Embodied Design of Full Bodied Interaction with virtual humans

1st Author Name
Affiliation
Address
e-mail address
Optional phone number

2nd Author Name
Affiliation
Address
e-mail address
Optional phone number

3rd Author Name
Affiliation
Address
e-mail address
Optional phone number

ABSTRACT

This paper presents a system that allows end users to design full body interactions with 3D animated virtual character through a process we call Interactive Performance Capture. This process is embodied in the sense that users design directly by moving and interacting using an interactive machine learning method. Two people improvise an interaction based only on their movements, one plays the part of the virtual character the other plays a real person. Their movements are recorded and they label it with metadata that identifies certain actions and responses. This labelled data is then used to train a Gaussian Mixture Model that is able to recognized new actions and generate suitable responses from the virtual character. A small study showed that users do indeed design in a very embodied way using movement directly as a means of thinking through and designing interactions.

Author Keywords
Embodied interaction; virtual humans; full body interfaces

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

General Terms
Human Factors; Design;

INTRODUCTION

Non-verbal communication is a vital part of our social interactions. While the often quoted estimate that only seven percent of human communication is verbal is contested, it is clear that a large part of peoples communication with each other is through gestures, postures, and movements. This is very different from the way that we traditionally communicate with machines. Creating computer systems capable of this type of non-verbal interaction is therefore an important challenge. Interpreting and animating body language is challenging for a number of reasons, but particularly because it is something we do subconsciously and we are often not aware of what exactly we are doing and would not be able to describe it later. Experts in body language (the people we would like to design the game) are not computer scientists but professionals such as actors and choreographers. Their knowledge of body language is embodied: they understand it by physically doing it and often find it hard to explicitly describe it in words (see Kirsh[16] for a discussion of embodied cognition in the area of dance). This makes it very hard for people to translate it into the explicit, symbolic form needed for computer programming.

The last few years have seen introduction of new forms of user interface device such as the Nintendo WiiMote, the Microsoft Kinect and the Sony Move go beyond the keyboard and mouse and use body movements as a means of interacting with technology. These devices promise many innovations, but maybe the most profound and exciting was one that appeared as a much hyped demo prior to the release of the Microsoft Kinect. The Milo demo (http://www.youtube.com/watch?v=CPIbGnBQcJY) showed a computer animated boy interacting with a real woman, replying to her speech and responding to her body language. This example shows the enormous potential for forms of interaction that make use of our natural body movements, including our subconscious body language. However, this demo was never released to the public, showing the important challenges that still remain. While sensing technology and Natural Language Processing have developed considerably in the 5 years since this demo there are still major challenges in simulating the nuances of social interaction, and body language in particular. This is very complex work that combines Social Signal Processing [33] with computer animation of body language[11]. Perhaps the greatest challenge is that body language is a tacit skill [27] in the sense we are able to do it without being able to explicitly say what we are doing or how we are doing it; and it is a form of embodied (social) cognition [16, 3] in which our body and environment play a fundamental role in our process of thought. The physicality of movement and the environment is an integral part of cognition and so a movement-based interaction is best understood through embodied movement. Kirsh[16] therefore argues that the next generation of interaction techniques should take account of this embodiment, part of a larger trend towards embodiment in interaction design [24, 9]. This raises an important challenge for designing computational systems because they traditionally must be programmed with explicit rules that are abstract and disembodied (in the sense that body movement is not an innate part of their creation). The problem of representing the embodied, tacit skills of body lan-

MOCO'15, August 14–15, 2014, Vancouver, Canada.

guage and social interaction requires us to develop computational techniques that are very different from the explicit and abstract representations used in computer programming.

In Fiebrink's evaluation of the Wekinator, a system for designing new gestural musical instruments one of the participants commented: "With [the Wekinator], it's possible to create physical sound spaces where the connections between body and sound are the driving force behind the instrument design, and they *feel* right. ...it's very difficult to create instruments that feel embodied with explicit mapping strategies, while the whole approach of [the Wekinator] ...is precisely to create instruments that feel embodied." [7, p. 249]. This shows that the wekinator uses a new approach to design gestural interfaces that not only makes it easier to design but changes the way people think about designing, from a explicit focus on features of the movement (e.g. shoulder rotation) to a holistic, embodied view of movement. This approach is called Interactive Machine Learning (IML): the use of machine learning algorithms to design by interactively providing examples of interaction. This "embodied" form of design taps into our natural human understanding of movement which is itself embodied and implicit. We are able to move and recognise movement effectively but less able to analyse it into components. IML allows designers to design by moving rather than by analysing movement.

