User-Defined Body Gestures for an Interactive Storytelling Scenario

Felix Kistler, Elisabeth André

User-Defined Body Gestures for an Interactive Storytelling Scenario

Felix Kistler, Elisabeth André

Human Centered Multimedia, Augsburg University,
Universitätsstr. 6a, 86159 Augsburg, Germany

{kistler, andre}@informatik.uni-augsburg.de

Abstract. For improving full body interaction in an interactive storytelling scenario, we conducted a study to get a user-defined gesture set. 22 users performed 251 gestures while running through the story script with real interaction disabled, but with hints of what set of actions was currently requested by the application. We describe our interaction design process, starting with the conduction of the study, continuing with the analysis of the recorded data including the creation of gesture taxonomy and the selection of gesture candidates, and ending with the integration of the gestures in our application.

Keywords: User Defined Gestures, Kinect, Full Body Tracking, Depth Sensor, Interaction, Interactive Storytelling

1 Introduction

Creating intuitive interaction has always been a difficult, but important task for developers of games or interactive storytelling applications. As depth cameras have become broadly available consumer products with the Microsoft Kinect sensor[1], opportunities emerge for such systems to make use of novel full body interaction techniques. This in turn poses new challenges for the interaction designer. Various researchers have already started to integrate this new kind of interaction in their interactive storytelling system [8, 1], but usually the gesture set for interacting with the system is chosen by the developers themselves according to their imagination and preferences. However, a gesture that is intuitive for the developers does not necessarily have to be intuitive for the majority of users. A different approach that creates a user-defined gesture set has been addressed by several other domains, such as surface computing [18], public displays [11], or human robot interaction [14].

In this paper we adapt the method for creating a user-defined gesture set for full body gestures in an interactive storytelling scenario. We describe the full process from conducting the user study, analysis of the recorded data, to the integration of the gesture set in our application.

[1] http://www.xbox.com/KINECT

1.1 In-Game Actions and Input Gestures

In computer games and interactive storytelling exist several types of in-game actions users can trigger via interaction. Two common action types are: *navigation* and *dialogue*. Basic navigation includes changing the position and orientation. Dialogue actions are usually involved when embodied conversational agents exist in the scenario that users can speak with. There are also other action types, e.g. manipulation actions which change the properties of an object, but our work will focus on navigation and dialogue. Furthermore, we focus on full body gestures that users need to perform for triggering those two types of actions. For navigation, this might be a quite straightforward choice, as body movements are also used for navigation in real-life. For dialogue the main interaction modality in real-life is speech and it therefore might seem a bit awkward to use body gestures for it. However, conversational gestures are used in real-life for emphasizing or enhancing speech utterances, and sometimes even to replace them, e.g. when performing a head nod instead of saying "yes". In addition, we want to keep the variety of different input modalities as small as possible because full body interaction itself can be quite difficult for users that are used to mouse and keyboard interaction and Kurdyukova et al. [10] have also shown that too many different input modalities can be more distracting than engaging for users. Overall, interactive storytelling applications can include a huge variety of specific actions for navigation and dialogue. It therefore seems quite impossible to find a generic set of actions for those two types. Instead, we investigate an action set created for our specific scenario. Nevertheless, it should represent a combination of actions typical to interactive storytelling scenarios and we have the hope that our findings also apply to other scenarios without major differences.

2 Related Work

In the following, we first describe work on classifying human gestures that helps us build our gesture taxonomy. Afterwards, we summarize an influential approach for acquiring a user-defined gesture set that we will adapt and use for our own work. The last area of research we analyze is that of full body gesture recognition, as we eventually aim to implement the recognition of our user-defined gesture set.

2.1 Human gestures

There exist different taxonomies for the classification of conversational human gestures. One of the first was introduced by Efron [3] who presented five categories: *physiographics*, *kinetographics*, *ideographics*, *deictics*, and *batons*. Further, Kendon [5] tried to link his gesture taxonomy to the relation with speech and defined the following categories: *gesticulation*, *language-like gestures*, *pantomime*, *emblems*, and *sign language*. Another popular taxonomy was proposed by McNeill [12] who presented five types of gestures: *iconic*, *metaphoric*, *deictic*, *cohesive*, and *beat* gestures. The three taxonomies have a considerable overlap in their covered concepts, and they all focus on hand gestures during a conversation. However, the properties they de-

scribe can also be used to categorize full body gestures for human computer interaction as in our case. None of the three taxonomies perfectly suits this purpose, but we have decided to use the terminology by McNeill and therefore describe the items of this taxonomy in more detail in the following. *Iconic* and *metaphoric* gestures both try to convey information by visually depicting an icon. However, they do this at different abstraction layers: *iconic* gestures are more concrete and directly represent a physical, spatial or temporal property of a real-world referent, e.g. when moving two fingers to indicate somebody is walking. *Metaphoric* gestures refer to abstract properties of a referent. For example, somebody might depict a container to refer to the contents of a story. *Deictic* gestures are pointing gestures that indicate a position or direction. The last two types of gestures do not convey meaning but accompany speech to emphasize parts of it (*beat gestures*) or to keep up the continuity (*cohesive gestures*). Therefore, those two types are less suited for direct interaction, whereas the other three categories seem to be good candidates for representing in-game actions. That is why we use them in our later described taxonomy. McNeill also defined three phases of a gesture: *preparation*, *stroke*, and *retraction*. *Preparation* is the phase in which the body is brought from its rest to a position that is suitable for executing the gesture. The *stroke* phase contains the main part of the gesture, while in the *retraction* phase the body is brought back to its rest. In terms of gesture recognition, the *stroke* phase is the most important part of a gesture as it contains the actual information.

