003], such as each of the o_D reference motions, in order to have the same length as the input motion o_j. Hence, the winning segment is the one that minimises D_M.

5 Motion Synthesis

In this section, an optimisation of the searching over the motion graphics transition process is introduced. The presented method treats the transition process separately for different hand parts (local transitions), rather than computing a single transition for the whole hand (global transition). The local transition process assumes that there is an optimal transition time that lies within the transition time computed with the standard motion graphs [Kovar et al. 2002] methodology. In this case, the minimisation of the distance between each independent part, which should be within the initial path computed with the standard motion graphs approach, is considered. Therefore, a smoother transition for each independent hand part is achieved as it is indicated by evaluating the transition cost between two consecutive motion sequences (see Section 6.3).

More specifically, the transition process is separated into six different parts over the character's hand: the thumb, index, middle, ring, and little fingers and the palm, as illustrated in Figure 4. Each independent part, during the local transition process of the character's hand, has its own origin frame, l_O, and target frame, l_T. Additionally, during the global transition process the character's hand has its own origin frame, t_O, and target frame, t_T. Figure 5 illustrates an example between the global and the local transition processes.

Figure 4: The different parts of the hand used for computing the transition process.

Based on this representation, the optimal time steps for the transition process of each independent part of the character's hand must be calculated. First, the global transition time, $T(t_O, t_T)$, between

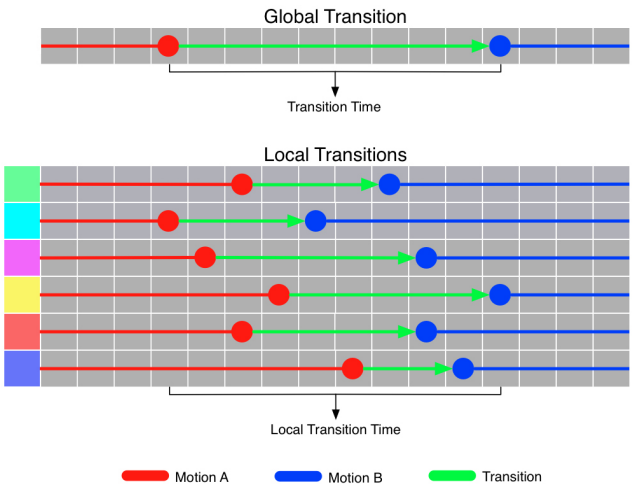

Figure 5: The transition process as it is computed using the standard motion graphs (upper diagram) and the local transition process for each part of the character's hand (lower diagram). Each colour on the side denotes the part of the hand (see Figure 4).

t_O and t_T is assessed using the standard motion graphs methodology. Having found the response frame, t_O, the transition time for each independent part is computed as follows:

$$T(l_O^i, l_T^i) = \min \|l_O^i - l_T^i\| \tag{11}$$

for

$$t_O \leq T(l_O^i, l_T^i) \leq t_O + T(t_O, t_T)$$

where $l_O^i \in l_O$ and $l_T^i \in l_T$ denote each independent part of the characters hand on the origin and on the target position, respectively.

6 Evaluations and Results

In this section, the results of the evaluation performed during each step of the motion estimation and synthesis process are presented.

6.1 Evaluating the Motion Phase Estimation

Given an input segment, in a first step, the system is responsible for estimating whether this segment belongs to the active or to the passive phase. This is achieved by using the character's wrist average orientation and distance. As mentioned earlier, an incorrect computation of the motion phase directly affects the motion estimation process. For example, consider an input segment that belongs to the active phase and that the system recognises it as a passive; then, it will directly search for a passive motion to assign to the character's hand.

In this case, to evaluate the accuracy of the motion phase estimation, a leave-one-out cross validation process was conducted. More specifically, by leaving out one of the segments (a segment that belongs to either the active or the passive phase), the correct estimation of the gesture type is computed. The same procedure is iterated for each of the segments (197 segments) that are contained in the database. In addition, it should be noted that for every iteration the new centroid values of the motion features were calculated again. The results obtained from this evaluation process are presented in Figure 6 and are quite promising. The proposed method can estimate correctly the active phase with approximately 98% accuracy.

To evaluate how accurate such a method could be, very close results were obtained while using the eight gesture types for the training process with motion segments from the other datasets (directions, debates, and conversations databases), which can be found in [Jörg 2015]. Analytically, while the gesture database is used for the training process, the estimation of the correct gestures is approximately 93% in the directions dataset, 96% in the debate dataset and 95% in the conversation dataset. Comparing the difference between the gestures dataset and any of the other datasets, the following can be stated. These differences result from the inability of the motion data to describe the large variety of motions that can be produced by the character. However, because the results remain quite close, it can be stated that the method works well.

