

Avelino J. Gonzalez*, Jason Leigh, Ronald F. DeMara, Andrew Johnson, Steven Jones, Sangyoon Lee, Victor Hung, Luc Renambot, Carlos Leon-Barth, Maxine Brown, Miguel Elvir, James Hollister and Steven Kobosko

Passing an Enhanced Turing Test – Interacting with Lifelike Computer Representations of Specific Individuals

Abstract: This article describes research to build an embodied conversational agent (ECA) as an interface to a question-and-answer (Q/A) system about a National Science Foundation (NSF) program. We call this ECA the LifeLike Avatar, and it can interact with its users in spoken natural language to answer general as well as specific questions about specific topics. In an idealized case, the LifeLike Avatar could conceivably provide a user with a level of interaction such that he or she would not be certain as to whether he or she is talking to the actual person via video teleconference. This could be considered a (vastly) extended version of the seminal Turing test. Although passing such a test is still far off, our work moves the science in that direction. The Uncanny Valley notwithstanding, applications of such lifelike interfaces could include those where specific instructors/caregivers could be represented as stand-ins for the actual person in situations where personal representation is important. Possible areas that come to mind that might benefit from these lifelike ECAs include health-care support for elderly/disabled patients in extended home care, education/training, and knowledge preservation. Another more personal application would be to posthumously preserve elements of the persona of a loved one by family members. We apply this approach to a Q/A system for knowledge preservation and dissemination, where the specific individual who had this knowledge was to retire from the US National Science Foundation. The system is described in detail, and evaluations were performed to determine how well the system was perceived by users.

Keywords: Embodied conversational agents, chatbots, animated pedagogical agents, dialogue management, automated question-and-answer systems.

*Corresponding author: **Avelino J. Gonzalez**, Electrical Engineering and Computer Science, University of Central Florida, PO Box 162362, 4000 Central Florida Boulevard, HEC 346, Orlando, FL 32816-2362, USA, Phone: +1-407-823-5027, Fax: +1-407-823-5835, e-mail: avelino.gonzalez@ucf.edu

Avelino J. Gonzalez, Ronald F. DeMara, Victor Hung, Carlos Leon-Barth, Miguel Elvir, James Hollister and Steven Kobosko: Intelligent Systems Laboratory, University of Central Florida, Orlando, FL, USA

Jason Leigh, Andrew Johnson, Steven Jones, Sangyoon Lee, Luc Renambot and Maxine Brown: Electronic Visualization Laboratory, University of Illinois at Chicago, Chicago, IL, USA

1 Introduction

Humans are a highly social species that readily communicate with one another. Brief communications have been historically done best via the spoken word, whereas longer, deeper, and more complex communications have been done via the written word. Newly emerging trends toward electronic communication notwithstanding (emails, SMS, online chats, etc.), most people still prefer to communicate via spoken speech to maintain a high degree of personal interaction. Face-to-face communication via spoken words and accompanied by appropriate gestures and expressions is often preferred in order to convey information not as effectively expressed via the more impersonal written communications.

We are particularly interested in extending the normal interactive communication between two humans to one that is between a human and a computer. The notion of such interactive agents has existed since the inception of the computing age. Idealistic visions of these agents are often endowed with extraordinary capabilities. Yet, state-of-the-art technology is only capable of delivering a small fraction of these expectations. The popular media is full of such notional conversational characters, some embodied, others not. The *Star Wars* robotic characters R2D2 and C3PO became cultural icons in the 1970s, even though R2D2 was not particularly articulate. HAL, the intelligent but disembodied computer character in *2001: A Space Odyssey*, hails back to 1984. Another disembodied but highly articulate conversational agent was KIT, the talking Pontiac Trans Am from the early to mid-1980s TV series *Knight Rider*. Yet another popular embodied agent from the same era was the talking computer-generated British talk show host *Max Headroom*. More recently, movies such as *D.A.R.Y.L.* (1985) and *AI Artificial Intelligence* (2001) likewise presented the concept of robotic entities that could communicate with humans in a natural and lifelike manner. The ultimate lifelike machine in the popular media, however, was Commander Data from *Star Trek: The New Generation*, a TV series that ran in the USA between 1987 and 1994. Of course, some of these characters were played by human actors, whereas for others, humans acted in the background, lending only their

voices. In any case, these were certainly not true entities capable of intelligent, computer-generated conversations. Nevertheless, these figments of the popular imagination provide glimpses of our fascination with intelligent and conversational nonliving agents. They also provide us with notional models to emulate, if only in small parts.

We base our research on the concept that interpersonal communications is best done via the spoken word in a face-to-face interchange. Furthermore, we assert that such communication is most effectively done when the interlocutor is an entity that is someone known to us – preferably someone trusted and/or loved or at least someone known and/or respected. Therefore, when communicating with a computer, it would follow that having a computer representation of a specific individual with whom the conversation could take place would be more effective than with an embodied but generic entity and certainly much more effective than with a disembodied entity such as Hal or KIT. If computer interaction is to be optimized for effectiveness, we believe that it must simulate such personal, face-to-face conversation with a familiar entity via spoken speech. This is particularly important for applications where knowing the individual being represented adds value to the communication exchange.

Embodied conversational agents (ECAs) have been the main research discipline in pursuing such objectives. Other commonly used names for ECAs have been *virtual humans* and *avatars* (our preferred name). If done sufficiently well, an avatar that was able to speak in, and understand natural spoken speech, and looked very much like its human counterpart, would be able to pass a (greatly) enhanced version of the Turing test [57]. In this conceptual extension of the classic Turing test, one would converse through spoken speech with an image of a known individual about a deep subject and then be asked to pass judgment on whether one was conversing with the person him/herself via a videoconference medium or merely interacting with his/her avatar. Such a test has been already suggested in the literature – Barberi's ultimate Turing test [2], although with somewhat different objectives and in different format. Clearly, we are very far from achieving that at this point, and we certainly do not claim that our LifeLike Avatar passes this test. Nevertheless, our research described here clearly takes us in that direction.

Our work is not the first or the only one on such avatars; we discuss the state of the art in Section 2. Nevertheless, our work is an attempt to put together a synergistic set of components that has passing the above-defined Enhanced Turing Test as an ultimate objective. This article describes our work, our accomplishments, our evaluations, and a discussion of the extensive future research that yet remains to be done to accomplish this goal.

2 State of the Art in ECAs

Chatbots (or chatterbots) are the evolutionary ancestors of ECAs. The original of these, of course, was the seminal ELIZA [61], a disembodied conversational representation of a Rogerian therapist that started this research area in earnest. In 1972, Colby [13] created the most well-noted successor to ELIZA, a chatbot named PARRY, which simulated conversations that one might hold with a paranoid schizophrenic individual [25].

In 1990, Mauldin [38, 39] created a bot named Julia, the first of what would be called *verbots*, as described by Foner [22]. Verbots have the ability to process natural language as well as some behavioral rules that guide their responses [38, 39].

Cassell et al. [11] provided some insights into human – computer interaction in a physically immersive environment with their conversational playmate Sam. Sam was not an autonomous avatar, but its creators were able to show that a child could interact with an artificial being in a reasonably effective manner. Bickmore and Picard [5] created Laura, a personal trainer agent that also is able to communicate via typed text to answer questions from a human user. Laura's interactions with the user were one-sided, where Laura controlled the dialogue at all times.

Wallace's [60] ALICE and ALICE's Program D implemented sentence-based template matching, an expansion of Eliza's keyword matching concept. Lee et al. [33] created an animatronics penguin named Mel to study the use of robots as conversational agents. Lee et al. [33] reported that humans were able to effectively interact with a physical and conversationally interactive robot. Kenny et al. [32] furthered the concept of interacting with ECAs with their Sergeant Blackwell conversational agent, which provided the user with a more natural human – computer interaction. Two other ECAs – Sergeant Star [1] and Hassan [56] – followed and were based upon SGT Blackwell's architecture.

Composite bots emerged also in the 2000s. These bots consist of a combination of techniques, such as pattern matching, machine learning, Markov models, and probabilistic modeling. One of the most notable achievements in composite chatterbots follows the Jabberwacky approach [10], which differs from other chatbots in that it learns to talk by exploiting the context of previous conversations it has stored in a large database.

In the midst of all these advances, Mori's [44] Uncanny Valley introduced some perspective into how humans perceive lifelike agents. Mori found that if the avatar is human-like but not too much so, then it is viewed as positive. However, if a lifelike avatar is too human-like, then a feeling of revulsion arises in the human interlocutor. The Uncanny Valley has produced much discussion, controversy, and follow-up research. Nevertheless, our work hopes to move past the Uncanny

Valley – to the point where the agent is so real and lifelike that a user would not know for certain that it is artificial.

Although the above discussion is far from being exhaustive and we have left out many important achievements in chatbots and avatars, it is fair to say that no chatbot/avatar described in the literature combines a lifelike appearance along with intelligent spoken dialogue in a deep subject matter. These are some of the required features in an avatar that would pass the enhanced Turing test. We do not pretend to have a solution at the moment. We only report here what we have done toward that end. We call the results of our work the LifeLike system, and it is described below.

3 The LifeLike System

To successfully fool a human user into thinking that an ECA is in fact a human, first and foremost requires a lifelike visual appearance, preferably one that strongly resembles a known individual. Without that, the avatar has no hope of ever passing the Extended Turing Test as defined here. The specific vehicle that we use for imposing such subterfuge on a human user is an avatar that we call the LifeLike Avatar. In its first implementation as part of the sponsoring grant from NSF, it used the image of Dr. Alex Schwarzkopf, a long-time program manager at the NSF (see Figure 1 for a look at the latest version of the LifeLike Avatar).

Figure 2 depicts the LifeLike avatar in its “office,” ready to field questions from the users. The whiteboard on the left of the screen serves to display answers that are long or that have graphics and therefore cannot be easily articulated orally. The avatar in Figure 2 is the next-to-the-last version of the avatar. Note



Figure 1. The LifeLike Avatar.



Figure 2. The Lifelike Avatar in its Environment, Ready to Talk to the Users.

how its facial features appear significantly more artificial than the latest version shown in Figure 1.

Compare the avatar in Figure 1 with a photo of the real Alex Schwarzkopf as shown in Figure 3. Although no sane person would confuse them at this time, our anecdotal evidence reports that students who had worked with the avatar but neither met Dr. Schwarzkopf in person nor had seen actual photos of him recognized him right away in a crowd of people when seeing him for the first time. The process of creating the Lifelike Avatar is covered in Section 4 of this article.

Also essential for passing the Enhanced Turing Test is an intelligent dialogue that is both natural and knowledgeable – to the same extent as the person being represented. Furthermore, it should sound like him/her and show the same gestures and emotions. Therefore, we posit that there are three specific characteristics that can make an advanced avatar seem natural to humans. These are



Figure 3. Comparison of the Lifelike Avatar to its Human Counterpart.

- A visually lifelike virtual embodiment of a specific human being that has a strong resemblance to that person.
- Communication in spoken natural language – input and output – combined with generation of nonverbal gestural expressions. This requires at the very least an automated speech recognition (ASR) system and a dialogue manager (DM) capable of understanding the words spoken and able to compose a response.
- A knowledgeable backend that can respond intelligently to questions on a specific topic. Ideally, the avatar would also be able to remember details about the conversation and learn about that topic through the interaction.

Our LifeLike system, therefore, has two basic components: The visual avatar framework component that encompasses the first item above (discussed and described in Section 4) and the intelligent communication system component that encompasses the last two items (discussed in Sections 5 and 6). This article describes the entire project in a capstone sense, with the deep details left for some of our other publications that are referenced here.

4 The Visual Avatar Framework

Currently, the most extensive use of avatars has been in computer/video games. The current state of the art in gaming technology for creating realistic responsive avatars follows the model of a finite-state system that responds to menu-based dialogue selections by initiating a prerecorded narrative that is synchronized with a specific motion-captured sequence. This section of the article describes the graphical/visual aspect of the LifeLike avatar. We begin with describing the avatar framework.

4.1 LifeLike Responsive Avatar Framework

The LifeLike Responsive Avatar Framework (LRAF) is essentially the main system by which all the components that drive the avatar are tied together to create a realistic representation capable of receiving speech input and providing an emotive response and a vocal response. The key goals are to understand what are the essential main components of this framework and to then develop such a framework for creating a realistically behaving avatar.

