Authoring rules for bodily interaction: From example clips to continuous motions

Klaus Förger, Tapio Takala, and Roberto Pugliese

Department of Media Technology,
School of Science,
Aalto University,
Espoo, Finland
{klaus.forger,tapio.takala,roberto.pugliese}@aalto.fi

Abstract. We explore motion capture as a means for generating expressive bodily interaction between humans and virtual characters. Recorded interactions between humans are used as examples from which rules are formed that control reactions of a virtual character to human actions. The author of the rules selects segments considered important and features that best describe the desired interaction. These features are motion descriptors that can be calculated in real-time such as quantity of motion or distance between the interacting characters. The rules are authored as mappings from observed descriptors of a human to the desired descriptors of the responding virtual character. Our method enables a straightforward process of authoring continuous and natural interaction. It can be used in games and interactive animations to produce dramatic and emotional effects. Our approach requires less example motions than previous machine learning methods and enables manual editing of the produced interaction rules.

Keywords: animation, motion capture, bodily interaction, continuous interaction, authoring behavior

1 Introduction

Virtual characters are common in modern games and their bodily motions can reflect the emotions and attitudes between characters. Real-time motion synthesis and synchronization with external stimuli enables making the characters interactive. The possibilities for using bodily interaction have increased as even consumer level sensor technology allows capturing bodily motions. Sometimes a motion does not mean much out of context, but can have a lot of meaning if displayed as a synchronized reaction to an action [1]. A good example is a virtual character taking a step backwards in isolation versus taking a step backwards as a reaction to aggressive behavior of another character.

Expressive motion based interaction between two virtual characters is possible with a library of recorded and annotated motions. For example, if a character moves in a way that was annotated as angry, then another character could react by selecting a motion that was annotated as scared. Similar interaction between a human and a virtual character requires ability to evaluate the style of previously unseen motions in real-time. Only part of the emotional content in a motion library can be annotated in advance as it can vary depending on the context of the motion.

In this paper we explore how motion captured examples of human interaction can be used in authoring interaction rules for virtual characters. These rules allow real-time generation of expressive behaviours in a continuous interaction loop. The idea is that there should be no frozen pauses during an interaction sequence as can happen in task-based approaches, but instead all idle moments could be used to reflect the attitudes of the participants. Continuous interaction scheme could be also used for subtle control over the style of motion. For example a walking character could immediately react to an observed aggressive action by changing the style of the walk from neutral to careful. The amount of visible aggression could be continuously mapped to the amount of carefulness.

We concentrate on the case where humans and virtual characters have equal amount of information from each other. The virtual characters observe humans through features that characterize different qualities of human motion. The features, from now on referred as motion descriptors, can be calculated in real-time. We show how using example motion pairs makes authoring of interaction rules a straightforward process. The example motions also help avoiding impossible combinations of motions descriptors. Moreover, we show that selective use of motion descriptors allows solving the curse of dimensionality, that arises from modeling human motion simultaneously with several descriptors.

We next present related work, and then describe our implementation in three parts. First is the calculation of motion descriptors. The second is the interaction rule authoring where observed descriptors are mapped to descriptors of desired reactive behaviours. The last part is motion synthesis that turns the descriptors to actual motions of a virtual character. In the fourth section, we present a use case of the process of authoring behaviours. The last sections contain discussion, conclusions and future work.

2 Related work

Real-time motion synthesis can be done by creating a motion graph from captured motions and playing one motion segment after another according to the graph [2]. Furthermore, if the captured motions are annotated, it is possible to control the motion synthesis [3]. This can happen by selecting motion segments from the graph that correspond to attributes used in the annotation. We use motion synthesis that is based on a motion graph. Our graph includes information about the motion style that is used in controlling the motion synthesis. Therefore, it is similar to the metadata motion graphs, which have been used for synchronizing human motion with beats in streaming music [4].

Manual annotation of motion can be very time consuming. Therefore automatically calculated motion descriptors for human motion have been developed

[5, 6]. The descriptors can measure for example the amount of motion, acceleration or qualities of the pose of a character. Similar values have been also calculated from the relational motion of hands representing two entities such as small animals [7]. We use motion descriptors for annotation of recorded motion in a motion graph and real-time motions. We also extend the relative descriptors from relations between hands to the case of relations between two human characters.