This paper presents a first attempt at applying Fiebrink's method to full body interaction with animated virtual characters, allowing an emodied form of designing by doing as suggested by Kleinsmith *et al* [17]. We call this approach *Interactive Performance Capture*. Performance capture is the process of recording actors' performances for mapping into a 3D animation. This is able to bring the nuance of the performance to the animation, but it works for static animations, not interactive systems. We use interactive machine learning as a way of capturing the interactions between two performers, as well as their movements.

RELATED WORK

Expressive Virtual Characters

This work builds on a long tradition of research in interactive virtual characters that aims to simulate people's non-verbal behaviour and how they interact with eachother [32]. Machine learning has been applied to virtual characters in a number of ways. It has been used most often in contexts where there is a clear score that the system has to optimize to produce the best possible character at playing a game (e.g. [21] on reinforcement learning for a boxing character). Others have used learning algorithms to approximate examples of real behavior. For example, [20] learn Dynamic Bayesian Networks that reproduce the interaction between two fighting characters. Similarly, [8] has created characters that respond expressively to game events (in this case, spectators at a football match). This was learned from motion capture of a person responding to example events. [22] and [26] use a similar approach for learning to control gestures to accompany spoken conversation. These examples underpin our current work. However, none of the above systems have experimented with how machine learning can be used interactively as a design tool rather than a batch process. We extend this approach to use IML, in which the human is not simply a source of data but is actively involved in guiding the learning process.

Full body interaction

Full body interaction covers a range of interaction styles that make greater use of body movement than traditional mouse and keyboard input. The most common kind of bodily interaction is gesture interaction, where an interface is based on recognizing particular gestures, normally made with the hands and arms [2, 7]. However, it can include broader forms of interaction such as body activated art installations [29] or dance-like interfaces for controlling music[1].

Bodily interaction in video games is now a mass-market phenomenon with the launch of controllers such as the Nintendo Wii, Sony Move and particularly the Microsoft Kinect. This opens up the possibility of video games in which players interact with characters through their natural body language and other body movements. This is only beginning to be explored in video games but builds on a long tradition of research into virtual character that can engage in dialogs with humans [14], including many experiments into developing virtual characters that respond to gestures and body movements, from Thórisson's early work [31] to Huang et al's [12] sophisticated responsive listening agent and Xiao et al's machine learning based gesture recognition and response [34].

Interactive Machine Learning

Interactive Machine Learning is a new approach to machine learning in which user interaction is central to the learning process. Rather than simply providing a fixed data set to a batch process, users add data and tune parameters interactively, progressively refining the machine learning model based on interactive testing. This approach has the potential to fundamentally change the use of machine learning in interface design. The interaction and progressive refinement can make machine learning into a genuine design tool in which a designer has fine control of the resulting interface, as opposed to a batch approach where the designer has to simply prepare the data and hope that the result comes out as expected. Interactive machine learning builds on a long tradition of programming by example (e.g. [25, 23]) but it also differs in focusing on statistical learning algorithms and also in not attempting to model general programming, but limiting training by example to specific elements such as classifiers that are well modeled by current learning techniques. The term Interactive Machine Learning (IML) was introduced by [6] who saw it as a way of involving users more closely in the machine learning process by interactively supplying and editing training data. Their Crayons system enables non-expert users to create image processing classifiers in an iterative process by drawing on images that they provide as training data. Other systems also allow users to manipulate other aspects of the learning process, for example, Talbot et al's [30] EnsembleMatrix which allows users to combine classifiers and interactively explore the relationships between them. ManiMatrix by [15] allows users to specify preferences for classification by manipulating a confusion matrix. Most of the above systems treat machine learning algorithms as black boxes without considering

