Eurographics Ireland (2009)
Yann Morvan and Veronica Sundstedt (Editors)

Exploring the Dimensionality of Finger Motion

S. Jörg and C. O'Sullivan

Trinity College Dublin, Ireland

Abstract

The animation or capture of a virtual character's hands is a complicated task, mostly due to the high number of degrees of freedom involved. However, the movements of different digits are often correlated with each other. In this paper we use Root Mean Square (RMS) deviations as a distance metric to analyse the correlations between different degrees of freedom of hands. Using the findings of our analyses, we develop a method that allows us to simplify the task of animating or capturing hands with minimal error in the resulting motions.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Human hands are used for a multitude of gestures and tasks. Motion capturing hand motions differs from the capturing of body motions because of its high number of small bones, which allow only a limited number of small markers to be used. When more accuracy is needed, the cameras of an optical motion capture system need to be closer to the markers, which is likely to reduce the capture space. Postprocessing of hand motion data is characterised by labelling errors and occlusions resulting in time-consuming manual labour. Therefore, in most capturing sessions for games and movies, only the body movements are captured whereas the hands are animated manually.

The hand consists of an elaborate structure of joints, tendons, and muscles leading to a large number of degrees of freedom (DOFs). This makes the task of animating hands particularly complicated. However, from observation it can be seen that single rotations of specific joints are strongly correlated, meaning that they are often performed together. For example, the ring finger and the little finger are often bent at the same time. Furthermore, other joints barely move at all in specific directions. The aim of this study is to analyse correlations in hand motions and thereby to identify methods to simplify the generation of hand movements. Such methods will allow us to simplify the task of animating hands considerably. Additionally, they will enable us to measure fewer joint rotations during motion capture sessions and thus to reduce the number of markers and the time needed for postprocessing. Furthermore, a keyframe animator would only need to animate a few joints with the remaining rotations being generated automatically.

Figure 1: *Frames from the staged argument.*

Hands are used for manipulative tasks where the digits interact with an object or a person, as well as for gestures that might or might not include interaction or self-interaction. Hand motions that include contacts depend to a large degree on the task and on the object of interaction. They cannot be generated accurately without knowledge of the task. We therefore exclude any type of contacts and focus on conversational gestures. For this exploratory analysis we use the hand motions of a male actor during 27.3 seconds depicting a staged argument (see Figure 1). A scatter plot, shown in

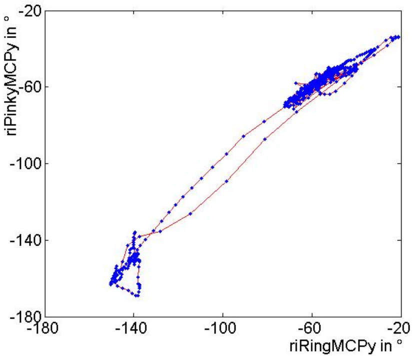

Figure 2: *The scatter plot of the flexion of the MCP-joint of the right ring finger and the flexion of the MCP-joint of the right little finger indicates a linear relationship. Each point represents a frame; successive frames are connected by a line.*

Figure2, of the flexion of two joints indicates the existence of linear relationships between some of the rotation coordinates, which implicates that the motion of one of the joints can be approximated by a linear transformation of the other motion.

Our proposed method is to use Root Mean Square (RMS) deviations as a distance metric in order to identify which joint rotations are most strongly correlated. We then use the rotation curve with the larger standard deviation to approximate the curve with the smaller standard deviation. We also remove rotations with a small range as they are barely visible. In summary, our technique takes advantage of redundancy (correlations) and irrelevance (barely visible rotations). Depending on the goal, it is possible to choose the requested quantity of dimensionality reduction. We implement different levels of simplification, and we discuss the resulting motions.

2. Background

Motion capture enables the creation of large databases of realistic motions. The amount of data in character animation has increased massively since the use of motion capture technology has become more widespread. To understand, classify, and synthesise human motion, a multitude of techniques have been developed and adapted. Although other researchers have used distance metrics and dimensionality reduction techniques to handle motion capture data [GM08, BSP*04], we apply them to a novel field and purpose, neither of which have been explored before.