Earlier systems that allowed interacting with virtual creatures were often targeted at goal oriented interaction [8] or interaction scripted with if-else clauses [9]. A more fluid model of interaction was allowed by a probabilistic method that uses pairs of recorded actions and reactions to learn how to react to human movements [10]. A similar system based on example motions has been used for teaching cleaning robots socially acceptable motion styles [11]. Our method for defining interaction fits in between these older methods as it takes advantage of example action-reaction motions and manual definitions.

The importance of usability has been noted in earlier works that present tools for authoring behaviours of virtual creatures [12, 13]. These tools assume that a range of low level behaviours such as wandering, following and actions that display emotional states are available. The tools allow applying the low level behaviours to crowds and joining them together to form more complex patterns. Our method could be used for authoring the said low level behaviours. One requirement for the earlier tools has been that using them should not require coding experience or understanding complex models that govern the behaviours. This requirement is taken into account in the design of our method.

Publications related to virtual characters include works on embodied conversational agents (ECAs) [1]. Considerable effort has been taken towards a unified Behaviour Markup Language (BML) that can be used when creating ECAs [14]. These works mainly view bodily motion as a way to make verbal conversations more believable. In this paper, we consider varied situations beyond conversations, and study non-verbal interaction where bodily motion is the only channel of communication. We also extend the scope of behaviours from friendly and believable characters to ones that could be considered anti-social and annoying as those can be required in for example games containing dramatic sequences.

A proposal has been made to extend the BML from describing the behavior of a single virtual human to the case of continuous interaction between two characters [15]. Continuous interaction is a core aspect of our paper, but we have a different point of view on what part of motion we want to control. The proposal suggests developing an XML based approach that could be useful in defining and controlling discrete reactions and gestures. However, we concentrate on motion style that we abstract with motion descriptors which have a continuous range of values that vary from frame to frame. For this reason we use a more continuous control mechanism.

Our work builds on earlier work about defining interaction rules using motion descriptors [16]. The earlier system showed that Radial Basis Functions (RBF) [17] can be used to map the input motion descriptors to output descriptors. The output descriptors where then used to control a motion synthesis engine. These parts created a virtual character that reacts to observed human motions. The system used only two input and output descriptors and therefore had limited capability to create interesting interaction. In this paper, we show that using more than two descriptors allows much more varied interaction. However, it also introduces the curse of dimensionality as the number of the combinations of the descriptors grows exponentially compared to the number used descriptors. That in turn forced us to find new ways to create the interaction rules.

3 Implementation

Our system is based on an interaction loop where two characters can observe each other and react to the actions they witness. The steps from the observed motions to the synthesized motion are shown in Figure 1. First, the values of the input descriptors are calculated from the observed motion of another character (or human) and from the character's own motion. Secondly, the input descriptor values are mapped according to the interaction rules into the desired output descriptor values. Then the motion synthesis engine creates a motion that fits the output descriptor values as well as possible. Next, the newly synthesized motion can be observed by another virtual character or by a human.

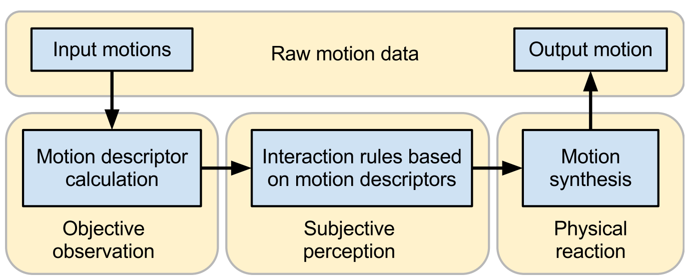

Fig. 1. Model of tasks performed by an interactive virtual character system and the flow of information in the system.

3.1 Motion descriptors

In our implementation, a motion descriptor is a function that takes as input the observed 3D motion data and calculates a value between zero and one for each frame of the motion. Ideally, the descriptors would tell all about the motion style of an action and ignore unimportant aspects. In practice, the descriptors are limited to what can be easily defined mathematically and calculated in real-time. The descriptors act as objective measures that are not affected by any internal states of the characters.