Figure 4 illustrates the LRAF functional architecture. The LRAF has two separate input sources. One is the LifeLike Dialogue Manager (see Section 5), which provides sentences that are intended to be spoken by the avatar. The other is the

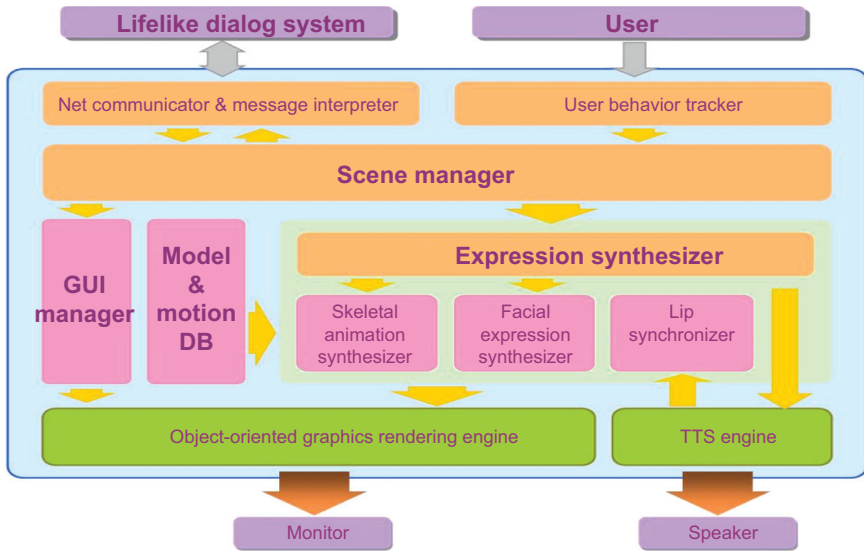


Figure 4. LifeLike Responsive Avatar Framework.

user's behavioral information, such as eye gaze. The most significant component of the LRAF is the Expression Synthesizer, which is responsible for taking the 3D facial models and applying the motion-capture data to produce a sequence of facial and body animations that fit the context of what has been spoken. Three major components of the Expression Synthesizer are (1) the Skeletal Animation Synthesizer, (2) the Facial Expression Synthesizer, and (3) the Lip Synchronizer.

The scene of our prototype system was displayed on a large 50-in. screen so that the avatar would appear close to life-sized (see Figure 5). It consists of the



Figure 5. LifeLike Responsive Avatar Framework Displayed on a 50-in. Screen for a Life-Sized Appearance.

avatar sitting behind a desk in a typical office setting. To interact with the avatar, a user wears a headset microphone or uses a directional desktop microphone to speak to the avatar, then the avatar will respond to the user's request via speech, with auxiliary information on the display above its right shoulder, along with natural nonverbal gestures including body motion and facial expressions.

The development of the Graphical Asset Production Pipeline (GAPP) and of the LRAF lays a foundation for methodically investigating two issues: (1) What is necessary to create a believable avatar? (2) For what types of tasks are human–avatar interactions most suited. We discuss these issues next.

4.2 Modeling the Human Subject

A GAPP was developed that encapsulates the tasks needed to create a visual representation of a human character. Investigations were conducted to identify and test the interoperability of tools for facial modeling, rendering the real-time graphics, motion-capture, and text-to-speech (TTS) synthesis. Furthermore, we examined and evaluated the options and best practices for recording vocal mannerisms and nonverbal mannerisms. We conducted observation and recordings of our subject (Alex Schwarzkopf).

Before developing the method, we surveyed and evaluated several existing open-source and commercial software and hardware tools that could form the foundation of our work. Software tools included libraries for realistic digital facial modeling, motion-capture data reconstruction, real-time graphics rendering, and speech synthesis. Hardware tools included a motion-capture system and display configurations for representing the avatar. We discuss these next.

4.2.1 Facial Modeling

FaceGen Modeller [20] is a tool for generating 3D head and face models using front and side photographic images. The technique was used to develop the highly acclaimed video game, Oblivion [49]. Figure 3 shows the resulting 3D head of Alex Schwarzkopf generated by FaceGen. FaceGen provides a neutral face model that can be parametrically controlled to emulate facial expression (Figure 6). In addition, FaceGen enables the user to control the gender, age, and race of the model. Although this is a good initial prototype, much can still be done to improve the visual realism by improving skin texture and developing high-quality skin-rendering techniques such as the subsurface light scattering properties of skin tissue.

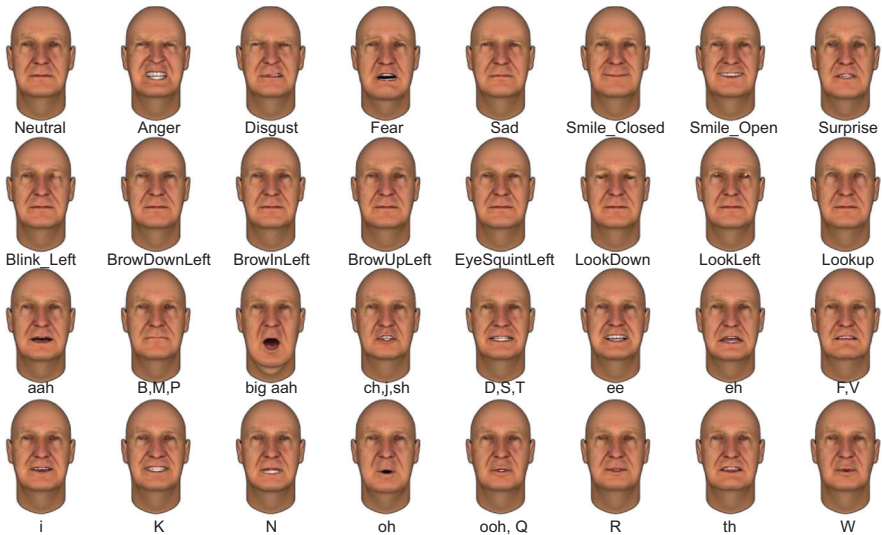


Figure 6. Facial Expression Morph Model.

4.2.2 Texturing

Visual improvements to the avatar’s head model were created by more accurately mirroring the key facial poses of the avatar with video and photographic images of the real Alex rather than using the default poses provided by the FaceGen software. In addition, texture blending was improved to enable more of the captured photographic imagery of Alex’s skin to show through, rather than relying on a computer-generated skin texture. However, there is still a “waxy” quality to Alex’s skin tone and a general lack of visible depth from imperfections such as pores and scars. We can improve this rendering result even further with advanced rendering techniques such as bump mapping, subsurface scattering, and displacement map.

Diffuse mapping typically provides the base color for a 3D object in a scene. Our Alex avatar’s prior diffuse map was automatically generated by FaceGen using a front and side photograph of the actual person. Two severe limitations of this software were that the resulting resolution is relatively low and the color blending was biased toward more cartoon-like or video game depictions (see Figure 7, left, and Figure 8, left). By reconstructing a high-resolution diffuse texture from numerous photographs of the subject by projecting 2D photos onto a 3D face model, we were able to obtain significantly enhanced depiction of variations in skin tone (see Figure 7, right, and Figure 8, right).

After the generation of head model, it is exported into a modeling tool as static shapes. The LifeLike visualizer uses these models to create weighted facial



Figure 7. Comparison of FaceGen Low Diffuse Map (Left) vs. our Custom Map (Right).

morphing animation in real time. Shape animation is similar to the bone controlled parametric model that is common in many studies about virtual characters or avatars. In general, morph animation based on shape results in better quality because all target meshes are well designed and precise in real-time control.

4.2.3 Body Modeling

Even with off-the-shelf tools at our disposal, the process of creating a virtual human is still a very time-consuming task. We formalized the process in terms of a GAPP (Figure 9). The first part of pipeline is to design a polygonal model of the character consisting of the head/face and body. The second part of the pipeline is to acquire and edit recorded motions of the character. In the film and video gaming industries, motion capture is still much preferred over algorithmic



Figure 8. Comparison of FaceGen Diffuse Map (Left) vs. our Custom Map (Right) When Applied to the Avatar Model.

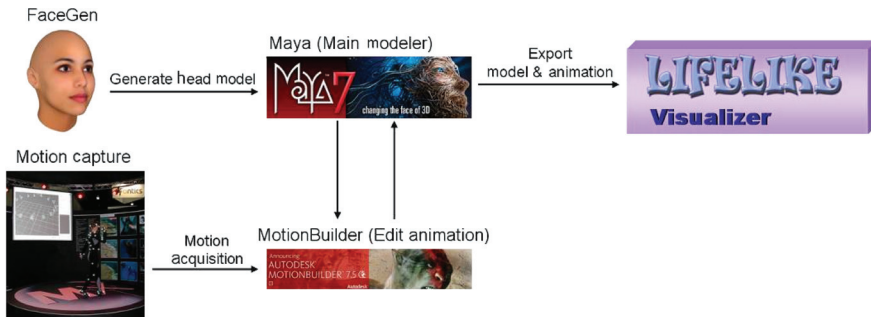


Figure 9. Graphical Asset Production Pipeline.

creation of motion because it is still extremely difficult to codify naturalistic, and especially subtle, human gestures.

Tools such as Autodesk Maya [40] and MotionBuilder [45] form the middle portion of the pipeline where the motion-captured animations are attached (or “rigged”) to the 3D character models. Once completed, the resulting model is ready for application-based control using the LRAF. The most important aspect of full production pipeline is that data had to be easily exchanged among different tools. The FBX file format is used to solve this compatibility issue [21]. FBX supports all polygonal mesh models including material properties (color and texture) and skeleton animation without any data loss.

The correct dimensions of Alex Schwarzkopf were gathered from photographs and measurements taken. Schwarzkopf’s body model consisted of 30,000 triangles (including the head) with 72 bones rigged (Figure 10). During the design of the full-body model, we developed a custom Maya Mel script to automate the integration of FaceGen head models with full-body models. The script exchanges the current head model with any new head design and automatically updates skeleton and morph shapes so that the designer can use the new head model immediately without any manual work.

4.3 Animation

We now discuss the animation of the LifeLike Avatar.

4.3.1 Motion Capture

Motion capture is the most widely used approach for acquiring realistic human figure animation for the film and video game industries. The focus in the project

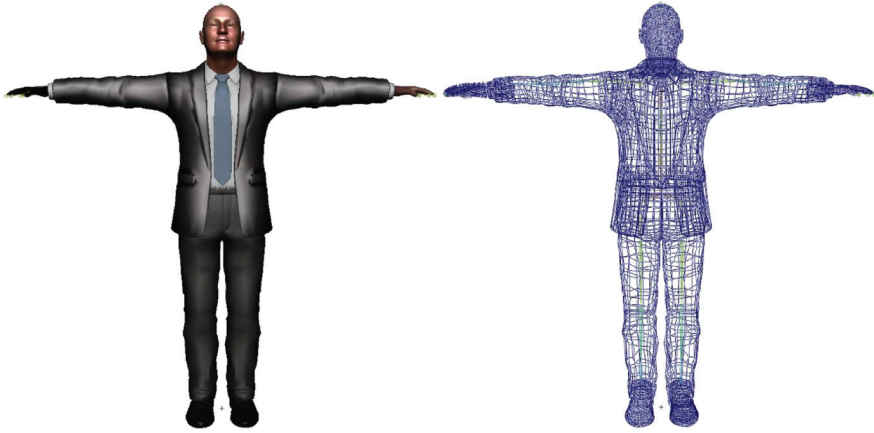


Figure 10. Full-Body Model with Skeleton Rigging (T-Pose of Full-Body Model).

is to capture a series of simple motions and enable our avatar to “re-enact” them. In December 2007, Dr. Schwarzkopf participated in a motion-capture session. We used Vicon motion-capture system with eight high-resolution (MX-F40, 4 megapixels) infrared tracking cameras (Figure 11) [58].

After the modeling phase, the animated avatar is created by a combination of motion-capture and manual fine-tuning using the Maya and MotionBuilder tools. Motion-capture data can be converted into either marker transformation data or kinematic fitted bone transformation data. When using marker data (Figure 12,

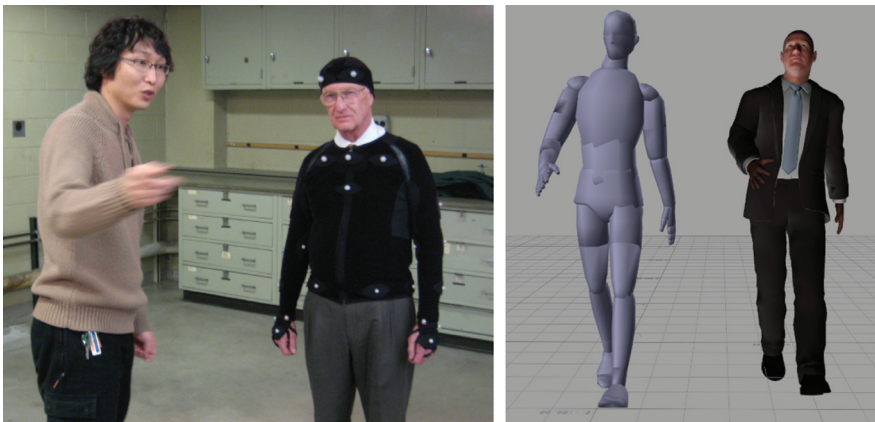


Figure 11. Performance Capture: Dr. Schwarzkopf in Motion-Capture Session (Left) and Motion-Capture Data Retargeting (Right).

