From: AAAI Technical Report WS-96-03. Compilation copyright © 1996, AAAI (www.aaai.org). All rights reserved.

Computer Theater: Stage for Action Understanding

Claudio S. Pinhanez **Aaron F. Bobick**
Perceptual Computing Group – MIT Media Laboratory
20 Ames St. – Cambridge, MA 02139
pinhanez | bobick@media.mit.edu

Abstract

This paper argues that computer theater are an interesting domain for research in action representation, recognition, and generation. Here, *computer theater* describes experiences and systems where performers use computers in theatrical performances, both as a means to expand their bodies and as partners. Some of the most recent experiences in computer theater are surveyed and classified according to criteria borrowed from computer music research. The possible uses of action understanding are then exemplified by examining our research on script-based control of TV cameras in a cooking show.

Introduction

Action is the basis of theater[1] and, as such, needs to be fully incorporated in whatever model a computer is running during a computer-based theatrical performance. We believe the lack of good models for action is one fundamental reason for the relative absence of experiments involving theater and computers. The attempts to wire up stages or performers have been in general concerned with dance (Lovell & Mitchell 1995), only using information about the position and attitude of the actors/dancers on the stage.

The main argument of this paper is that computer theater not only requires action representation and recognition but it is also an interesting domain for action research. To support our argument we begin by examining the multiple possibilities of using computers in theatrical performances, concerning both explored and unexplored developments. Recent theatrical experiences are prefered for citation rather than old ones in order to draw a picture of the current research. Some attempts to represent and recognize actions are examined in the second part of the paper, and, particularly, the work we are doing in using and recognizing action information from scripts (Pinhanez & Bobick 1996).

[1](Langer 1953), chapter 17, contains an interesting discussion about the basics of theater.

Computer Theater

As much as museums and art galleries seem to depend on the physical presence of objects, a performance requires the performers and audience to share a common physical space. Otherwise the fundamental relation, *the suspension of disbelief*, does not take place.

In this paper we consider as *computer theater* only environments which involve human performers and audience in the same physical space, therefore excluding the idea of "distributed theater" (as proposed in (Krueger 1990), pg. 221). We also restrict the usage of the term computer theater to performance situations (ruling out, for instance, user browsing and storytelling). Computer theater, in our view, is about providing means to enhance the artistic possibilities and experiences of professional and amateur actors, or of audiences clearly engaged in a representational role in a performance (Schechner 1988).

The classification of interactive computer music systems proposed in (Rowe 1993) is an interesting starting point for the understanding of the possibilities of computer theater.

Hyper-Actors

(Rowe 1993) classifies an interactive musical system as following the *instrument* paradigm if the basic concern is to construct an extended musical instrument. For instance, in the *hyperinstruments* project led by Tod Machover at the MIT Media Laboratory, (Machover 1992), musical instruments were built which sense a virtuoso musician's gestures, enabling him/her to control and modulate a computerized counterpart to the acoustic sound produced by the instrument.

An actor's instrument is his body — including voice and facial expression. "Virtuosi" actors are able to control their bodies in extraordinary and different ways. Through the centuries actors have relied on masks, make-up, and costumes to alter their bodies, or in the extreme case, on puppets and marionettes.

We suggest the term *hyper-actor* to denote computer theater systems which aim to enhance an actor's body and therefore his expressive capabilities. A hyper-actor expands an actor's body so he is able to trigger lights,

sounds, or images on a stage screen; to control his final appearance to the public if his image or voice is mediated through the computer; to expand its sensor capabilities by receiving information on earphones or video goggles; or to control physical devices like cameras, parts of the set, robots, or the theater machinery.

The idea has been more explored in dance and music than in theater. Body suits wired with sensors having been widely explored with, recently in the works of Troika Ranch[2]. Other examples include the performances of Laurie Anderson involving the processing of her voice and singing through a Synclavier, (Anderson 1991); George Coates' experimentation with actors receiving the script from Internet users during the live performance of *"Better Bad News"*[3]; and Christopher Janney's performances where a musician and a dancer played with the sound of their heartbeats[4].

