Multimodal Acting in Mixed Reality Interactive Storytelling

Marc Cavazza, Fred Charles, and Steven J. Mead
University of Teesside

Olivier Martin
Université Catholique de Louvain

Xavier Marichal and Alok Nandi
Alterface

An experimental mixed reality using a multimodal approach lets users play characters in interactive narratives as though acting on a stage. Users interact with characters through speech, attitude, and gesture, enhancing their immersion in the virtual world.

Interactive storytelling immerses users in fantasy worlds in which they play parts in evolving narratives that respond to their intervention. Implementing the interactive storytelling concept involves many computing technologies: virtual or mixed reality for creating the artificial world, and artificial intelligence techniques and formalisms for generating the narrative and characters in real time.

As a character in the narrative, the user communicates with virtual characters much like an actor communicates with other actors. This requirement introduces a novel context for multimodal communication as well as several technical challenges. Acting involves attitudes and body gestures that are highly significant for both dramatic presentation and communication. At the same time, spoken communication is essential to realistic interactive narratives. This kind of multimodal communication faces several difficulties in terms of real-time performance, coverage, and accuracy.

We've developed an experimental system that provides a small-scale but complete integration of multimodal communication in interactive storytelling. It uses a narrative's semantic context to focus multimodal input processing—that is, the system interprets users' acting (the multimodal input) in the mixed reality stage in terms of narrative functions representing users' contributions to the unfolding plot.

System overview: The mixed reality installation

Figure 1 shows the mixed reality system architecture. The system uses a "magic mirror" paradigm, which we derived from the Transfiction approach.[1] In our approach, a video camera captures the user's image in real time, and the Transfiction engine extracts the image from the background and mixes it with a 3D graphic model of a virtual stage, which includes the story's synthetic characters. The system projects the resulting image on a large screen facing the user, who sees his or her image embedded in the virtual stage with the synthetic actors.

We based the mixed reality world's graphic component on the Unreal Tournament 2003 game engine (http://www.unrealtournament.com). This engine not only renders graphics and animates characters but, most importantly, contains a sophisticated development environment for defining interaction with objects and character behaviors.[2] It also supports integration of external software through socket-based communication.

We use the Transfiction engine to construct the mixed environment through real-time image processing.[3] A single (monoscopic) 2D camera analyzes the user's image in real time by segmenting the user's contours. The segmentation's objectives are twofold:

- It extracts the user image silhouette and injects it into the virtual setting on the projection screen (without resorting to chroma keying).

- At the same time, the Transfiction engine analyzes the extracted body silhouette to recognize and track user behavior (position, attitude, and gestures) and influence the interactive narrative accordingly.

A detection module segments the video image in real time and outputs the resulting image together with other data, such as gesture recog-
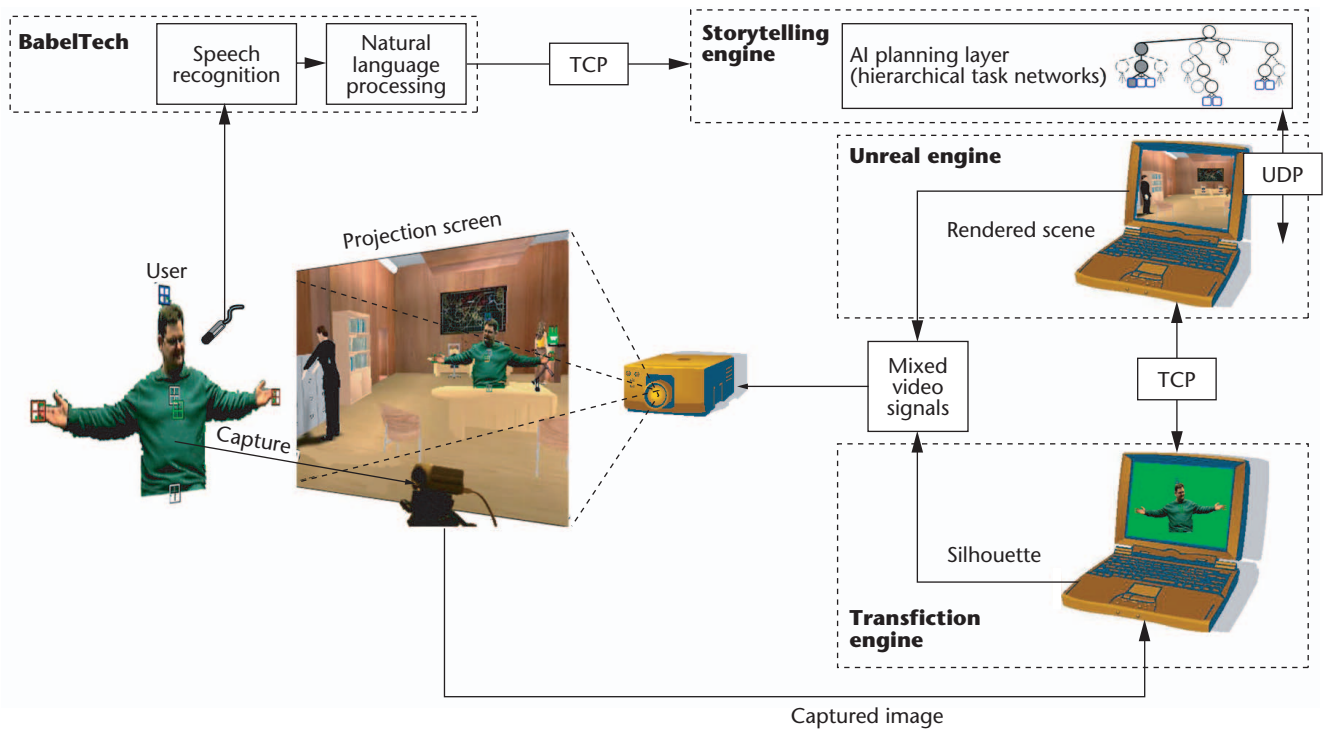
Figure 1. Mixed reality system architecture. The Transfiction engine extracts the user's image from the film captured by the video camera and mixes it with a 3D graphic model of a virtual stage. The user views the resulting image on a large screen.

