Towards Bi-directional Dancing Interaction

Dennis Reidsma, Herwin van Welbergen, Ronald Poppe,
Pieter Bos, and Anton Nijholt

Human Media Interaction Group
University of Twente, Enschede, The Netherlands
{reidsma,welberge,poppe,anijholt}@ewi.utwente.nl
http://hmi.ewi.utwente.nl/

Abstract. Dancing is an entertaining form of taskless interaction. When interacting with a dancing Embodied Conversational Agent (ECA), the lack of a clear task presents the challenge of eliciting an interaction between user and ECA in a different way. In this paper we describe our Virtual Dancer, which is an ECA that invites a user to dance. In our system the user is monitored using global movement characteristics from a camera and a dance pad. The characteristics are used to select and adapt movements for the Virtual Dancer. This way, the user can dance together with the Virtual Dancer. Any interaction patterns and implicit relations between the dance behaviour of the human and the Virtual Dancer should be evoked intuitively without explicit appeal. The work described in this paper can be used as a platform for research into natural animation and user invitation behavior. We discuss future work on both topics.

1 Introduction

Embodied Conversational Agents are usually sent into the world with a task to perform. They are asked to provide information about theater performances, engage the user in a training activity, sell a mortgage or help a user to successfully complete a hotel reservation. Users are often interested in interacting with these ECAs since they have an interest in the task to be performed. Since the user's focus is on the task, any nonverbal behavior exhibited by the ECA that aims at engaging the user will have a relatively low impact.

Our Embodied Agent, the Virtual Dancer (Fig. 1) tries to invite and engage the user, with the sole purpose of having an interaction. Existing dance-related entertainment applications usually introduce a task. The user should hit targets, stamp in certain patterns on a dance pad or mimic sequences of specific poses, gaining high scores by doing it fast, precise, or to the beat. We drop even that incentive. The user is simply invited to dance together with the Virtual Dancer; any interaction patterns and implicit relations between the dance behaviour of the human and the Virtual Dancer should be evoked intuitively without explicit appeal.

Viewing dancing as a taskless interaction gives us the opportunity to investigate more subtle aspects of engaging and inviting behavior in isolation, without

the distraction of a concrete task that must be performed. Letting go of the goal-directed task presents us with the challenge of eliciting an interaction between user and ECA in a different way. The user should first be seduced to enter into interaction. When the user is involved with the application, the system should establish progressively more complex interaction patterns with the user, without explicit 'game rules' or commands, yet in a way that is clear enough to be able to say when the interaction is 'successful' or not. Achieving that will be a significant contribution to the field of engaging and entertaining ECAs.

Fig. 1. Screenshot of the Virtual Dancer

The basic idea of our application is to monitor global movement characteristics of the user, and then use those characteristics to select and adapt movements for the Virtual Dancer. We describe the modules that we have built, including the animation system, the beat detection and the computer vision observation. We also describe the initial interaction models with which we try to achieve interesting interaction patterns. Furthermore, we present our short-term plans to extend these interaction models and evaluate their impact.

2 Related Work

Applications of dancing avatars exist in several variations. In some cases, the main reason for working with dance is the fact that dancing provides an interesting domain for animation technology. Perlin *et al.* and Mataric *et al.* focus

on animation specification and execution [1, 2] within the dancing domain. Shiratori *et al.*, Nakazawa *et al.* and Kim *et al.* [3–5] research the dance as a whole. They describe the regeneration of new dance sequences from captured dances. Captured sequences are segmented into basic moves by analysis of the movement properties and, optionally, the accompanying music. Then they are assembled into sequences using motion graphs, aligning the new dance to the beat structure of the music. Chen *et al.* use the traditional Chinese Lion Dance as domain for their function based animation, focussing on the possibilities for style adaptation: exaggeration, timing and sharpness of movements [6].

While above work focusses on the dance itself, we take this research one step further and look at the interaction with a human dancer. Ren *et al.* describe a system where a human can control the animation of a virtual dancer [7]. Computer vision is used to process the input from three cameras. The fact that they use a domain with a limited collection of known dance forms (swing dancing) allows them to obtain a very detailed classification of dance moves performed by the human dancer. The classified dance moves are used to control the animation of a dancing couple. For the physical dancing robot Ms DanceR [8], a dance robot that can be led through the steps of a waltz, the interaction between human and artificial dancer focusses on the mutually applied forces between a dancing couple. Detection of applied forces is used to determine the appropriate movements for the robot.