In this work we concentrate on behaviours that include standing, walking, jumping and generally moving around on a flat floor. For these types of motions, relevant motion descriptors for an isolated character include Quantity of Motion (QoM) [6], turning left/right and moving backwards/forwards. Examples of motions corresponding to high and low values of the descriptors are shown in Figure 2. The QoM estimates the total amount of energy in the motion and is calculated as the sum of instantaneous velocities of all body parts. We also use a variant of QoM called non-transitional QoM (NtQoM) that estimates the energy used only for body language or other expressive motions, disregarding locomotion.

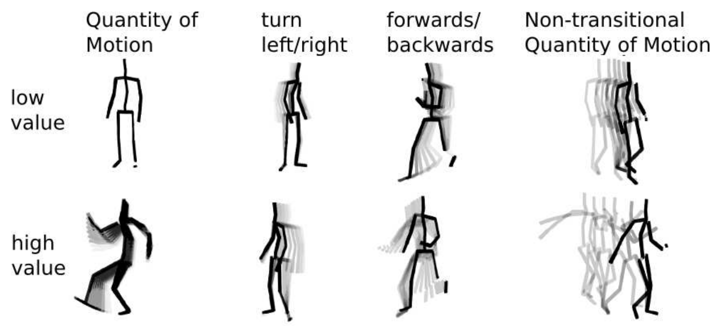

Fig. 2. Examples of motion descriptors calculated from an isolated character.

Another class of descriptors are relational descriptors, i.e. those that compare motion of two characters. Of these we use the distance between the characters, their facing angle, and approach/retreat as illustrated in Figure 3. The facing angle is normalized to be zero for face-to-face characters and one when a character's back is turned towards the other. The values of approach/retreat are linearly relational to the velocity along the direction to the other character. The extreme values of approach/retreat are normalized in order to get zero when the character is moving towards the other character at 7 m/s and one when moving away at the same speed. The value 7 m/s is reasonable limit for bodily interaction as a forceful jump without a run can have approximately that velocity. All the other descriptors are also normalized in a similar manner. For the distance there is only one common value for both characters, but the characters have their own values for the facing angle and approach/retreat.

We used a commercial motion tracking system (NaturalPoint OptiTrack) with 24 cameras working at 100 frames per second to capture the motions from which we calculated the descriptors. We were also able to calculate all the used descriptors from the data Microsoft Kinect provides using OpenNI library. However, the lower accuracy and higher amount of errors in the Kinect data made

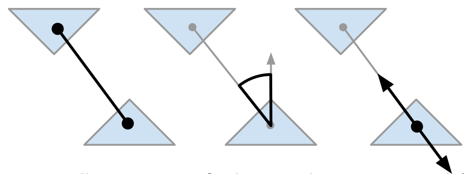

Fig. 3. Relational motion descriptors between two characters.

using it impractical. Especially calculating QoM from the noisy data is unreliable.

While the motion descriptors estimate different aspects of the motion, they are not independent from each other. Some dependencies are direct. For example, if QoM is zero, it limits all other descriptor values to those corresponding to no movement. Another type of dependency is dynamic. For example, between QoM and facing angle all value combinations are physically possible, but changing facing angle becomes impossible if the QoM remains zero. Because of these physical constraints, it is useful to pick descriptor values from recorded motions. A randomly selected set of descriptor values might be impossible to synthesize into motion of a virtual character.

3.2 Interaction rule authoring

The interaction rules define the reactions of a virtual character to observed motions. The reactions can vary depending on the own position and motion of the virtual character. A rule can reflect physical properties such as being strong or weak. Also, the mental state of the character, such as sadness or aggressiveness, can be built in the behavior produced with a rule.

In practice, the rules are used for projecting a frame of an observed motion to a desired reaction. Here we consider the frame to include positional information and instantaneous velocities of body parts. Our method uses motion descriptors to abstract the frames of input and output motions. In an earlier publication the mappings were created with manually defined example point pairs [16]. A mapping consisted of one point in the input corresponding to one point in the output space. After the mappings for the rule were defined, projecting a point from the input to the output was done with Radial Basis Functions (RBF), that is a sparse data interpolation method [17]. This process of creating a rule required filling the input space evenly with points along every dimension.