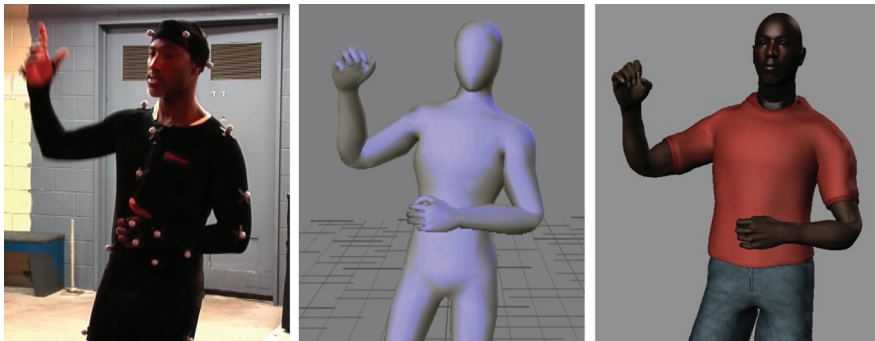


Figure 12. Motion-Capture Performance (Left), Character Template Data (Center), and Character Retargeting Data (Right).

left), one can map markers onto a dummy character template (Figure 12, center). This then enables the creation of a character that can re-enact the motion-captured movements. (Figure 12, right).

4.3.2 Motion Composition and Blending

One of the key challenges to creating realistic motion animation for an avatar is ensuring that the transitions between the individual motion-captured sequences appear smooth. We realized that many of the sequences that we captured of Alex were very dissimilar, making motion interpolation difficult. A well-known approach to overcoming this problem is to exhaustively evaluate the similarity of a given pose to the next in a large motion database [35]. Using lengthy motion clips from a variety of behaviors such as walking, running, and jumping, one can construct a comprehensive motion graph from which any natural motion can be synthesized [54]. Although this approach is suitable for non-real-time applications, it is not suitable for real-time applications because as the database grows, the search space of possible animations leading to the final goal state grows dramatically. In LRAF, we took the approach used by many video games. That is, the avatar's motions are classified into several major categories such as sitting, turning, pointing, etc., and within each category, a similarity measure (e.g., based on distance between morph points) is computed whenever one motion from one category (such as sitting) needs to transition to the next (such as pointing). This helps to dramatically prune the search space, making it possible to compute in real time (at least 60 frames per second).

Avatars intended to mimic human behavior need to behave somewhat non-deterministically, or else they will appear unnatural and mechanistic. To accommodate this, we devised the concept of a semi-deterministic hierarchical finite-state machine (SDHFSM). An SDHFSM is a hierarchical finite-state machine where a substate is chosen either based on a set of constraints or randomly given multiple possible options. For example, the highest level of hierarchy of an SDHFSM to model avatar behavior may consist of states for speaking, idling, or body motion. Within the body motion substate, there are multiple substates that consist of a variety of behaviors such as pointing left or pointing right. When an avatar transitions from an idle state to a body motion state, it must select a state whose motion-capture information is kinematically compatible with the avatar's current motion state (Figure 13). By kinematically compatible, we mean that an avatar can transition from one sequence of motions to another in a way that appears to be within the capabilities of physics. As each kinematically compatible substate may have multiple possible motions (there are many different ways to point to the left, for example), the choice is made randomly and nonrepetitively to avoid appearing robotic or mechanistic.

4.3.3 Facial Expression

Autonomous facial expressions (AFE) were implemented in LRAF as a means for the avatar to alter its facial expressions to control lip synchronization or emotion. The AFE works by either receiving explicit events from the context-based dialogue manager (DM) or through randomly generated expressions when the avatar is in an idle state. The latter is used to create involuntary actions such as blinking and small movements of the mouth.

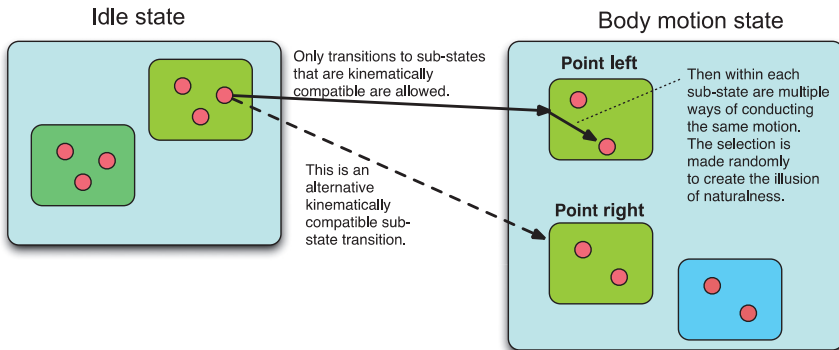


Figure 13. Semideterministic Hierarchical Finite-State Machine.

Through an avatar’s XML definition file, the developer can customize a neutral facial expression as well as any desired random expressions (including mouth movements) that may be idiosyncratic to the character being mimicked. Furthermore, multiple individual expressions can be grouped into hierarchies to simplify the creation of combined expressions. The following XML snippet illustrates one example of idle mouth animation that includes three different unit shapes in one group (“Mouth0”).

```
<IdleExpression>
  <IEGroup name="Mouth0" initialOffset="2.0" frequencymin="4.0"
    frequencymax="10.0" durationmin="0.4" durationmax="1.0">
    <Shape name="Phoneme__aah" wmin="0.2" wmax="0.4" offset="0.0"/>
    <Shape name="Phoneme__R" wmin="0.1" wmax="0.45" offset="0.5"/>
    <Shape name="Phoneme__oh" wmin="0.1" wmax="0.35" offset="0.7"/>
  </IEGroup>
</IdleExpression>
```

In addition to autonomous random generation of facial expression for involuntary motion, a human expresses emotions in terms of facial feature movement upon its mental status. Profound researches on this subject have been conducted in multiple literatures. One well-established study about emotional facial expression is Ekman’s categorical expressions [16]. Ekman proposed six basic emotion categories – anger, disgust, fear, happiness, sadness, and surprise. These definitions of human emotion have been adopted in the vast amount of research that followed it. Our Facial Expression Synthesizer implementation is also relying on this work.

Our first prototype framework used these six basic emotions tied with corresponding facial expressions generated by FaceGen software. One drawback in this approach is that facial expression mechanism always uses the same morph shape for the given emotion, which makes our avatar not as natural as a real person. The advanced emotional expression generator in LRAF incorporates empirical facial expression database to solve this unnaturalness. LRAF uses the CK+ facial expression database [30, 36] to decompose lower-level facial feature sets for each expression with over 100 different subject data found in the database. Figure 14 shows one example from the database. Facial features in CK+ database are encoded in Facial Action Coding System (FACS) [17] to describe each expression. Extended LRAF facial animation data includes individual action unit (AU) as separate morph shape to realize this variation in facial animation. When the Facial Expression Synthesizer receives an elicited emotion for the avatar, the synthesizer samples the database and recomposes the corresponding facial expression with the retrieved AU information.

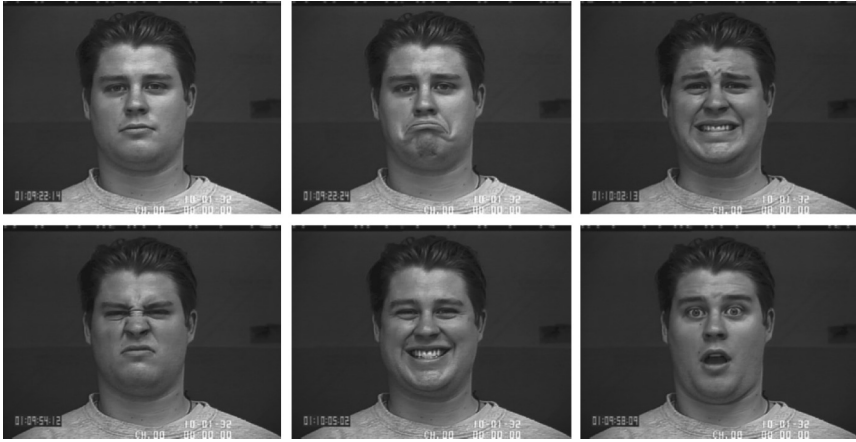


Figure 14. Cohn-Kanade Facial Expression Database (Subject 132 ©Jeffrey Cohn).

4.3.4 Lip Synchronization

Lip synchronization with voice is a crucial aspect of facial animation. The LRAF supports two methods to control lip shape. The first model uses Microsoft Speech API (SAPI) [42] TTS and its visual lip shape (viseme) events that are synchronized with the TTS engine to select the appropriate mouth shapes to depict. The second model uses a recorded audio file and its amplitude and frequency information to drive lip shapes. This enables the avatar to use both computer-synthesized speech as well as directly recorded speech.

Microsoft's SAPI was chosen as the API for enabling text-to-speech (TTS) synthesis. It provides an event-generation mechanism to report the status of the phoneme and its corresponding viseme during the synthesis of voice. These events can be used to create realistic phoneme-based lip animations. Furthermore, a number of commercial speech systems provide an interface to SAPI so that an application can transparently leverage a multitude of speech systems. As phoneme information is a type of discreet event, Lip Synchronizer interpolates it to co-articulate lip shape-morphing values in between events. For example, when one phoneme event arrives, it starts the fade-in phase by increasing the related shape value; then, it initiates the fade-out phase upon receiving the next one. This simple yet effective linear interpolation of individual lip shape animation gives natural and smooth transition for the synchronous lip animation for a given speech synthesis.

Lip synchronization for prerecorded speech is designed to modulate main speech amplitude together with subdivided frequency bands. Our approach continuously monitors 28 frequency bands of an audio stream and uses them to

select different lip shapes. This per-frequency band mapping method is based on the premise that certain frequencies of a speech can only be produced with specific mouth shapes. However, since a perfect one-to-one mapping is not possible and likely to be unrealistic, we use multiple group mappings between frequency bands and mouth shapes, e.g., many bands to single mouth shape; single band to many shapes; many bands to many shapes depending on programmable criteria, such as which band(s) came before (Figure 15).

In October 2009, three members of the team visited Alex Schwarzkopf for a day to record a large bulk of corpus materials for the new lip synchronization technique. By recording Dr. Schwarzkopf's voice, we are able to make the avatar more closely resemble the real person and create a more realistic interface users will associate with a human. Six hours of recording based on a script derived from the knowledge system was cleaned, manipulated, and produced into a database of utterances that can be combined to create a number of phrases determined by the dialogue system.

4.4 Rendering

We evaluated two open-source graphics engines for real-time rendering implementation (Blender3D and OGRE (Object-oriented Graphics Rendering Engine)). Although Blender3D [6] provides a rich set of development capabilities, including a 3D modeling, animation, and scripting system, it did not have sufficient data exchange support to enable the loading of motion-captured data sets from commercial tools such as Maya and MotionBuilder. Furthermore, Blender3D does not allow applications to manually control bone-level animation, which is necessary to enable the avatar to do more than simply play back a prerecorded motion-

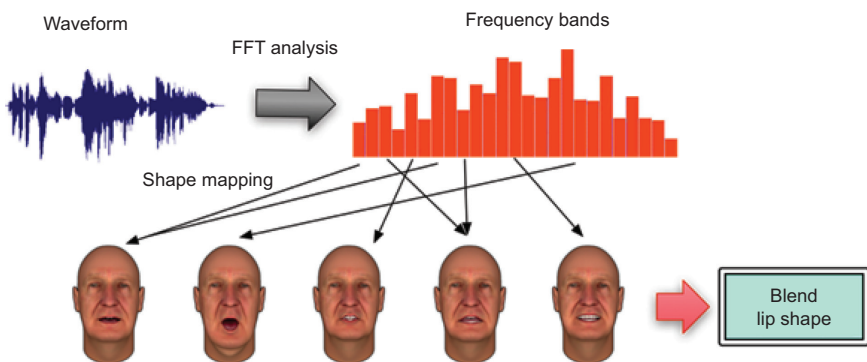


Figure 15. Recorded Voice Lip Synchronization Model.

capture sequence. OGRE [51] proved to be a better fit. Its plug-in-based architecture provides greater ability to interoperate with other software tools. OGRE provides both a high-level interface for interacting with graphical objects and a low-level Shader control to create specialized visual effects to create more realistic avatars.