2.2 User defined gestures

In the last couple of years, gestural interfaces have become more and more popular, ranging from small multitouch phones to large interactive tables or walls, but also to freehand and full body interaction using a depth sensor as the Microsoft Kinect. Nevertheless, gesture sets are often designed without sufficiently taking into accounts the preferences, habits, and needs of the actual users. For this reason, several researchers started to involve the user into the design process of the gestural interaction. Wobbrock et al. [18] presented a seminal approach to develop intuitive gestures for surface interfaces. They gathered a gesture set by presenting the wanted effect within the system to the user, and then asking the user to perform a gesture that should trigger this effect. After that, they determined gesture candidates by looking at all gestures performed by the users for a specific effect, and calculating an agreement score based on how often the same gestures were used. Researchers already adopted this process for other areas, e.g. Kurdyukova et al. [11] used it to design gestures for transferring data between tablet computers and a multi-display environment, and Obaid et al. [14] applied the process to create a full body gesture set for the navigational control of humanoid robots. In this paper, we adapt the process by Wobbrock et al. to identify intuitive gestures for an interactive storytelling scenario.

2.3 Full Body Gesture Recognition

A lot of work was already done in the field of gesture recognition in general, and different approaches were developed and extensively tested, including statistical clas-

sifiers [15], Hidden Markov Models [16], dynamic programming [13], and many more. Recognition algorithms of this kind were used in various application areas, although most of those methods are quite complicated to implement, are computational expensive, or need to be trained with a lot of example data [19]. An effort to make gesture recognition more accessible to developers without a strong background in pattern recognition is the $1 recognizer presented by [19]. However, this approach was – as the aforementioned ones – still targeted at gestures defined by the movement of a single point in 2D space. It also relied on manual data segmentation, so the start and end of a gesture had to be indicated to the algorithm, either in an implicit way, e.g. by touching the surface of a PDA with the pen [19], or in an explicit way, e.g. by pressing a button on a Wiimote [9]. There have also been approaches to get rid of those shortcomings by porting them to 3D space [9], or removing the need for manual data segmentation [20]. However, the research for those methods is partly still in an early phase, and they are quite complicated to apply for practitioners as well.

The release of the Kinect sensor again motivated researchers to investigate gesture recognition for the more complex motion capturing data provided by full body tracking. The challenge with this data is that it contains multi-point data in 3D space (position and orientation of multiple important joints of one or more users). In addition, there is no obvious way to apply manual data segmentation (no device in the users' hands to press a button on), and the data itself also is rather noisy [6]. On the other hand, this means that the data itself already contains more information as in other interaction modalities. In this way, one single data frame for one tracked user already can be seen as a gesture, or more precisely a posture, as it defines a specific configuration of the user's skeleton. For this reason, researchers developed easy to use techniques for full body gesture recognition in application prototypes. One of the first was the Flexible Action and Articulated Skeleton Toolkit (FAAST) [17] that bound gestures defined by simple text scripts to key and mouse events, and therefore enabled to control arbitrary applications via full body interaction. We developed a similar approach in our FUBI framework[2] of which an earlier version was already presented in [7]. It tries to achieve more powerful gesture recognition by giving more complex configuration options in an XML-based definition language. We use this framework to implement the user-defined body gestures in section 4.

3 Interactive Storytelling Scenario and Gesture Study

In this section, we describe the scenario of our application and its intended user interaction. We further explain our interaction study and present the results of its analysis.

3.1 Scenario and User Interaction

Our interactive storytelling scenario aims to provide intercultural training for young adults (18-25 year olds). The users learn by participating actively in the narrative in

[2] http:/www.hcm-lab.de/fubi.html

which they have to interact with virtual characters from different cultures. However, the characters do not represent real cultures, but synthetic ones as defined by Hofstede [4]. The users adopt the role of a character that has not traveled too much for most of his life. The scenario starts at the café of the character's grandmother, in which he receives a letter from his deceased grandfather. In this letter, the grandfather, who liked to travel the world, promises the grandson a "lost treasure" that he should find in a journey through different countries. In each country the grandson has to interact with locals in so-called critical incidents to progress. To be successful, the users have to select the correct interaction options depending on the agents' simulated synthetic culture. The selection itself should eventually be done by performing a corresponding full body gesture. In the final country the users will find out that the promised treasure is the experience that the grandfather had while travelling. The scenario is implemented using the cross-platform game engine Unity3D[3] and an agent architecture for culturally adaptable behaviors [2].

We conducted our interaction study for the introduction in the café and the first country of our scenario which at this state included two critical incidents. The users' first task was to find out the way to his hotel by interacting with people in a bar (first critical incident, cf. Fig. 1 left-hand side). In the subsequent incident users had to find the responsible supervisor in a nearby museum in order to receive entry permission for a park (second critical incident, cf. Fig. 1 right-hand side). The scene in the grandmother's café and the mentioned two critical incidents together included the following in-game actions to be triggered by the users: *yes, no, sit at bar and wait, approach group, ask for directions, leave the bar, ask about supervisor, ask guard to talk to supervisor, approach supervisor, ask permission.*

Fig. 1. Virtual environment of the two investigated critical incidents.