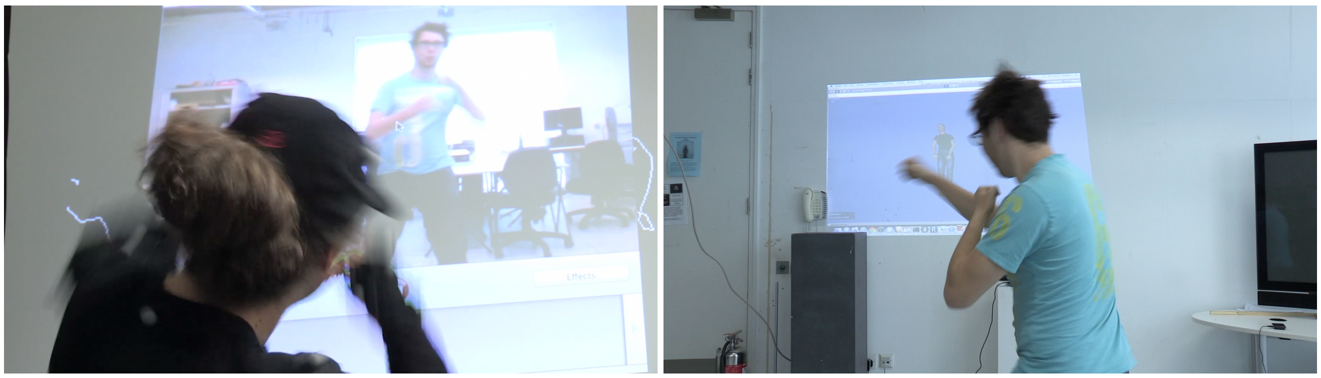

Figure 1. The actor being motion captured (above) and a participant interacting with the virtual character.

how users understand them. While ManiMatrix and EnsembleMatrix allow users to visualize and edit how the outputs of classification are used, they treat the algorithms themselves as a black box and do not encourage users to understand how they work. [19]'s research, on the other hand, centers on supporting users in understanding how their classifier reaches it's classification. They focus on end user debugging of machine learning systems for spam filtering by allowing users to make queries relating to why the system makes certain classifications. [7] has used Interactive Machine Learning for the design of gestural and bodily interfaces for the control of electronic movement. Her users found this form of design to be more natural and engaging than approaches that required them to analyze gestures in terms of specific features. This seems to have been because the process was more embodied, allowing users to concentrate on their movements, and in particular on the physical performance of those movements, rather than analyzing abstract features of the sensor data.

Machine learning has been applied to virtual agents in a number of ways. It has been used most often in contexts where there is a clear score that the system has to optimize to produce the best possible character at playing a game (e.g. [21] on reinforcement learning for a boxing character). Others have used learning algorithms to approximate examples of real behavior. For example, [20] learn Dynamic Bayesian Networks that reproduce the interaction between two fighting characters. Similarly, [8] has created characters that respond expressively to game events (in this case, spectators at a football match). This was learned from motion capture of a person responding to example events. [22] and [26] use a similar approach for learning to control gestures to accompany spoken conversation. These examples underpin our current work. However, none of the above systems have experimented with how machine learning can be used interactively as a design tool rather than a batch process. We extend this approach to use IML, in which the human is not simply a source of data but is actively involved in guiding the learning process.

DESIGNING BY MOVING

We have developed a process and software for designing body movment interactions with virtual agents. It allows users to design the interaction directly by moving rather than by programming, by using an interactive machine learning approach. The process consists of three phases: **improvising** movements to design the interaction **labelling** the movments that have been recorded in order to train a machine learning classifier and **testing** the resulting system.

PROCESS

The aim of this research is to create a software and hardware application that could make the creation of full body interaction interfaces accessible to independent game developers and ordinary game players. In particular it was to support the design of full body interaction with a virtual character, including both recognition of players' bodily actions and animation of a character's response. Users can design characters that can recognize actions performed by a player and then perform a suitable response.

Our application uses an Interactive Machine Learning approach, in which users design by giving examples of particular interactions. A learning algorithm is used to create a classifier that can recognize players' actions and which is then used to select responses. The learning algorithm is trained using the examples of interaction created by the users. The aim is to capture interactions that are as natural as possible, so our application is built around capturing interactions between two users as they happen.

The design process is therefore based on users providing examples of interaction via motion capture systems. This requires them to act out interaction, but does not require specific technical knowledge of programming. It focuses on users designing actions by doing those actions, using their implicit knowledge of behavior without having to formally analyze their actions.

The design process is split into three major phases: improvisation, labeling and testing.

Improvisation

The improvisation set up enables two users to improvise interaction in order to design the virtual character. One user plays the part of the character while the other plays the part of the player. The set up is shown schematically in figure 2. The two users are in different spaces and their movements were streamed live to each other. The movements of the the character were motion captured using an Optitrack^{TM}optical motion

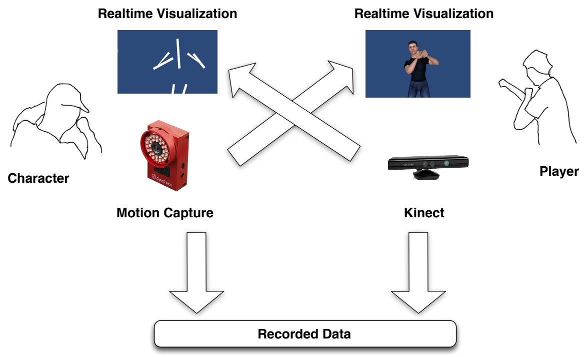

Figure 2. The improvisation and capture system

capture system and the player's movements were recorded using a Microsoft Kinect consumer motion tracking device.