Another possibility is having the actor not on stage, and providing him the means to control the physical appearance of his image to the audience. Mark Reaney's "Virtual Theater", (Reaney 1996), is a curious illustration of this concept. In a typical scene an actor on stage plays with an off-stage actor whose image is seen by the audience on two large stereo-graphic video screens (the audience wears special 3D-goggles). The off-stage actor's image expands and contracts according to the play events and is used to symbolize and enrich the power struggle portrayed in the scene.

Computer-Actors

The *player* paradigm in interactive music systems corresponds to situations where the intention is to build *"... an artificial player, a musical presence with a personality and behavior of its own..."*[5]. In the computer theater realm the player paradigm corresponds to the *computer-actor*, a computer program which interacts with human actors, assuming the role of one of the characters of the play. In this case the computer displays its actions using an output device such as video screens, monitors, speakers, or physical devices.

The distinction between hyper- and computer-actors is important because computer-actors require a control system which decides what to do "independently" of the desires of the human partners. A computer-actor must be able to follow the script (if there is one) and react according to its own role; here, the issues of action recognition and automatic control of expressiveness seem to be more relevant than in the case of hyper-actors. In contrast, hyper-actors are likely to require much better sensing of the human performer movements than computer-actors.

A straightforward implementation of computer actors would be human-like or cartoonish characters displayed on a stage screen. Most of the interesting cases come from the research oriented towards direct user interaction with computer-generated characters for game-like systems. Worthy of mention is the work of Bruce Blumberg (Maes *et al.* 1995) in building a computer graphics generated dog which interacts with the user, not only obeying simple gestural commands (sit, catch the ball) but also having its own agenda of necessities (drinking, urinating).

An interesting alternative is being developed by Flavia Sparacino, (Sparacino 1996), who is incorporating behavior-based interaction into text, pictures, and video, constituting what she calls *media-creatures*. The project also involves exploring media-creatures in dance and theater performances — *media-actors*. Similarly, computer-actors can be computer-generated objects which do not exist in the real world (or do not normally interact with people). For example, Sommerer and Mignonneau[6] developed an art installation where fractal-based images of plants grow when the user touches a real plant in the space. Actors and dancers can also be embodied in robots[7].

Rehearsal vs. Performance

Ensemble rehearsing is a key part of the artistic process of theater. Compared to music, the ensemble rehearsal process in theater is longer and richer in experimentation. Characters are usually built through the interaction between actors on the stage with decisive supervision and guidance coming from the director.

Performing with a bad actor is bad, but rehearsing with a bad actor is quite worse. An unmotivated or limited actor in rehearsal can stop the creative process of the whole company. The importance of rehearsal is a major point that most of the computer theater experiments so far have inadequately addressed, especially in the case of computer-actors.

According to this view, one of the biggest challenges in computer-theater is to build hyper- or computer-actors which can actively respond to variations in the staging of a script as they are discovered and proposed during rehearsal time by the other actors and by the director. Such *rehearsable* computer-actors probably require more action representation and recognition than *performance-only* computer-actors, as is explained later.

Scripts and Improvisations

(Rowe 1993) also distinguishes between *score-* and *performance-driven* computer music, which we map to the concepts of scripted and improvisational computer theater. *Scripted* computer theater systems are supposed to follow totally or partially the sequence of actions described in a script. During the performance

[2] `http://www.art.net/Studios/Performance/Dance/Troika_Ranch/TroikaHome.html`

[3] `http://www.georgecoates.org/`

[4] `http://www.janney.com/heartb.htm`

[5] (Rowe 1993), pg. 8.

[6] `http://www.mic.atr.co.jp/~christa/`

[7] `http://guide.stanford.edu/people/curtis/machoreo.html`

the system synchronizes the on-going action with the script, giving back its "lines" as they were determined during the rehearsal process or, less interestingly, in an off-line mode by the director.