3.2 Study Setup, Procedure, and Participants

The experiment was arranged in a room of about 3 meters width and 6.5 meters depth. The participants were standing at a distance of about 2.5 meters in front of a 50 inch plasma display. A camera was placed in a height of about 1.5 meters left of the display to record the users' front from a slightly tilted view. The participants were told

[3] http://unity3d.com

that they should place themselves at the initial position, but that they were still allowed to freely move within the camera's field of view during the study. The experimenter was sitting to the left of the participant and controlling the application running on the display via mouse and keyboard.

After a short introduction and a demographic questionnaire that also included a question about the users' experience with body gesture based interaction, the experimenter explained the participants their role in the study. The experimenter ran through the story script of the application and as soon as a user input would have been requested by the application, text boxes with the currently available in-game actions were displayed as overlays on the virtual scene as depicted in Fig. 1. At this point, the participants' task was to invent and perform a gesture for each displayed action, one after the other. The participants were told that they were allowed to use their full body for gesturing, but that the gesture itself should mainly be intuitive for them to trigger the requested action. It should, however, have a semantic relation to the action and not consist of simply pointing towards the action label on screen. To keep the process as reproducible as possible, the experimenter always spoke out the action that the user should investigate next and also gave a short explanation about the meaning of the action to avoid misunderstandings. After performing their invented gesture, the participants should indicate on a questionnaire how easy it was for them to come up with that gesture on a 7-point Likert scale.

22 participants took part in the study including 4 females and 18 males. Their age ranged from 22 to 35 with an average of 26.23 (SD 3.80). All except for one were right-handed. The participants were recruited from our university campus and therefore all had a computer science background. They stated themselves a medium experience with body gesture based interaction of 2.18 (SD 0.85) on a scale from 0 (no experience) to 4 (practically daily usage).

3.3 Results

The next chapters depict the results of our study, including our gesture taxonomy, a description of the gesture set, user ratings, agreement scores and time performances.

Table 1. Full body gesture taxonomy for our interactive storytelling scenario.

Form	static gesture	A static body gesture is held after a preparation phase.
	dynamic gesture	The gesture contains movement of one or more body parts during the stroke phase.
Body parts	one hand	The gesture is performed with one hand.
	two hands	…with two hands.
	full body	…with at least one other body part than the hands.
Gesture type	deictic	The gesture is indicating a position or direction.
	iconic	The used gesture visually depicts the meant in-game action or a part of it directly.
	metaphoric	The gesture visually depicts an icon and describes the in-game action in an abstract way.

Gesture Taxonomy.
The recorded videos were analyzed and annotated using the ELAN annotation tools[4] to extract the stroke phases of all gestures performed by the study participants for each in-game action. We manually classified all performed gestures according to three dimensions: *form, gesture type,* and *(involved) body parts*. Each dimension consisted of multiple items as shown in Table 1. The dimensions were based on the taxonomy we had used in [14], we only left out the view-point dimension, as our interactive storytelling scenario always kept the user in a first-person perspective and there were no changes in the view-point for this reason. Furthermore, we changed the wording of the *gesture type* dimension to be closer to McNeill [12].

Fig. 2 displays the overall taxonomy distribution for the 251 performed gestures. The frequency of static and dynamic gestures was quite similar. Users tended to perform few deictic gestures, but more metaphoric ones. They only seldom chose two hand gestures, but roughly an equal number of one hand and full body gestures. The gestures we categorized as iconic according to McNeill [12] in fact were very concrete, which means that most of them were directly miming the meant in-game action, e.g. *approach group* was often expressed by actually walking a step forward.

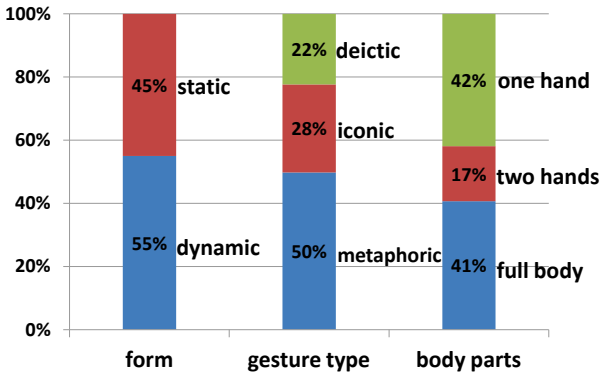


Fig. 2. Overall taxonomy distribution.

Gesture Set.
We selected suitable gesture candidates for each in-game action as follows:

1. For each action a we identified a set $M(a)$ containing all proposed gestures.
2. The gestures in $M(a)$ were then grouped into subsets of identical gestures $M_i(a)$ with $i \in 1..n_a$ and n_a being the total number of identified subsets for action a.
3. The first candidate $c_1(a)$ was determined by the largest subset $M_i(a)$, i.e.:

$$c_1(a) = MAX_{i \in 1..n_a}(M_i(a)). \quad (1)$$

4. The second candidate $c_2(a)$ was determined by the second largest subset $M_i(a)$:

[4] http://www.lat-mpi.eu/tools/elan/

$$c_2(a) = MAX_{i \in 1..n_a,\, M_i(a) \neq c_1(a)}(M_i(a)). \quad (2)$$

There can been more than one first or second candidate if there are multiple largest subsets, but this was not the case for our data. Further, we only took the second candidate into account if the size of the corresponding subset was at least half the size of the subset of the first candidate. This was done to ensure that the considered candidates always had been performed by a relevant percentage of users.