Both sets of movement data were streamed into our software application and used to display a visualization on a screen in front of each user. Each user could see the movement of the other user, as well as their own. Both motion tracking devices represented the movement in terms of joint rotations. These rotations were mapped onto articulated skeletons consisting of a hierarchy of transformations. The rotations were written directly to the transformation corresponding to the same joint. The articulations were then used to drive an animated figure. In the case of the character, the figure was a moderately realistic mesh of a man. The player's movements were represented in a more abstract way as a stick figure. In our first prototype the user representing the player was directly visible to the other user via a live video stream (figure 1, left). However, this caused problems. A live video feed gives a holistic idea of a person's behavior based on a combination of many complex and subtle factors ranging from facial expression to muscle tension. However, the data used as input from the Kinect is only able to capture a small range of features, essentially only joint rotations and even then only of certain joints. Users' understanding of a pose would therefore rely on a number of factors that were not available to the software. The video feed made visible many aspects of the user's movement including many that were not available to the algorithm. On reviewing the poses used during the workshops it was clear that the users were making use of distinctions that were not visible in the joint rotation data. For example, they used a strong man pose, in which users raised their arms and flexed their biceps and also a celebration pose in which arms were raised without flexing the biceps. The two were easy to distinguish from video but looked very similar based on joint rotations. The final prototype used a new representation of the Kinect data as a stick figure (Figure 3). This representation aimed at making the system's representation of action as Kinect data more explicit to users. This minimal representation was designed to show only the joint rotation features available to the algorithm. To ensure consistency, this new representation was used both to display the user's movement live during improvisation and in the labeling interface.

The movement data was also recorded for later playback and as an input to the machine learning algorithm. The data was saved as an animation object consisting of a sequence of timed keyframes. Whenever new motion capture data was received, and after it had been mapped onto the skeleton, a keyframe was saved. The keyframe consisted of a time and a list of quaternion rotations, one for each joint. A single animation object was used for both the character and player data, to allow for synchronized playback.

Labelling

After the improvisation phase, the users use an editing interface (Figure 3) to select particular movements and label them as examples of particular actions and reactions. This labeling process is divided into two stages, selecting actions made by the player (using the interface shown in figure 3, left) and responses by the character (figure 3, right). The first are individual poses from the movement data of the user representing the player. Responses are sequences of movement by the player representing the character, which can be used to animate the character.

Both actions and responses are labeled with textual tags representing the type of action/response pair. Actions labeled with a certain tag will trigger one of the responses labelled with the same tag. Users are free to choose their own textual labels (with the exception of a special label "nothing" corresponding to no action). By allowing a free choice of tags the application aims to avoid imposing a fixed set of semantics on behaviour. Instead it aims to encourage users to use the functionality to develop their own meanings and representations of behavior, inline with research on designing systems for appropriation by users [5, 4]. Labels could be added using the interface labeled 4 in figure 3 (left). Each label is assigned a unique color and all items tagged with that label are shown in that color, to make the labeling clearly visible throughout the interface.

At the center of the action interface is the representation of the kinect data from the player with the character next to it for reference. As discussed in the previous section the kinect data is shown in a minimal stick figure (labeled 1 in figure 3, left) that only displays the joint rotation information that is actually available from the kinect, thus preventing the users from reading in extra detail that is not available to the algorithm. At the top of the screen is a list of recorded sets of movement data that the users can select (2 in figure 3, left). Once a movement data set has been selected users can move through it using the slider at the bottom of the screen (3 in figure 3, left). This allows them to select a particular pose as an action. This action can then be labeled with a tag, shown in the colored buttons above the stick figure (4 in figure 3, left). Each tag has a unique color which is used to represent it consistently across the interface. All of the labeled actions are shown as smaller "thumbnail" stick figures below the main stick figure (5 in figure 3, left). Each action is displayed in the color that represents its tag. The function of the thumbnails will be discussed in more detail in section **??**. The data selected using the interface above was used to train a classifier, which will be discussed in more detail in the next section.