Principal Component Analysis (PCA) is probably the most common method to reduce high-dimensional data [SFS98,CGA07,BZ04]. Contrary to our approach, when applied to character animation, PCA results in a new set of basis poses that include rotations of multiple joints. Our goal is to reduce dimensions while we keep the basic rotations of the hand as basis vectors, so that we have an intuitive system that an animator can work with.

A variety of metrics have been developed to measure the similarity of curves. Examples are correlation coefficients, time warping techniques, the search for the longest common subsequence [VGK02], or the extraction and comparison of features [OFH08]. Our rotation curves all have the same length. We expect neither extreme outliers nor different speeds in our data. Therefore, we choose to use RMS deviations as a metric.

Studies that are particularly concerned with hand motions can be found in the fields of biology, neuroscience, robotics and computer vision. Especially the analysis and development of grasping techniques is an extensive subject of study. Santello et al. [SFS98] show that the movements of 15 measured degrees of freedom of the hand performing 57 different grasps could by represented to more than 80% accuracy by the first two components of a PCA. Braido and Zhang [BZ04] find that as much as 98% of the variance can be explained by the first two componants of PCA, though only four fingers and two types of gestures were analysed. Ciocarlie et al. [CGA07] use PCA and *eigengrasps* to plan grasp motions for robotic hands within a highly reduced dimensionality. Even though these studies apply dimensionality reduction techniques to hand motions, none of them applies such methods to motion synthesis in character animation.

The synthesis of gestures based on language as described in [SDO*04] and [KNKA07] includes hand motions, though these are an inherent part of the gestures and are not considered individually. Majkowska et al. [MZF06] fit separately recorded hand motions to body motions using a dynamic time warping algorithm. The goal of our approach is to reduce the dimensions of hand motions and thereby to simplify the capture, animation, and synthesis of hand motions.

3. Data acquisition and processing

We captured the motion of two actors performing a staged argument. Fifteen reflective, spherical markers of 4mm-diameter were attached to the digits of each hand (3 on each digit). An additional 4 markers with 9mm-diameter were fixed on each palm, resulting in a total of 38 markers. The body and facial motions of both actors were captured during the same session, though we use only the hand data from the male actor in the study described in this paper. The markers' positions were recorded by an optical motion capture system, consisting of 18 infrared and near-infrared vicon cameras. The data was subsequently post-processed and smoothed, and the skeleton and the joint rotations were computed based on the resulting marker positions. The rotations were then checked for outliers and two sequences containing

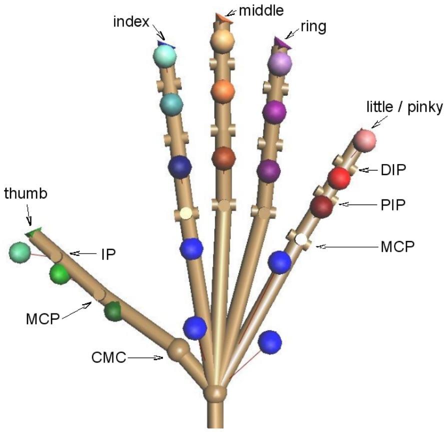

Figure 3: *Hand Model*

fewer than 10 frames each were corrected on a few rotation coordinates.

Figure 3 shows the joint model of the right hand that is used in this study. To the best of the author's knowledge, this model corresponds to the commonly used hand model in character animation. Palastanga et al. [PFS06] describe the anatomy of the human hand, on which the model is based. To avoid confusion between the words "finger" and "digit", we specify that a hand features four fingers and a thumb, or five digits. Each hand is represented by 16 joints, 3 for each digit and one for the palm. The analysis comprises the 15 joints of the digits. Each joint originally has 3 directions of rotation that we call x, y and z, though some of them are constrained. The three joints of the index, middle, ring, and little finger are called the metacarpophalangeal joint (MCP), the proximal interphalangeal joint (PIP) and the distal interphalangeal joint (DIP). The thumb has a slightly different representation as it is rotated by 90 degrees with respect to the other digits. The three joints of the thumb are the carpometacarpal joint (CMC), the metacarpophalangeal joint (MCP), and the interphalangeal joint (IP). All of the interphalangeal joints as well as the MCJ-joint of the thumb are limited to one degree of freedom: the flexion or extension. This is the y-axis for the fingers and the z-axis for the thumb. The MCP-joints of the fingers are reduced to two degrees of freedom in the skeleton model. However, the motion capture data processing software calculates values for all three degrees of freedom. The adduction or abduction of the MCP-joints of the fingers is represented by the x-axis.