Improvisational theater relies on well-defined characters and/or situations. This type of computer theater has immediate connections with the research on developing characters for computer games and software agents (Bates, Loyall, & Reilly 1992; Maes *et al.* 1995). However, the distinction between actor/situation is not present in most cases, impoverishing the theatrical interest in such creatures since a major source of dramatic conflict in improvisations is the setting of characters in unexpected situations and environments.

Yet more important is the fact that good improvisation requires recognition of intentions. Knowing what the other character wants to do enables interesting and rich counteracting behavior. Otherwise, the resulting piece is flat, structurally resembling a "dialogue" with an animal: the sense of immediacy dictates most of the actions.

Performers and Users

Most of the research on building interactive computer characters has been targeted towards non-actors, people unfamiliar with the computerized environment/space. This is not the case in computer music, where there has been a concentration in providing tools for people with some musical training.

Although the development of ideas and methods to concretely engage users is very important, we believe it is also very important to concentrate some effort on understanding and reacting to actors and audience in a performance situation. There is also an important reason to do so: users are boring from the action point of view. Users are motivated by curiosity, and their repertoire of displayed actions and reactions is normally very restricted.

It is also very hard to develop a performance with people who are not committed to being engaged — as street performers well know. A system assuming non-engaged users must rely on story-telling techniques (and therefore, narrative) to create an interesting environment. Theater and, in general, performance can go beyond story-telling by assuming that performers and audience know their roles, actions, and reactions.

Computerized Stages

It is worth mentioning another dimension of computer theater which is concerned with the expansion of the possibilities for the stage, set, props, costumes, light and sound. The fundamental distinction with the hyper- and computer-actors is that elements of *computerized stages* are not characters or representations of characters.

A stage can react by changing illumination, generating visual and special sound effects, changing the appearance of backdrops and props, or controlling machinery. An example is the Intelligent Stage project at Arizona State University, (Lovell & Mitchell 1995), which enables the mapping of volumes in the 3D space to MIDI outputs. Movement and presence are monitored by 3 cameras, triggering music and lights accordingly.

Action-Based Computer Theater

It is certainly possible to have a computer theater system which just produces output in pre-determined and pre-timed ways. Although human actors (and especially dancers) can adjust their performances to such situations, the results normally are devoid of richness and life. Computer theater seems to be worthwhile only if the hyper- or computer-actor follows the actions of its human partners and adjusts its reactions accordingly.

In the case of scripted theater the computer system must be able to recognize the actions being performed by the human actors, and to match them with the information from the script. Minimally, the computer can use a list describing mappings between sensory inputs and the corresponding computer-generated outputs. The list can be provided manually by the "director" or technical assistants, and, during performance, the recognition consists in synchronizing live action and the list according to the sensory mappings.

Although the "simple" system just described is hard to implement in practice due to noisy sensors and performance variability, we believe there is a much more interesting approach to computer theater based on *action understanding*. Instead of providing a computer-actor a list of sensor-reaction cryptic mappings, the challenge is to use as input the actions and reactions as determined by a script or by the director.

Textual description of actions in the script corresponding to the human part can then be analyzed by the computer producing visual and auditory components which can be detected by sensory routines. On the other hand, the hyper- or computer-actor's actions can be used to directly generate low-level instructions for computer-graphics or external physical devices. According to this view, a computer-actor should be instructed by words like "shout" or "whisper", and be able to recognize automatically an action described simply as "actor walks to the chair".

A positive feature of action-based verbal descriptions is precisely their vagueness. A description like "actor walks to the chair" does not specify from where the actor comes, the path taken, etc. Instead, it highlights only the final destination enabling the actor to explore different ways of performing it without disengaging the recognition system. Similarly, describing the computer-actor's actions in textual mode provides room for reactive mechanisms during rehearsal and performance time.

However, in the case of improvisational computer-actors it is also necessary to deal with the recognition of the intentions behind the actions of the human actor. For example, consider an improvisational act where an actor is in jail, and the computer-actor is the prison guard. In this situation, the computer system must recognize whether the prisoner is trying to escape from the jail and which means he intends to use. Those intentions are embodied, translated into the actor's physical activities and the recognition system must infer their occurrence in order to react properly. Perceptual recognition of intentions has been hardly explored, and we believe it constitutes a major challenge for the design of interesting improvisational computer-actors.

