Towards Augmented Choreography

Diego Bernini[1], Giorgio De Michelis[1], Mauro Plumari[1], Francesco Tisato[1], and Michele Cremaschi[2]

[1] DISCo, University of Milano-Bicocca, viale Sarca 336, 20126 Milano, Italy
{bernini,demichelis,plumari,tisato}@disco.unimib.it
[2] Residenza Teatrale InItinere, via Valmarina 25, 24129 Bergamo, Italy
michele.cremaschi@gmail.com

Abstract. Choreographers are interested in enriched performances where virtual actants play together with live performers. Augmented Choreography can be viewed as the definition of how perceptions generated from the environment turn into commands that influence the environment itself and, in particular, virtual actants. This paper introduces a modular and extensible architecture that supports the flexible and dynamic definition of augmented choreographies and presents an experimental application.

Keywords: augmented choreography, interactive performance, multimodal interaction, performance design, virtual puppetry.

1 Introduction

Choreography [1] is the art of designing sequences of movements (*choreographies*) performed by actants. In ballet and stage dancing choreographies define sequences of dance steps that are synchronized with music beats and, in general, with musical events: for example, choreography might prescribe that during a musical phrase one dancer has to withdraw from another dancer. Choreography may also prescribe how actants behave (for example, moving their arms or jumping) according to the behavior of a coryphaeus. Finally, choreography may prescribe stage effects (for example, lighting) in correlation with music or movements. Ultimately, choreography dictates how perceptions are mapped into performers' actions.

New media technologies enhance the potential of performing arts by introducing virtual actants and by exploiting multifarious sensing technologies. For example, Latulipe et al. [2] explore the design space of dance and technology presenting a specific show of interactive dance "Bodies/AntiBodies". James et al. [3] describe Lucidity, a show of "interactive choreography".

Augmented Choreography is choreography where perceptions, actions and mapping rules are augmented by exploiting hardware and software technologies. For example, one or more virtual actants i.e., computer-projected body animations, move according to music, movements of real dancers and clapping.

A.L. Brooks (Ed.): ArtsIT 2011, LNICST 101, pp. 10–17, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

A major challenge is how to design and develop cost-effective IT systems that allow artists to realize Augmented Choreography (ITAC). At a first glance, they seem close to widespread real-time and interactive systems, ranging from automation to video-gaming, which ultimately turn input stimuli into output actions. However, these systems are mostly "vertical" in the sense that they are designed and optimized once and for all to perform a specific task and to exploit specific technologies.

Augmented choreographies should adaptively exploit heterogeneous perception and actuation flows, which not only depend on the rapid technological evolution of sensing and actuation devices, but also rely on different conceptual models of the environment and, in particular, of the performance space. This leads to a high degree of technological dependency, which combinatorially explodes as the system must integrate heterogeneous technologies and computational models. On the other side, there is a wide semantic gap between the languages and cultures of technicians and artists. The result is that the design, development and evolution of augmented choreographies involve a close and painful collaboration between artists and technicians.

A sound architectural approach may reduce the combinatorial explosion of complexity by carefully separating technological issues from choreographic aspects. A further step is to devise a technology-neutral language that allows choreographers to define augmented choreographies in a seamless and user-friendly way.

Section 2 presents basic ideas about architecture and linguistic issues. Section 3 presents an experiment in the virtual puppetry area. Section 4 compares our system to significant related work, while Section 5 proposes some conclusions and future developments.

2 Augmented Choreography

2.1 Architecture

The architecture of an ITAC system should carefully separate technological issues, which are wrapped by peripheral software components, from mapping functions, which conceptually define the choreography and are up to a Mapper component.

Sensing Wrappers encapsulate device-specific technicalities and produce Perceptions, i.e., symbolic representations of events localized in Perception Spaces. For example (Figure 1), a Microsoft Kinect wrapper provides perceptions localized in the 3D space of a stage, a camera wrapper provides perceptions localized in the 2D space of an image and a microphone wrapper provides perceptions localized in a 1D space modeling sound loudness.