Our Virtual Dancer is not *controlled* by a human, but actively participates in the interaction process in which both the human and the Virtual Dancer influence each other and let themselves be influenced in turn.

3 Architecture

The architecture of our system is shown in Fig. 2. In our setup, the Virtual Dancer is projected on a screen. A user is observed by a camera that is placed above the screen, monitoring the area in front of the screen. A dance pad is placed in front of the screen. Our setup further includes a sound system with speakers to play the music to which both the user and the Virtual Dancer can dance. The different components of the architecture are described in this section.

3.1 Beat Detection

Both tempo and beats are detected from the music using a real-time beat detector. From a comparison of detectors in [9] it was found that the feature extracting and periodicity detection algorithm of Klapuri [10] performs best. The first part of this algorithm is an improvement of the algorithm of Scheirer [11]. Both algorithms use a number of frequency bands to detect accentuation in the audio signal, and employ a bank of comb filter resonators to detect the beat. Klapuri improved the accentuation detection and comb filter banks. The biggest difference between the two is the way these comb filter banks are used to detect periodicity. Scheirer's algorithm uses filter banks that can be used to detect

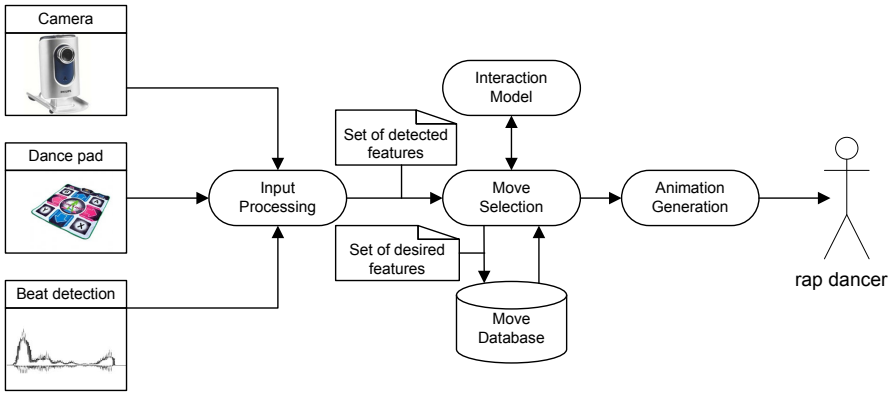

Fig. 2. Architecture of the Virtual Dancer system

tempo and beat directly. The comb filter with the highest output is selected. Klapuri uses many more, and slightly different, filters which detect periodicity in a broad range. A probabilistic model is used to detect the tactus, tatum and measure. For the Virtual Dancer we implemented Klapuri's algorithm.

3.2 Video Analysis

A single video camera is used to observe the user. Ideally, one would like to have complete knowledge about the movements of the user. This requires recovery of the pose of the user, usually described in terms of joint angles or limb locations. However, this is too demanding for our application for a number of reasons. Firstly, since only a single camera is used, no depth information is available. This makes it hard, if not impossible, to fully recover a complete pose. Secondly, there can be large variations in appearance and body dimensions between users. These can be estimated from the video, but this is hard since no pose information is present at first. An alternative is to add an initialization phase, in which these parameters can be estimated. However, such a phase prevents the more spontaneous use of our application that we aim for. Finally, when the movements of the user are known, our Dancer needs to extract certain characteristics and react to them. When poses are described in great detail, it is non-trivial how these can be used in the dancer's move selection phase (see also Section 3.5). Therefore, in our approach we use global movement features. These have a couple of advantages: they can be extracted more robustly, model variations between persons implicitly and can be used to determine selection criteria in the move selection phase. The set of characteristics U that we extract from the video are summarized in Table 1. We distinguish between discrete values, that are either 0 or 1, and continuous values, that can have any value in the $[0 \ldots 1]$ interval.

As a first step, we extract the user's silhouette from the video image (Fig. 3(a)). This method requires a known background model, but it is computationally inexpensive. Moreover, silhouettes encode a great deal of information about the

Characteristic	Type	Source
BODY_HIGH	discrete	center of mass detector
BODY_LOW	discrete	center of mass detector
HORIZONTAL_ACTIVITY	continuous	center of mass detector
HAND_LEFT_TOP	discrete	radial activity detector
HAND_LEFT_SIDE	discrete	radial activity detector
HAND_RIGHT_TOP	discrete	radial activity detector
HAND_RIGHT_SIDE	discrete	radial activity detector
RADIAL_ACTIVITY	continuous	radial activity detector
FEET_MOVE_INTENSITY	continuous	dance pad

Table 1. Summary of user characteristics, their types and input source

user's pose. We employ two image processes to recover the movement characteristics. We describe these below.