nition, that enable further processing. The current detection module uses a 4×4 Walsh function Hadamard determinant and calculates the transform on 4×4-pixel elements. Sliding the box of two pixels aside allows taking decision on 2×2-pixel blocks. As a result, it can segment and adequately detect objects' boundaries and offers some robustness to luminance variations. Figure 2 gives an overview of the change-detection process with the Walsh-Hadamard transform. First, the detection module calculates the background image's Walsh-Hadamard transform. It then compares the transform's values for the current and background images. When the rate of change is higher than an established threshold, the module sets the area as foreground. Because shadows (which can be problematic because of variable indoor lighting conditions) can corrupt segmentation results, we remove them using invariant techniques.[4]

Next, we composite the resulting video image with the virtual environment image by mixing the video channels captured by a separate computer running a DirectX-based application. The first stage involves isolating the user image from its background using basic chroma keying. The remaining stage attempts to solve the occlusion
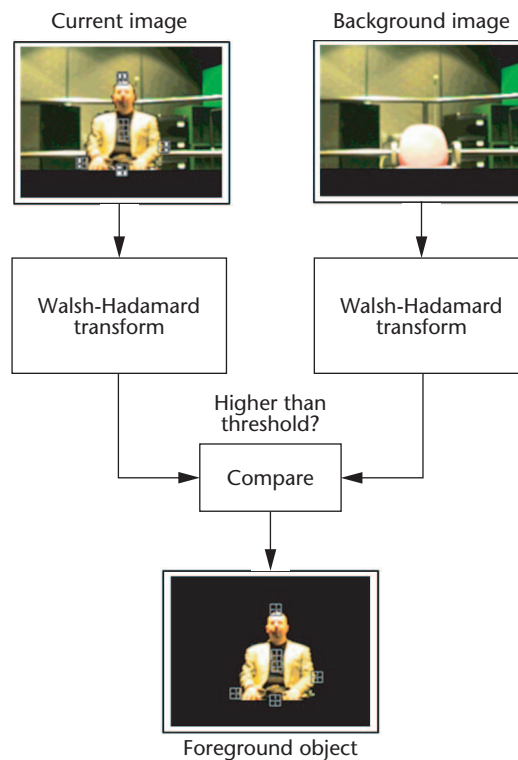

Figure 2. Extracting the user's image from the background. The detection module uses a Walsh-Hadamard transform to compare the background and current images.

problem by blending the user image with the virtual environment image using empirical depth information. The gesture-recognition module

Figure 3. Constructing the mixed reality environment. The mixed reality system blends the user image and the vitual environment image to produce the final image, which it projects on a large screen.
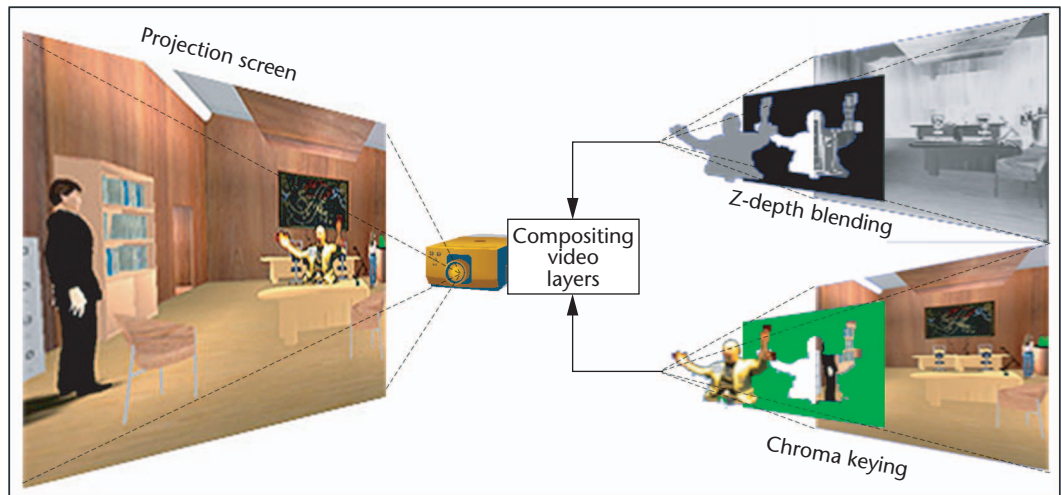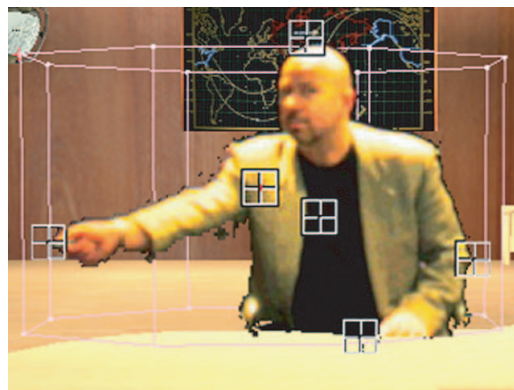
Projection screen