Even when watching the most sophisticated computer-generated humans today (such as in the recent film, *Beowulf* [3]), many in the media have commented on the apparent “hollowness” in the digital character’s eyes. Furthermore, others have noted that an older avatar appears to look more realistic than a younger avatar, perhaps because younger avatars have fewer blemishes that would tend to make it resemble the traditional smoothly polished computer-generated image. We noticed this phenomenon in the creation of our Schwarzkopf avatar where, because of a limitation in the commercial FaceGen software, the exported head had considerably less pronounced skin texturing than the real person, which tended to make the avatar look plastic and doll-like.

Investigations in high-quality rendering included the development of an improved skin rendering method, in particular one that was better able to capture the appearance of more elderly subjects; the image space normal mapping to enhance skin details; the development of reflection and refraction techniques for depicting realistic eyeglasses; the creation of soft shadows to heighten the illusion of spatiality/depth of the depicted avatar; and the stereoscopic rendering techniques for the 3D display system to create immersive lifelike avatar representation.

4.4.1 Realistic Skin Rendering – Subsurface Scattering

The realistic rendering of skin is an active area of research, largely motivated by Hollywood and the video game industry. Prior work has shown that precise modeling of translucent multilayered human skin (subsurface scattering, or SSS) can produce very realistic results [15, 29]. Recent approaches have taken advantage of graphics processing units (GPU) to enable them to run in real time [19]. By implementing a prototype of these algorithms, we discovered a limitation of existing algorithms – namely that current GPU-based approaches to SSS only work for static faces, not animated ones. Figure 16–18 show the result of our implementation of GPU-based SSS shader (intermediate screen space rendering result during shading pipeline).

The reason for the limitation on static mesh rendering is that the SSS approach uses texture-based computation to create the complex surface geometry. Because animated faces are created by morphing between geometric shapes, one must recompute the vertex normal whenever there is a change in the vertex position of the geometric control points of the face. This problem is solvable by applying additional computation in the geometry shader and providing new vertex data to



Figure 16. Predefined Map Source for SSS Rendering: Color Texture Map (Left), UV Coordinate Snapshot (Center), and Object Space Normal Map (Right).

the next shading stages (vertex and fragment shader). In addition, the UV stretch map also needs an update at each frame as correlation between the UV coordinate and vertex position changes as the mesh deforms on the fly.

A second major challenge arises at the seams between two geometric objects – for example, the face and the neck – which are often depicted as two separate geometries with their own texture material information. Because the two geometries have discrete textures to begin with, there is often a visible seam between them during convolution filtering. To solve this issue, we extended neighboring texture boundaries so that resulting rendering minimize these texture seams.

4.4.2 Skin Normal Map

Normal mapping is one of the most common rendering techniques for enriching polygonal surface-rendered images with fine-grained details without incurring

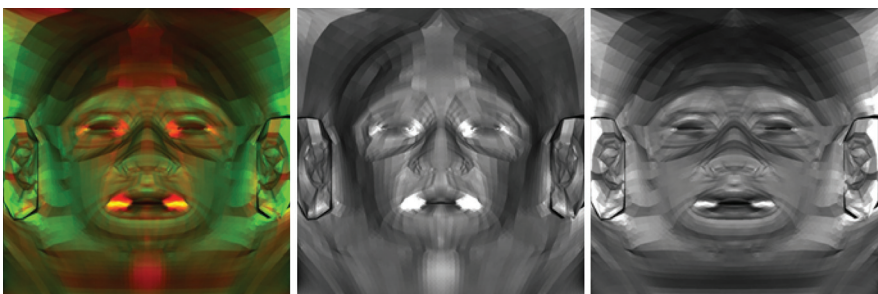


Figure 17. UV Stretch Map: Used to Adjust Texture Map Stretching Effect: UV Stretch Map in Four-Component Color Space (Left), U Coordinate Values (Center), and V Coordinate Stretch (Right).

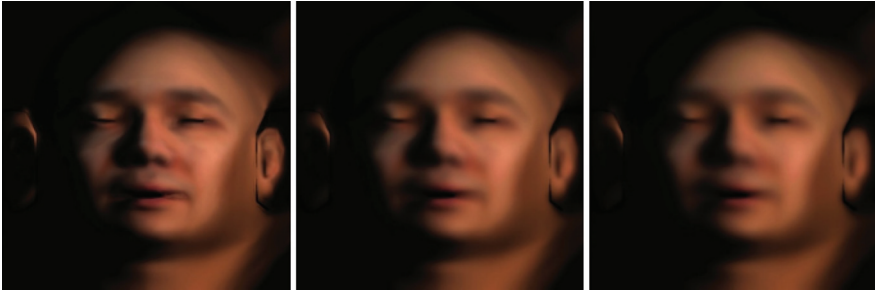


Figure 18. Multiple Gaussian Convolution Filtering: SSS Requires Six Separate Convolution Filter Operation on Texture Space Render Result to Compute Skin Diffuse Profile. Three Pictures Show the First, Third, and Sixth Filtering Result.

the cost of increased geometry complexity. This therefore allows for an image to appear to have much greater visual detail and yet can operate in real-time on modest computing hardware. Pixel-based (image-based) lighting from normal maps contribute greatly to the reduction of polygon complexity without losing detail. In particular, the parallax normal mapping algorithm supports view-dependent surface unevenness in addition to per-pixel bumpy lighting effects [31].

In our prototype application, a high-resolution tangent space normal map texture was extracted from a diffuse color map by analyzing the gray-scale height map by [47]. Figure 19 shows partial maps from a 4096×4096 full-sized normal map of Alex Schwarzkopf's skin. This technique is an excellent method for estimating the “bumpiness” of skin, in particular older skin, without the need to use expensive imaging hardware.

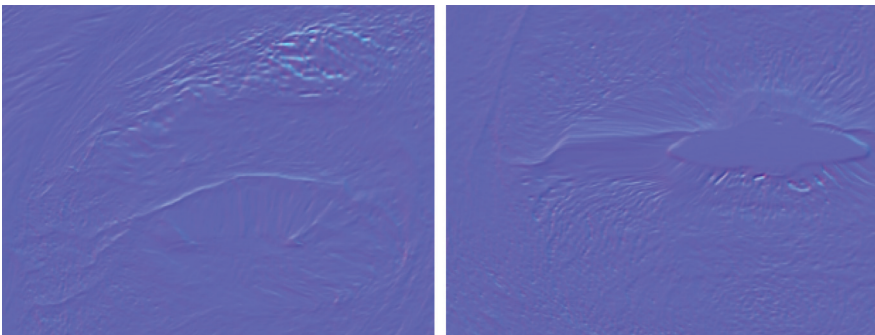


Figure 19. Tangent Space Normal Map Extracted from Diffuse Color Texture: Region Near the Eye (Left) and Region Near the Lip (Right). In a Normal Map, the Red, Green, and Blue Components of an Image Correspond to the x, y, and z Vectors of a Normal to the Surface of the Skin.

Figure 20 shows the result of the application of the parallax normal map rendering method. Note that the result without normal mapping lacks naturalness at the wrinkles and skin pores even though skin tone is properly depicted. Normal mapping enables both heavy wrinkles and pores to be more realistically visualized.

4.4.3 Wrinkle Map

The normal map technique in the previous section is a good low-bust method to improve skin details. It is mostly suitable for the static model; however, it lacks the ability to show dynamic wrinkles during facial muscle movement because the static skin normal map stays same regardless of facial animation status. This is especially an issue in areas of heavy wrinkles such as nose wrinkle and mouth furrow. Therefore, it is necessary to decompose the skin normal map into two or more maps to support dynamic wrinkle generation for real-time facial expression changes.

The method in the dynamic wrinkle texturing is to split the normal map into one for more permanent fine details such as pores and scars and another for heavy wrinkles such as nose wrinkle, mouth furrow, and eye corner wrinkles. The latter ones can be controlled by the facial muscle contraction caused by the facial morphing for various expressions. For example, as an avatar moves the upper lip up, it is likely showing deepened mouth furrow. This will activate a wrinkle map region for mouth furrow to tell the wrinkle shader to increase blending factor for the given wrinkle normal map. Figure 21 shows our wrinkle map rendering result for an anger expression. Applied wrinkle map shader in Figure 21 (center) presents improved facial details over nonwrinkle map rendering (Figure 21, left).



Figure 20. A Comparison of the Avatar with (Right) and without (Left) a High-Resolution Parallax Normal Mapping.



Figure 21. Wrinkle Map Example: Simple Shading Model (Left), Wrinkle Map Applied (Center), and Used Tangent Space Wrinkle Normal Map (Right).

There are a few recent studies that deployed this method to depict more natural dynamics of facial wrinkle generation. Dutreuve proposed a dynamic wrinkle control via skeletal animation poses as it controls facial animation with bones [14]. Another example is a region-based mask to activate a certain area of the wrinkle map when an application triggers an expression associated with the wrinkle region [48]. The latter approach is more appropriate to our framework because we use the morph target method instead of the skeleton to animate an avatar face.

We further devised the wrinkle map region masks for AUs in FACS so that the wrinkle map shader properly activates each region upon individual facial feature on the fly. This method saves texture memory space by using a low-resolution mask map with a high-resolution single wrinkle map. The blending factor for the wrinkle map is determined by the weight values for each AU. The assignment to connect an AU and a wrinkle region is defined in the avatar XML specification file.

4.4.4 Soft-Edged Shadow

The depiction of dynamic real-time shadows is another important technique for enhancing the illusion of depth of a scene or a 3D object in a scene. Recent advances in GPU enable the generation of high-quality soft-edged shadows. Soft shadow techniques [62] were investigated into our avatar-rendering framework producing the final version of the avatar (Figure 22).

The depiction of reflective materials in 3D objects requires complex and multiple rendering pipelines to draw the reflected image on a mapped surface (in this case, the lenses of the avatar's eyeglasses and eyeballs).



Figure 22. Result of Real-Time Soft-Edged Shadow Rendering.

4.4.5 Reflective Material

Techniques such as environment mapping have been frequently used to minimize rendering overhead. In this technique, a predefined reflection image is taken and mapped to vertex coordinates, providing a suitably realistic simulation of a reflection. This technique also works well for dynamic objects, as it uses camera position, reflection point, and an inverse ray to the scene to determine the appropriate mapping coordinates. Our avatar rendering system supports these shading methods. However, we have also enhanced these methods using various texture blending techniques to provide a more accurate depiction of the reflection material itself.

Figure 23A shows the first test results to investigate refractions on a nonflat surface. Utilizing a grid texture, we can see non-uniform light reflections upon each normal vector at the vertices.

After computing the proper UV coordinates for a reflection map, a single opaque texture is applied to the object's surface. Figure 23B shows a sample texture taken in common office environment to capture the florescent lights on a ceiling (Figure 23B, left).

Because our avatar's glasses are not heavily coated sunglasses, a second texture layer was incorporated to increase transparency. The advantage of using a transparency texture instead of traditional parameter-based fixed transparency is that it provides the possibility for varying transparency over an area. In this particular example, a gray-scale image (Figure 23C, left) controls the intensity of the transparency – darker areas are considered more transparent.