Center of Mass Detector The center of mass detector uses central moments to determine the 2D location of the silhouette's center of mass (CoM). Most changes in the silhouette due to pose changes will have only a small effect on the CoM. However, jumping or stretching the arms above the head will result in a higher CoM, whereas bending and squatting will lower the CoM considerably. Two thresholds are set on the vertical component of the CoM: a low threshold and a high threshold. If the CoM is below the low threshold, the BODY_LOW value is set. Similarly, if the CoM is above the high threshold, the BODY_HIGH value is set. The values of the thresholds are determined empirically. Furthermore, the average difference in successive values of the horizontal component is a measure for the HORIZONTAL_ACTIVITY value. This value is normalized with respect to the silhouette's width.

Radial Activity Detector When the CoM is calculated, we can look at the distribution of silhouette pixels around the CoM. We are especially interested in the extremities of the silhouette, which could be the legs and arms. Therefore, we look at foreground pixels that lie in the ring centered around the CoM (Fig. 3(b)). The radius of the outer boundary equals the maximum distance between silhouette boundary and CoM. The radius of the inner boundary equals half the radius of the outer boundary. The ring is divided into 12 radial bins of equal size (see also Fig. 3(c)). A threshold on the percentage of active pixels within a bin is determined empirically. If the threshold within a bin is exceeded, the HAND_LEFT_SIDE, HAND_LEFT_TOP, HAND_RIGHT_TOP and HAND_RIGHT_SIDE values are set, for the corresponding bins. In addition, the RADIAL_ACTIVITY value is determined by the normalized average change in bin values between successive frames.

(a) (b) (c)

Fig. 3. (a) Extracted silhouette (b) Center of mass with ring (c) Radial activity bins

3.3 Dance Pad

To monitor feet movement we use a Dance Dance Revolution (DDR) pad. This pad contains eight 'buttons', that are pressed if a foot is placed on them. We do not force users to restrain their movement to the floor area covered by the pad. If the pad is used, we determine the FOOT_MOVE_INTENSITY characteristic by looking at the number of button presses that occurs in a given period of time.

3.4 Move Database

A human pose is described by setting the rotation values of the joints. Animations are defined as a number of keyframes that describing poses, and interpolation between them. The keyframes can be specified manually or obtained from motion capture. We can also use the location of end effectors to describe a pose. Using inverse kinematics (IK), we determine the rotation of joints involved in the animation. For example, we could describe the path of a hand and automatically calculate the rotation of the shoulder and elbow needed to place the hand on this path. Figure 4 visualizes the movement paths for the hands as defined in the 'car' move. Both hands move along a segment of an ellipse. Those paths are defined as a set of functions over time with adaptable movement parameters ($x(t,a)$, $y(t,a)$ and $z(t,a)$). The parameter t ($0 \leq t \leq 1$) indicates the progress of the animation. The parameter a can be seen as an amplitude parameter and is used to set the height of the hand's half-ellipse move. In a similar way, we defined formulae that describe joint rotation paths. We combine keyframe animation, rotation formulae for the joints and path descriptions for limbs and body center. Currently, we do not combine motion capture data with the other animation types.

For each move we stored key positions in time, that are aligned to the beats in the animation phase. Key points can have different weights, according to how important it is that they are aligned to a musical beat. For example, the time instance where a hand clap occurs is stored as a key position with high weight since we would like our Dancer to clap to the beat rather than between just anywhere.

Fig. 4. Samples of the 'car' move, in which the hands are rotated in a driving movement. The path of the hands is shown by the white spheres.

3.5 Move Selection

The move selection is built to choose moves based on the current state of the Dancer and the characteristics of the dancing behaviour of the human (see Table 1). A mapping from this information to information stored about each move determines the selection of the next move of the Dancer. To support this mapping, each move m in the database is annotated with its type (e.g. 'dancing' or 'bored') and the default duration. Furthermore, we manually set values for the each possible move characteristic $B^m \in M$. Currently, M (the set of characteristics that a dance move can have) contains only a few components (HIGH_LOW, ACTIVITY, SYMMETRY, HAND_POSITION, REPEATING and DISPLACEMENT) but the set can be extended at any time.