We denominate a joint on the right hand with *ri* and a joint on the left hand with *le*. The little finger is denoted by *Pinky*. Thus, *riRingDIPy* characterises the curve of the right ring finger's DIP-joint's y-rotation, i.e. its flexion or extension. *lePinkyMCPx* denotes the curve of the left little finger's MCP-joint's x-rotation, i.e. its abduction or adduction.

Each rotation coordinate is represented as a sequence of frames with 30 frames per second. The rotations are specified in degrees, relative to their parents in the hierarchy. The animation consists of 27.3s or 820 frames with a keyframe at every frame. In total, taking the constrained degrees of freedom of the joints into account, this sums to 25 rotation curves for each hand.

4. Proposed Method

The goal of our algorithm is to reduce the dimensions of hand motions while keeping the rotation coordinates of the different joints as a base. To achieve this goal we investigate which joint rotations are either barely noticeable (irrelevant) or accurately represented through a linear transformation of a different joint rotation (redundant).

Our procedure is based on the following rules:

1. If the range of a joint rotation is smaller than a specified threshold, we replace the angular values of each frame with the mean value of the whole motion.
2. If the root mean square deviation of two joint rotations is less than a threshold, the curve a with the smaller standard deviation is created based on the curve b with the larger standard deviation in the following way: $a = (b - mean(b))/std(b) * std(a) + mean(a)$.
3. If the root mean square deviation of one joint rotation and of the inverse of a second joint rotation is less than a threshold, the curve a with the smaller standard deviation is created based on the curve b with the larger standard deviation in the following way: $a = (mean(b) - b) = std(b) * std(a) + mean(a)$.

4.1. Find irrelevance

First, we calculate the range, the mean, and the standard deviation of the values of the frames for each curve. Figure 4 shows the distribution of the ranges of the rotation curves. As can be seen in the histogram, the majority of the ranges are small. Out of 50 joint rotations, 19 have a range of less than 5; for 11 coordinates the range is even less than 1. As a rotation of 1° is barely visible, we can delete those rotations without losing too much information. We replace the curves of the 11 joints that have a range of less than 1 with a constant value: the mean of the curve. A rotation of 5° is visible, but it does not change the meaning of the gesture. In a later step, we replace the coordinates that have a range of less than 5 in order to reduce the degrees of freedom of the hands even further.

Figures 2 and 5 show three possible scatter plots in each of which two curves are plotted against each other. We can see that a line with either a positive or negative slope is a good approximation for the relationship between some pairs of rotation curves.

To find which curves are most strongly correlated, we use

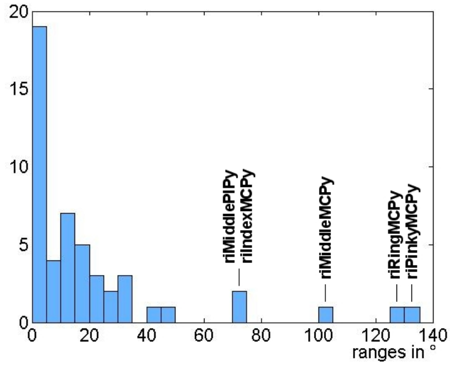

Figure 4: *Distribution of the ranges of the 50 rotation curves.*

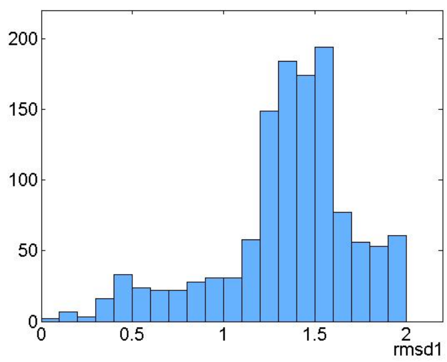

Figure 6: *Distribution of the 1225 values for $rmsd_1$, the distance metric for linear relations with a positive slope.*

the RMS deviation as a distance metric. First, we standardise each rotation curve by subtracting the mean from the value at every frame and dividing by the standard deviation, resulting in a set of curves that all have mean 0 and standard deviation 1.