Once implemented, the results were impressive; however, the image appeared too “perfect.” To incorporate imperfections, such as fingerprints or dust on the eyeglasses, a dust texture created from random noise filter was also blended with the previous series of textures. This dust texture multiplies the intensity of the

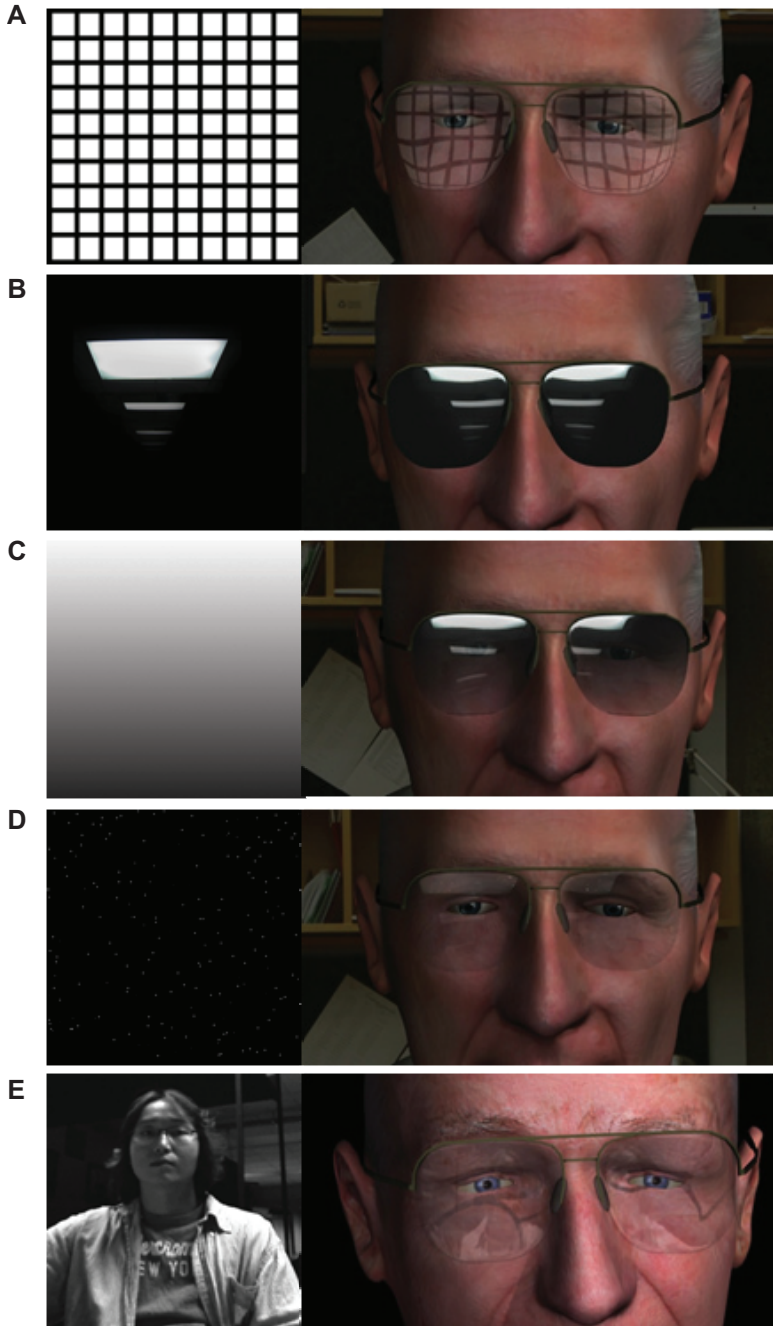


Figure 23. Iterative Design of Reflective Material Rendering Technique.

computed color with the previous two textures, resulting in subtle highlights that appear like dust specs on the lenses surface Figure 23D.

The last improvement made to the model was to use a real-time image captured of the environment rather than a presampled image (such as the one depicted by the florescent lights). This enables the person actually speaking with the avatar to appear directly in the reflection of the eyeglasses as well as the avatar's eyeballs. To achieve this, the live image from a web camera in gray scale was used to produce the dynamic reflection map (Figure 23E).

4.4.6 Stereoscopic Rendering

In the past, the stereo display system had been a specialized system for a certain applications; however, it is becoming more prevalent in various domains. Most of recent commercial TV is now capable of 3D display. As the popularity of 3D display increases, it is necessary for our framework to accommodate such capability with appropriate stereoscopic rendering techniques. This will therefore allow our avatar to participate in fully 3D immersive environments.

The LRAF implementation of stereoscopic rendering supports various stereoscopic separation schemes such as anaglyphic, side-by-side, and interlaced pair of images for different types of display system. Asymmetric frustum parallel axis (off-axis) projection method is implemented in the LRAF-rendering pipeline to generate a user-centric stereo images (Figure 24).

The LRAF rendering engine draws the scene twice at each frame by alternating camera position and view frustum for each eye. Then, the two images are merged at the final composition stage to generate frame buffer to display. The composition method is specified upon the stereo display system type. Figure 25 shows one example of this composition result using interlaced method that is

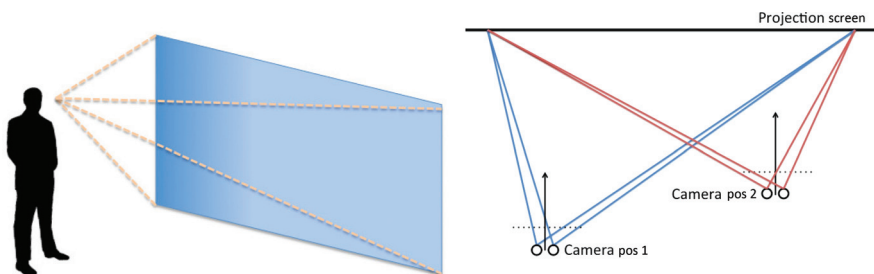


Figure 24. Head-Trackered View Frustum for Off-axis Stereoscopic Rendering. View Frustum for Stereo Rendering is Computed Based on User's Head Position so that Application Perspective Corresponds to the Through the Window Style Immersive Environments.



Figure 25. Stereoscopic Rendering Image Using Interlaced Scheme.

supported by the passive 3D display monitor (the stereo separation is exaggerated to demonstrate composed stereo effect in the figure).

4.5 Synchronous Event Generator and Handler

An avatar's behavior is based on a programmable state machine. Transitions between states are effected by events that trigger multiple actions synchronously such as speech generation or movement of mouth and limbs. Synchronization is therefore important because without it, the avatar's spoken words and actions will appear uncoordinated.

LRAF provides two different methods to implement this synchronous event trigger mechanism. One is the animation-driven event (ADE) trigger and the other is the speech-driven event (SDE) trigger. We can encode all necessary common LRAF events in those two types.

The ADE is defined as any common LRAF event that needs to synchronously trigger a long avatar animation playback. For instance, effect sound "A" plays for 2 s after starting motion "B". We can specify such an event in an animation specification file (external XML file). Each animation clip includes all those events and its timing information in addition to its own animation description. The following shows one example of such an event encoding. This example of animation (idle animation) has two event embedded with "ADE" tag. The first one triggers effect sound id "0" when animation reaches 0.5 s. The second one triggers the next action change event at 6.3 s.

```
<Anim id="0" name="idle" blend="1.0">
  <ADE type="AE_SOUND" time="0.5" param="0"/>
  <ADE type="AE_ACTION" time="6.3" param="CAT_IDLE"/>
</Anim>
```

The SDE has two difference types because LRAF supports two speech-generation methods, TTS synthesis and recorded voice. A TTS-driven event consists of a “bookmark” that can be embedded within real-time voice synthesis. When LRAF encounters a bookmark event during speech synthesis, it will immediately trigger the corresponding event. In the following example, the speech-synthesized utterance of “hey, look to your right” will trigger the avatar to point to its right.

“Hey, Look to your <bookmark mark="SE_ACTION"][POINT_RIGHT]"/> right.”

A recorded voice-driven event is similar to an animation-driven method. Within a definition for recorded voice file specification, we can encode generic LARF event with the “SDE” tag. The example XML code fragment below describes one instance of recorded speech definition and will trigger “POINT_RIGHT” action in 1.0 s once the speech wave file starts to play.

```
<FFT speech="Hey, Look to your right." file="lookright.wav"/>
  <SDE type="SE_ACTION" time="1.0" param="POINT_RIGHT"/>
</FFT>
```

5 The Intelligent Communication System

The LifeLike Avatar’s intelligent communication system is founded on the concept of context. The idea behind this is that communication among humans is closely tied to the perceived context of the communicants. Knowing and accepting a common context by all participants in a conversation permits them to dispense with the need to define all terms by making some reasonable assumptions about the topic of conversation. For example, the word “skiing” has vastly different connotations in winter than in summer – in the mountains of Colorado than in Miami Beach. Knowing the context in which the word “skiing” is used eliminates the need to further define it as alpine or water-skiing.

Context-based methods refer to the techniques to drive behavior based on the current context. Although it has been used successfully to direct the behavior of agents performing a task in an environment, we use it here to help the avatar understand the query it is being presented.

Resolving semantic ambiguity remains a classic problem in natural language processing (NLP). One particular research avenue involves reducing the word (or phrase) identification search space by incorporating clues from the ambient conversational contexts. Contextualization effectively adds an extra layer of knowledge to a reasoning system. Semantic analysis methods have been enhanced by

introducing contextual information into their training routines [43]. In general, NLP problems can be enhanced with contextually driven methods [52], such as those found in spoken language translation [34] and knowledge modeling [53].

Perhaps, the subdiscipline within NLP that has benefited the most from contextualization is the ASR community. Much work has been done in this area that deserves to be mentioned here; however, a detailed and exhaustive discussion of these remains outside the scope of this article, and the reader is referred to Hung [27]. Although most of the published work uses context as supplemental and often peripheral knowledge to disambiguate utterances, our work uses context as the basis of the communication process. We describe this further in the following subsections.

The intelligent communication system is itself subdivided into two major components: the speech recognizer and the dialogue manager. A third, albeit less important, component is the interrupt system, and it is also described.

The objective of the DM used in our LifeLike Avatar is to promote open dialogue, that is, manage an unscripted dialogue that could be initiated by either the avatar or the human. The direction that the dialogue is to take is assumed unknown a priori and the DM is designed to handle any enunciations by the human interlocutor as long as they are in the domain of interest. The speech recognizer provides the inputs to the DM, and we discuss that first.

5.1 The Speech Recognizer

The function of the speech recognizer is to listen and convert audio inputs from the human user through the microphone into text. The LifeLike speech recognizer uses the commercially available Chant [12] and Microsoft SAPI [42] systems. Figure 26 describes the speech recognizer within the overall LifeLike system. The video feed at the bottom comes from a web camera and is used to detect movement in the user's mouth and his/her gaze directed onto the avatar (or the microphone) to enable the user to interrupt the avatar in the middle of its talk. The interruption feature is described in Section 5.2. The output of the speech recognizer is passed to the DM via a network connection.

The original speech recognizer used in our work was grammar-based. It provided good performance for heavily scripted inputs, such as were the dialogues used in the early stages of our research. However, in our subsequent pursuit, to open the dialogue to initiative from the human user, as mentioned above, required dictation-based speech recognition. Both versions used the same commercially available tools mentioned above. Unfortunately, as we will see later, this also resulted in a significant decrease in proficiency in recognizing spoken

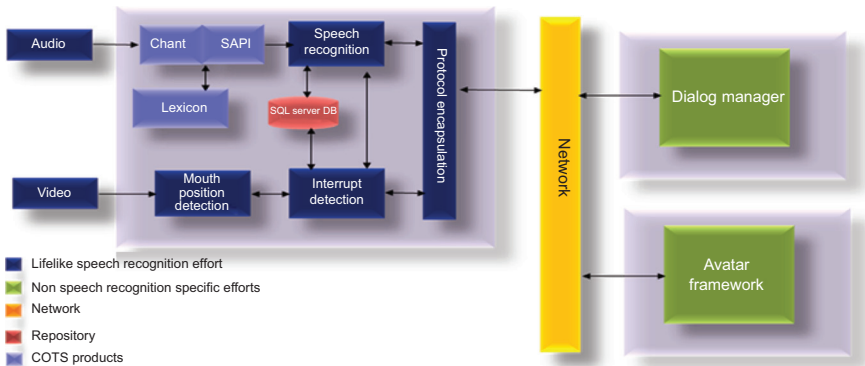


Figure 26. LifeLike Speech Recognition Architecture.

input and led to the DM being redesigned to better tolerate the high word error rates (WER) faced.

This latest version of the speech recognizer presented many challenges in all development areas. Furthermore, the new changes affected the areas where grammar was used as the principal lexicon for recognition. This was so that speech recognition can respond to the user's natural language based on SAPI dictation mode, where the capability to convert the audio input into the correct words becomes constrained by the Microsoft SAPI lexicon. The ability to perform user-independent speech recognition is diminished because of the uncertainty inherent in the larger lexicon search space and speech recognition engine while selecting similar word sounds.

In particular, this was addressed using the Speech Recognition Grammar Specification (SRGS), which is a W3C standard for how speech recognition grammars are specified [28]. A speech recognition grammar is a set of word patterns that are used primarily to indicate to the speech recognizer what to expect from the user; specifically, this includes words that may be spoken, patterns in which those words occur, and the spoken language surrounding each word. The syntax of the grammar format can be specified in two forms:

- ABNF (augmented Backus–Naur form) – this is a non-XML plain-text representation similar to traditional BNF grammar [28].
- XML (extensible markup language) – this syntax uses XML elements to represent the grammar constructs [28].

Both the ABNF and XML forms have the expressive power of a context-free grammar (CFG) and are specified to ensure that the two representations are

semantically mappable to each other. It is possible to convert from one form to the other and achieve identical semantic performance of the grammars.