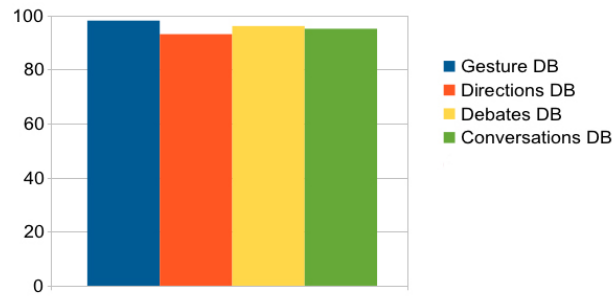

Figure 6: The results obtained from the motion phase estimation process when using different datasets.

6.2 Evaluating the Gesture Type Estimation

Given an input segment that belongs to the active phase, the system is responsible for estimating first the type of gesture, and then for estimating the most appropriate motion segment. This goal is achieved by using the methodology presented in Section 4.2. This section evaluates the accuracy of the methodology to estimate the correct gesture type. As in the previous evaluation process, the leave-one-out cross validation technique is considered. For understanding how each of the features reacts during the motion estimation process, the results of the evaluation process that computes the correct estimates for each feature separately are shown in Figure 7. Moreover, to understand deeper how the addition of each feature in Equation 9 influences the motion estimation process, in Figure 8, the approximate correct estimation when an additional feature is added, is presented.

The complete methodology, as presented in Section 4, was evaluated by using the leave-one-out cross validation process. Specifically, by keeping a motion segment out of the database the system was trained with the rest of motion segment. Then, by using the excluded motion segment the gesture type was estimated. The class confusion matrix (Figure 9) represents the results that were obtained from the evaluation process. The proposed methodology is responsible for correctly estimating more of the relative finger gestures compared with the solution proposed in [Jörg et al. 2012], which estimates on average 80%. The methodology achieves an overall 90% correctness, which is a significant improvement. Example synthesised postures of the character based on the proposed methodology are illustrated in Figure 10. Finally, examples of finger gestures that were synthesised based on the proposed solution are shown in the accompanying video.

In showing the efficiency of the proposed methodology, by using other finger motion datasets (conversation, debate, and direction

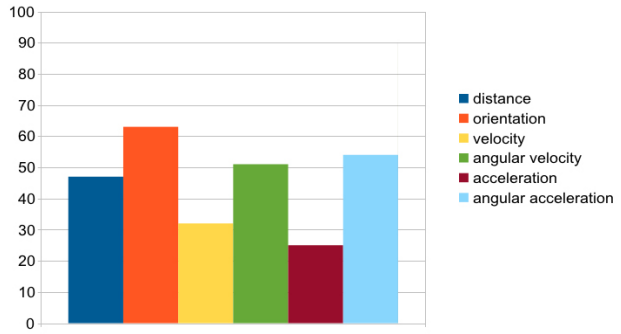

Figure 7: The correct estimation of a motion segments while using each of the examined features independently.

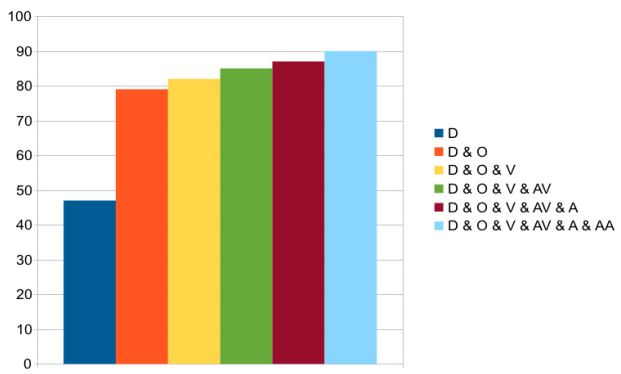

Figure 8: How Equation 9 is influenced while a new feature is added. D, O, V, AV, A, AA denote the distance, orientation, velocity, angular velocity, acceleration, and angular acceleration motion features, respectively.

datasets) that can be found in [Jörg 2015], various example motions were synthesised. Examples of different postures that were synthesised are illustrated in Figure 11. Additional results can be found in the accompanying video. The ability of the system to estimate correctly a gesture of a character while using three different datasets were computed by using the leave-one-out cross validation technique. The results obtained while evaluating the presented solution against the one proposed by Jörg et al. [Jörg et al. 2012] are illustrated in Figure 12.