Compositing video layers

Z-depth blending

Chroma keying

Figure 4. The 3D bounding cylinder determines physical interactions in the Unreal Tournament 2003 engine.

provides this information for the user as the user's relative distance to the camera; the game engine provides it for the virtual environment. Figure 3 illustrates the overall process whereby the system composites several video image layers in real time to produce the final image, which it projects onto a screen in front of the user.

In the first prototype, the gesture detection and recognition components share a normalized system of coordinates, which we obtain through calibration prior to running the system. This prototype doesn't deal with occlusion in mixed reality, which is also set at calibration time. We're currently developing an occlusion management system, which uses depth information provided by the transfiction engine.

The shared coordinates system lets us not only position the user in the virtual image, but also determine the relations between user and virtual environment. To do this, we map the 2D bounding box produced by the transfiction engine—which defines the contour of the segmented user character—to a 3D bounding cylinder in the

Unreal Tournament 2003 environment, which represents the user's position in the virtual world (as Figure 4 shows). Relying on its basic mechanisms, the Transfiction engine automatically generates low-level graphical events such as collisions and object interaction.

The two subsystems communicate via TCP sockets: the image-processing module, working on a separate computer, regularly sends two types of message to the graphic engine. The messages update the user's position and any recognized gestures. The Transfiction engine transmits the recognized gesture as a code for the gesture (for example, a 2D vector indicating the direction of pointing represents a pointing gesture). However, contextual interpretation of the gesture occurs within the storytelling system.

The storytelling scenario in our experiments is a James Bond adventure in which the user plays the villain (the Professor). The narrative properties of James Bond stories make them good candidates for interactive storytelling experiments; Barthes used them as a supporting example in his foundational work in contemporary narratology.[5] In addition, their reliance on narrative stereotypes facilitates both narrative control and users' understanding of the roles they're to play. The basic storyline represents an early encounter between Bond and the Professor. Bond's objective is to acquire some essential information, which he can obtain by searching the Professor's office, asking the Professor's assistant, or, under certain conditions, deceiving or threatening the Professor himself. The user's actions as the Professor interfere with Bond's plan, altering how the plot unfolds.
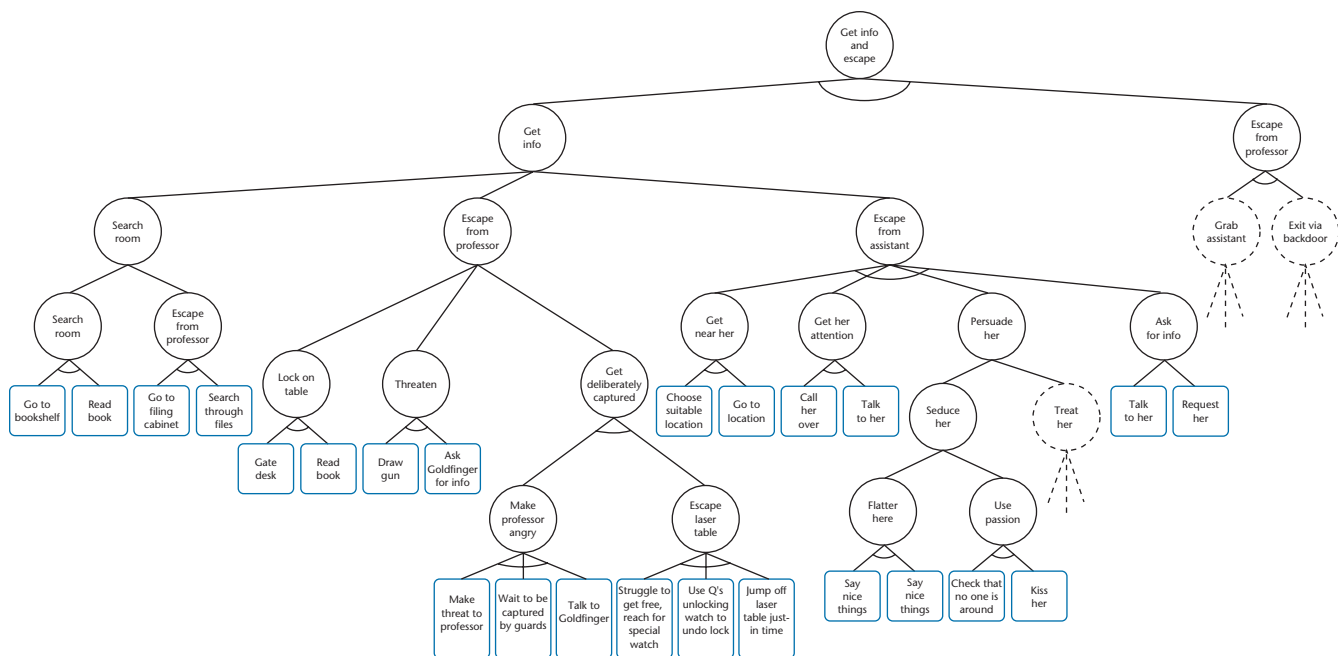
Interactive storytelling