As the research originally focuses on the use of the XML-based grammar format, it was influenced by the use of SAPI version 5 SRE. Thus, Microsoft SAPI 5 specifies a CFG structure and grammar rule format using XML. A grammar compiler transforms the XML grammar into a SAPI 5 binary format for the SAPI 5-compliant SRE. A SAPI 5 grammar text file is composed of XML grammar elements and attributes that express one or more rules, i.e., recognizable utterances. Speaker-independent audio input from a microphone headset is passed to the speech recognizer where it is processed by an SRE. There are two forms of recognized speech data. The first form uses the grammar XML file, where a context-specific rule is made active and the speech utterance is matched against the phrases in that rule. This narrow scope of words allows for a more precise match and better recognition results. If the recognizer did not find a match of high confidence within the grammar, the second form is used. In this form, the SR uses a generic noncustomized grammar-free lexicon against which the utterance is matched.

Because the grammar XML file allows for better recognition rates, our focus was on this option. Initially, a small prototype grammar file was built and tested with much success. However, as the knowledge base grew, building and testing the file manually proved futile and lacked ingenuity. The need for an autonomous audio regression testing system resulted. Many tests performed with the original LifeLike speech recognizer executing under the Windows XP operating system and SAPI 5.3 failed to provide a specific lexicon that supported the topics of interest to our application (NSF program management). The Windows 7 operating system integrated a different lexicon training method using documents as well as an improved SAPI engine. After providing the NSF-based ontology [46] as the basis for speech recognition training, the results were positive and encouraging. Recognition rates improved dramatically to a WER of 12.4% while using NSF-related sample text to test speech recognition. WER is defined as the addition of the word substitutions, deletions, and insertions divided by the number of words in the reference document [63]. The following equation [41] describes the relationship:

$$WER = \frac{S+D+I}{N},$$

where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions, and N is the number of words in the reference.

However, the out-of-vocabulary word recognition did not perform as well, even after the system was trained with common language responses. The overall

speech recognition decreased considerably, as noted in user testing, to somewhere near 50% (see the exact test results in Table 7, Section 6). As might be expected, the recognition sweet spot was NSF lexicon training with less common speech language. This combination proved to be suitable for the needs of the system but only after nontrivial compensation in the design of the DM.

Three users were given three sets of grammar phrases, G_1 , G_2 , and G_3 , from the LifeLike domain to conduct a series of recognition tests. The first set of phrases, G_1 , comprises 15 randomly chosen names from different universities that receive funding from NSF. The recognition rates using the users' natural voice was compared with the recognition rates when their recorded voice was used. The recognition observed with recorded voice was obtained using the regression testing abilities of ART to see how well the system could use a recorded voice sample to do speech-to-text. Table 1 shows the raw data collected from the three users for G_1 . A checkmark in the table indicates that the name was correctly recognized.

Table 2 shows the recognition data for 15 randomly chosen university names. Table 3 contains the data accumulated after the users were asked to test the system with the sets G_1 and G_2 as well as the acronyms of 15 different university names (set G_3).

Another improvement was the addition of different profiles for women and men because the ASR engine can recognize the different pitches that exist between the genders. The speech recognition favors speech input within average tones. Consequently, low-frequency tones in deep male voices and high-pitched

Table 1. Recognition Data for Directors' Names (G_1).

| Director name (G_1) | User 1 | User 2 | User 3 |
|-------------------------|--------|--------|--------|
| Betty Cheng | ✓ | | ✓ |
| Charles Petty | ✓ | ✓ | ✓ |
| David Goodman | ✓ | ✓ | ✓ |
| Frank Allen | ✓ | ✓ | ✓ |
| Jay Lee | ✓ | | ✓ |
| Shah Jahan | | ✓ | |
| Balakrishna Haridas | ✓ | ✓ | |
| Don Taylor | ✓ | | ✓ |
| Samuel Oren | | | ✓ |
| Ram Mohan | ✓ | ✓ | ✓ |
| Nikos Papanikolopoulos | | | |
| Richard Muller | ✓ | | ✓ |
| Rahmat Shoureshi | ✓ | ✓ | ✓ |
| Steven Liang | ✓ | | ✓ |
| Sami Rizkalla | ✓ | | |

Table 2. Recognition Data for University Names (G_2).

| University name (G_2) | User 1 | User 2 | User 3 |
|---------------------------------|--------|--------|--------|
| University of Central Florida | ✓ | | ✓ |
| University of Texas at Austin | ✓ | | ✓ |
| North Carolina State University | ✓ | ✓ | |
| Oregon State University | ✓ | | ✓ |
| Purdue University | | ✓ | ✓ |
| University of Utah | ✓ | | |
| Ohio State University | ✓ | ✓ | ✓ |
| Michigan State University | ✓ | | ✓ |
| Clemson University | ✓ | ✓ | ✓ |
| Iowa State University | ✓ | ✓ | ✓ |
| University of Maryland | ✓ | ✓ | ✓ |
| University of New Mexico | ✓ | | |
| George Washington University | ✓ | | ✓ |
| Carnegie Mellon University | ✓ | | ✓ |
| University of Houston | ✓ | ✓ | ✓ |

tones in women and children will not fare well with our speech recognizer unless a speech profile is created for each group and the system is trained accordingly. The solution may seem easy, but it is against our objective to make the system speaker independent. Furthermore, it would be rather complex and cumbersome because it would be necessary to automatically detect the interlocutor's gender before he or she speaks to enable the speech recognizer to switch to the appropriate speech profile and grammar.

5.2 The Interrupt System

One early complaint in our testing was that once the avatar began to enunciate its response, often a somewhat lengthy one, it was impossible to interrupt it if

Table 3. Recognition Rates for Three Different Grammar Sets.

| Grammar set | Recognition rate (%) | | |
|-------------|----------------------|--------|--------|
| | User 1 | User 2 | User 3 |
| G_1 | 80 | 46.7 | 73.3 |
| G_2 | 93.3 | 46.7 | 80 |
| G_3 | 100 | 86.7 | 80 |

the answer was out of context or otherwise irrelevant. Therefore, to add naturalness as well as efficiency to the dialogue, the ability of the human speaker to interrupt the LifeLike Avatar in the midst of its response was deemed necessary. The interrupt system developed for our LifeLike system consists of two parts. The first uses a microphone to detect any audio that might be spoken by the human user when attempting to interrupt the avatar. The second involves a web camera to visually determine whether the user is purposely trying to interrupt the avatar or whether the verbal inputs are the result of background noise or the user speaking to a third party. This visual input detects movements of the mouth of the human speaker. It does not, of course, read lips – it merely recognizes when the human interlocutor appears to be addressing the avatar, thereby making the interrupting words credible, as opposed to the speaker turning around to speak to someone else.

For the first part, a separate speech recognition engine becomes activated to only detect interrupts. This feature permits the recognizer engine to have its microphone activated permanently to detect interruptions. In the second part of the interrupt system, the web camera monitors the user while the avatar is speaking to the user because this is the only time an interruption is relevant. The user closest to the web camera is assumed to be the primary speaker. The user with the largest face in the image is assumed to be the one sitting closest to the camera and thereby the interlocutor. All others in the image are ignored to reduce the processing time and permit real-time operation. Once the primary interlocutor has been identified, the web camera subsystem processes the image to first determine whether the interlocutor is paying attention to the avatar by observing certain facial features. The most important of these features is the mouth. Using weak classifiers and a blob tracker, we can determine the location and state of the mouth. Mouth and face information is used to estimate the orientation of the user with respect to the camera. An assumption is made here that the camera is located near the avatar. If the interlocutor is paying attention to the avatar, the web camera subsystem then attempts to determine whether the user's mouth is open or closed. If the microphone subsystem detects a noise and the web camera subsystem detects the user's mouth as open, then it confirms that the user is trying to interrupt the avatar.

The detection stage of the interrupt is done in six steps.

1. The web camera is turned on. This is done with the EMGU libraries because they support C# manipulation of OpenCV functions [18]. OpenCV, an image-processing library created by Intel, provides the image classification functionality [7].
2. An image from the camera is processed. To save processing time, the image is reduced to 320×240 gray scale as shown in Figure 27A. This means the system need not waste processing time on a full-resolution color image.



Figure 27. Image of Human Interlocutor for Interruption Detection.

3. The blob tracker locates the closest human to the web camera. The use of the blob tracker limits the search space in the image, giving faster performance (see Figure 27B).
4. An attempt is made to find a face in the search space marked by the blob tracker. To detect the face, common face models are compared with every area of the search space. A face may not always be detected because a user may not be looking into the camera at all times (see Figure 27D).
5. If a face can be identified, the mouth is located using weak classifiers. These classifiers work with the basic knowledge that the mouth is in the lower part of the face, below the nose, and can be open, closed, or partially opened.
6. A determination is made as to whether the mouth is open or closed. This is done using a new set of weak classifiers. A final image can be seen in Figure 27B. For our purpose, there are only two states for the mouth (open and closed).

One problem with using a web camera to detect a face is the available lighting. An object that is poorly lit looks different from an object overexposed to light. To eliminate this problem, a threshold value was introduced. The threshold value determines how close a match must be made in order to classify if there is a face and the state of the mouth. In other words, the threshold value serves to normalize the differences in lighting and skin tone variations. Other problems can occur when there is a bright light in the background or when there is too bright a light in the foreground. When the threshold value is not properly set, this scenario can

cause the incorrect recognition of the face in the image, as seen in Figure 27E. If there is an object blocking the face or mouth, like a microphone, then the face models and/or classifiers have trouble locating the face and/or mouth, as in Figure 27C.

The interrupt recognizer was dramatically improved by adding features from the open-source library OpenCV [50] to include computer vision detection of mouth positioning to sense human distractions as part of the interrupt detection. The latest version of the interrupt recognizer not only has the capability to detect interruptions while the avatar is speaking, but it also can perceive when the user is distracted.

5.3 Dialogue Manager

The DM parses the incoming speech converted into text to understand its meaning and composes and controls the messages passed back to the interlocutor in response. It directs intelligent responses to the Avatar Framework for the TTS operations. The operation of the DM, called CONCUR (for CONTEXT-centric Corpus-based Utterance Robustness), was inspired by how humans communicate reasonably effectively when one of the communicants speaks the common language very poorly. Understanding is typically enabled by recognizing keywords in the conversation that might identify the context of the conversation. Unlike in ELIZA and ELIZA-based chatbots, CONCUR seeks to understand the context of the conversation by recognizing only a few words. This is approach taken in CONCUR to compensate for the poor WER observed in the dictation-only version of the speech recognition module. Once a context is recognized, the descriptions associated with that context contain the information requested by the human user and the avatar enunciates it via a TTS system.

LifeLike tries to capture an open-domain feel, with special regard to robustness. This latter requirement was the result of preliminary testing that revealed the need to provide a proper safety net for errant user inputs. To provide a more stable, open-domain conversation style, a greater grasp of context-based methods was emphasized for the second version of the prototype's DM development. By embracing context-based reasoning (CxBR) techniques [24], the DM can attain a more complex state-transition infrastructure, a benefit enjoyed by more traditional CxBR-based behavioral modeling efforts.

For the LifeLike system, two major DM components were created: the goal manager and the knowledge manager. Both components were developed under the tenets of the CxBR paradigm. The next sections describe each component in detail.

5.3.1 Goal Manager

Goal management in a dialogue system involves processes that recognize and satisfy an interlocutor's needs as conveyed by his or her utterances. Within any conversation, regardless of the presence of machine agents, there exists some sense of goal-oriented activity on the part of all participants. Often, these activities are characterized as some form of knowledge transfer, such as requesting or delivering information. Every participant contributes utterances, or speech acts, to drive the conversation toward purposefulness. In a two-party conversation, both sides go into the conversation with the intention of getting something out of the interaction. The participants begin talking to one another in an initial state, only to end in a different state, a goal state. This model of conversation assumes that its conclusion occurs when both participants are satisfied with how much they have achieved from the session. Hence, under normal conditions, the goals of both speakers are accomplished when their conversation ends.

Figure 28 depicts the general architecture of the goal management system. To provide a goal management system is to offer a general approach to creating the effect of a natural, open-dialogue interaction experience. Open dialogue refers to a loose set of conversational input constraints, allowing the agent to handle a wide range of user utterances [26]. Additionally, one or more user goals can exist at any time during an open-dialogue interaction. This contrasts with the closed, highly constrained, and unnatural multiple-choice style of input expectation found in automated airline booking agents and telephone-based credit card payment systems. Moreover, these types of interactions can only accommodate one user task at a time. The open-dialogue style allows for a more natural flow of language.