The interface for selecting responses (figure 3, right) is similar. This time the character is in the center of the screen (1 in

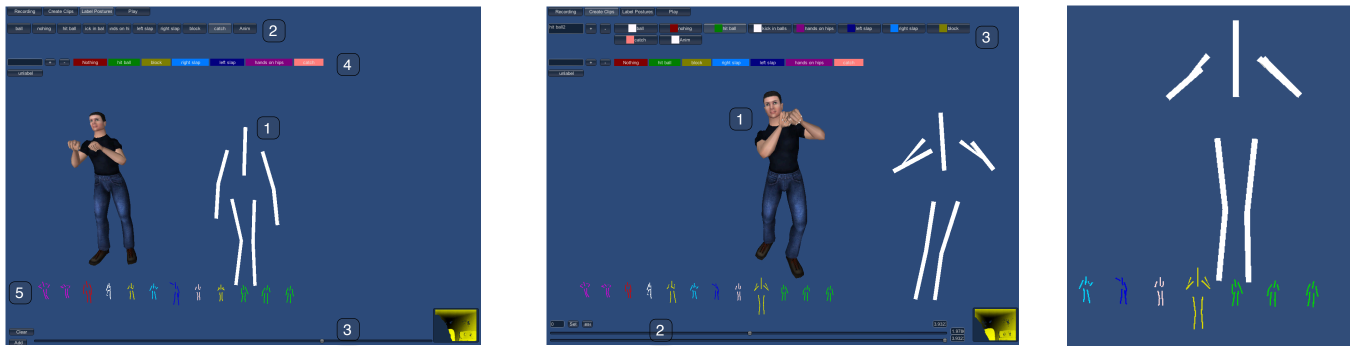

Figure 3. Left: The user interface for labeling action data. Center: The user interface for labeling response data. Right: The dynamic visualization of the nearest neighbor algorithm. Each data item is displayed as a stick figure. The figures are scaled in proportion to the amount of probability each contributes to the classification of the current posture.

figure 3, center). The responses are represented by a sequence of movement from motion capture data. These are created by selecting a range of movement from the data, using two sliders, one for the start and one for the end of the sequence (2 in figure 3, right). These sequences can be labeled with tags in the same way as the actions and then appear as buttons at the top of the screen (3 in figure 3, center).

The labeled movement data is used to train a machine learning classifier that is used to classify the behavior of the player and select appropriate responses for the character. The classifier is trained with data from the Kinect. The pose of the kinect data at each selected action is used as the training data and the textual tag is used as the class. The pose is represented as a vector of features consisting of the rotation of each joint represented as Euler angles (rotation angles about the x,y,z axes). We use the weighted nearest neighbor classifier proposed by Gillies *et al*[10]. They suggest that this classifier is particularly suited to interactive machine learning with body movement as it makes use of the training examples directly rather than creating an abstract model from the data. This maximizes the value of the training instances, which is important because in interactive machine learning there are typically fewer instances than in batch learning [7], but they are carefully chosen. The use of the training examples also makes the algorithm easier for users to understand, as it is directly acting on examples they are using, and it is easy to visualize in terms of these examples (see below for a discussion of the visualization).

Interaction

When interacting with the virtual character, the player's set up was identical to the improvisation set up: they interacted with a virtual character via the Kinect. However, in this case the character's actions were selected by our software rather than being controlled live by a user in motion capture.

When the software is controlling the character, the player's movements are tracked by the Kinect and converted into joint rotation features in real time. These features are classified by the Gaussian Mixture Model classifier and the result is used to select an animation clip to play. After the labeling process, each class label has been assigned to a number of motion capture clips of the character's movement. When the current clip has finished playing, the class label resulting from the latest classification is used to choose a new clip. A clip is chosen randomly from the clips tagged with the latest class label. This new clip is then smoothly blended into the previous animation using Unity's built in animation blending system. Once an animation clip has been selected the classification is re-set to the special label "nothing" which indicates that no action has been recognized. Thus, if no action is recognized while the current clip is playing, one of the clips labeled "nothing" will be played.

This set up allows users to test their virtual character and verify whether it produces the desired responses and detect errors. In our early tests, one of the biggest challenges for users was what to do when the system did not respond as expected. This requires users to debug the result of the learning algorithm, a very challenging task for end users and a well known issue for interactive machine learning [19]. In order to support this we have provided a visualization that supports users in understanding the working of the system. This visualisation is based on the one suggested by Gillies *et al.* [10] and is shown in figure 3 (right). Thumbnails of all of the training examples are shown at the bottom, each shown in a color corresponding to its class label. Each training example is shown as a stick figure. This is a deliberately simple representation that is constrained to only show the information available in the kinect data. These are scaled based on the probability weight they contributed to the classification of the current pose. This allows users to see which training examples are responsible for the current classification.