Table 2. Gesture candidates for each action.

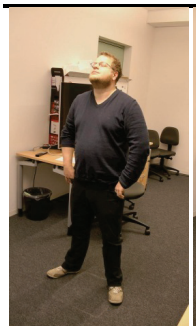
In-game action	Gesture candidates	Occurrences	Form	Gesture type	Body parts
yes	head nod	68%	dynamic	metaphoric	full body
no	head shake	68%	dynamic	metaphoric	full body
sit at bar and wait	sit down	56%	static	iconic	full body
approach group	step forward	56%	dynamic	iconic	full body
ask for directions	arms out	34%	static	metaphoric	two hands
leave bar	turn away	45%	dynamic	iconic	full body
	step backward	27%	dynamic	iconic	full body
ask about supervisor	arms out	50%	static	metaphoric	two hands
ask guard to talk to supervisor	point at one after the other	38%	static	deictic	one hand
	point to front	21%	static	deictic	one hand
approach supervisor	step forward	56%	dynamic	iconic	full body
ask permission	arms out	23%	static	metaphoric	two hands
	tip on shoulder	19%	dynamic	iconic	one hand

Table 2 summarizes the gesture candidates for all ten in-game actions. The third column includes the percentage of how often this candidate was performed among all gestures proposed for this action, and the last three columns depict the candidate's taxonomy. A second candidate was taken into account only in three cases (*leave the bar*, *ask guard to talk to supervisor*, and *ask permission*).

The gesture candidates are further exemplified by images of users performing them in Table 3. For the actions *yes* and *no*, most users chose a *head nod* or *head shake* as gestures. The action *sit at bar and wait* was in most times represented by actually adopting to a sitting position (=*sit down*). Similarly, we found gesture candidates that represented the action quite directly for *approach group*, *leave bar*, and *approach supervisor*, in which the users did a *step backward* or a *step forward*. For the action *leave bar* a second gesture candidate was *turn away* which meant the user actually turned around as if going away. *Ask permission* was additionally expressed by the gesture *tip on shoulder* that was chosen because the supervisor – that participants should ask for permission to enter a park – stood there with the back to them (cf. Fig. 1 right-hand side: the virtual character at the back), so they assumed they first needed to get his attention. For the ask actions we often got the gesture *arms out* that always

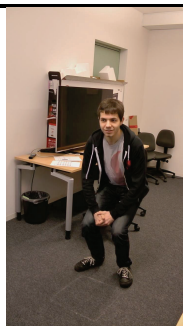
included moving the arms to an outward position with open hands, often accompanied by raising the shoulders. The only action for which the gesture candidates were pointing gestures was *ask guard to talk to supervisor*. Participants either chose to point in the direction of the supervisor (*point to front*), or to point at the guard first and only afterwards to the supervisor (*point at one after the other*).

Table 3. User images of the gesture candidates.

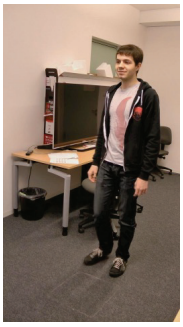


head nod

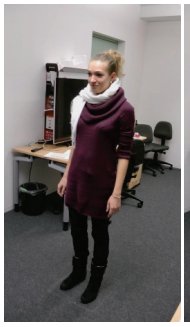
head shake



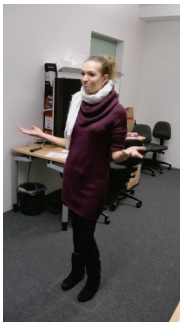
sit down



step forward (from left to right) / *backward* (from right to left)



turn away



arms out

point to front (left image) / *to one after the other* (both images)

tip on shoulder

User Ratings.
Fig. 3 depicts the average user ratings for the easiness to invent the gestures for the 10 in-game actions on a scale from 0 (very hard) to 6 (very easy). Error bars represent the standard error. The actions are ordered according to their user rating.

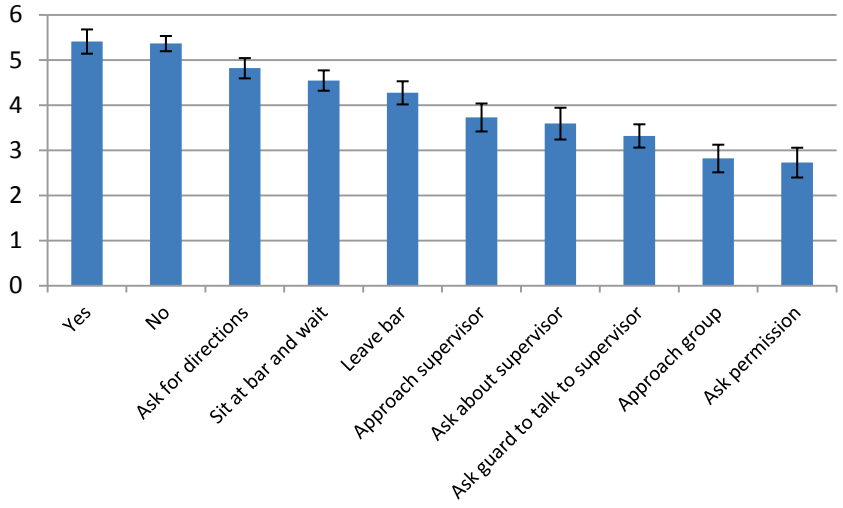


Fig. 3. User difficulty ratings of the 10 actions.