Actuation Wrappers encapsulate device-specific technicalities and are controlled by Actions i.e., symbolic representations of expected actions localized in device-specific Action Spaces. For example, a screen wrapper receives actions defining where graphical objects must be visualized in the 2D space of the screen.

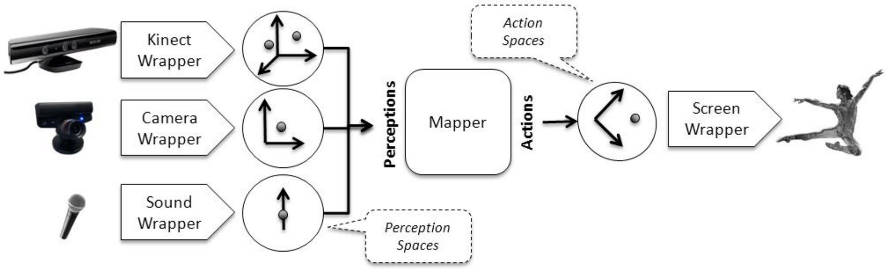

Fig. 1. ITAC architecture

The Mapper encapsulates the choreography i.e., how localized Perceptions are mapped into localized Actions. For example, it defines how the position of a hand of a physical performer, which is perceived in the 3D Kinect space, must be mapped into the position of a foot of a virtual performer in the 2D screen space.

The proposed architecture clearly separates technological issues from conceptual aspects. Therefore it reduces the technology dependency of artists and improves the flexibility and extensibility of the ITAC system. New devices can be added by realizing proper wrappers that generate Perceptions (or receive Actions) expressed in a symbolic and technology-neutral style. Conversely, choreographies can be modified without dealing with technological details.

2.2 Choreography as Translation

Though the proposed architecture enhances the separation between technological and conceptual concerns, defining the behavior of the Mapper is still a complex issue as long as the Mapper is a software component that must be explicitly programmed to realize a specific choreography. The next step is to devise a generalized Mapper, which acts as interpreter of choreographies defined in linguistic, artist-oriented terms. For example, the choreography of a dance should be defined by stating how beats (i.e., perceptions) expressed in musical notation are mapped into movements (i.e., actions) expressed in choreutic notation.

The role of Mapper is to *translate* Perceptions into Actions. This can be generalized in linguistic terms:

- An input language L_I is defined by a grammar G_I whose tokens t_I model Perceptions localized in Perception Spaces.
- An output language L_O is defined by a grammar G_O whose tokens t_O model Actions localized in Action Spaces.
- A choreography $C_{I,O}$ defines translation rules from well-formed strings S_I^i of L_I (i.e., G_I-compliant sequences of t_I tokens) to well-formed strings S_O^J of L_O (i.e., G_O-compliant sequences of t_O tokens).

Established results from the area of Language Theory can be exploited to model more and more sophisticated choreographies. For simplicity and according to the experiment presented in Section 3, in the following "performer" and "puppet" denote a physical and a virtual performer respectively.

Direct Mapping Choreography. In the simplest case both L_I and L_O are context-free languages and the translation rules (i.e., the choreography) just translate perceptions into actions. For example, the positions of the skeleton joints of a performer turn into positions of the corresponding skeleton joints of a puppet. Mirroring or more complex effects can be easily defined. For example, an Action describing the position of a puppet joint can depend both on the Perception of the position of a performer joint in a Performer Space and on the Perception of a sound in a Sound Space.

Behavioral Choreography. More complex behaviors can be achieved by translating an input token into a sequence of output tokens (in Language Theory parlance, L_O is a regular language). For example, the perception of a new position of the performer's hand might turn into a sequence of positions of the puppet's hand. Moreover, if the input language L_I too is a regular language whose legal strings are modeled by a state automaton, performers' behaviors (i.e., specific strings of L_I) can be recognized. For example, a sequence of repeated movements from left to right of the performer's hand can turn into a puppet's movement that causes it disappearing.

Time Sensitive Choreography. Choreography is intrinsically tied to the concept of movement, therefore to the concept of *time* [4]. Timing is relevant both to recognize specific input behaviors (for example, fast movements) and to drive output behaviors.