The standard RBFs approximate a function from a high dimensional space to a single dimension using example points were the output value is predefined [17]. In practice, if the point that is being projected is close to only one of the example points, the output will have the value that is linked to that example point. Should the point be in the middle of two example points, the output would be an average of the linked output values weighted by the inverse of distances to those example points. The case of projecting from a high dimensional space to another high dimensional space with RBFs requires only repeating the standard case for each of the output dimensions [17]. The RBFs were chosen as the interpolation method as it is a simple approach to code and cheap to calculate.

The projection using point pairs and RBFs works well up to two descriptor dimensions, but faces the curse of dimensionality in the combinatorial sense with a higher number of dimensions. This means that the amount of point pairs required to cover the input space grows exponentially with the number of dimensions. High dimensionality also hinders visualizing and editing points as three spatial dimensions is the limit on human vision. Growing dimensionality also makes motion synthesis harder as each descriptor dimension sets new requirements for the produced motions.

When experimenting with high dimensional interaction rules, we observed that creating the mappings rarely requires using more than three dimensions simultaneously. In fact, many interesting and useful mappings require only using one or two dimensions, but the set of required dimensions varies between mappings. This observation lead us to the conclusion that the problems caused by a high dimensionality could be solved by allowing the author of the rules to select which dimensions are relevant to each of the mappings individually.

Mathematically, our solution requires pairing each set of descriptor values with a scaling vector and a modification to the projection done with RBFs. The scaling vector indicates how important the related descriptor dimensions are. The input data we need to consider includes the point we want to project p and the example point pairs indexed as $[1...k...K]$. A point pair consists of an input point i_k and the related input scaling vector s_k, an output point g_k and the related output scaling vector u_k. The data we want to calculate is the output descriptor values o and the output scaling vector h. During real-time interaction p is the observed motion, o is the desired reaction and h tells how the output descriptors should be prioritized in the motion synthesis.

Next, we go through the changes needed in the standard case of RBF. Let us consider an input space with dimensions indexed as $[1...n...N]$ and an output space with dimensions indexed as $[1...m...M]$. The scaling vector s_k of an example input point i_k affects the calculation of the distance d_k between the point p, that is being projected, and the point i_k as follows:

$$d_k = \left\| (i_k - p) \begin{bmatrix} s_{k_1} & 0 & \dots & 0 \\ 0 & s_{k_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_{k_N} \end{bmatrix} \right\| . \tag{1}$$

If the scaling vector s_k is all ones, then the distance calculation returns to the unmodified case. If one of the values in the scaling vector is zero, then that input descriptor dimension is effectively ignored by the mapping. The output scaling vectors $u_{1...K}$, which are paired with example output descriptor values $g_{1...K}$, enable the creation of mappings that can be independent also on the output side of the projection. In practice, this means that the influence of a mapping (a point pair) can be limited to only part of the output descriptors $o_{1...M}$. The values for the example output scaling vectors $u_{1...K}$ and the example output points $g_{1...K}$ of the mappings are used in the RBF interpolation resulting in the output descriptor values $o_{1...M}$ as follows:

$$o_m = \frac{\sum_k (g_{k_m} \cdot u_{k_m} \cdot (1 - d_k))}{\sum_k (u_{k_m} \cdot (1 - d_k))} \ . \quad (2)$$

Finally, the values $h_{1...M}$ of output scaling vector that forms a pair with output descriptor values $o_{1...M}$ is:

$$h_m = \sum_k (u_{k_m} \cdot (1 - d_k)) \quad (3)$$

As the values for descriptors and the scaling vectors are limited to the range from zero to one we also limit all the values in the calculations to the same range. The o and the h are given as input to the motion synthesis engine. The descriptor values o control type and style of synthesized motion and scaling vector h tells how the output descriptors should be prioritized and which can be ignored.