Computer Theater as a Domain for Action Understanding

We claim that the action approach is not only appropriate to computer theater, but also that computer theater is a good domain for research in action understanding. In the simplest analysis, it is easy to see that the gestures employed by actors are more explicit and determined than ordinary human activity. For instance, if holding a glass of whiskey is important for the dramatic structure of the play, the actor makes sure that the audience notices when the glass is picked up. Theater also naturally provides a wider range of gestures, postures, and situations than normal life. Moreover, the actions are staged such as every member of the audience can see them, and minuscule gestures are rarely used. Therefore, visual action recognition can employ long-shot, wide-angle cameras which correspond to the audience field of view, avoiding the problems of having different image resolution needs.

There are more interesting and deeper reasons to use theater as a domain for action understanding. Theater defines clear and defined contexts which provide natural limits for reasoning and recognition processes. The context is described in the script, as well as the basic repertoire and sequence of actions and movements of the actors. Also, the mechanics of the dramatic text causes attention to be driven by actions of the performers, and only quite rarely by non-human caused events.

However, the greatest hope is that computer theater might enable action research to start tackling the hard issue of intentionality. In theater, intentions must be translated into physical activity. The process of transforming the intentions of the play-writer in the script into physical actions (including voice punctuation and intonation) is normally a joint effort of the director and the actors. Traditionally, the director analyzes the text and assigns intentional actions to parts of the script, and general objectives to different characters (see (Clurman 1972), chapter 7 and part IV). During rehearsal, the actors are guided to find physical activities which correspond to the intention of the characters. Therefore, theater enables us to assume that every action is intentional and the result of the conflict between the character's objectives and the other actors' actions: intentions can be expected to be explicit and present in every activity.

Thus, systems without any capabilities for representing intentions tend to be inadequate for computer theater — and particularly, for rehearsal — because they lack the ability to react purposefully to a given situation or to an action from one of the human actors.

Action: Representation, Recognition, and Generation

In this section we try to examine the different aspects and challenges for AI research posed by action representation and recognition: in particular we detail some work we have been doing in terms of representing, using, and recognizing actions in a scripted performance (Pinhanez & Bobick 1996). Most of this research considers the domain of TV cooking shows, and has been applied in the development of *SmartCams*, automatic cameras for TV studios (Pinhanez & Bobick 1995). We believe that most of this work extends naturally to computer theater.

Representation

Representing actions has been the object of research of linguistics (Jackendoff 1990; Schank 1975; Pinker 1989), computer graphics (Kalita 1991), and computer vision (Siskind 1994). As part of our work with automatic cameras, (Pinhanez & Bobick 1996), a representation scheme has been developed based on Schank's *conceptualizations*. The representation uses *action frames*, a frame-based representation where each action is represented by a frame, whose header is one of Schank's primitive actions — `PROPEL`, `MOVE`, `INGEST`, `GRASP`, `EXPEL`, `PTRANS`, `ATRANS`, `SPEAK`, `ATTEND`, `MTRANS`, `MBUILD` — plus the attribute indexes `HAVE` and `CHANGE`, and an undetermined action `DO`.

Figure 1 contains examples of action frames. The figure contains the representation for two actions of a cooking show script, "`chef talks about today's recipe`" and "`chef mixes bread-crumbs and basil in a bowl`". In the first example, "talking" is translated into the action of "mentally transporting" (`MTRANS`) the text `today-recipe-text` through `sound` going into the direction of `camera-2`. "Mixing" is translated as an unknown action (or group of actions) which puts bread crumbs and basil inside a bowl and in physical contact with each other. A better description of the meaning of each slot and other examples of action frames can be found in (Pinhanez & Bobick 1996).