This implies that both input and output tokes are *time-stamped*. Timestamps are referred to a unique reference time. On the input side timestamps are exploited to recognize specific behaviors. On the output side timestamps are exploited to generate output commands according to a proper timing. For example, like in a physical dance performance, the movements of the orchestra conductor are perceived as timed input tokens and analyzed to recognize beats, whose frequency leads the timing of the commands delivered both to sound sources (be they humans or software) and to dancing puppets. Time sensitivity [4] supports the realization of choreographies that include advanced domain-specific issues. For example, the choreography might include the concept of laziness of a puppet to drive how fast it reacts to a command. The choreography could also include a dynamic model instead of a simple kinematic model to take into account the mass of the puppet.

Choreographers as Performers. The Mapper in the basic architectural scheme of Figure 1 is an interpreter of choreographies that define translation rules from well-formed strings S_I^i of L_I to well-formed strings S_O^J of L_O. Choreographies, though sophisticated, are assumed to be statically defined by a Choreographer *before* the performance takes place. The ultimate step is to lift up the role of the Choreographer to that of a full-fledged Performer. This means that she/he is a Performer that dynamically changes the choreography i.e., the translation rules *during* the performance. Ultimately, the separation between Performer and Choreographer vanishes.

3 An Experiment: Augmented Puppetry

Puppetry is a very ancient form of art [5]. *Computer puppetry* [6] turns in real-time the movements of a performer to the movements of an animated character (*puppet*). Coutrix et al. [7] observe that computer-mediated puppetry has been used extensively for animation production rather than for live public performances.

On December 2010 the InItinere theatrical residency contacted us to realize a system to support their future shows. Quoting their needs: *"We would like to produce a staged theatrical performance, drawing on the tradition of mimes, clowns and visual comedy augmented with digital technologies. One or more virtual puppets will be put on stage through a character that is hand drawn and video-projected rather than physical. The movements of an actor-performer should determine the virtual puppet animation. We imagine different interaction scenarios. First there could be a direct, real-time connection by the performer and the puppet motion. Then the connection could be made less directed introducing time delays…The performer should interact with the puppet through different modalities, even simultaneously…For example by gestures recognized by a camera-based system; or by a tangible cross bar…".*

Starting from these needs, we developed a modular system for augmented puppetry, which is instantiated according to artists' requirements and whose architecture is strongly based on the openness, multiplicity and continuity qualities [8]. The development and experimentation activities allowed us to highlight the problems and to define the architecture introduced in the previous section.

The current implementation includes Sensing Wrappers for Microsoft Kinect, for Nintendo Wii Balance Board and for microphones. Microsoft Kinect provides 2D images with distance information for each pixel. The Kinect Wrapper exploits open source middleware to provide perceptions that model the 3D position of fifteen skeleton points of a performer moving in front of the camera. The Nintendo Wii Balance Board Wrapper computes the projection of the barycenter of the performer in the bi-dimensional Board Space. Finally, the Microphone Wrapper produces the sound intensity detected by a microphone.

The Puppet Visualization Wrapper executes Actions defining positions of puppet joints. It exploits the Animata real time animation software, an open source project maintained by the Kitchen Budapest lab. Puppets are animated according to the *skeletal animation model* [9]. A character is represented in two parts: a surface representation used to draw the character (*skin* or *mesh*) and a set of *joint*s connected through *bones* (*skeleton*) used to animate the mesh. Various computational methods can be exploited to animate the skeleton. Our system exploits inverse (*goal-directed*) methods allowing the position of some joints only to be specified. The positions of other joints are automatically computed by taking into account previous positions, bone lengths and joint angles.

The Mapper component is at the heart of the system. Different versions of Mapper realize different scenarios i.e., different choreographies.

The first scenario realizes a direct mapping choreography by projecting the perceived 3D positions of the skeleton points of the performers into 2D positions on the screen of the corresponding joints of the puppet.

The second scenario too realizes a direct mapping choreography, but the Mapper simulates a cross bar by projecting the 3D positions of the performer hands into 2D positions of the puppet hands; the movements of other puppet joints are computed by Puppet Visualization Wrapper according to the skeleton animation model.

The third scenario realizes a time sensitive choreography, which is similar to that of the first scenario, but actions that control puppet positions are delayed.