We adapted the interactive storytelling technology used in these experiments from our previous work, described in detail elsewhere.[6] Thus, we only briefly overview the approach here, focusing on the aspects most relevant to the system's mixed reality implementation, in particular multimodal user interaction.

Interactive storytelling involves the real-time generation of narrative actions such that the consequences of user intervention result in the interactive storytelling system regenerating the story with a modified environment. *Narrative control* dictates that user intervention should modify the story's course, but only within the limits of the story's genre. Narrative control generally relies on a baseline plot that defines possible character actions, but imposes no unnecessary constraints on how the actions can be combined to constitute a plot.

Our approach, *character-based interactive storytelling*,[6] centers on the virtual actors' roles. We based the artificial intelligence mechanism supporting character behavior on a planning technology using hierarchical task networks (HTNs), as illustrated in Figure 5.[7] These representations describe the character's role as a plan using a hierarchical decomposition of tasks into subtasks. (Formally, HTNs are AND/OR graphs and we can represent the solution plan as a subgraph of the HTN.)

For instance, we can decompose an information-gathering task into several options for gaining access to that information, such as searching files or getting it from another character. Each of these tasks can be further decomposed—for instance, to get information from another character, the user's character must approach it, establish a relationship with it, convince it to handle the information, and so on. HTN task decomposition continues until reaching the terminal-action level—that is, the level at which the synthetic character can visually perform actions in the virtual world. The system thus uses an HTN planner to select in real-time each character's actions. Our module implemented within the Unreal Tournament 2003 sends an action's failure back to the planner, which produces an alternative solution. This mechanism is essential in interactive storytelling because user intervention often causes a character's planned action to fail, leaving it to produce an alternative solution that will lead the story into new directions.

To accommodate the mixed-reality context, we adapted our previous character-based storytelling framework to the new user-interaction paradigm derived from the magic mirror metaphor,[8,9] which assumes greater user involvement than other interactive storytelling approaches. This greater involvement calls for more flexible narrative control. In our supporting example, each situation represents a stage in the encounter between Bond and the villain: introduction, negotiation, and separation. In this version, we've defined one HTN for each situa-
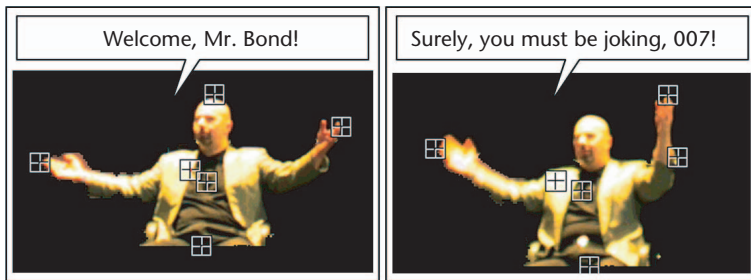
Figure 6. Ambiguous gestures. This attitude, characterized as the user sitting with his arms raised and partly extended, can represent greetings, denial, challenge, and so on. Interpreting the correct meaning requires joint analysis of the user's utterance and attitude.

tion rather than a single HTN encompassing the entire scene. Figure 5 depicts the discussion phase HTN.

We adapted basic interactive storytelling mechanisms to the greater user involvement. HTNs are still based on the Bond role, but they give a more explicit status to user intervention to allow for the user's regular, but unpredictable, interaction, as the HTN representing the conversation between Bond and the Professor illustrates. This HTN incorporates several extensions to the HTN used in our previous work. Some extensions involve a novel use of the representation; others required modifying the underlying planning algorithm that uses the HTN to animate the virtual character. For example, the HTN in the current system uses mixed nodes that have both an AND and an OR, letting us incorporate optional actions while still limiting the representation's complexity (for example, in Figure 5, the HTN makes the threat situation optional). We've also incorporated the possibility of some user intervention in the HTN itself. One required extension allows a character to attempt an action only after the planner tests for the compatible user action (in this case, the Professor giving away the information). This doesn't prevent the user from performing actions other than the one expected, which will impact the character's plan at another level. In other words, this representation departs from a strict character-based approach to incorporate plot representation to accommodate the higher level of user involvement.

Multimodal interaction

The user intervention is a multimodal input consisting of a spoken utterance and an optional body gesture interprets this multimodal unit in context (that is, using knowledge about the plot progression) to determine what kind of response it is to the virtual characters' actions.