To select a move, we first calculate the set of observed characteristics $O \in \wp(U)$ displayed by the human dancer. These characteristics are then mapped to a set of desired characteristics in the dance move ($D \in \wp(M)$) using mapping G:

$$G := U \longrightarrow M \tag{1}$$

By comparing the desired values D_i with the value of the corresponding characteristic B_i^m for each move m in the database the most appropriate move is determined. The mapping G is defined by the interaction model. A matching score s_m is calculated for each move:

$$s_m = \sum_i (1 - |D_i - B_i^m|) w_i \tag{2}$$

w_i is the weight of characteristic i. The weights are normalized to make sure they sum up to 1. The probability that a certain move m is selected is proportional to its score s_m.

3.6 Animation Generation

Dancing to the Beat One important feature in any dance animation is the alignment of the dance movements to the beat of the music. Our approach to this is as follows. Whenever a new move is being planned, the beat detector module is queried for the current tempo and beat pattern of the music. This information is used to produce a vector of predictions of beats in the near future. The set of key points from the selected move and the beats from the beat prediction vector are time-aligned to each other using an algorithm inspired by the event-aligner from [12] (see Fig. 5). This algorithm takes into consideration the importance of the key points, the relative position of key points in the move, the beats in the vector and the strength of the beats.

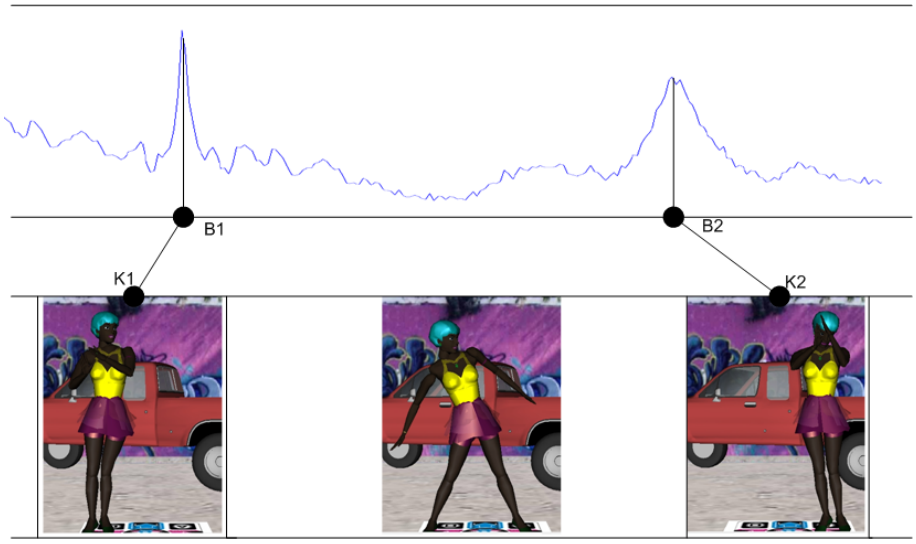

Fig. 5. Move alignment to the beat: beat B_1 is aligned to keyframe K_1; beat B_2 is aligned to keyframe K_2

Interpolation To generate the transition from one dancing move to the next, we make use of a simple interpolation algorithm. The root position is linearly interpolated from the end position of the previous animation to the start position of the next animation. If there is no significant feet displacement, all joint rotations are interpolated. If significant feet displacement is needed to get from the previous animation to the next, the Dancer makes two intermediary steps. The movement of the feet and the vertical movement of the root are specified by the step formula described in [13].

3.7 Interaction Model

The interaction model is implemented as a state machine. Currently it has the states 'bored' 'invite' and 'dance'. During the 'bored' state, the Dancer exhibits bored behavior such as scratching her head or inspecting her fingernails. If the presence of a human is detected by the video analysis system, she tries to invite him or her to dance with her. This behavior is performed using nonverbal invitation gestures. Once the user steps on the dance pad, the dancing starts.

We implemented the dancing process as alternating phases of the ECA following and leading the user (or at least attempting to lead the user). 'Following' means dancing with movement properties that are similar to what the user shows. 'Leading' involves varying the movement properties considerably in one or more dimensions. The implicit intention is to get the the user to adapt in reaction. Based on the state of the Dancer, the mapping G and the weights w_i are adapted. This gives us a system which allows for all kinds of different dimensions of interactivity. The human and the Virtual Dancer will have a chance to influence the other. The can also observe the reactions to thatinfluence as well as the attempts at influencing by the other and can signal their reaction to that.