EMBODIED DESIGN IN USE

We performed a small evaluation of the final prototype to understand whether it better supported users. We recruited participants from applicants to a computer science programme, all of whom identified themselves as keen gamers. We ran three workshops with participants working in groups to design a virtual character with our system. Two workshops had two participants each and another had three.

All groups were asked to design a movement based game in which they interacted with a virtual character that responded to their movements. They could design whatever game they liked. They were shown how to use the system and then left

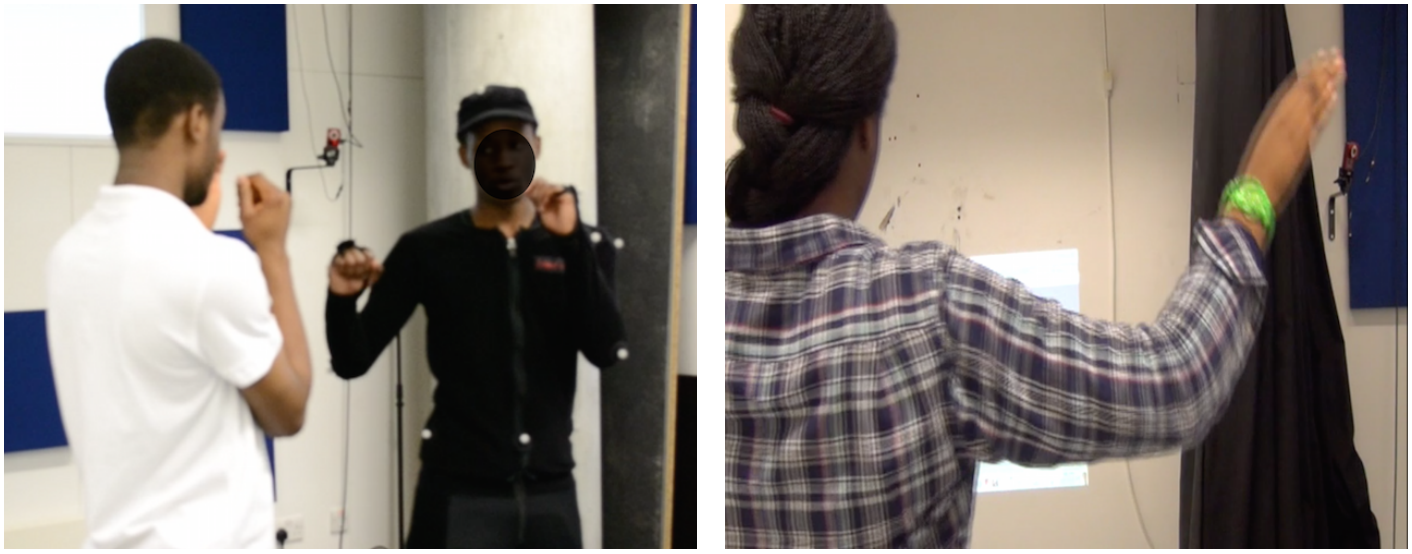

Figure 4. Embodied Design in practice. Left: two participants practicing an interaction together, they are constantly moving while discussing an interaction. Right: moving in front of the systesm, a playful dancing movement turns into a design idea.

to work independently. s groups were asked to improvise and act out interactions that would be suitable for their virtual characters. They recorded these improvisations in the process shown in figure 1 and then labelled the resulting data using the interface shown in figure 3. After the initial training phase, all groups were asked to test their games live using the Kinect. After this testing all groups were asked whether it worked as they wanted, and if not whether they knew what to do in order to fix it. All groups reported problems but immediately reported that they knew how to fix them. This implies that they had all formed a (not necessarily correct) model of the problem and had a strategy to fix it. They all proceeded to refine the model and in the final interview they all reported that they were happy with the end result. One participant explicitly noted that *"the second time around [the results were] much better"*. This suggests that participants were able to debug effectively. The testing did seem to be a key part of this: *"it wasn't fully clear until I tried it again"*.

Designing by moving

As we had hoped, movement was an integral part of the design process for our participants. They were in motion throughout the process. They moved while the discussing the design of the interfaces, creating a multimodal discussion combining speech with acting out movements (figure 4, left). There was also a lot of playful movement. Participants would joke and laugh while acting out movements and watching their motion capture. A number of participants also danced spontaneously, watching themselves. This might, in part be due to the novelty of the technology but the play also resulted in design ideas. One participant began dancing and made a waving movement (figure 4 right), in the middle of which she turned to her partner and said *"Ooh, we could do a wave"*, indicating a new potential idea for their interface. Our observation of participants' behaviour therefore supports our idea that movement is important to the design process and by allowing participants to design with movement they will design in new and embodied ways.