To detect values that are highly positively correlated, we calculate the square deviations of the corresponding points of two curves and compute the root of the mean of those deviations. Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ denote two curves with n the number of frames. Then our distance metric is calculated as:

$$rmsd_1(a,b) = \sqrt{\frac{\sum_{i=1}^{n}(a_i - b_i)^2}{n}}$$

We repeat this process for every pair of curves, resulting in $50 * 49/2 = 1225$ values. Figure 6 shows the distribution of those values. The smaller the value of this distance metric, the more accurately one curve can be estimated as a linear transformation of the other curve, the slope of the transformation being positive.

For values that are highly negatively correlated, we adapt the distance metric to calculate the RMS deviation of one curve with the second curve mirrored across the x-axis.

$$rmsd_2(a,b) = \sqrt{\frac{\sum_{i=1}^{n}(a_i + b_i)^2}{n}}$$

Again, we compute this metric for every set of two rotations. Figure 7 graphs the distribution of the 1225 resulting values. Analogous to $rmsd_1$, the smaller the value of this distance metric, the better one curve can be estimated based on the other one. In this case the slope of the linear transformation is negative.

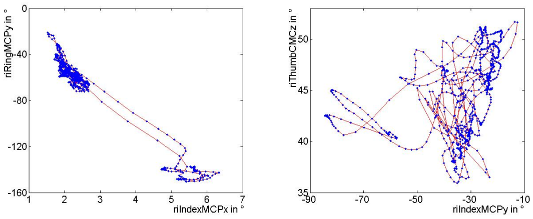

Figure 5: *Two scatter plots of pairs of rotation curves. Each point represents a frame; successive frames are connected by a line.* ***Left****, the relationship between riIndexMCPx and riRingMCPy can be approximated by a line with negative slope.* ***Right****, a line would not be an accurate approximation for the relationship of riIndexMCPy and riThumbCMCz.*

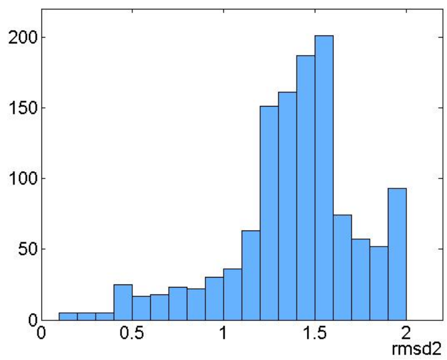

Figure 7: *Distribution of the 1225 values for $rmsd_2$, the distance metric for linear relations with a negative slope.*

4.2. Simplify curve using rmsds

When the distance metric is less than a chosen threshold, we compute the curve with the smaller standard deviation as a linear combination of the curve with the larger standard deviation using the following equations for $rmsd_1$ and $rmsd_2$, respectively.

$$a_i = \frac{b_i - mean(b)}{std(b)} * std(a) + mean(a)$$

$$a_i = \frac{mean(b) - b_i}{std(b)} * std(a) + mean(a)$$

This computation results in curves having the same mean and the same standard deviation as the original curve.

Figures 8 and 9 show an original rotation curve and the same rotation computed as a linear combination of a different curve.

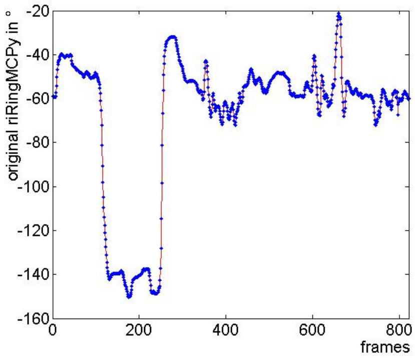

Figure 8: *Original rotation curve riRingMCPy.*

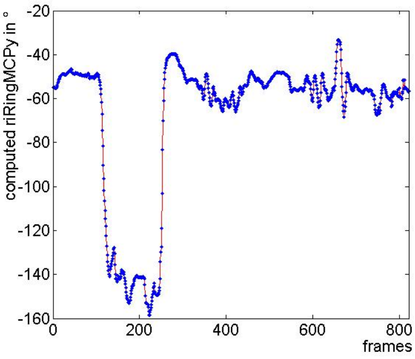

Figure 9: *Rotation riRingMCPy computed as a linear combination of the curve riPinkyMCPy.*