4 Results

The system described in this paper has been implemented and was exhibited on several smaller and larger occasions[1]. It has proved to be very robust. At the CHI Interactivity Chamber the program had been running non stop for two days in a row without needing any other intervention than occasionally making new snapshots of the changing background. The system currently runs on two 'average' laptops, one running the computer vision processes and the other running the real-time beat detection and all other processes for controlling and animating the Virtual Dancer, including the interaction algorithms.

During those demonstration events, many people interacted with the installation. Some of the interactions were recorded on video. The resulting recordings will be used to get a first idea of the interaction patterns to which people react as well as of the types of reactions. Then we will use this knowledge to improve the user observation modules and the interaction models to get closer to our aim of a system where interaction is not enforced but enticed.

5 Future Work

The work described in this paper can be used as a platform for research into natural animation, mutual dance interaction and user invitation behavior. This section describes our ongoing work on these topics.

[1] See Figure 1 for a screenshot and `http://hmi.ewi.utwente.nl/showcases/The Virtual Dancer/` for demos and movies.

5.1 Animation

Merging animations described by mathematical formulae with animations derived from motion capture by simply applying animating some joints with the one, and some with the other specification, results in unrealistically looking animations. The formula-based approach looks very mechanical, compared to the movements obtained by motion capture, which contain a high level of detail. However, the formula-based animation gives us a high level of control on joint movements, which allows us to modify the path of movement and the amount of rotation of joints in real time. We have less control over motion captured movements. Currently, we can only align motion capture movement to the beat of the music and adapt its velocity profile. We would like to be able to modify not only the timing, but also the position of body parts in the animation.

The lack of control is a general problem in motion captured animation. There is much ongoing research in the domain of adaptation of motion capture data. A motion capture frame can be translated to IK data for certain body parts, so that the translation path of these parts can be adapted [14]. Motion capture data can be divided in small portions. Then, transitions between motions that show many similarities can be defined, which results in a motion graph [5, 15]. Suitable animations are created by selecting a path through the graph that satisfies the imposed animation constraints. Motion capture can also be used as 'texture' on generated or handmade keyframe animations [16], which improves the detail and naturalness of the animations. Different motion capture animations could be blended together to create new animations [17]. The movement style obtained from motion capture can be used to enhance animation generated by bio-mechanical models [18]. We plan to adapt our motion capture data to gain expressiveness of and control over our animations using such techniques as mentioned above.

5.2 Mutual Dance Interaction

Many issues still need to be resolved if we want to achieve the kind of interaction patterns that we are aiming for. Amongst others, the system should be able to detect when its attempts at leading are successful (see e.g. [19], where this is partially done for two dancing humans), the system should have a (natural) way to signal acknowledgement and approval to the user when the user reacts appropriately to the leading attempts of the system, the system should be able to detect situations when the user is attempting to lead, the interaction pattern should become progressively more complex when the first interaction is established and we should determine which dimensions in the dance moves are most suitable for variation. Such topics will shape some of our short-term future work on this project.

5.3 Invitation

In our ongoing work centered around the Virtual Dancer installation, one of the themes is the *invitation of users to join the dance*. Because there is no practical

application associated to the installation, users will have no compelling reason to engage in interaction with it. At the moment, the Virtual Dancer initiates the interaction by making inviting gestures to the user. This is a kind of 'enforced' interaction: without warning or consent the user finds herself in the middle of an ongoing interaction. This is about as subtle as a television advertisement or an outbound sales agent who still needs to meet his quota. In real life, interaction often starts in a more subtle way. For example, as described in [20], people use all kinds of mechanisms to signal their willingness and intention to interact, even before the first explicit 'communication' is started. Peters describes a theoretical model for perceived attention and perceived intention to interact. Primarily gaze and body orientation, but also gestures and facial expression, are proposed as inputs for synthetic memory and belief networks, to model the level of attention directed at the agent by an other agent, virtual or human. The resulting attention profile, calculated over time, is used to determine whether this other agent is perceived as 'intending to interact'. Quote from [20]: "For example, peaks in an otherwise low magnitude curve are interpreted as social inattention or salutation behaviors without the intention to escalate the interaction. A profile that is of high magnitude and increasing is indicative of an agent that has more than a passing curiosity in an other and possibly an intention to interact. Entries regarding locomotion towards the self actively maintain the level of attention in cases where the profile would otherwise drop due to the eyes or head being oriented away."