The practical work that the author of the rules must do when creating a mapping for a rule includes defining the values for input and output descriptors and scaling vectors. This can be done manually with the sliders of a graphical user interface shown in Figure 4 (E, F). However, we have found that it is not always easy to see which movement would correspond to given descriptor values or vice versa. This problem can be overcome by capturing an example action and reaction and selecting the descriptor values from them with the user interface. The time line/frame counter (fig. 4 D, upper slider) can be used to simultaneously browse through the values (fig. 4 B, E, F) and view an animation (fig. 4 A) of the example motions.

In example motions the action and the reaction do not always happen at exactly the same moment. Therefore, there can be a need to scroll the reaction forward in order to find the right descriptor values for it. This can be done with the lower slider in the user interface in Figure 4 (D). This offsetting is necessary especially for the relational descriptors that have only one value such as the distance between the characters. Without offsetting, those descriptors would by definition have the same value for both input and output.

Picking descriptor values from example motions helps the process of authoring rules, but picking the values for scaling vectors cannot be done in the same way especially in a high dimensional case. If more than one example pair of motions displaying the same interaction would be available, then some of the scaling values could be estimated based on the correlations in the examples. However, this would add much work in capturing the examples. For this reason the author of the rules needs to have a vision of the intended interaction that guides the selection of the scaling values.

Compared with the old method of creating rules [16] the new method allows rules to be made with much less mappings as the scaling can be used to ignore those descriptor dimensions which are not relevant. When using the old method,

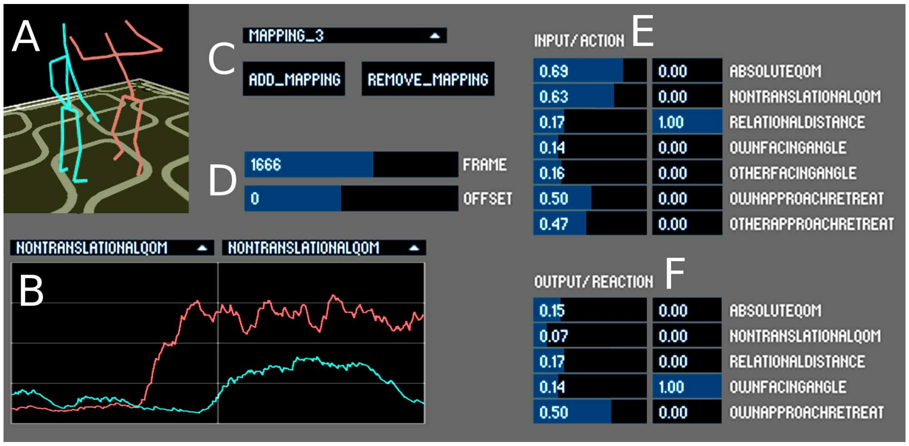

Fig. 4. A graphical user interface for authoring interaction rules that includes animation of the example motions (A), view of descriptor values over time (B), selection of mappings (C), sliders for the animation time and offset between motions (D), input/output descriptors (usually picked from animation) (left side E, F) and input/output scaling (manually defined) (right side E, F).

it would have been necessary to define mappings for all the combinations of descriptor dimensions, even those that should not affect the end result.

3.3 Motion synthesis

During real-time interaction, the motion synthesis engine takes the desired output descriptor values and creates a continuous motion following the values as closely possible. We use a motion graph based on recorded motions for the synthesis. All the motions in the motion graph are annotated using the motion descriptors. After this we can synthesize new motions by concatenating motion clips that fit well to the desired descriptor values.

The motion graph we used contains a motion library divided into motion clips and the possible transitions between the motion clips. The motion clips are half to two seconds long samples from a nine minutes recording. The recordings contained motions that are required for moving on a flat surface (standing, turning, walking, running) and a few expressive motions (waving hands, jumping). The motions were recorded many times with differing styles to get versions with both high and low QoM similarly as shown in Figure 2.

We create the motion graph by finding all transitions from a frame to another where the pose and velocities do not differ too much with the restriction that the transitions are at least half a second apart. We do not prune any transitions from the motion graph as it would increase the reaction time of the virtual character and reduce the amount of possible reactions. During real-time interaction, we evaluate as many motion clip sequences as is possible during half a second, usually a few tens of thousands, and then select the sequence of clips that matches the desired descriptors best. The high number of possible clip sequences makes motions synthesis the most computing intensive part of the whole system.