Visualising the algorithm is "really helpful"

The visualisation of the nearest neighbour algorithm seemed to play an important part in supporting their understanding. One participant rated the visualisation as *"really helpful"* with his partner adding *"especially when you are testing it over there [the live testing area]"*. In their interviews participants reported clearly understanding the function of the visualisation *"If it triggers it, it comes up, so that makes you know that movement triggers that posture"* and *"both the tags and the labelling went well, the fact that you can see them . . . bigger . . . it's good because when you move the bar around . . . you can see when it recognises it and when not, so you can see when a pose is clear and when not"*. These quotes indicate that participants were able to create conceptual models of how the classifier worked based on the visualisation. These were not completely accurate, they did not mention anything like weighting or probabilities, but they were sufficient to debug. They understood which postures were being triggered by a particular movement, and therefore which needed correction. One participant identified the visualisation as key to identifying problems: *"because as soon as we did the posture thing we understood what was the problem. So after you do that, it's easier to go and do movements . . . because you know how it works, the little stick man, let's say"*.

Visualising features is problematic

The participants reported problems with the stick figure representation of the Kinect data for actions: *"sometimes [the stick figure] can be a bit less clear, because it doesn't have depth perception, so you can have your arm out but it just thinks it's just there"*. One participant stated that she would have preferred to see a real camera view: *"If you have a real camera while you are recognising the movements and the poses . . . it's easier, you can recognise the movements*

...just for working out". The representation was purposefully designed to be impoverished relative to a live interaction or camera feed, in order to display only what the Kinect is able to recognise. However, this clearly made things difficult for participants. The first quote is simply about the lack of depth in the representation. This could be fixed without providing more information than the Kinect perceives. However, the second quote indicates that participants wanted a richer view, and that a more impoverished view was not sufficient to recognise movements.

CONCLUSIONS

Our participants' use of the system show that they do naturally use movement as the primary means of designing interaction, if the design tools allow this. This suggests that considering the embodied aspects of interaction is important not only for final systems, but also in the design process. This supports Hummels et al.'s belief that *"[in order to design movement interfaces] one has to be or become an expert in movement, not just theoretically, by imagination or on paper, but by doing and experiencing while designing"* [13]. Our system has show that interactive machine learning can potentially be a way of enabling this type of embodied design.

However, there is considerably more research to be done. While our participants did find visualizations useful in debugging the system, they did not like the stick figure visualization used for the training examples. This is problematic, as visualization was designed to only show the information that is available in the kinect sensor data, and so not mislead users by showing features such as muscle tension, which cannot be recognized in the system. However, users felt the lack of the rich information they normally have about movement. This shows that finding ways of communicating to users the nature of the sensor data, and how it differs from what we see, will be very important.

The types of interaction that our system supports are relatively simple: and action by the player directly generates a response by the character. This type of interaction is well suited to a machine learning classifier. However, human body language interaction is in general much more complex, with continuous exchange of much more subtle cues, with responses that have long term temporal effects. This type of behavior will require more sophisticated learning models such as multivariate regression (e.g. Gaussian Processes[28]) or complex probabilistic models[18]. It will therefore require considerably more research to achieve the type of interaction we outlined in our introduction.