As it can be seen in Figure 12, the estimation rate comparing while using the gesture dataset is reduced. Specifically, the system is able to estimate correctly on average 84% a direction, 72% a debate, and 77% a conversation finger gesture. Comparing the use of the gesture database with the use of the three different databases it should be mentioned that it is difficult to identify repeating motion sequences. Hence, even if it is not possible to generalise specific finger gestures, the proposed methodology provides better results compared to the metric proposed by Jörg et al. [Jörg et al. 2012] (see Figure 12(b)).

gesture type of the closest segment

segment taken out of the database	att.	big	d-s	OK	PP	small	turn	wipe
att.	83	0	1	7	0	5	0	4
big	0	94	0	0	2	0	4	0
d-s	2	0	78	2	3	5	10	0
OK	6	0	0	91	0	0	0	3
PP	0	0	1	0	93	1	4	1
small	0	0	1	0	0	97	1	1
turn	0	1	1	0	2	0	93	3
wipe	0	1	0	0	2	2	5	90

Figure 9: The class confusion matrix shows the percentage of gestures that are estimated using the proposed finger motion estimation process.

6.3 Evaluating the Local Transitions

In addition to the aforementioned evaluation process the transition process provided by the local transition methodology as it was presented in Section 5 was evaluated. Specifically, the proposed local transition process was evaluated against the global transition as initially introduced in [Kovar et al. 2002]. The transition process was computed for each pair of corresponding motion sequences that belong to the gesture dataset. For the transition cost the transition between the last frame, t_{last}, of an origin segment A and the first frame, t_1, of a target segment B are considered. Thus, the transition cost is measured by:

$$c_T = \frac{1}{D}\sum_{d=1}^{D}(A_d(t_{last}) - B_d(t_1))^2 \tag{12}$$

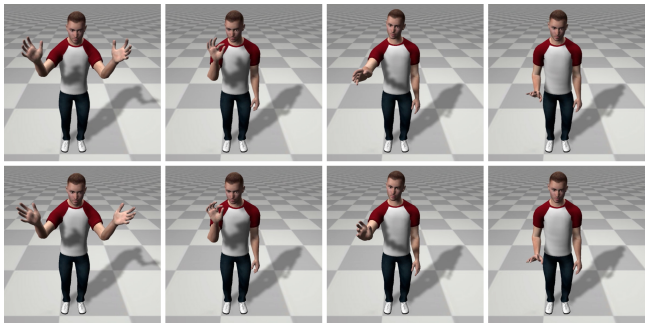

Figure 10: Given a reference motion (upper row), the system estimates and synthesises the most appropriate gesture of the character's finger (lower row).

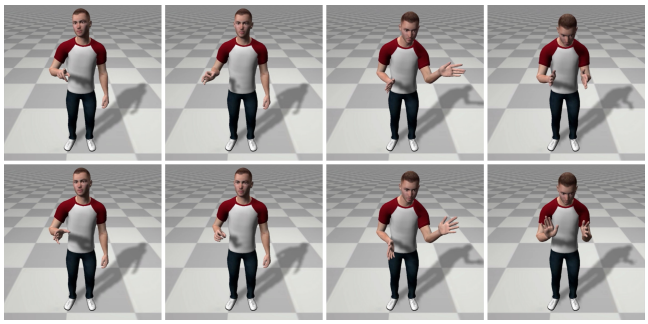

Figure 11: Using different datasets that contains motion sequences related to debate and directions (upper row), new finger motion gestures are synthesised (lower row).

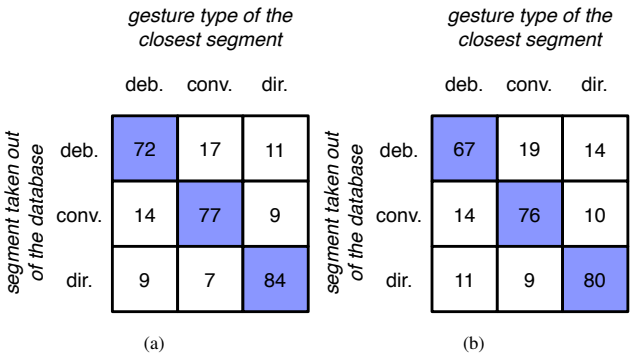

Figure 12: The class confusion matrices show the percentage of gestures that are estimated while using three different datasets. (a) the presented methodology, and (b) the methodology proposed by Jörg et al. [Jörg et al. 2012].

where $d = 1, ..., D$ is the total number of degrees of freedom of the character's hand skeleton.