Consider the joint recognition of a multimodal speech act comprising an utterance analyzed through speech recognition and a body gesture processed by the transfiction engine. We categorize the user's attitude shown in Figure 6 as the user sitting with his arms raised and partly extended (other categories include pointing gestures, which can mean showing, giving, and so on). This attitude is compatible with different interpretations, including greetings ("Welcome, Mr. Bond!"), denial ("You must be joking, Mr. Bond"), or challenge ("Shoot me and you'll never know, 007!"). We interpret the correct meaning through joint analysis of the user's utterance and attitude.

Traditional literature on multimodality focuses on the use of deictic gestures in natural language instructions or dialogue[10] and of gestures in nonverbal communication.[11] The narrative context of interactive storytelling creates new forms of gesture use, which in turn create new multimodal combinations. The system supports deictic gestures, such as when the user indicates an object or a location in a multimodal utterance ("Take a seat, Mr. Bond").

Another type of gesture—physical gestures—is on-stage physical interventions, such as grasping an object, slapping a character, or standing in front of an object or character. To implement physical gestures effects, we use the main mixed reality mechanism—that is, a single coordinate system that controls interaction through bounding boxes in the virtual environment.

The most important gesture type is semiotic gestures. Semiotic gestures include opening one's arms to welcome someone, raising a hand to attract attention or call someone, raising both arms in wonder or disbelief, and opening arms to indicate ignorance. What distinguishes these gestures from other nonverbal behaviors is that they constitute isolated units associated with a precise communicative function; in particular, a function that can be mapped to a narrative context (unlike beat and other continuous and dynamic gestures).

Much controversy over the status of the various modalities in terms of their semantic content exists in the multimodal literature. A distinguishing characteristic of the interactive storytelling context is that speech and semiotic gestures can have comparable semantic content.

Speech recognition

Speech is the only practical mode of communication between the user and the virtual actors in an interactive storytelling context, in addition to its being part of the narrative itself. Of course,

formidable challenges associated with speech understanding exist. In this context, we aim for a modest level of robustness, although a principled approach would use the specific interactive storytelling context to guide the speech interpretation strategy.

We based the speech recognition component on BabelTech's Ear software developer's kit system, shown in Figure 7. The SDK can be used in various modes, including multikeyword spotting in an open loop. Multikeyword spotting involves dynamically recognizing keywords from a predefined set in any user utterance, regardless of the utterance's other contents. It provides robust recognition of the utterance's most relevant topics in context without imposing constraints on the user (such as the use of a specific phraseology). One necessary step when using multikeywork spotting is to provide a more integrated definition of keywords as meaning units (for example, "be_careful" or "pay_attention").

User utterances occurring in this narrative context are a specific type of *speech act*—that is, an utterance with a specific impact on the hearer's behavior. More importantly, a good mapping exists between speech acts and narrative actions (greetings, threats, requests, denials, and so on) such that they constitute direct input into the narrative representation.

Categorizing user utterances in terms of speech acts—that is, recognizing the relevant speech act from the speech recognition output, be it a set of keywords or a more complex structure—is the key problem. No universally agreed-on method to identify speech acts exists. Practical approaches in speech understanding have sought to either detect surface-form cues (such as the occurrence of "welcome" in greetings or *wh*-markers in questions) or derive the speech act from the utterance's casual structure (that is, identify the action verb and its parameters).

Our implementation uses both approaches in parallel, while extending the shallow approach to cue detection. When the natural-language processing module doesn't recognize an action verb around which to instantiate an action template, it looks for surface cues, which it maps to a coarse-grained semantic category, such as approval/disapproval, confirmation/denial, or friendly/hostile (surface patterns such as "you won't," "you'll never," and so on). The underlying principle is to use coarse-grained semantic categories that the module can directly map to a speech act, which in turn the previously men-
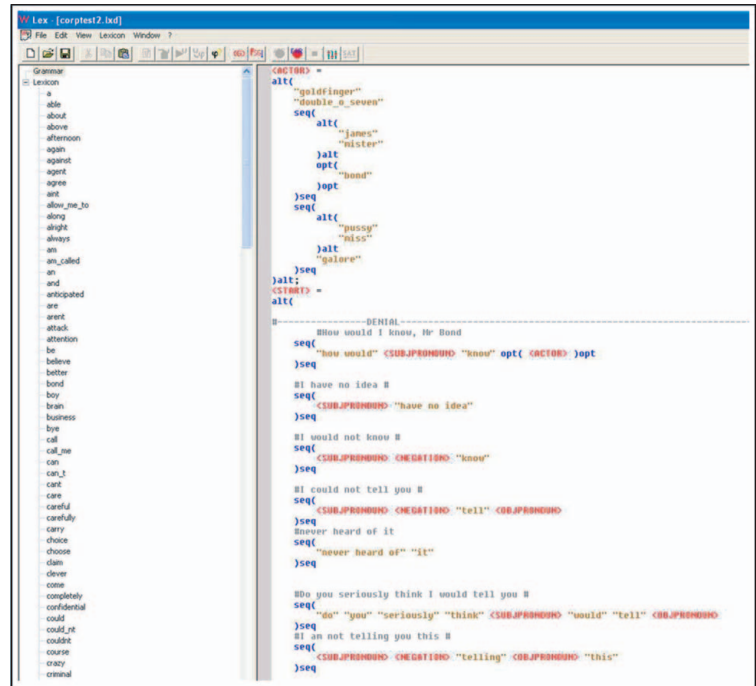

Figure 7. BabelTech's Ear software developer's kit environment.

tioned speech act can assimilate into a narrative function according to its impact on the interactive story. Although the use of coarse-grained categories doesn't let us extract an occurrence's complete meaning, it does let us identify a global meaning in context, which should trigger an appropriate behavior from the virtual actor.