A one-way repeated measures ANOVA revealed that the ratings differed significantly between the different actions with $F(9, 21) = 15.90$, $p < 0.01$, $\eta^2 = 0.43$. Post-hoc tests with Bonferroni correction showed that the more complex conversational actions were perceived as more difficult to invent a gesture for them. Accordingly, all actions that include asking character(s) something were rated as the most difficult ones. In particular, all those actions were rated significantly lower ($p < 0.01$) than the actions *yes* and *no*. *Approach supervisor* and *leave bar,* were also rated significantly higher ($p < 0.05$) than the ask actions, except for *ask permission*. *Approach group* was only significantly higher rated ($p < 0.05$) than *ask about supervisor*. We found no significant difference between *sit at bar and wait* and the ask actions, and there was also no significant difference between the ratings of the different ask actions.

Agreement Scores.
To further investigate the level of agreement among the participants, we calculated an agreement score based on the process defined and used by Wobbrock et al. [18]. For an action *a*, the agreement score $AS(a)$ is defined as:

$$AS(a) = \sum_{i \in 1..n_a} \left(\frac{|M_i(a)|}{|M(a)|} \right)^2 \qquad (3)$$

An agreement score $AS(a)$ for an action a is represented by a number in the range $[1 / |M(a)|, 1]$, with a higher value corresponding to a higher agreement, and 1 representing a perfect agreement (all participants chose the same gesture for this action).

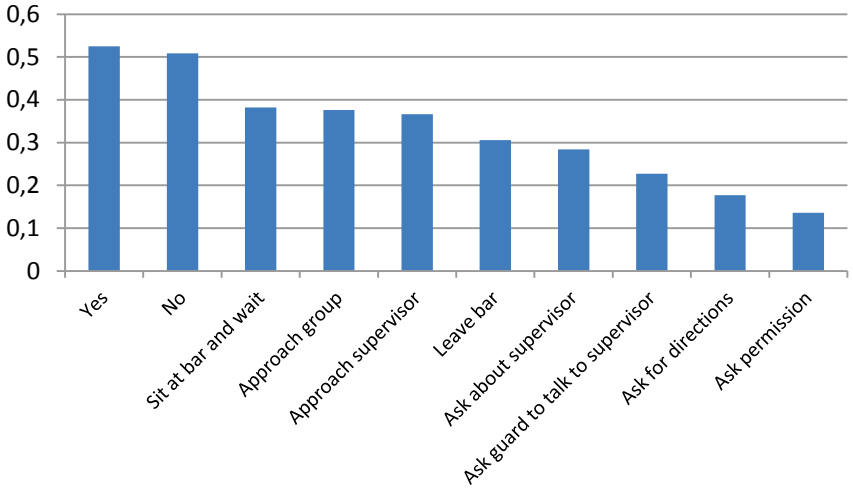


Fig. 4. Agreement scores for the 10 actions.

The overall agreement we got for our action set was 0.329 (SD 0.129). Fig. 4 depicts the agreement scores of the different actions. They are ordered from the action with the highest agreement to the one with the lowest agreement. Similar to the user ratings, more complex actions (i.e. all "ask …" actions) caused lower agreement between the users and we got a large number of different gestures for those.

In fact, the results reveal that the level of agreement between the participants strongly correlates to the easiness to invent gestures (Pearson's $r = 0.812, p < 0.01$). When the participants thought it was easy to find a gesture for an in-game action, more participants chose the same gestures, and in opposite, when the participants thought it was difficult to invent a gesture for an action, we got a higher variation of gestures as well.

Timings.

Table 4 depicts the times it took the users to perform one stroke of the gesture candidates. One stroke means e.g. for the head nod gesture that the user moves the head from the resting position upwards, then downwards under the resting position, and upwards to the resting position again. In other words, one stroke consists of the minimal gesture that can be found when dividing the gesture into equal sub gestures. For static gestures, one stroke consists only of a hold phase in which the user holds the relevant posture until moving back to a resting position. The table enlists the average times as well as the standard deviation, the minimum, and maximum among the different gesture performances.