The results obtained from this evaluation process are summarised in Figure 13. Specifically, as it can be seen, the local transition process minimises the transition cost on average by 11.4% while the transition process evolves from phase 1 to phase 2 (from passive to active phase), 13.7% while the transition process evolves from phase 1 to phase 3 (from passive to passive phase), and 19.3% while the transition process evolves from phase 2 to phase 2 (from active to active phase). Based on the aforementioned results it can be stated that the presented methodology achieves smoother transition between two consecutive motion segments.

7 Conclusions and Future Work

In this paper, a novel methodology for estimating the motion of a character's fingers was introduced. The advantage of the proposed methodology is based mainly on its ability to analyse the existing motion data by computing various features provided by a character's hand. Therefore, rather than directly and individually searching for each segmented motion that is contained in a database, a method that splits the motion estimation process into sub-problems was presented.

By evaluating the proposed methodology, it was shown that better results could be achieved when the estimation process is split into sub-problems. However, the accuracy of such a method is mainly based on the amount of data that is used. Moreover, a larger

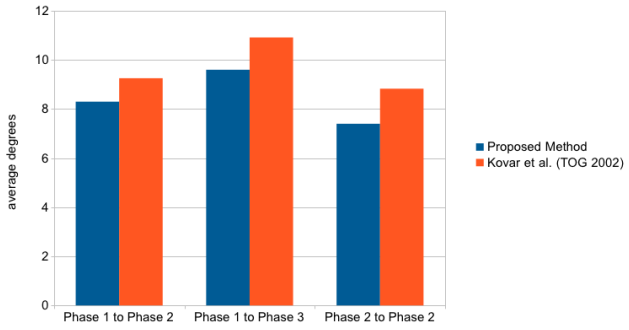

Figure 13: The results obtained from the transition cost evaluation between the proposed local transition and the global transition process as described in Kovar et al. [Kovar et al. 2002].

collection of motion features can increase the robustness of such a methodology. Conversely, as the number of gestures to be estimated increases, it is assumed that the estimation of the correct gesture may decrease. This is because a larger number of gesture types can increase the intersection of the motion features and the common characteristics that the different gesture types have. Thus, as the gestures to be estimated increase in number, it will be necessary to add more parameters of human motion features into the gesture estimation process.

Among other similar challenges, the finger motion estimation process remains an open challenge for us. Thus, our future plans will employ research on exploring and evaluating a vast number of features that can be used for the finger motion estimation process. Finally, even if off-line computation is quite beneficial in the animation community, for us, an open challenge remains in finding a method to estimate the motion of the character's fingers in real-time during the performance capture process. In this case, it is assumed that the estimation of the character's finger gestures based on the analysis of the motion features, as presented in this paper, is the first step in solving such a problem.

References

ANDREWS, S., AND KRY, P. G. 2013. Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics 37*, 7, 830–839.

BAI, Y., AND LIU, C. K. 2014. Dexterous manipulation using both palm and fingers. In *IEEE International Conference on Robotics and Automation*, IEEE, 1560–1565.

BRAY, M., KOLLER-MEIER, E., AND VAN GOOL, L. 2007. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding 106*, 1, 116–129.

CHAI, J., AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics 24*, 3 (July), 686–696.

CLAVEL, C., PLESSIER, J., MARTIN, J. C., ACH, L., AND MOREL, B. 2009. Combining facial and postural expressions of emotions in a virtual character. In *Intelligent Virtual Agents*, Springer Berlin Heidelberg, 287–300.

COURGEON, M., BUISINE, S., AND MARTIN, J. C. 2009. Impact of expressive wrinkles on perception of a virtual charac-

ter's facial expressions of emotions. In *Intelligent Virtual Agents*, Springer Berlin Heidelberg, 201–214.

ELKOURA, G., AND SINGH, K. 2003. Handrix: Animating the human hand. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 110–119.

HYDE, J., CARTER, E. J., KIESLER, S., AND HODGINS, J. K. 2013. Perceptual effects of damped and exaggerated facial motion in animated characters. In *IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 1–6.

JIN, G., AND HAHN, J. 2005. Adding hand motion to the motion capture based character animation. In *Advances in Visual Computing*, Springer Berlin Heidelberg, 17–24.

JÖRG, S., HODGINS, J., AND O'SULLIVAN, C. 2010. The perception of finger motions. In *ACM Symposium on Applied Perception in Graphics and Visualization*, ACM, 129–133.