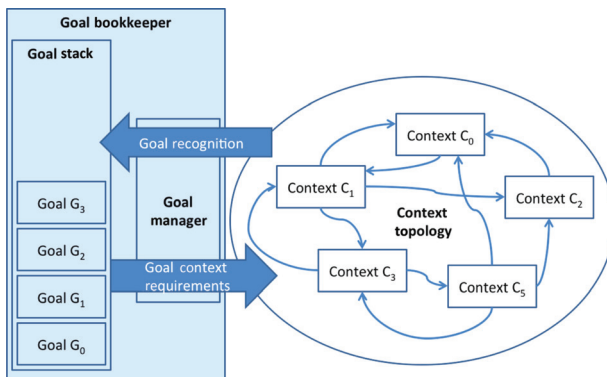


Figure 28. Goal Management Block Diagram.

To realistically accomplish the illusion of open dialogue through goal management, the following assumptions were made for the LifeLike Avatar dialogue system:

- The dialogue system is limited to an expert domain and the user is cognizant of the dialogue system's role as an expert entity. This constrains the user to a topical context with which the LifeLike Avatar is deeply familiar, without jeopardizing the open-dialogue style.
- The user's goals are limited to those related to the avatar's expertise. This assumption dictates that the user understands the agent's limitations as a domain-specific entity.

Goal management in the LifeLike Avatar DM involves three parts: (1) goal recognition, (2) goal bookkeeping, and (3) context topology. Goal recognition refers to the process of analyzing user utterances to determine the proper conversational goal to be addressed. This is analogous to the context activation process in CxBR methods, where production rules determine the active context according to the state of the agent and of the environment. The difference with the goal recognizer, however, is that the latter identifies the proper context to activate using keywords and phrases that are extracted from a parts-of-speech parsing of input responses. Armed with the knowledge manager, the user utterance is interpreted and the context associated with this understanding is activated.

Goal bookkeeping is the process of servicing every identified goal in the order that it is presented. Immediately after recognizing a goal, it is placed in a goal bookkeeping stack, a similar structure to that of the discourse stack model [8]. In the LifeLike Avatar, complex interruptions may occur, including switching to entirely different contexts. Thus, the goal stack was especially designed to handle conversation paths that experience drastic shifts between context changes. Furthermore, a transitional speech act must be executed to smooth over these context shifts.

Context topology refers to the entire set of spoken behaviors of the chatting avatar. This structure also includes the transitional actions when moving between contexts when a goal shift is detected. The context topology carries out the responses needed to clear out the goal bookkeeping stack. Upon receiving the activated goal to be addressed from the goal stack, the context topology operates on this signal to provide the proper agent response. Each context within the context topology corresponds to a certain conversational task, whether user motivated (external) or agent motivated (internal). Most of these conversational tasks adhere to a specific user task goal. These are known as user goal-centered contexts. The remaining conversational tasks constitute the agent goal-driven contexts. The inclusion of all user goal-centered contexts and agent goal-driven contexts constitutes the entire LifeLike Avatar context topology.

5.3.2 Knowledge Manager

In conjunction with LifeLike’s goal manager, a strong knowledge base must be in place for proper goal recognition. The information in this knowledge base is analogous to the rote knowledge that a human learns and manipulates to make decisions. For this project, three knowledge models were built: domain-specific knowledge, conversational knowledge, and user-profile knowledge (see Figure 29).

5.3.2.1 Domain-Specific Knowledge

The scope and depth of domain-specific knowledge were modeled after a traditional expert system, where a domain specialist meticulously adds information to a machine by hand. The relevant domain of the LifeLike Avatar is information about the NSF I/UCRC program.

In most expert systems, knowledge exists as a set of if–then statements to make decisions [23]. For the LifeLike Avatar, however, the domain-specific knowledge was organized in a linear, encyclopedia-style corpus. A knowledge parser was developed to fetch these data and organize them in a context-layered fashion. For each piece of knowledge, the parser also generates a key phrase list to assist in the context identification process needed for the context topology infrastructure. This key phrase generation employs the heavy use of an NLP-based toolkit.

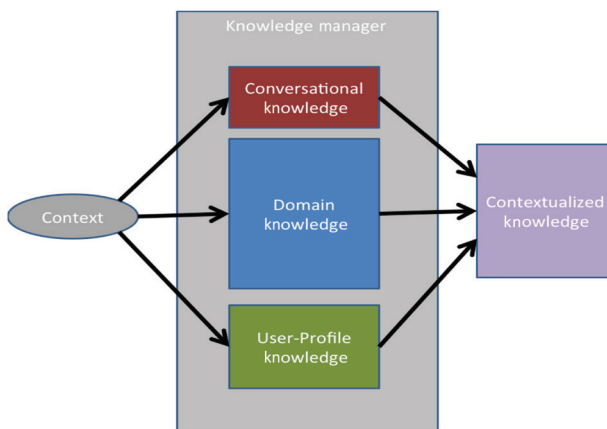


Figure 29. Knowledge Manager Block Diagram.

5.3.2.2 Conversational Knowledge

Alongside expert knowledge, conversational knowledge is needed to deploy the unique behaviors of the LifeLike Avatar. The conversational knowledge database only relates to the transitional speech actions to be interspersed among the domain knowledge deployments, such as sentences like “hi,” “keep the peace,” or “what else would you like to know?”

5.3.2.3 User-Profile Knowledge

MacWhinney et al. [37] describe the importance of memory during a conversation. They claim that memory structure is key when dealing with natural dialogues, as it provides an extra layer of interactive immersion. In addition to domain-specific and conversational knowledge, the user-profile knowledge database addresses the concept of memory, albeit in a limited fashion. All that the agent knows about the human with whom it is communicating exists in this body of information. Once the user has identified himself or herself, the knowledge manager can immediately retrieve his or her individual profile. This is particularly important in providing an HCI experience that escalates the level of realism and conveys an effect of personalization.

5.3.2.4 Contextualized Knowledge

Contextualized knowledge refers to a cross section of all three knowledge sources that is relevant for the active context of the conversation. Each piece of information within the knowledge manager is annotated with a context tag. Once the dialogue system determines the context of the conversation, knowledge that is labeled with the current context is elicited as valid information for the conversation and funneled into the contextualized knowledge database. Once this information is established, the DM can then work with a manageable portion of the entire knowledge base. This is especially useful when performing goal management, which may require memory-intensive processes.

The concept of contextualized knowledge is a novel feature of the LifeLike Avatar DM. The idea that only a portion of an agent’s entire knowledge is needed at any given time reflects how a human makes decisions, that is, he or she relies on a small subset of its knowledge base to make the decision. The contextualization of the knowledge base makes identifying this appropriate subset of knowledge feasible and practical based on matching the context being experienced by the agent with the appropriate contextualized portion of the knowledge base. A CxBR-based architecture lends itself to this concept because the determination of an active context, and therefore an active set of contextualized knowledge, is a built-in function of CxBR.

(01/14/10 18:36:56) *Avatar to User*: I'm Alex Schwarzkopf. What's your name?

(01/14/10 18:37:02) *User to Avatar*: harm run the legal (HI I'M RODNEY KWICKO)

(01/14/10 18:37:05) *Avatar to User*: It's good to see you. I don't think we've met. I am the Project LifeLike Avatar, a 3-year collaborative effort sponsored by the NSF between the University of Central Florida and the University of Illinois at Chicago. What do you want to talk about?

(01/14/10 18:37:28) *User to Avatar*: of a letter of interest

(01/14/10 18:37:28) *Avatar to User*: A letter of interest is one in which company states it is interested in joining the center if it is created. This is not to be confused with a letter of commitment or a letter of support. Do you have questions about that information?

(01/14/10 18:37:47) *User to Avatar*: no

(01/14/10 18:37:47) *Avatar to User*: How else can I help you out?

(01/14/10 18:37:53) *User to Avatar*: know what about the marketing plan (UM, WHAT ABOUT THE MARKETING PLAN)

(01/14/10 18:37:53) *Avatar to User*: The marketing plan explains how the center shall make itself attractive to potential industrial members. See the current solicitation for guidelines to preparing the marketing plan. Do you have questions about that information?

(01/14/10 18:38:11) *User to Avatar*: no

(01/14/10 18:38:11) *Avatar to User*: I guess we're done now. Keep the peace.

The dialogue displayed above is an example of how the LifeLike Avatar DM (CONCUR) interacts with a human user. The capitalized letters following some of the user responses are the actual utterances that were transcribed from a voice recording, whereas the lowercased text represents the actual ASR output. This particular example lasted just over a minute. It suffered a WER of 29%, yet scored 100% in both conversational accuracy and goal completion accuracy. Although this particular dialogue uses keywords to define the context, the DM system can use more complex inputs than simply keywords to identify the context.

5.4 Summary of Intelligent Dialogue

The large WERs encountered were a significant problem in this project that required innovative means to overcome and still provide a useful and satisfying conversation to the human user. The approach taken was to mimic the situation when a person who is not fluent in a language attempts to communicate with a native speaker. Generally, the conversation involves short, badly pronounced words from which the native speaker must extract the intended context. Once this context is discovered

(correctly), then he or she can more easily respond to the request. The LifeLike Avatar compensates for large WERs by contextualizing the conversation. This means that the context of the conversation, once identified, is matched to a part of the conversational knowledge base that is relevant to this conversation, thereby reducing the effort in finding and providing an appropriate response to the user.

6 Evaluation

A prototype of the LifeLike Avatar that encompasses the above components was built. The resulting prototype was evaluated quantitatively and qualitatively with human test subjects. This section summarizes this extensive set of tests and reaches a conclusion about the viability of our research efforts. Full details can be found in Hung [27]. Note that we only discuss the evaluation of the Avatar Intelligent Communication System.

6.1 Assessment of the Avatar Intelligent Communication System

We sought to collect data supporting the hypothesis that the presented LifeLike Avatar with the CONCUR DM provides an HCI experience that is natural and useful.

In Experiment 1, a fully animated, speech-based lifelike avatar used a search-based system as its backend knowledge to answer questions about the NSF I/UCRC program domain. The search-based backend for this avatar was the result of a prior NSF-sponsored research called AskAlex, which approximated a traditional expert system-based question-and-answer (Q/A) system, except it did not use productions to represent its knowledge, but rather, contextual graphs (CxG) [9]. The data for this experiment were collected at the 2009 Annual I/UCRC Director's Meeting as well as at the Intelligent Systems lab at the University of Central Florida (UCF) in 2010. Manual transcription of all the 20 speech recordings were performed, whereas 30 user surveys were collected.

Experiment 2 assessed the ultimate version of our LifeLike Avatar and the specific combination of technologies developed as part of this project. This was the ultimate objective of our project. It is the performance of this version that we sought to validate by comparing it with other versions reflected in Experiments 1, 3, and 4. In Experiment 2, the fully animated, speech-based LifeLike Avatar was combined with the CONCUR DM and the NSF I/UCRC knowledge base to obtain

the data for Experiment 2. The collection of these data took place at the 2010 Annual I/UCRC Director’s Meeting as well as at the Intelligent Systems Laboratory (ISL) at the University of Central Florida in 2010. This use of dual testing sites allowed for a wider distribution of user demographics. Thirty data points were collected. Additionally, 30 user surveys were included in this set.

Experiment 3 used a text-based, disembodied chatbot in lieu of the LifeLike Avatar used in Experiment 2. In this experiment, text inputted from the keyboard was used to interact with the chatbot using CONCUR. The objective was to eliminate the ASR WER by inputting text from the keyboard and thereby gauging the effect of the high WER on performance. A Google Chat chatbot was developed using the JABBER middleware. The transcribed responses from Experiment 1 were fed into the chatbot of Experiment 3. The resulting 30 responses from this CONCUR-based chatbot were then recorded.

Experiment 4 used this same text-based CONCUR Chatbot interaction model as in Experiment 3 but instead coupled with a “current events” knowledge domain, rather than the NSF I/UCRC. This was done to determine how easily the domain knowledge could be replaced in the CONCUR DM while still providing appropriate performance. This new domain was constructed from various news articles pertaining to the United States, world affairs, sports, science, and health. Collecting data points for this set was conducted online using a set of 20 Google Chat participants, who ranged in levels of education and types of profession. Twenty user surveys were obtained in this experiment.

Table 4 summarizes the data sets resulting from the four experiments described. The evaluation process featured in this work is derived from the PARADIGM for Dialogue System Evaluation (PARADISE) [59]. A multimodal version of this system exists in PROMISE [4], but our work references PARADISE for simplicity’s sake. Sanders and Scholtz [55] affirm that ECA and chatbot goals for interaction are essentially the same.