We intend to use these ideas to experiment with behavior that entices people in an implicit way into interaction with the Dancer. Simulations and models for eye contact and attention of the type described above will be implemented using robust computer vision and the eye contact detection technology of [21].

Acknowledgements

The authors would like to thank Moes Wagenaar and Saskia Meulman for performing the dance moves that are used in this work. Furthermore, we thank Hendri Hondorp, Joost Vromen and Rutger Rienks for their valuable comments and their contributions to the implementation of our system.

References

1. Perlin, K.: Real time responsive animation with personality. IEEE Transactions on Visualization and Computer Graphics **1**(1) (1995) 5–15
2. Mataric, M., Zordan, V., Williamson, M.: Making complex articulated agents dance. Autonomous Agents and Multi-Agent Systems **2**(1) (1999) 23–43
3. Shiratori, T., Nakazawa, A., Ikeuchi, K.: Rhythmic motion analysis using motion capture and musical information. In: Proc. of 2003 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. (2003) 89–94
4. Nakazawa, A., Nakaoka, S., Kudoh, S., Ikeuchi, K.: Digital archive of human dance motions. In: Proceedings of the International Conference on Virtual Systems and Multimedia (VSMM2002). (2002)

5. Kim, T., Il Park, S., Yong Shin, S.: Rhythmic-motion synthesis based on motion-beat analysis. ACM Transcactions on Graphics **22**(3) (2003) 392–401
6. Chen, J., Li, T.: Rhythmic character animation: Interactive chinese lion dance. In: Proc. of International Conference on Computer Animation and Social Agents. (2005)
7. Ren, L., Shakhnarovich, G., Hodgins, J.K., Pfister, H., Viola, P.: Learning silhouette features for control of human motion. ACM Transcactions on Graphics **24**(4) (2005) 1303–1331
8. Kosuge, K., Hayashi, T., Hirata, Y., Tobiyama, R.: Dance Partner Robot –Ms DanceR–. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2003). (2003) 3459–3464
9. Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., Cano, P.: An experimental comparison of audio tempo induction algorithms. IEEE Transactions on Speech and Audio Processing (2006) In press.
10. Klapuri, A., Eronen, A., Astola, J.: Analysis of the meter of acoustic musical signals. IEEE Transactions on Speech and Audio Processing (2006)
11. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. Journal of the Acoustical Society of America **103**(1) (1998) 558–601
12. Kuper, J., Saggion, H., Cunningham, H., Declerck, T., de Jong, F., Reidsma, D., Wilks, Y., Wittenburg, P.: Intelligent multimedia indexing and retrieval through multi-source information extraction and merging. In: 18th International Joint Conference of Artificial Intelligence, Acapulco, Mexico (2003) 409–414
13. Meredith, M., Maddock, S.: Using a half-jacobian for real-time inverse kinematics. In: International Conference on Computer Games: Artificial Intelligence, Design and Education. (2004)
14. Meredith, M., Maddock, S.: Adapting motion capture using weighted real-time inverse kinematics. ACM Computers in Entertainment (2005)
15. Kovar, L., Gleicher, M., Pighin, F.H.: Motion graphs. ACM Transcactions on Graphics **21**(3) (2002) 473–482
16. Pullen, K., Bregler, C.: Motion capture assisted animation: texturing and synthesis. In: 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'02), New York, NY, USA, ACM Press (2002) 501–508
17. Safonova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Transcactions on Graphics **23**(3) (2004) 514–521
18. Liu, K.C., Hertzmann, A., Popovic, Z.: Learning physics-based motion style with nonlinear inverse optimization. ACM Transcactions on Graphics **24**(3) (2005) 1071–1081
19. Boker, S., Rotondo, J.: Symmetry building and symmetry breaking in synchronized movement. In Stamenov, M., Gallese, V., eds.: Mirror Neurons and the Evolution of Brain and Language. (2003) 163–171
20. Peters, C.: Direction of attention perception for conversation initiation in virtual environments. In Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T., eds.: Intelligent Virtual Agents, 5th International Working Conference. (2005) 215–228
21. Shell, J., Selker, T., , Vertegaal, R.: Interacting with groups of computers. Special Issue on Attentive User Interfaces, Communications of the ACM **46**(3) (2003)