One challenge here is that the desired values might require actions that cannot be performed simultaneously. For that problem, the scaling vector of the output values (Equation 3) helps as it tells which descriptors need to be prioritized and which can be ignored. The constraints set by the human body and physics offer another challenge as they should not be broken when natural looking motions are desired. The concatenative synthesis we use always produces physically plausible motions, but allows only approximate following of the desired descriptors.

During real-time interaction, selecting the next motion clips requires searching for the sequence of clips where the deviation from the desired motion descriptors is minimal. For the descriptors that can be calculated from an isolated character, this can be done by using the descriptor values that were calculated when preparing the motion graph. The relative descriptors cannot be calculated in advance as they vary depending on the other character. Also, the relative descriptors cannot be used directly in the search as the virtual character can only decide its own motion. Therefore, the relational values for distance, facing angle and approach/retreat are transformed into position, direction of movement and facing in the absolute coordinate system. These values can then be used as parameters to be optimized in the search for the next motion clip.

4 Example cases of authoring behaviours

The simplest type of examples of authored behavior contains only one mapping between the input and the output descriptors. Let us consider a character that turns its face to another character. For this behaviour, the input should have all the scaling values of the descriptors set to zero and the output facing angle with value zero and weight one. During real-time interaction, having only this mapping would set the desired facing angle to zero. All the other descriptors would be ignored by the motion synthesis engine as they would have zero scaling values. A character following this rule would create an impression that it is aware of the position of the other character, but it would ignore all other aspects of the motion.

A more interesting character could be one that acknowledges the other character by turning to face it, but would get offended and turn away if the other character misbehaves. To create this behavior, an example pair of an action and a reaction shown in Figure 5 can be used. The behaviour rule can be created by scrolling through the action and reaction and by creating mappings from all the significant parts of the motion pair.

The motions start by having the characters far away form each other (fig. 5 A). A desired reaction in this case could be to ignore the far away character. This can be turned into a mapping where the input has the distance between the characters with weight one and the output has all the descriptors weighted zero.

In the next part of the motions, the characters have come near each other and the reaction character has turned to face the action character (fig. 5 B). This can be turned into a mapping that has a low distance and low non-transitional QoM on the input side and low facing angle on the output side. The last part of motion has the action character waving hands forcefully and the reaction character responds by turning away (fig. 5 C). This part corresponds to a mapping where the input side has a low distance and high non-transitional QoM and the output has a high facing angle.

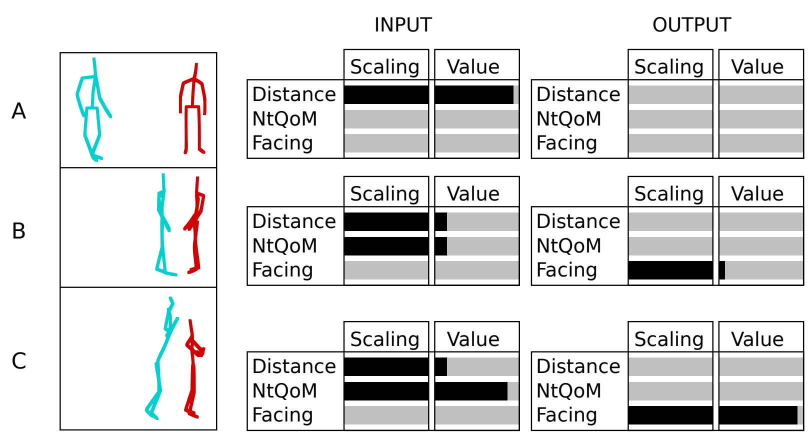

Fig. 5. Significant frames (A-C) from an example action (on the left side) and reaction (on the right side) motions, descriptor values picked from those parts of the motions and the scaling values decided by the author of the rule.