Table 4. Times for one stroke of the gesture candidates.

in-game action	gesture candidates	mean time	SD	MIN	MAX
yes	head nod	0.822	0.374	0.420	1.560
no	head shake	0.687	0.233	0.474	1.420
sit at bar and wait	sit down	0.661	0.635	0.120	1.970
approach group	step forward	1.508	0.702	0.422	2.770
ask for directions	arms out	0.707	0.392	0.295	1.375
leave bar	turn away	1.738	0.908	0.557	3.400
	step backward	2.041	0.593	1.295	3.000
ask about supervisor	arms out	0.589	0.341	0.200	1.280
ask guard to talk to supervisor	point at one after the other	1.013	0.257	0.540	1.410
	point to front	0.560	0.558	0.130	1.625
approach supervisor	step forward	1.444	0.659	0.517	2.470
ask permission	arms out	0.759	0.578	0.190	1.770
	tip on shoulder	0.356	0.066	0.257	0.410

4 Integration of the Gesture Set

After collecting the data for our gesture set, we aimed to enable the gestural interaction in our application. In the following, we describe the recognition framework we used to achieve this task, the implementation of the gesture recognizers, and how we integrated them into our interactive storytelling scenario.

4.1 Recognition Framework

For implementing the recognition of the gesture candidates in our application, we used our framework for full body interaction (FUBI) [7]. The recognition framework uses the Kinect for Windows SDK[5] for getting the depth data provided by the Kinect sensor, and for applying full body user tracking on that data. In this way, we get positions and orientations for 20 different user joints. The joint data is analyzed in the recognition framework for detecting gestures that are defined via XML files. Those XML files first can contain three types of basic gesture recognizers:

1. *Joint orientation* recognizers are defined by a minimum and/or a maximum angle for a specific joint.
2. *Joint relation* recognizers look at the position of a joint in relation to a second one, e.g. whether and how much a joint is above a second one or how large the distance between those two is.

[5] http://www.microsoft.com/en-us/kinectforwindows/

3. *Linear movement* recognizers are defined by a specific direction and a minimum and/or maximum speed.

In addition, those three types of basic recognizers can be combined to sequences in so-called *combination* recognizers. A *combination* recognizer consists of several states that contain sets of the above mentioned basic recognizers. For each state there is a minimum and maximum duration defined that those recognizers have to be fulfilled for the recognition process to get into and stay in this state. For the transition to the next state there is also a maximum duration defined. Fig. 5 depicts the XML definition for a *combination* recognizer that is meant to recognize a head nod gesture.

```
<!--HeadNod-->
<JointOrientationRecognizer name="HeadDown">
  <Joint name="head"/>
  <MaxDegrees x="-13"/>
</JointOrientationRecognizer>
<JointOrientationRecognizer name="HeadUp">
  <Joint name="head"/>
  <MinDegrees x="-5"/>
</JointOrientationRecognizer>
<CombinationRecognizer name="HeadNod">
  <State maxDuration="1" timeForTransition="0.4">
    <Recognizer name="HeadUp"/>
  </State>
  <State maxDuration="1" timeForTransition="0.4">
    <Recognizer name="HeadDown"/>
  </State>
  <State maxDuration="1" timeForTransition="0.4">
    <Recognizer name="HeadUp"/>
  </State>
  <State>
    <Recognizer name="HeadDown"/>
  </State>
</CombinationRecognizer>
```

Fig. 5. XML definition of the recognizer for a head nod.

4.2 Integration of Gestures in the Application

For implementing the recognizers for the gesture candidates, one first has to determine by which sequence of basic recognizers a gesture candidate can be described and how the recognizers' parameters need to be adjusted. This is done by studying sample videos of the gesture in more detail and approximating the parameters. For determining the time constraints of a combination recognizer, the measured timings as depicted in Table 4 can be used. However, it has to be mentioned that in most cases, the timings cannot be used directly, but should rather serve as a basis for understanding a gesture and its temporal variance between the users.

To implement the gesture candidate *head nod* for the action *yes*, we used the XML definition as shown in Fig. 5. The definition includes two different *joint orientation* recognizers that observe the x angle of the head. Those *joint orientation* recognizers

were combined in a sequence of four states. Note that all states do not have a minimum duration, so it would be sufficient if the recognizers of one state are fulfilled only for one frame of the tracking data stream. However, each state has a maximum duration of 1 second and a time for transition to the next state of 0.4 seconds. That means a head nod could include a head tilted upwards for at most 1 second, and then a movement downwards that lasts at most 0.4 seconds follows. The tilted downwards head could be hold for up to 1 second. After that follows an upwards movement for again at most 0.4 seconds, and the whole sequence needs to be repeated a second time. Therefore, a head nod is recognized as soon as two single nods are detected. This was sufficient for our purpose as we were not interested in distinguishing different kinds or numbers of nods.

The rest of the gesture candidates were implemented in a similar way. We used *combinations* of joint orientation recognizers for the gesture candidates *head shake*, *sit down*, and *turn away* to check the orientations of the joints included in the gesture. For the candidate *arms out* we used a recognizer that combines two *joint relation* recognizers in one state. The first *joint relation* recognizer observed the left hand and shoulder and waited for the hand to be in a height similar to the shoulder (y difference smaller than 30 cm) and that the hand was at least 35 cm left of the shoulder (using the x coordinate). The second recognizer looked for the same properties with the right hand and shoulder. The gesture candidate *point to front* as well used a *joint relation* recognizer for checking the hand position. In addition it ensured that the right elbow was not too far away from the line from the shoulder to the hand (at max 12 cm) and that the hand was not moving with more than 0.5 m/s. The last condition was implemented using a *linear movement* recognizer. The candidate *pointing at one after the other* was defined in the same way but with an additional state that allowed hand movement in between the two pointing states. We also implemented the second gesture candidate for the action *ask permission* that was the gesture *tip on shoulder*. This was basically realized the same way as point to front, but in addition, it waited for a sequence of *linear movements* in upward and downward direction of the hand. The gesture candidates *step forward* and *step forward* were implemented as *linear movement* recognizers looking at the torso joint and waiting for a movement in z or –z direction after a short phase of standing still.