5. Results

In this study we present four steps of simplification based on the following criteria:

1. range < 1
2. $\text{rmsd}_1 < 0.3$ or $\text{rmsd}_2 < 0.3$
3. range < 5
4. $\text{rmsd}_1 < 0.5$ or $\text{rmsd}_2 < 0.5$

Step 1 and 3 affect rotation curves with small ranges, step 2 and 4 are based on the proposed distance metric.

5.1. Step 1: range<1

The 11 rotation curves that have a range of less than 1, as well as their ranges and means, are listed in Table 1.

curve	range	mean
leIndexMCPx	0.20	0.5
leIndexMCPz	0.42	-3.8
leIndexDIPy	0.12	11.9
leMiddleMCPx	0.32	-0.7
leMiddleMCPz	0.95	-6.3
leMiddleDIPy	0.007	13.9
leRingDIPy	0.001	15.8
lePinkyDIPy	0.94	-0.5
riIndexDIPy	0.08	-8.0
riMiddleDIPy	0.08	-11.6
riPinkyDIPy	0.31	-16.4

Table 1: *Rotation curves with a range of less than 1.*

Each curve is replaced by a constant value: its mean. The listed rotation curves either describe the flexion (y-axis) of the DIP-joints or represent adduction, abduction (x-axis) or rotation motions (z-axis) of the left hand. These joint rotations are expected to be small. The left hand is moving less than the right hand in the analysed animation sequence, which explains why the majority of the simplifications affect the left hand.

5.2. Step 2: rmsd< 0.3

Once the curves with a range less than 1 are excluded, 11 $rmsd_1$ values and 10 $rmsd_2$ values are less than 0.3. However, within these relationships, there are overlaps, e.g. if curve a is controlled by curve b, but curve b is already controlled by curve c. In this case we compute curve a based on curve c, even if the distance metric exceeds the threshold (this happens for only two joint rotations in Step 4). If the distance metric of curve a and curve b as well as the distance metric between curve a and curve c are less than 0.3, curve a could be controlled by curve b or curve c. We then calculate curve a based on the curve with the larger standard deviation. Table 2 lists the remaining relationships after simplification according to the above criteria.

dependent curve	controlling curve	rmsd
leRingMCPx	leRingMCPy	$\mathrm{rmsd}_2 = 0.25$
leRingMCPz	leRingMCPy	$\mathrm{rmsd}_2 = 0.19$
lePinkyMCPx	lePinkyMCPy	$\mathrm{rmsd}_2 = 0.18$
lePinkyMCPz	lePinkyMCPy	$\mathrm{rmsd}_2 = 0.21$
riThumbMCPz	riThumbIPz	$\mathrm{rmsd}_1 = 0.17$
riIndexMCPx	riPinkyMCPy	$\mathrm{rmsd}_2 = 0.18$
riMiddleMCPy	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.19$
riRingMCPz	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.13$
riRingDIPy	riRingPIPy	$\mathrm{rmsd}_1 = 0.29$
riRingMCPx	riPinkyMCPz	$\mathrm{rmsd}_2 = 0.22$
riRingMCPy	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.13$
riPinkyMCPx	riPinkyMCPz	$\mathrm{rmsd}_2 = 0.25$

Table 2: *Relationships of rotation curves when all root mean square deviations of less than 0.3 are taken into account.*

These 12 simplifications – one is represented in Figure 9 – cover a diverse set of joint rotations. We are therefore able to retain variation within the motion.