After the mappings are defined, the rule is ready to be tested. The testing can happen by seeing if changing the values of input descriptors produce sensible output values. This can show if any errors were made while defining the rules. However, testing the rules with real-time interaction is also important as it can show if there are problems in the synthesis of the output descriptor values. The synthesis can fail if the virtual character is not able to find any possible motions that would fit the output descriptors quickly enough. This can be a problem especially if the motion graph has long motions that cannot be interrupted. Another possible problem is that an action can be so short that the input descriptors show the action for too short time to cause a reaction. This calls for more careful descriptor design and possibly descriptors that are calculated as an average over a period of time instead of just individual frames of motion.

Variations to the presented example behaviour could be that instead of turning away when provoked the character would start to be aggressive. The roles could be also swapped and then the virtual character would be the one starting the provocation.

5 Discussion

The example case shows that using recorded action and reaction motions guides the work flow of authoring interaction rules. The rules can be authored and tested with graphical and bodily user interfaces. Therefore, the requirements for the author creating the interaction rules do not include coding experience or learning an XML dialect. To further develop the usability of the system, user tests should be done with the users authoring new rules and interacting with virtual characters following those rules.

The used reactive interaction rules work in real-time and they are good for interactive background characters. Characters that need more intelligence could be built by adding reasoning capabilities and internal state into the virtual character and selecting the interaction rules based on the internal state. However, the interaction rules alone are not enough as the believability of the authored behaviours is heavily dependent on the capabilities of the motion synthesis engine.

It is challenging to synthesize motions that are realistic and balance the sometimes conflicting demands of expressiveness in real-time. The used motion graph approach is not ideal as it only allows playing motions a clip at a time, while perfect following of the desired output descriptors would require a more continuous synthesis method with a shorter reaction time. The situation could be helped by real-time filtering of the produced motion or using physics based motion synthesis when a sudden reaction is needed.

In other than research applications, using only bodily motions is not enough for creating a complete virtual character. We feel that modalities such as facial expressions and tone of voice could be added to the current authoring system. Also, defining musical interaction could be possible. The main requirement is that it must be possible to create continuous signals to describe the medium and to drive a synthesis engine with those signals. Combining the continuous interaction with discrete gestures could be more challenging. That would require deciding whether the continuous changes in the motion style only modulate the gestures or could they also interrupt or prevent the gestures.

One shortcoming in the presented approach is that the author of the behaviours has to assume the connections between the descriptor values that represent motion style and actual emotions visible in human motion. For example angry motions are likely to have high QoM, but there are many motions with high QoM that might not look angry. A possible solution could be to develop new descriptors that are learned from annotated data with machine learning techniques. The new descriptors could be estimates for visibility of emotions like anger and sadness. Simultaneous use the planned emotional descriptors and the ones that we have presented could allow more precise authoring of emotional reactions.

6 Conclusions and future work

In this paper we introduced a new way to author behavior rules for interactive virtual characters using bodily motions as the medium. We showed that the simultaneous use of several motion descriptors enables creation of expressive interaction rules. Since the descriptors are continuous in both the time and motion style domains, the produced interaction has a chance to be fluid and natural. The problems that emerge from the use of several descriptors include the curse of dimensionality and increased risk of physically impossible descriptor combinations. We solved the curse of dimensionality by adding scaling vectors for each set of descriptor values in each mapping. This reduced the amount of required mappings per interaction rule dramatically.

We also introduced a way to create mappings based on an example with action and reaction motions. This approach can reduce the risk of defining impossible descriptor combinations. The example also guides the creation of the interaction rules and makes the process a straight forward one. Even when using example motions, our method allows manual fine tuning of the rules.

In the future we intend to develop motion descriptors that measure visibility of emotions like sadness and anger from human motions. A promising approach is to create descriptors from annotated motion data by machine learning techniques. We see the development of these descriptors as a required step in order to go from expressive bodily interaction to emotional bodily interaction.

Acknowledgments. This work has received funding from the projects Enactive Media (128132) and Multimodally grounded language technology (254104) which are both funded by the Academy of Finland.