The examples shown in fig. 1 show the minimum information produced by a direct translation of the sentences in the script. They typify the notion of "vagueness" of action mentioned before. To effectively use

```
;; "camera-2:  chef talks about today's recipe"
(mtrans (actor chef)
 (to public)
 (object today-recipe-text)
 (instrument
  (speak (actor chef)
   (object sound)
   (to (direction camera-2)))))

;; "chef mixes bread-crumbs and basil in a bowl"
(do (actor chef)
 (result
  (change
   (object (group bread-crumbs basil))
   (to (contained bowl))
   (to (phys-cont
        (group bread-crumbs basil))))))
```

Figure 1: Action frames corresponding to two actions from a script.

this information — both to recognize and generate action — it is necessary to expand it by inference mechanisms.

In our research on automatic cameras, we implemented a simple inference system based on Rieger's inference system for Schank's conceptualizations (Rieger III 1975). In a typical case, the inference system, using as its input the action frame corresponding to the sentence "**chef mixes bread-crumbs and basil in a bowl**", deduces that the chef's hands are close to the bowl.

Part of our current research is focused on designing a better representation for actions than the action frames described in this paper. We are still debating the convenience of using Schank's primitives to describe every action. Also, action frames need to be augmented by incorporating visual elements, as in (Kalita 1991), and time references, possibly using Allen's interval algebra, (Allen 1984). Another important element missing in our representation system is a mechanism to specify the intensity of an action. For computer theater purposes, the difference between "talking" and "shouting" is crucial.

Recognition

Research in visual action recognition has been restricted to recognizing human body movements as described in (Israel, Perry, & Tutiya 1991; Polana & Nelson 1994; Bobick & Davis 1996). (Kuniyoshi & Inoue 1993; Siskind 1994) are among the few works which actually examined some of the issues related to understanding actions and their effects in the world.

Bruce Blumberg's dog mentioned above uses the recognition capabilities of ALIVE (Maes *et al.* 1995) to react to commands like "go", "sit", and "catch the ball". The limited vocabulary and precise context enables a trivial translation of hand positions directly into actions: an extended arm into the ground is recognized as a "sit" command independently of any other factors, as for instance, the actual shape of the hand or the direction of sight.

We have been conducting research (unpublished) addressing visual action recognition based on action descriptions similar to those of fig. 1. The key idea behind the proposal is to represent time constraints using Allen's interval algebra (Allen 1984), enabling vaguely specified relationships among the actions, sub-actions, and visual features. The visual features are obtained in a dictionary of action verbs which translates the actions into information about attributes detectable by visual routines.

Speech recognition can be a simple way to synchronize performance and scripts, by matching the spoken words with the lines in the script. However, a system based purely on speech matching is limited in its reactivity, especially in the case of improvisational theater where some understanding seems to be required. Another dimension of voice is the expression of emotions: recognition research in this case is only beginning (Pittam, Galois, & Callan 1990).

Generation

There has been a significant amount of work to incorporate action into computer-graphics: (Perlin 1995; Kalita 1991; Thalmann & Thalmann 1990) are good examples. Perlin's work is particularly interesting because the computer-actor receives commands directly as action verbs.

The synthesis of facial expressions for human-like computer characters has also received significant attention from the computer graphics community. (Terzopoulus & Waters 1993) is a typical example where the modeling of facial muscles and tissue enables a variety of facial expressions.

A good example of attributing expressiveness to media objects is the work in (Wong 1995) with expressive typography. In this case, text dynamically changes its shape, typeface, color, and screen position in order to convey temporally the expressive dimension of the message.

Final Remarks

Throughout this paper we have stressed the importance of action in computer theater. Action and reaction are essential to the vitality of theatrical performance and should be incorporated, implicitly or explicitly, into any computer theater system. It is an open question whether action should be incorporated symbolically using the techniques developed in the research in action understanding.

The classification of computer theater developed in the first half of the paper is intended to clarify and compare different techniques and approaches. Rather than being exhaustive, the enumeration of different possibilities of computer theater has the aim of guiding

the design of new systems targeting specific scientific or artistic concerns.

Finally, action representation and recognition research may be able to profit from the computer theater domain. Defined contexts, exaggerated gestures, controlled environments, known and reliable mappings between symbols and real world, and richness of different situations can provide a fertile environment for action research. The disadvantages are the likely "toy" domains, and the difficulties on devising evaluation methods.