We based the natural language interpretation on a template-matching procedure, because the use of multikeyword spotting precludes more complete forms of syntax-based parsing. In template filling procedures, the natural-language processing module looks for certain action verbs or substantives. Recognizing one of these words activates a template, which then searches for keywords corresponding to the action parameters. To find a subject or object, for example, the template might look for pronouns or proper names.

To identify a relevant speech act, the natural-language processing module uses the information in the template, any surface cues encountered, and the narrative context—that is, the current stage of the plot. Implementing speech act identification requires a set of production rules. The narrative actions represented in the plot model determine the number of target speech acts, which are thus limited in number, facilitating the mapping process.

We've so far associated speech acts with spoken utterances; in practice, they correspond to

Figure 8. Example body gestures. A gesture lexicon contains as many as 15 body attitudes, such as the four illustrated here.

so on. A set of feature-value pairs describes each gesture. While the Transfiction engine constantly outputs tracking point coordinates, the gesture recognition system derives in real time the values for each gesture feature from the tracking points' coordinates. The system uses each feature's set of values to filter candidate gestures from the gesture repository. Whenever it encounters a satisfactory match, it outputs the candidate semiotic gesture or a set of candidate gestures, such as {welcoming, denial}. The gesture recognition system can then unify this semantic category with those categories produced by the speech recognition component. Figure 9, which represents the temporal histograms corresponding to certain gesture features aligned with the spoken utterance, "You must be joking, Mr. Bond!" illustrates this process.

By processing speech and gestures jointly, the system implies that the open arms attitude serves a denial narrative function. Users will thus interpret the attitude as a negative answer to Bond's question, which corresponds to a failure of the task in the corresponding HTN, shown in Figure 10 (on p. 38), leading to a new course of action.

multimodal input, which includes both speech and gesture. Although this doesn't alter the nature of the target speech acts, it requires processing both modalities simultaneously, which in turns supports more robust recognition.

Gesture recognition and multimodal processing

To a large extent, the gesture recognition software follows a philosophy that is not unlike that of multikeyword spotting used for the speech component. That is to say, a fixed set of parameters is extracted from the image, which can be mapped to previous characterization of user attitudes. A set of semiotic gestures constitutes a *gesture lexicon*, containing as many as 15 body attitudes, some of which are represented in Figure 8.

The gesture collection has a variety of sources: literature, actors in relevant movies, and so on. For each attitude, we collected and associated data from the transfiction engine to the gesture in the lexicon. We similarly associate the semiotic interpretation to the gesture. A gesture that is ambiguous out of context receives several possible interpretations (Figure 6).

The representation of each semiotic gesture in the lexicon is a set of descriptive features: distance between arm extremities, hand height, and

Conclusion

Mixed reality is a significant departure from other paradigms of user involvement, such as pure spectator (with the ability to influence the story)[6] or Holodeck,[12] in which an actor is immersed in first-person mode. Although we've yet to explore the practical implications of such involvement, mixed reality interactive storytelling brings new perspectives for user interaction as well, with an emphasis on multimodal interaction.

Human–computer interface research describes the mode of appropriation used in our system (in which the context leads the user to rediscover modes of expression previously described) as *habitability*. We can therefore conclude that acting creates the condition for multimodal habitability.