The integration of the gestures in the application was done similarly to how we conducted the study. Instead of only displaying text boxes with the currently available actions, we additionally displayed symbols that visualized how the gestures for these actions should be performed. The symbols consisted of either a single image that is displayed constantly or multiple images that are displayed as an animation sequence. The Unity3D integration the FUBI framework only needs a recognizer definition that is named the same way as the image (sequence) that is used as a symbol for it. As soon as the symbol is displayed on screen, the recognition framework automatically checks the corresponding recognizer for users tracked with the depth sensor, and - if the recognition has been successful - it triggers an event related to the symbol. The event for each symbol can be defined in the same way as it is done for default interface buttons in Unity3D. Fig. 6 depicts a scene of the first critical incident, now displaying four symbols of the new gesture set that can currently be selected by the user.

Although we implemented the recognizers for all gesture candidates, we had to decide which of the candidates we would actually use in the three cases in which two candidates were determined. As they seemed to fit a bit better to the parallel and surrounding gestures and because of their partly more reliable recognizers, we chose the gesture *turn away* for the action *leave bar*, *point to front* for *ask guard to talk to supervisor*, and *tip on shoulder* for *ask permission*.

Fig. 6. Gesture symbols in the first critical incident.

We preliminarily tested the integration of the new gesture set in an informal study with the same setup as in the previous study, but this time a Kinect sensor was placed centered below the screen to enable the interaction. Three users took part in the study, two males and one female. They had already participated in the study for the creation of the gesture set, thus they were used to the setting. They again ran through the story script, but this time the gestural interaction was enabled, so they always had to perform one of the requested gestures to continue. We had to omit the part in which the two gestures *head nod* and *head shake* appeared due to (later solved) technical issues with the Kinect face tracking SDK. For the rest of the gestures, the detection already worked quite robust. Out of 15 performed gestures, 11 were directly recognized by the first attempt of the user, while four of them had to be executed a second time to be recognized by the system. Thus, we had four false negatives and 15 true positives, which can be summarized to a preliminary recognition rate of 79 %. In addition, most of the false negatives were due to a wrong gesture performance by the participant. This was caused by the fact that the gestures were not explained to the participants, but they had to guess how to perform them by looking at the symbols displayed on screen as depicted in Fig. 6. The system never recognized a wrong gesture, so there were no false positives. Overall, we got positive feedback about the integration, and the participants liked to interact by using the new gesture set.

5 Discussion

As far as the taxonomy distribution of our gesture set is concerned, we got quite different results in comparison to our previous work that investigated user-defined gestures for navigating a humanoid robot [14]. We got much less deictic gestures as in the previous work, but more metaphoric ones. This is due to the target of the gestural interaction. While deictic gestures seem to be more suitable for navigational actions, our action set also included conversational actions that need to be described with metaphoric gestures due to their increased semantic complexity. We also had a closer look at the taxonomy distribution of each action itself that revealed that the users never used iconic gestures for the conversational actions (all *ask actions* plus *yes* and *no*) except for the gesture *tip on shoulder* of the action *ask permission*. All other actions – which can be categorized as navigational actions – included all three types of gestures, except for *leave bar* for which we only observed iconic and deictic gestures, but no metaphoric ones. For further increasing the information content of their gestures, the participants more often used other body parts than their hands in opposite to the robot navigation task. However, they used less dynamic gestures, which indicates that full body gestures often provide enough information in a static version. As full body gestures were overall frequently chosen quite often, it can be said that this kind of gestures is worth to be used in interactive storytelling scenarios in general. It seemed that full body gestures are especially intuitive for triggering the in-game actions that occur in this kind of scenarios.

We proposed to select the gesture candidates according to how many users chose a gesture for one in-game action. However, this does not always have to be the best choice. For example, it makes no sense to give the user the choice between two actions represented by the same gesture at the same point in time. In this case it is better to select a less often chosen gesture candidate for at least one of the actions. There are also other cases in which it is helpful to select a different gesture, e.g. if the recognition software is not able to detect the gesture in a robust way. A more specific reason for doing this is also given in our application. As we aim at intercultural training, we want the users, at a later point in our scenario, to be confronted with gestures that are unfamiliar to them, as this can occur when travelling to different countries. For this purpose it might also be worth to conduct the study with participants of different cultural background to get a different gesture set.

Another challenge we faced was the problem of potentially too complex in-game actions, and especially the difficulty to represent verbal actions with non-verbal gestures. This can be seen in the relatively low agreement scores and user ratings we got for all actions that involve asking virtual characters about something. At the state of the study, our scenario never included multiple ask actions in parallel, so we had no problems with their ambiguous gestures, but as soon as this occurs, it might be necessary to refine the actions for making them easier to be related to gestural input. Otherwise, it might also be necessary to include different kinds of interaction, e.g. using a graphical user interface or other modalities such as speech input.