5.3. Step 3: range<5

Nine rotation curves have a range between 1 and 5 and are therefore simplified in this step, as shown in Table 3.

curve	range	mean
leRingMCPx	1.7	-1.8
leRingMCPz	2.2	-15.8
leThumbCMCx	4.4	45.5
riIndexMCPx	4.8	1.2
riMiddleMCPx	4.7	-49.5
riRingMCPx	3.9	-0.3
riRingDIPy	1.5	-11.8
riPinkyPIPy	4.4	-12.2

Table 3: *Rotation curves with a range between 1 and 5.*

Similar to Step 1, these values continue to reduce the flexion (y-axis) of the DIP-joints and adduction, abduction (x-axis) and rotation values (z-axis), which all have low ranges in natural human hand motion. An exception is the PIP-flexion of the right little finger. The small range of this joint rotation is probably due to approximation within the model that does not take the motion of the fifth carpometacarpal joint, which allows the bending of the palm on the side of the little finger, into account. This simplification causes an increase in the range of the MCP-joint of the little finger and a decrease of the range of the PIP-joint.

5.4. Step 4: $\mathrm{rmsd}_1 < 0.5$ or $\mathrm{rmsd}_2 < 0.5$

In a fourth step, we choose all pairs with *rmsd* < 0.5. Table 4, which lists the added or changed relationships, shows that, after the computation, most joint rotations only depend on a few original rotations:

dependent curve	controlling curve	rmsd
leThumbCMCz	leThumbCMCy	$\mathrm{rmsd}_2 = 0.46$
leIndexPIPy	leMiddlePIPy	$\mathrm{rmsd}_1 = 0.50$
leRingMCPy	leMiddleMCPy	$\mathrm{rmsd}_1 = 0.39$
leRingPIPy	leMiddlePIPy	$\mathrm{rmsd}_1 = 0.45$
riThumbMCPz	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.51$
riThumbIPz	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.46$
riIndexMCPy	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.49$
riMiddlePIPy	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.35$
riRingPIPy	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.63$
riPinkyMCPz	riPinkyMCPy	$\mathrm{rmsd}_1 = 0.39$
riPinkyMCPx	riPinkyMCPy	$\mathrm{rmsd}_2 = 0.56$

Table 4: *Relationships of rotation curves when all root mean square deviations of less than 0.5 are taken into account.*

5.5. Summary

Within four different stages, we continually reduce the degrees of freedom of the hand. The amount of remaining DOFs are listed in Table 5.

condition	DOFs left
original	50
range< 1	39
range< 1 + rmsd< 0.3	27
range< 5	31
range< 5 + rmsd< 0.5	15

Table 5: *Remaining degrees of freedom after each simplification step.*

From the original 50 DOFs for both hands we reduced the motions to 15 DOFs, which shows the potential of our approach.

6. Discussion

Our results show that there are no strong correlations between the motions of the right hand and the motions of the left hand. In both hands, many adduction/abduction or rotation curves of the MCP-joints are reduced. This is not surprising as the range of those rotations is small in natural human hand motion.

For the right hand, numerous joints' flexions of the middle, ring and little finger are created based on the flexion of the MCP-joint of the little finger, which has the largest standard deviation. This seems plausible: Firstly, the range of motion of the flexion of the MCP-joints increases from the index to the little finger in a human hand. Secondly, the chosen hand model is likely to further increase this difference. The CMC-joint of the thumb is completely independent from the other digits and keeps its three degrees of freedom. Further curves that are independent are two rotation curves of the index as well as the *riMiddleMCPz* rotation. A closer look at the latter shows that the range of this joint is 7.1 only because of a few outliers in fewer than 10 frames. Without these outliers, the range would be less than 4.0 and the curve would have been simplified in the third step.

The correlations are less explicit for the left hand, which keeps more variation in the motion. The flexions of the fingers are partly based on the middle finger's PIP-joint, and partly on the middle finger's MP-joint. Similar to the right hand, the motion of the thumb is independent, keeping three DOFs. Furthermore, the rotations *leIndexMCPy*, *lePinkyMCPy*, and *lePinkyPIPy* are kept unchanged. The rotation curves that become simplified are similar for the left and the right hand, but not exactly the same. We therefore deduce that our findings may vary slightly but not substantially depending on the handedness of the actor.