Finally, the fourth scenario (Figure 2) simultaneously exploits all the three Sensing Wrappers. The performer wears a microphone and stands in front of a Kinect sensor on a Nintendo Balance Board. Performer hands are mapped to puppet hands as in the second scenario. The position of the performer barycenter perceived by the Balance Board Wrapper is mapped to actions that control the tilt of the boat, thus the position of the puppet foots. Moreover the vocal intensity sensed by the Microphone Wrapper is linked to the character mouth: when the voice intensity overcomes a threshold, the mouth moves.

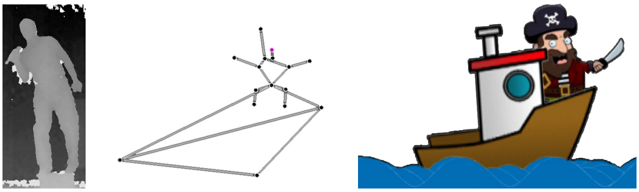

Fig. 2. Multi-modal animation

4 Related Work

Several live digital animation systems have been proposed in literature.

A first class of works aims at translating the interaction of the user/performer with different interfaces to corresponding actions of a virtual character in a virtual world.

Mazalek et al. [10] present an embodied puppet interface that translates the performer body movements to a virtual character, focusing on the fine grained control of the character. Mazalek and Nitsche [11] show a system which exploits a tangible marionette in order to control the correspond virtual one.

Liikkanen et al. [12] present PuppetWall, a multi-user, multimodal system intended for digital augmented puppetry. It provides functionalities to control puppets and manipulate playgrounds including background and puppets. PuppetWall exploits hand movement tracking, a multi-touch display and speech recognition.

Shin et. al. [13] propose an approach to generate realistic motion for a character in real-time while preserving the characteristics of captured performance motions as much as possible.

These solutions are focused on digital animation reproducing human motion. Our approach is more general because it aims to augment the performance through a virtual puppet, whose behavior is influenced by a wide range of perceptions including, as a particular case, the movements of a human performer.

Other systems are more flexible with respect to the correspondence between performer motion and puppet animation. For example, Dontcheva et al. [14] present an animation system where the relation between the animator and the character can be either explicitly defined or inferred from similarities. CoPuppet [15] is a system for multimodal, collaborative puppetry where performers, or even audience members, affect different parts of a puppet through gestures and voice.

Finally, several works share our vision of an open system accessible to artists and choreographers.

Neff et al. [16] propose an approach for mapping 2D mouse input to high-dimensional skeleton space via *correlation maps* which transform the input to meaningful abstract output.

Samanci et al. [17] propose a framework for interactive storytelling. It exploits an interaction technology based on computer vision and full-body tracking. The framework provides a rich set of interactions and supports simultaneous multi-point and multi-user input.

Vasilakos et al. [18] present a system offering to the performance artists a creative tool to extend the grammar of the traditional theatre. Actors and dancers at different places are captured by multiple cameras and their images are rendered in 3D form so that they can play and dance together on the same place in real-time.

Kuşcu and Akgün [19] propose an approach to interactive performance systems which is very close to our proposal. They define a component-based architecture which resembles our architecture, though they do not highlight the possible different kinds of mappings and the role of suitable choreographer-oriented languages. Like us, they highlight the need of taking timing into account and propose a timed state machine allowing choreographers to edit the audiovisual behavior of a choreography system.

5 Conclusions and Future Work

The ITAC framework we developed is a good starting point for radically changing the way technologists interact with choreographers: from trying to realize what is requested by an artist, to providing she or he with the possibility of autonomously defining augmented choreographies.

In order to bring our system to completion, we plan to work on two parallel directions: on the one side, we plan to continue our collaboration with InItinere and to collaborate with other choreographers in order to augment our experience; on the other side, we plan to do an ethnography of choreographers at work in order to deeply understand the nature of their work. In fact, an ITAC system must be evaluated from two viewpoints: its capability of orchestrating movements so that the effects required by the artist are reached and its capability to be directly exploited by the choreographer without distracting them from their way of working.