References

Allen, J. F. 1984. Towards a general theory of action and time. *Artificial Intelligence* 23:123–154.

Anderson, L. 1991. *Empty Places.* New York, New York: HarperPerennial.

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1992. An architecture for action, emotion, and social behavior. In *Proceedings of the Fourth European Workshop on Modeling Autonomous Agents in a Multi-Agent World.*

Bobick, A. F., and Davis, J. W. 1996. An appearance-based representation of action. Technical Report 369, M.I.T. Media Laboratory Perceptual Computing Section. To appear in ICPR'96.

Clurman, H. 1972. *On Directing.* New York, New York: Collier Books.

Israel, D.; Perry, J.; and Tutiya, S. 1991. Actions and movements. In *Proc. of the 12th IJCAI,* 1060–1065.

Jackendoff, R. 1990. *Semantic Structures.* Cambridge, MA: The M.I.T. Press.

Kalita, J. K. 1991. *Natural Language Control of Animation of Task Performance in a Physical Domain.* Ph.D. Dissertation, University of Pennsylvania, Philadelphia, Pennsylvania.

Krueger, M. W. 1990. *Artificial Reality II.* Addison-Wesley.

Kuniyoshi, Y., and Inoue, H. 1993. Qualitative recognition of ongoing human action sequences. In *Proc. of IJCAI'93,* 1600–1609.

Langer, S. K. 1953. *Feeling and Form.* New York, New York: Charles Scribner's Sons.

Lovell, R. E., and Mitchell, J. D. 1995. Using human movement to control activities in theatrical environments. In *Proc. of Third International Conference on Dance and Technology.*

Machover, T. 1992. Hyperinstruments: A progress report. Technical report, M.I.T. Media Laboratory.

Maes, P.; Darrell, T.; Blumberg, B.; and Pentland, A. 1995. The ALIVE system: Full-body interaction with autonomous agents. In *Proc. of the Computer Animation '95 Conference.*

Perlin, K. 1995. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics* 1(1):5–15.

Pinhanez, C., and Bobick, A. F. 1995. Intelligent studios: Using computer vision to control TV cameras. In *Proc. of IJCAI'95 Workshop on Entertainment and AI/Alife.*

Pinhanez, C. S., and Bobick, A. F. 1996. Approximate world models: Incorporating qualitative and linguistic information into vision systems. To appear in AAAI'96.

Pinker, S. 1989. *Learnability and Cognition.* Cambridge, MA: The M.I.T. Press.

Pittam, J.; Galois, C.; and Callan, V. 1990. The long-term spectrum and perceived emotion. *Speech Communication* 9:177–187.

Polana, R., and Nelson, R. 1994. Low level recognition of human motion. In *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects,* 77–82.

Reaney, M. 1996. Virtual scenography: The actor, audience, computer interface. *Theatre Design and Technology* 32(1):36–43.

Rieger III, C. J. 1975. Conceptual memory and inference. In *Conceptual Information Processing.* North-Holland. chapter 5, 157–288.

Rowe, R. 1993. *Interactive Music Systems.* Cambridge, Massachusetts: The MIT Press.

Schank, R. C. 1975. Conceptual dependency theory. In *Conceptual Information Processing.* North-Holland. chapter 3, 22–82.

Schechner, R. 1988. *Performance Theory.* London, England: Routledge.

Siskind, J. M. 1994. Grounding language in perception. *Artificial Intelligence Review* 8:371–391.

Sparacino, F. 1996. Directive: Choreographing media creatures for interactive virtual environments. Technical Report 377, M.I.T. Media Laboratory Perceptual Computing Section.

Terzopoulus, D., and Waters, K. 1993. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE PAMI* 15(6):569–579.

Thalmann, N. M., and Thalmann, D. 1990. *Synthetic Actors in Computer Generated 3D Films.* Berlin, Germany: Springer-Verlag.

Wong, Y. Y. 1995. Temporal typography: Characterization of time-varying typographic forms. Master's thesis, Massachusetts Institute of Technology.