Although the system is fully implemented, it remains a proof-of-concept prototype. Thus it's still too early to perform detailed user evaluations. However, grounding future evaluation procedures on the multimodal habitability notion is appropriate. This suggests including an early adaptation stage during which the user gets acquainted with the system interface with minimal guidance apart from the prior presentation of original film footage. The subsequent evaluation should be allowed to measure various forms
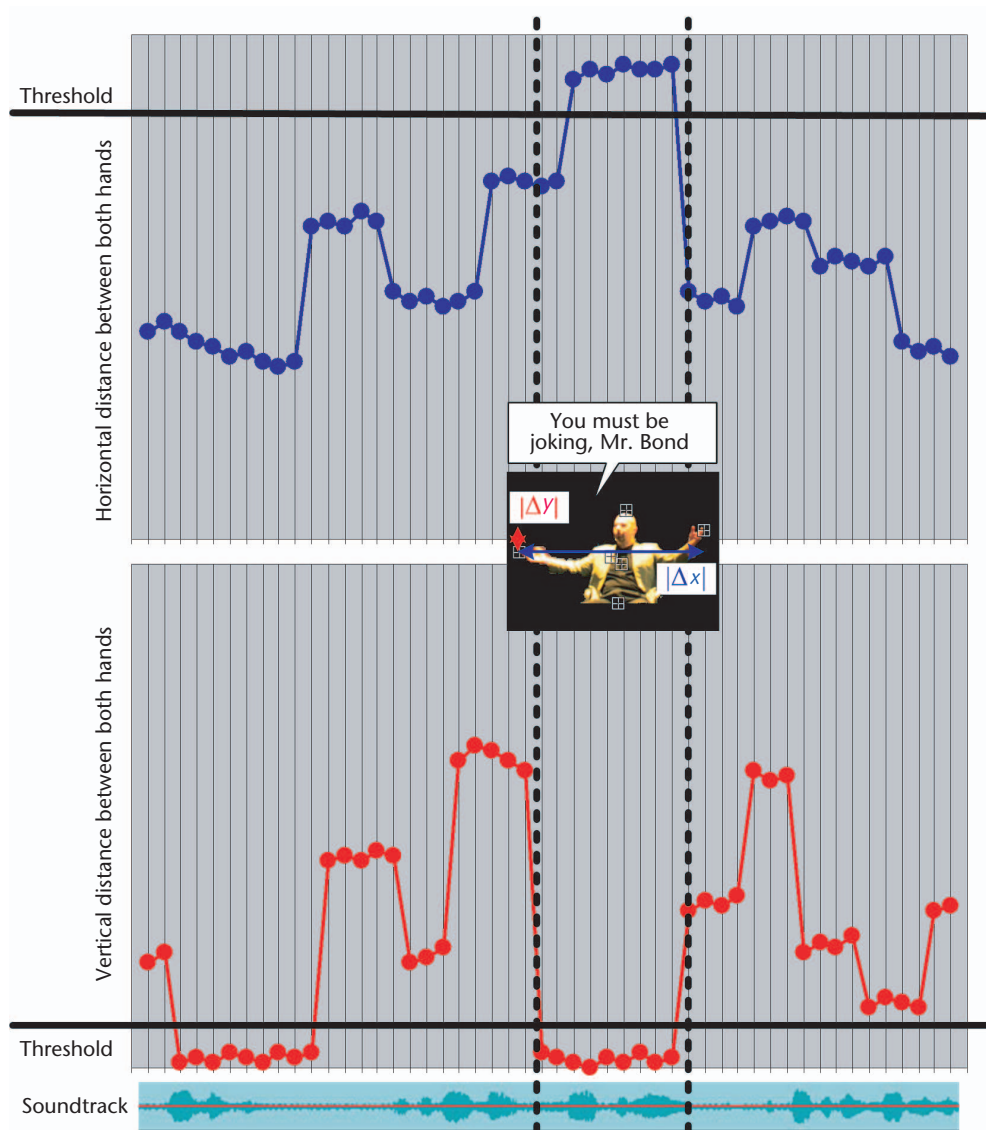
Figure 9. Multimodal detection: natural-language processing and gesture recognition for interpretation of user's acting.

of departure between the user's expressions and those characteristics of the original role within the story genre. **MM**

Acknowledgments

References

1. R. Barthes, "Introduction à l'Analyse Structurale des Récits (in French), *Comm.*, vol. 8, 1966, pp.1-27.
2. M. Cavazza, F. Charles, and S.J. Mead, "Character-Based Interactive Storytelling," *IEEE Intelligent Systems*, vol. 17, issue 4, 2002, pp. 17-24.
3. T. Darrell et al., "A Novel Environment for Situated Vision and Behavior," *Proc. IEEE Workshop Visual Behaviors*, IEEE CS Press, 1994, pp. 68-72.
4. R.S. Feldman and B. Rime, *Fundamentals of Nonverbal Behavior*, Cambridge Univ. Press, 1991.
5. X. Marichal, and T. Umeda, "Real-Time Segmentation of Video Objects for Mixed-Reality Interactive Applications," *Proc. SPIE,* vol. 5150, Visual Communications and Image Processing 2003, T. Ebrahimi and Thomas Sikora, eds., 2003, pp. 41-50.
6. A. Nandi, and X. Marichal, "Senses of Spaces through Transfiction," *Proc. Int'l Workshop Entertainment Computing* (IWEC 2002), Kluwer, 2002, pp. 439-446.
7. D. Nau et al., "SHOP: Simple Hierarchical Ordered Planner," *Proc. 16th Int'l Joint Conf. Artificial Intelligence*, AAAI Press, 1999, pp. 968-973.