References

1. Blom, L.A., Chaplin, L.T.: The Intimate Act Of Choreography. University of Pittsburgh Press (1982)
2. Latulipe, C., Lottridge, D., Wilson, D., Huskey, S., Word, M., Carroll, A., Carroll, E., Gonzalez, B., Singh, V., Wirth, M.: Exploring the design space in technology-augmented dance. In: Proceedings of the 28th of the International Conference on Human Factors in Computing Systems - CHI EA 2010, p. 2995. ACM Press, New York (2010)

3. James, J., Ingalls, T., Qian, G., Olsen, L., Whiteley, D., Wong, S., Rikakis, T.: Movement-based interactive dance performance. In: Proceedings of the 14th Annual ACM International Conference on Multimedia - MULTIMEDIA 2006, p. 470. ACM Press, New York (2006)
4. Micucci, D., Oldani, M., Tisato, F.: An Architectural Model for Time Awareness. Aracne editrice (2006), http://www.disco.unimib.it/go/215281425
5. Blumenthal, E.: Puppetry and Puppets: An Illustrated World Survey. Thames & Hudson (2005)
6. Sturman, D.J.: Computer puppetry. IEEE Computer Graphics and Applications 18, 38–45 (1998)
7. Coutrix, C., Jacucci, G., Spagnolli, A., Ma, L., Helin, M., Richard, G., Parisi, L., Roveda, S., Narula, P.: Engaging spect-actors with multimodal digital puppetry. In: Proceedings of NordiCHI 2010, p. 138. ACM Press, New York (2010)
8. De Michelis, G.: The Swiss Pattada. Interactions 10, 44–53 (2003)
9. Murray, C.: A user-based evaluation of skeletal animation techniques in graph interaction. In: Proceedings of the 2005 Asia-Pacific Symposium on Information Visualisation, vol. 45, pp. 7–14. Australian Computer Society, Inc., Darlinghurst (2005)
10. Mazalek, A., Chandrasekharan, S., Nitsche, M., Welsh, T., Clifton, P., Quitmeyer, A., Peer, F., Kirschner, F., Athreya, D.: I'm in the game. In: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction - TEI 2011, p. 129. ACM Press, New York (2011)
11. Mazalek, A., Nitsche, M.: Tangible interfaces for real-time 3D virtual environments. In: Proceedings of the International Conference on Advances in Computer Entertainment Technology - ACE 2007, p. 155. ACM Press, New York (2007)
12. Liikkanen, L.A., Jacucci, G., Huvio, E., Laitinen, T., Andre, E.: Exploring emotions and multimodality in digitally augmented puppeteering. In: Proceedings of the Working Conference on Advanced Visual Interfaces - AVI 2008, p. 339. ACM Press, New York (2008)
13. Shin, H.J., Lee, J., Shin, S.Y., Gleicher, M.: Computer puppetry: An importance-based approach. ACM Transactions on Graphics 20, 67–94 (2001)
14. Dontcheva, M., Yngve, G., Popović, Z.: Layered acting for character animation. In: Proceedings of ACM SIGGRAPH 2003, p. 409. ACM Press, New York (2003)
15. Adams, R., Gibson, S., Arisona, S.M., Bottoni, P., Faralli, S., Labella, A., Malizia, A., Pierro, M., Ryu, S.: Transdisciplinary Digital Art. Sound, Vision and the New Screen. Springer, Heidelberg (2008)
16. Neff, M., Albrecht, I., Seidel, H.-P.: Layered Performance Animation with Correlation Maps. Computer Graphics Forum 26, 675–684 (2007)
17. Samanci, Ö., Chen, Y., Mazalek, A.: Tangible comics. In: Proceedings of ACE 2007, p. 171. ACM Press, New York (2007)
18. Vasilakos, A., Wei, L., Nguyen, T., Thienqui, T., Chen, L., Boj, C., Diaz, D., Cheok, A., Marentakis, G.: Interactive theatre via mixed reality and Ambient Intelligence. Information Sciences 178, 679–693 (2008)
19. Kuşcu, H., Akgün, B.T.: A new approach to interactive performance systems. In: Proceedings of MULTIMEDIA 2005, p. 578. ACM Press, New York (2005)