References

1. Huang, L., Morency, L., Gratch. J.: Virtual Rapport 2.0. In: Vilhjálmsson, H., Kopp, S., Marsella, S., Thórisson, K. (eds.) IVA 2011. LNCS vol. 6895, pp. 68-79 (2011)
2. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Transactions on Graphics (SIGGRAPH '02), vol. 21, is. 3, pp. 473-482. ACM, New York (2002)
3. Arikan, O., Forsyth, D. A., O'Brien, J. F.: Motion synthesis from annotations. ACM Transactions on Graphics (SIGGRAPH '03), vol. 22, issue 3, pp. 402-408 (2003)
4. Xu, J., Takagi, K., Sakazawa, S.: Motion synthesis for synchronizing with streaming music by segment-based search on metadata motion graphs. Conf. on Multimedia and Expo (ICME), 2011 IEEE International, pp. 1-6 (2011)
5. Hachimura, K., Takashina, K., Yoshimura, M.: Analysis and evaluation of dancing movement based on LMA. In: IEEE International Workshop on Robot and Human Interactive Communication 2005 (ROMAN 2005), pp. 294-299. IEEE (2005)
6. Camurri, A., Mazzarino, B., Ricchetti, M., Timmers, R., Volpe, G.: Multimodal Analysis of Expressive Gesture in Music and Dance Performances. In: Camurri, A., Volpe, G. (eds.) GW 2003. LNCS (LNAI), vol. 2915, pp. 20-39. Springer, Heidelberg (2004)

7. Young, J., Ishii, K., Igarashi, T., Sharlin, E.: Puppet Master: designing reactive character behavior by demonstration. In Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08), pp. 183-191. Eurographics Association, Aire-la-Ville, Switzerland (2008)
8. Blumberg, B., Galyean, T.: Multi-level direction of autonomous creatures for real-time virtual environments. In: Mair, S.G., Cook, R. (eds.) Proc. of SIGGRAPH 1995, pp. 4754. ACM, New York. (1995)
9. Perlin, K., Goldberg, A.: Improv: a system for scripting interactive actors in virtual worlds. In: Proc. of SIGGRAPH 1996, pp. 205216. ACM, New York (1996)
10. Jebara, T., Pentland, A.: Action Reaction Learning: Automatic Visual Analysis and Synthesis of Interactive Behaviour. In: ICVS 1999. LNCS, vol. 1542, pp. 273-292. Springer, Heidelberg (1999)
11. Young, J., Ishii, K., Igarashi, T., Sharlin, E.: Style-by-demonstration: Teaching Interactive Movement Style to Robots. In ACM Conf. on Intelligent User Interfaces (IUI '12), pp. 41-50. ACM, New York, USA (2012)
12. Metaxas, D., Chen, B.: Toward gesture-based behavior authoring. In: Proc. of the Computer Graphics International 2005 (CGI '05), pp. 59-65. IEEE Computer Society, Washington, DC, USA (2005)
13. Ulicny, B., Ciechomski, P., Thalmann, D.: . Crowdbrush: interactive authoring of real-time crowd scenes. In: Proc. of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '04), pp. 243-252. Eurographics Association, Aire-la-Ville, Switzerland (2004)
14. Vilhjálmsson, H., Cantelmo, N., Cassell, J., Chafai, N. E., Kipp, M., Kopp, S., Mancini, M., Marsella, S., Marshall, A. N., Pelachaud, C., Ruttkay, Z., Thórisson, K. R., Welbergen, H., Werf, R. J.: The Behavior Markup Language: Recent Developments and Challenges. In: Pelachaud, C., Martin, J., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS vol. 4722, pp. 99-111. Springer, Heidelberg (2007)
15. Zwiers, J., Welbergen, H. V.: Continuous interaction within the SAIBA framework. In: Vilhjálmsson, H., Kopp, S., Marsella, S., Thórisson, K. (eds.) IVA 2011. LNCS vol. 6895, pp. 324-330. Springer Berlin / Heidelberg (2011)
16. Pugliese, R., Lehtonen, K.: A framework for motion based bodily enaction with virtual characters. In: Vilhjálmsson, H., Kopp, S., Marsella, S., Thórisson, K. (eds.) IVA 2011. LNCS vol. 6895, pp. 162-168. Springer, Heidelberg (2011)
17. Buhmann, M. D.: Radial Basis Functions : Theory and Implementations. Cambridge University Press, Cambridge, United Kingdom (2003)