Figure 10. Multimodal interaction. (a) Bond questions the Professor. (b) The Professor replies, "You must be joking, Mr Bond!" with a corresponding body gesture. The system interprets the multimodal speech act as a refusal (see Figure 9), causing the corresponding task in Bond's HTN to fail.
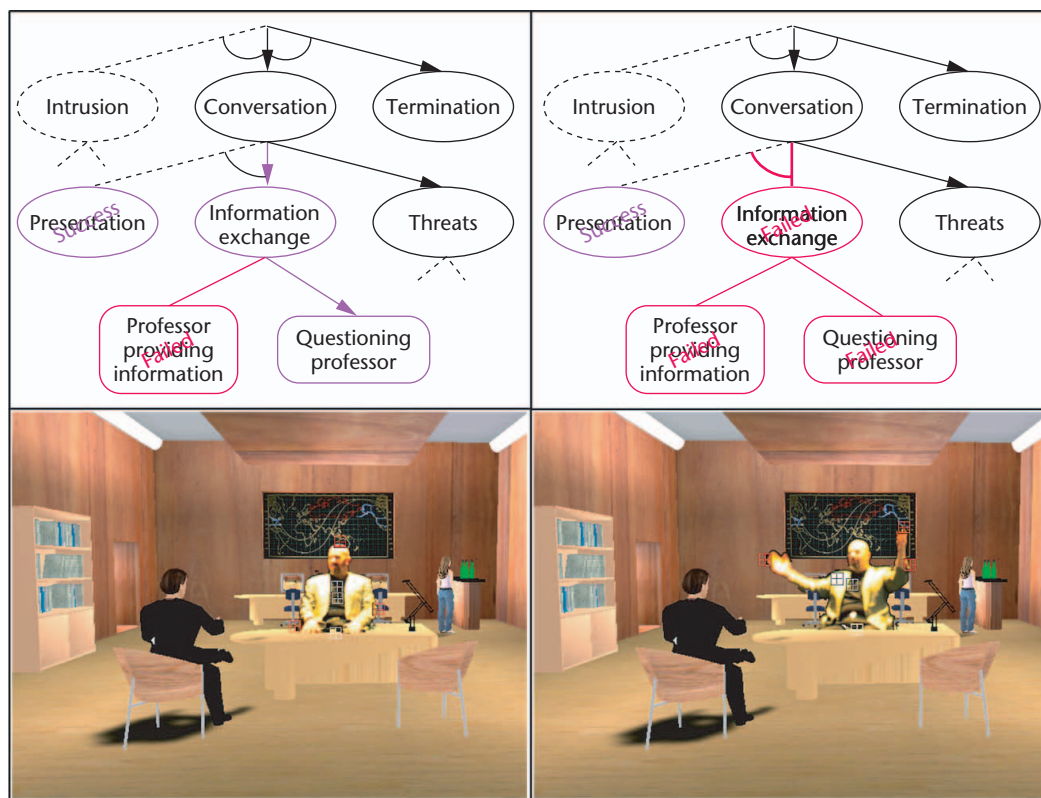
8. S. Oviatt, "Ten Myths of Multimodal Interaction," *Comm. ACM*, vol 42, no. 11, Nov. 1999, pp. 74-81.

9. T. Psik et al., "The Invisible Person: Advanced Interaction Using an Embedded Interface," *Proc. 7th Int'l Workshop on Immersive Projection Technology, 9th Eurographics Workshop on Virtual Enviroments*, ACM Press , 2003, pp.29-37.

10. E. Salvador, A. Cavallaro, and T. Ebrahimi, "Shadow Identification and Classification Using Invariant Color Models," *IEEE Signal Processing Soc. Int'l Conf. Acoustics, Speech, and Signal Processing* (ICASSP 2001), IEEE Press, 2001, pp. 1545-1548.

11. *Comm. ACM*, special issue on game engines in scientific research, vol. 45, no. 1, Jan. 2002.

12. W. Swartout et al., "Toward the Holodeck: Integrating Graphics, Sound, Character, and Story" *Proc. Autonomous Agents 2001 Conf.*, ACM Press, 2001, pp. 409-416.

working on interactive storytelling systems and dialogue formalisms for conversational characters. He received his MD and PhD from the University of Paris 7.

Olivier Martin is a research assistant in artificial intelligence at the Université Catholique de Louvain, Belgium. His main research interest is intelligent agent design for interactive applications, with an emphasis on gesture recognition and behaviour planning. He received his MD cum laude in electrical engineering from the Université Catholique de Louvain.
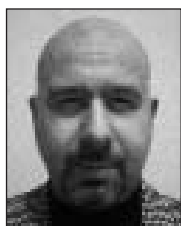
Marc Cavazza is a professor of intelligent virtual environments at the University of Teesside, UK. His main research interest is artificial intelligence for virtual humans, with an emphasis on language technologies. He is currently

Fred Charles is senior lecturer in computer games programming at the University of Teesside and is currently completing a PhD in the area of virtual actors in interactive storytelling. He has a BSc in computer science and an MSc in computer graphics from the University of Teesside.

Steven J. Mead is a lecturer in the computer games programming department at the University of Teesside and is currently undertaking an MPhil in intervention in autonomous agents' behaviors in interactive storytelling using natural language. He has a BSc in computer science and an MSc in computer graphics from the University of Teesside.

Alok Nandi is the co-founder of Alterface, a spin-off integrating real-time interactive applications. His main interests are storytelling (interactive and not) art direction, design, and visualization. He received a BSc in electrical engineering from the Université libre de Bruxelles and a Licence en Philosophie et Lettres, with a focus on cinematographic analysis and writing.

Xavier Marichal is the co-founder of Alterface, a spin-off integrating real-time interactive applications. His main interests concern motion estimation/compensation and image analysis in the framework of distributed interactive settings. He has a BSc and a PhD in electrical engineering from the Université Catholique de Louvain.

Readers may contact Cavazza at the University of Teesside, School of Computing and Mathematics, Borough Rd., Middlesbrough, TS1 3BA, UK; m.o.cavazza@tees.ac.uk.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

39