An alternative approach to reduce the dimensions would be to compute correlation coefficients of pairs of curves as a distance metric, which leads to very similar results. More sophisticated techniques such as PCA may provide interesting results, though they are likely to yield an unintuitive set of basis poses. Nevertheless, more complex models that retain basic rotations could include relationships with more than one joint rotation and (self-)relationships within time.

If oversimplified, for example when nearly all of the rotation curves are generated based on just one curve, the motion could seem artificial. Using a larger variety of relationships would add more variation to the resulting motions. Furthermore, it is possible to add plausible variation into the rotations by adding noise.

The initial model of the hand, which influences the calculations of the rotation angles, is a simplification of the complex structure of the real hand. Even though the chosen simplifications are common in character animation, they alter the real motions of the hand. Simplifications compared to a real human hand occur, especially at the palm, and influence the motions of the little finger and the thumb. A more accurate model would allow for a more exact analysis.

7. Conclusions and Future Work

We analysed conversational hand motions to identify barely noticeable movements and rotations that can be accurately computed as a linear relation of each other. Based on our findings, we reduce the dimensions of the motions of digits while keeping an intuitive base of rotations. The transformations are simple and do not require any elaborate computations.

In summary, we advise an animator or a person capturing motions to concentrate on the thumb and on the MCP flexions of the index and one additional finger, such as the middle or pinky, as their rotation curves convey the most essential information of the motion. In future work, we want to demonstrate the validity of our proposed techniques with a wider range of motions and actors.

8. Acknowledgements

We would like to thank Disney Research Pittsburgh and the graphics lab at Carnegie Mellon University, in particular Jessica Hodgins, Moshe Mahler, and Justin Macey, for providing the facilities and assistance in creating the motion data. This study has been supported by the Embark Postgraduate Scholarship Scheme of the Irish Research Council for Science, Engineering, and Technology (IRCSET).

References

[BSP*04] BARBIČ J., SAFONOVA A., PAN J.-Y., FALOUTSOS C., HODGINS J. K., POLLARD N. S.: Segmenting motion capture data into distinct behaviors. In *GI '04: Proceedings of Graphics Interface 2004* (2004), pp. 185–194. 2

[BZ04] BRAIDO P., ZHANG X.: Quantitative analysis of finger motion coordination in hand manipulative and gestic acts. *Human Movement Science 22*, 6 (2004), 661–678. 2

[CGA07] CIOCARLIE M., GOLDFEDER C., ALLEN P.: Dimensionality reduction for hand-independent dexterous robotic grasping. In *Intelligent Robots and Systems, 2007* (2007), pp. 3270–3275. 2

[GM08] GIBET S., MARTEAU P.-F.: Analysis of human motion, based on the reduction of multidimensional captured data – application to hand gesture compression, segmentation and synthesis. In *Articulated Motion and Deformable Objects* (2008), vol. 5098/2008, pp. 72–81. 2

[KNKA07] KIPP M., NEFF M., KIPP K. H., ALBRECHT I.: Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In *IVA '07: Proceedings of the 7th international conference on Intelligent Virtual Agents* (2007), pp. 15–28. 2

[MZF06] MAJKOWSKA A., ZORDAN V. B., FALOUTSOS P.: Automatic splicing for hand and body animations. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 309–316. 2

[OFH08] ONUMA K., FALOUTSOS C., HODGINS J. K.: FMDistance: A fast and effective distance function for motion capture. In *Eurographics Short Papers* (2008). 2

[PFS06] PALASTANGA N., FIELD D., SOAMES R.: *Anatomy and Human Movement – Structure and Function*, 5 ed. New York: Basic Books, 2006. 3

[SDO*04] STONE M., DECARLO D., OH I., RODRIGUEZ C., STERE A., LEES A., BREGLER C.: Speaking with hands: creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (SIGGRAPH '04) 23*, 3 (2004), 506–513. 2

[SFS98] SANTELLO M., FLANDERS M., SOECHTING J. F.: Postural hand synergies for tool use. *The Journal of Neuroscience 18*, 23 (1998), 10105–10115. 2

[VGK02] VLACHOS M., GUNOPOULOS D., KOLLIOS G.: Discovering similar multidimensional trajectories. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering* (2002), IEEE Computer Society, p. 673. 2