ACKNOWLEDGMENTS

removed for anonymity

REFERENCES

1. Antle, A. N., Corness, G., and Droumeva, M. What the body knows: Exploring the benefits of embodied metaphors in hybrid physical digital environments. *Interacting with Computers 21*, 1-2 (2009), 66–75.
2. Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F., and Rasamimanana, N. Continuous realtime gesture following and recognition. *Gesture in Embodied Communication and Human-Computer Interaction* (2010), 73–84.
3. Bishop, J. M., and Martin, A. O., Eds. *Contemporary Sensorimotor Theory*, vol. 15 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*. Springer International Publishing, 2014.
4. Dix, A. Designing for appropriation. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 2*, BCS-HCI '07, British Computer Society (2007), 27–30.
5. Dourish, P. The appropriation of interactive technologies: Some lessons from placeless documents. *Comput. Supported Coop. Work 12*, 4 (9 2003), 465–490.
6. Fails, J. A., and Olsen, D. R. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces - IUI '03*, IUI '03, ACM Press (2003), 39.
7. Fiebrink, R. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. PhD thesis, 1 2011.
8. Gillies, M. Learning finite-state machine controllers from motion capture data. *IEEE Transactions of Computational Intelligence and AI in Games 1* (2009), 63–72.
9. Gillies, M., and Kleinsmith, A. Non-representational interaction design. In *Contemporary Sensorimotor Theory*, J. M. Bishop and A. Martin, Eds. Springer-Verlag, 2014, 201–208.
10. Gillies, M., Kleinsmith, A., and Brenton, H. Applying the cassm framework to improving end user debugging of interactive machine learning. In *Intelligent User Interfaces* (2015).
11. Gillies, M., Pan, X., and Slater, M. Piavca: a framework for heterogeneous interactions with virtual characters. *Virtual Reality* (2010), 1–8.
12. Huang, L., Morency, L.-P., and Gratch, J. Learning backchannel prediction model from parasocial consensus sampling: a subjective evaluation. In *Proceedings of the 10th international conference on Intelligent virtual agents*, IVA'10, Springer-Verlag (2010), 159–172.
13. Hummels, C., Overbeeke, K. C., and Klooster, S. Move to get moved: a search for methods, tools and knowledge to design for expressive and rich movement-based interaction. *Personal Ubiquitous Comput. 11*, 8 (12 2007), 677–690.
14. Jung, Y., Kuijper, A., Fellner, D. W., Kipp, M., Miksatko, J., Gratch, J., and Thalmann, D. Believable virtual characters in human-computer dialogs: State of the art report. In *32nd Annual Conference of the European Association for Computer Graphics*, ACM Press (2011).

15. Kapoor, A., Lee, B., Tan, D., and Horvitz, E. Interactive optimization for steering machine classification. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, ACM (2010), 1343–1352.

16. Kirsh, D. Embodied cognition and the magical future of interaction design. *ACM Trans. Comput.-Hum. Interact. 20*, 1 (4 2013), 3:1–3:30.

17. Kleinsmith, A., and Gillies, M. Customizing by doing for responsive video game characters. *International Journal of Human-Computer Studies 71*, 7 (2013), 775–784.

18. Koller, D., and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*, vol. 2009. 2009.

19. Kulesza, T., Stumpf, S., Wong, W.-K., Burnett, M. M., Perona, S., Ko, A., and Oberst, I. Why-oriented end-user debugging of naive bayes text classification. *ACM Trans. Interact. Intell. Syst. 1*, 1 (10 2011), 2:1–2:31.

20. Kwon, T., Cho, Y.-S., Park, S. I., and Shin, S. Y. Two-character motion analysis and synthesis. *IEEE Transactions on Visualization and Computer Graphics 14*, 3 (2008), 707–720.

21. Lee, J., and Lee, K. H. Precomputing avatar behavior from human motion data. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (7 2004), 79–87.

22. Levine, S., Krähenbühl, P., Thrun, S., and Koltun, V. Gesture controllers. *ACM Transactions on Graphics 29*, 4 (7 2010), 1.

23. Lieberman, H., Ed. *Your wish is my command: programming by example*. Morgan Kaufmann Publishers Inc., 2001.

24. Magnusson, T. Of epistemic tools: musical instruments as cognitive extensions. *Organised Sound 14*, 02 (6 2009), 168.

25. Myers, B. A., Cypher, A., Maulsby, D., Smith, D. C., and Shneiderman, B. Demonstrational interfaces: Coming soon? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, ACM (1991), 393–396.

26. Neff, M., Michael, Albrecht, I., and Seidel, H.-P. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Trans. Graph. 27*, 1 (3 2008), 5:1–5:24.

27. Polanyi, M. The tacit dimension. *Knowledge in Organizations* (1966), 135–146.

28. Rasmussen, C. E., and Williams, C. K. I. *Gaussian processes for machine learning.*, vol. 14. 2006.

29. Snibbe, S. S., and Raffle, H. S. Social immersive media: pursuing best practices for multi-user interactive camera/projector exhibits. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, ACM (2009), 1447–1456.

30. Talbot, J., Lee, B., Kapoor, A., and Tan, D. S. Ensemblematrix: interactive visualization to support machine learning with multiple classifiers. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, ACM (2009), 1283–1292.

31. Thórisson, K. Real-time decision making in multimodal face-to-face communication. In *second ACM international conference on autonomous agents* (1998), 16–23.

32. Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, E., Pan, X., Loscos, C., and Slater, M. Building expression into virtual characters. In *Eurographics Conference State of the Art Reports* (2006).

33. Vinciarelli, A., Pantic, M., and Bourlard, H. Social signal processing: Survey of an emerging domain. *Image Vision Comput. 27*, 12 (2009), 1743–1759.

34. Xiao, Y., Yuan, J., and Thalmann, D. Human-virtual human interaction by upper body gesture understanding. *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology - VRST '13* (2013), 133.