As described in section 4, it was feasible for us to implement a prototypic recognition logic for the gathered gesture set within our recognition framework [7] and to

integrate the gestures in our storytelling scenario. The recorded videos of the gesture performances were very useful for this task as well as the measured timings for the gestures' stroke phases. Nevertheless, the creation of a gesture recognizer based on this data is still a challenging task that has to be done in a careful way to realize the gestures at least close to as they were intended by the users.

6 Conclusion and Future Work

In this paper, we presented how we produced a user-defined gesture set for an interactive storytelling scenario. We described the full process from conducting the study, over analysis of the data, to the implementation of the gesture candidates. During this process we obtained a taxonomy of full body gestures for our interaction set, user ratings and agreement scores for each in-game action, the time performances of all gesture candidates, and we finally integrated the gesture candidates in our applications using our open source full body interaction framework FUBI [7] .

A first validation for FUBI according to accuracy and usability was already done with a different interactive storytelling scenario [8] that included different kinds of iconic gestures. We plan to conduct a similar study with the new scenario in order to provide a more complete validation, also with more abstract metaphorical gestures. At the moment, we investigated the already implemented first part of our scenario. As soon as more of the application is ready, we want to continue the study with the additional in-game actions. In the meantime, it became clear that our scenario will have interactions that sometimes include multiple conversational actions in parallel that we will probably not be able to represent with unambiguous gestures. Therefore, we plan to include an additional interaction with a graphical user interface for those cases in our application.

Acknowledgments. This work was funded by the European Commission within FP7 under grant agreement eCUTE (FP7-ICT-257666). We would like to thank our project partners for the collaboration, and especially Nick Degens, Hélio Mascarenhas, Samuel Mascarenhas, and André Silva for their work on the storyboard, agent architecture, and asset creation that formed the background for our study.

References

1. N. Álvarez and F. Peinado. Exploring body language as narrative interface. In *Interactive Storytelling*, Lecture Notes in Computer Science, pages 196–201. Springer Berlin Heidelberg, 2012.
2. J. Dias, S. Mascarenhas, and A. Paiva. Fatima modular: Towards an agent architecture with a generic appraisal framework. In *Proc. of the Int. Workshop on Standards for Emotion Modeling*, 2011.
3. D. Efron. *Gesture and Environment*. King's Crown Press, Morningside Heights, New York, 1941.

4. G. J. Hofstede. Role playing with synthetic cultures: the evasive rules of the game. *Experimental Interactive Learning in Industrial Management: New approaches to Learning, Studying and Teaching*, page 49, 2005.
5. A. Kendon. How gestures can become like words. In *Cross-cultural Perspectives in Nonverbal Communication*, pages 131–141. Hogrefe, 1988.
6. K. Khoshelham and S. O. Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.
7. F. Kistler, B. Endrass, I. Damian, C. Dang, and E. André. Natural interaction with culturally adaptive virtual characters. *Journal on Multimodal User Interfaces*, 6:39–47, 2012.
8. F. Kistler, D. Sollfrank, N. Bee, and E. André. Full body gestures enhancing a game book for interactive story telling. In *Interactive Storytelling*, volume 7069 of *Lecture Notes in Computer Science*, pages 207–218. Springer Berlin Heidelberg, 2011.
9. S. Kratz and M. Rohs. A $3 gesture recognizer: simple gesture recognition for devices equipped with 3D acceleration sensors. In *Proc. IUI 2010*, pages 341–344, New York, NY, USA, 2010. ACM.
10. E. Kurdyukova, E. André, and K. Leichtenstern. Introducing multiple interaction devices to interactive storytelling: Experiences from practice. In *Interactive Storytelling*, volume 5915 of *Lecture Notes in Computer Science*, pages 134–139. Springer Berlin Heidelberg, 2009.
11. E. Kurdyukova, M. Redlin, and E. André. Studying user-defined iPad gestures for interaction in multi-display environment. In *Proc. IUI 2012*, pages 1–6, 2012.
12. D. McNeill. *Head and Mind: What Gestures Reveal About Thought*. University of Chicago University of Chicago Press, 1992.
13. C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, 1981.
14. M. Obaid, M. Häring, F. Kistler, R. Bühling, and E. André. User-defined body gestures for navigational control of a humanoid robot. In *Social Robotics*, volume 7621 of *Lecture Notes in Computer Science*, pages 367–377. Springer Berlin Heidelberg, 2012.
15. D. Rubine. Specifying gestures by example. *SIGGRAPH Comput. Graph.*, 25(4):329–337, 1991.
16. T. M. Sezgin and R. Davis. HMM-based efficient sketch recognition. In *Proc. IUI 2005*, pages 281–283, New York, NY, USA, 2005. ACM.
17. E. Suma, B. Lange, A. Rizzo, D. Krum, and M. Bolas. FAAST: The flexible action and articulated skeleton toolkit. In *Proc. VR 2011*, pages 247 –248, 2011.
18. J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-defined gestures for surface computing. In *Proc. CHI 2009*, pages 1083–1092, New York, NY, USA, 2009. ACM.
19. J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. UIST 2007*, pages 159–168, New York, USA, 2007. ACM.
20. A. Zinnen and B. Schiele. A new approach to enable gesture recognition in continuous data streams. In *Proc. ISWC 2008*, pages 33–40, Washington, DC, USA, 2008. IEEE Computer Society.