JÖRG, S., HODGINS, J., AND SAFONOVA, A. 2012. Data-driven finger motion synthesis for gesturing characters. *ACM Transactions on Graphics 31*, 6, Article No. 189.

JÖRG, S., 2015. Finger motion database. `http://people.cs.clemson.edu/~sjoerg/fms.html`.

KANG, C., WHEATLAND, N., NEFF, M., AND ZORDAN, V. 2012. Automatic hand-over animation for free-hand motions from low resolution input. In *Motion in Games*, Springer Berlin Heidelberg, 244–253.

KENDON, A. 2004. *Gesture: Visible Action as Utterance*. Cambridge University Press, Cambridge, UK.

KITA, S., VAN GIJN, I., AND VAN DER HULST, H. 1998. Movement phases in signs and co-speech gestures, and their transcription by human coders. In *Gesture and Sign Language in Human-Computer Interaction*, Springer Berlin Heidelberg, 23–35.

KOVAR, L., AND GLEICHER, M. 2003. Flexible automatic motion blending with registration curves. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 214–224.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics 21*, 3, 473–482.

KRY, P. G., AND PAI, D. K. 2006. Interaction capture and synthesis. *ACM Transactions on Graphics 25*, 3, 872–880.

LIU, H., HE, F., CAI, X., CHEN, X., AND CHEN, Z. 2011. Performance-based control interfaces using mixture of factor analyzers. *The Visual Computer 27*, 6–8, 595–603.

LIU, C. K. 2008. Synthesis of interactive hand manipulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 163–171.

LIU, C. K. 2009. Dextrous manipulation from a grasping pose. *ACM Transactions on Graphics 28*, 3 (July), Article No. 59.

MAJKOWSKA, A., ZORDAN, V. B., AND FALOUTSOS, P. 2006. Automatic splicing for hand and body animations. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 309–316.

MCNEILL, D. 1992. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press.

MOUSAS, C., NEWBURY, P., AND ANAGNOSTOPOULOS, C.-N. 2014. Analysing and segmenting finger gestures in meaningful phases. In *Computer Graphics, Imaging and Visualization*, IEEE.

MOUSAS, C., NEWBURY, P., AND ANAGNOSTOPOULOS, C. N. 2014. Data-driven motion reconstruction using local regression models. In *Artificial Intelligence Applications and Innovations*, Springer Berlin Heidelberg., 364–374.

MOUSAS, C., NEWBURY, P., AND ANAGNOSTOPOULOS, C. N. 2014. Efficient hand-over motion reconstruction. In *International Conference on Computer Graphics, Visualization and Computer Vision*, WSCG, 111–120.

MOUSAS, C., NEWBURY, P., AND ANAGNOSTOPOULOS, C. N. 2014. Evaluating the covariance matrix constraints for data-driven statistical human motion reconstruction. In *Spring Conference on Computer Graphics*, ACM, 99–106.

NEFF, M., AND SEIDEL, H.-P. 2006. Modeling relaxed hand shape for character animation. In *Articulated Motion and Deformable Objects*, Springer Berlin Heidelberg, 262–270.

POLLARD, N. S., AND ZORDAN, V. B. 2005. Physically based grasping control from example. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 311–318.

SAMADANI, A. A., DEHART, B. J., ROBINSON, K., KULIC, D., KUBICA, E., AND GORBET, R. 2011. A study of human performance in recognizing expressive hand movements. In *IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 93–100.

TINWELL, A., GRIMSHAW, M., NABI, D. A., AND WILLIAMS, A. 2011. Facial expression of emotion and perception of the uncanny valley in virtual characters. *Computers in Human Behavior 27*, 2, 741–749.

TSANG, W., SINGH, K., AND FIUME, E. 2005. Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 319–328.

WHEATLAND, N., JÖRG, S., AND ZORDAN, V. 2013. Automatic hand-over animation using principle component analysis. In *Motion in Games*, ACM, 175–180.

YE, Y., AND LIU, C. K. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics 31*, 4, Article No. 41.

ZHAO, W., ZHANG, J., MIN, J., AND CHAI, J. 2013. Robust realtime physics-based motion control for human grasping. *ACM Transactions on Graphics 32*, 6, Article No. 207.

ZHU, Y., RAMAKRISHNAN, A. S., HAMANN, B., AND NEFF, M. 2013. A system for automatic animation of piano performances. *Computer Animation and Virtual Worlds 24*, 5, 445–457.