Table 4. Dialogue System Data Set Collection Setup.

| | Experiment | | | |
|------------------------|-----------------|-----------------|------------|----------------|
| | 1 | 2 | 3 | 4 |
| Dialogue system | AskAlex | CONCUR | CONCUR | CONCUR |
| Interface type | LifeLike Avatar | LifeLike Avatar | Chatbot | Chatbot |
| Input method | Speech | Speech | Text | Text |
| Speech action engine | Search-driven | CxBR | CxBR | CxBR |
| Domain corpus | NSF I/UCRC | NSF I/UCRC | NSF I/UCRC | Current Events |
| Number of trials | 20 | 30 | 30 | 20 |
| User surveys collected | 30 | 30 | n/a | 20 |

6.2 Evaluation Metrics

The primary objective of this work was to provide a balance between dialogue performance (naturalness) and task success (usefulness) during a human–computer interaction. To determine usefulness, two types of metrics were used, efficiency and quality metrics. Efficiency metrics pertain to those interaction traits that can be empirically observed with no need for qualitative interjection. For the most part, the prototype software internally monitors these metrics. The ASR-related metric, WER, was measured by comparing the textual chat log from the agent with an audio recording transcript of the exchange. Quality metrics use both quantitative analysis and survey-based data and include metrics such as total number of out-of-corpus misunderstandings, total number of general misunderstandings, total number of inappropriate responses, total number of user goals, total number of user goals fulfilled, out-of-corpus misunderstanding rate, general misunderstanding rate, error rate (percentage of system turns that resulted in inappropriate response), awkwardness rate (percentage of system turns that resulted in general misunderstanding or inappropriate response), goal completion accuracy, and conversational accuracy (percentage of non-awkward responses from avatar).

To measure naturalness as well as further evaluate usefulness, each test subject was given an exit survey at the conclusion of his/her interaction with the avatar. This questionnaire directly addressed the remaining quality metrics that are impossible to assess without the user's personal input. The following list describes survey statements, which are answered using a Likert scale response system with a range of 1–7.

- Naturalness
 - Statement 1: If I told someone the character in this tool was real, they would believe me.
 - Statement 2: The character on the screen seemed smart.
 - Statement 3: I felt like I was having a conversation with a real person.
 - Statement 4: This did not feel like a real interaction with another person.
- Usefulness
 - Statement 5: I would be more productive if I had this system in my place of work.
 - Statement 6: The tool provided me with the information I was looking for.
 - Statement 7: I found this to be a useful way to get information.
 - Statement 8: This tool made it harder to get information than talking to a person or using a website.
 - Statement 9: This does not seem like a reliable way to retrieve information from a database.

Because of the way they are worded, Statements 4, 8, and 9 are negatively presented. This means that a score of 7 is the worst score that can be assigned. Hence, when aggregating the results, the assessments from these survey statements must be translated in a positive manner, such that the scores are reversed – a score of 1 becomes a score of 7, 2 becomes 6, 3 becomes 5, and 4 remains the same.

6.3 Summary of Results

This section provides a summary of results for all the experiments in the groups of metrics described above. We begin with the naturalness and usefulness surveys.

6.3.1 Summary of Naturalness and Usefulness

Table 5 displays the aggregate survey results for the survey parts of Experiments 1, 2, and 4 (there was no questionnaire in Experiment 3). Each column is labeled with an individual survey statement. Table 6 depicts the naturalness and usefulness results from averaging the normalized results from Statements 1–4 for naturalness and Statements 5 to 9 for usefulness.

From Table 6, we can conclude from the results of Experiments 1 and 2 that the LifeLike Avatars both obtained slightly positive responses from users in both naturalness and usefulness. A fairly poor rating of naturalness was given to the

Table 5. Normalized Survey Results.

| Experiment | Statement | | | | | | | | |
|-------------------|-----------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1. AskAlex Avatar | 3.20 | 4.10 | 4.73 | 4.10 | 4.57 | 5.07 | 3.77 | 4.83 | 4.03 |
| 2. CONCUR Avatar | 4.07 | 4.00 | 4.97 | 3.83 | 4.90 | 5.43 | 3.67 | 4.57 | 3.70 |
| 4. CONCUR Chatbot | 2.20 | 2.45 | 3.00 | 2.35 | 4.10 | 3.70 | 3.20 | 3.45 | 2.05 |

Table 6. Survey Results for Naturalness and Usefulness.

| Experiment | Naturalness | Usefulness |
|------------------------------|-------------|------------|
| 1. AskAlex LifeLike Avatar | 4.02 | 4.47 |
| 2. CONCUR LifeLike Avatar | 4.14 | 4.51 |
| 4. CONCUR text-based Chatbot | 2.40 | 3.38 |

lone text-based ECA from Experiment 4 while also achieving a slightly negative assessment of its usefulness.

6.3.2 Efficiency Metrics Results

Table 7 shows the aggregate efficiency metrics collected from the four experiments. These metrics deal with the measurable, nonqualitative results recorded by each agent.

The WER results report how well the ASR performed for each agent. Note that Experiments 3 and 4 did not use speech-based input; thus, they yielded perfect recognition accuracy (WER=0). These data reveal that each agent conversation was relatively similar in total elapsed times, ranging from nearly 3 min to just over 4 min. The AskAlex agent in Experiment 1 resulted in a slightly higher average turn count for both the user and the agent over the rest of the field. This is most likely caused by the scripted discourse manner in AskAlex that forces users to completely exhaust a particular topic path to its end. The text-based CONCUR DM of Experiment 4 saw a longer amount of time between turns. The text-based nature of this data set probably contributed to the lack of urgency by the user to respond between system responses. Both speech-based agents in Experiments 1 and 2 were virtually equal in recognizing user utterances at a 60% WER. This levels the playing field for any ASR-related metric comparison, as the agents from both experiments suffer from virtually identical WER.

6.3.3 Quantitative Analysis Metrics Results

Table 8 displays the aggregate results of the quantitative analysis of the quality metrics. In these metrics, each chat transcript was manually inspected for misun-

Table 7. Efficiency Metrics.

| Experiment | Total elapsed time (min:s) | Number of user turns | Number of system turns | Elapsed time per turn (s) | User words per turn | Agent words per turn | WER (%) |
|------------|----------------------------|----------------------|------------------------|---------------------------|---------------------|----------------------|---------|
| 1 | 3:36 | 13.4 | 14.4 | 4.2 | 2.8 | 28.6 | 60.9 |
| 2 | 3:20 | 10.9 | 11.9 | 6.1 | 4.9 | 29.1 | 58.5 |
| 3 | 2:52 | 10.1 | 11.1 | 6.1 | 5.0 | 28.2 | 0 |
| 4 | 4:03 | 8.9 | 9.9 | 9.4 | 4.2 | 35.8 | 0 |

Table 8. Quantitative Analysis of Quality Metrics.

| Experiment | Out-of-corpus misunder- standing rate (%) | General misunder- standing rate (%) | Misunder- standing rate (%) | Error rate (%) | Awkward- ness rate (%) | Goal completion accuracy (%) | Conver- sational accuracy (%) |
|------------|--|--|-----------------------------------|----------------------|------------------------------|---------------------------------------|--|
| 1 | 0.29 | 9.51 | 9.80 | 8.71 | 18.22 | 63.29 | 81.78 |
| 2 | 6.15 | 14.49 | 20.64 | 21.81 | 35.78 | 60.48 | 63.93 |
| 3 | 6.77 | 7.48 | 14.25 | 16.68 | 24.66 | 68.48 | 75.34 |
| 4 | 17.45 | 0.00 | 17.45 | 16.46 | 16.46 | 48.08 | 83.54 |

derstandings, erroneous agent responses, and context goal satisfaction. The final two columns, goal completion accuracy and conversational accuracy, provide indication of each agent's usefulness and naturalness, respectively.

Table 8 recounts how well each agent can handle the user input in terms of minimal conversational awkwardness and maximized assistive utility. The out-of-corpus misunderstanding rate assesses the percentage of time the agent must spend to react to a user requesting information that cannot be found in its knowledge base. In these results, it is shown that Experiment 4's chatbot experienced a substantial number of out-of-corpus misunderstandings, whereas the AskAlex agent in Experiment 1 saw very little. The explanation of this phenomenon is the simple fact that AskAlex's highly constrained input expectations from its menu-driven discourse serves as a preventative measure for out-of-corpus information requests. The CONCUR agent, meanwhile, maintains a higher amount of input flexibility, causing users to ask more questions that could potentially be out of the knowledge domain.

The general misunderstanding rate addresses the percentage of turns in which the agent is presented with situations that it could not handle, most often because of garbled ASR inputs or erratic user speech, such as stalling. The conversation agent in Experiment 4 did not have to deal with these issues, hence its lack of general misunderstandings. The CONCUR Chatbot of Experiment 3 also lacked ASR-related errors, but it still fell victim to user input errors because of its use of Experiment 2 inputs.

Error rate describes the percentage of turns where the agent returns a non-sensical response. The CONCUR agents all had similar error rates, whereas the AskAlex agent of Experiment 1 was the least error prone because of its menu-driven nature. The dialogue openness of the CONCUR system plays a part in causing erroneous system responses because the presence of specific Q/A information requests. This factor deals with the idea that users want very specific answers to questions, and it is discussed in further depth later.

In terms of usefulness, the goal completion accuracy metric indicates how effective an agent can service users' information requests. Although all of the NSF I/UCRC corpus-based agents (Experiments 1, 2, and 3) were able to complete over 60% of their users' goals, the current events CONCUR Chatbot in Experiment 4 was just under 50% for goal completion accuracy.

Awkwardness rate and conversational accuracy give a quantitative indication on the naturalness of the agent's dialogue. Essentially, conversational accuracy tells what percentage of the time the conversation agent gave an answer that can be perceived as natural. The awkwardness rate is simply the percentage of unnatural responses. Although each agent was able to demonstrate better than 60% conversational accuracy, the CONCUR ECA in Experiment 2 was far less conversationally accurate than the agents in Experiments 1, 2, and 4.

7 Summary and Conclusions

In summary, we have presented a new avatar that appears lifelike visually as well as audibly. The avatar system, called LifeLike, shows a strong resemblance to a specific human being and can be used to communicate with humans via spoken language. Our work represents a small but definitive step in our goal of developing a virtual human that can pass the enhanced Turing test – one that can fool a human into thinking he or she is speaking with the actual person via computer-based communication rather than a virtual representation of the person. We are still clearly far from reaching that goal, however. Nevertheless, there are many current applications for this type of interface; these include education, training, health-care support, and legacy creation.

The communication of our LifeLike Avatar takes place on an open-dialogue basis, where either the avatar or the human interlocutor can seize the initiative. The knowledge of the avatar in the domain of conversation is comparatively easy to create and set up. The avatar is also intolerant (to a degree) of high word recognition errors introduced by the ASR system.

We have tested the intelligent communication aspect of the LifeLike Avatar and find that it generally succeeds in its objective. We do not report tests of the graphical aspects of the avatar but leave it to the reader to infer our success or failure by their inspection of the quality of the graphics we include in this article.

Received March 21, 2013; previously published online May 27, 2013.

Bibliography

- [1] R. Artstein, S. Gandhe, J. Gerten, A. Leuski and D. Traum, Semi-formal evaluation of conversational characters, in: *Languages: From Formal to Natural: Essays Dedicated to Nissim Francez on the Occasion of His 65th Birthday*, pp. 22–35, Springer-Verlag, Berlin, 2009.
- [2] D. Barberi, *The Ultimate Turing Test*. <http://david.barberi.com/papers/ultimate.turing.test/>.
- [3] Beowulf [movie], Paramount (accessed 24 September, 2012). <http://www.paramount.com/movies/beowulf>.
- [4] N. Beringer, U. Kartal, K. Louka, F. Schiel and U. Turk, PROMISE – a procedure for multimodal interactive system evaluation, in: *Proceedings of the LREC Workshop on Multimodal Resources and Multimodal Systems Evaluation*, 2002.
- [5] T. W. Bickmore and R. W. Picard, Towards caring machines, in: *Proceedings of the Conference on Computer Human Interaction*, Vienna, April 2004.
- [6] Blender, *Blender Open Source 3D Contents Creation Suite* (accessed 24 September, 2012). <http://www.blender.org>.
- [7] G. Bradsk, *Open Computer Vision Library* (2009). <http://opencv.willowgarage.com/wiki/>.
- [8] K. Branting, J. Lester and B. Mott, Dialogue management for conversational case-based reasoning, in: *Proceedings of the Seventh European Conference on Case-Based Reasoning*, pp. 77–90, 2004.
- [9] P. Brézillon, Context in problem solving: a survey, *Knowledge Eng. Rev.* **14** (1999), 1–34.
- [10] R. Carpenter and J. Freeman, *Computing Machinery and the Individual: The Personal Turing Test*, Technical report, Jabberwacky (2005). <http://www.jabberwacky.com>.
- [11] J. Cassell, M. Ananny, A. Basu, T. Bickmore, P. Chong, D. Mellis, K. Ryokai, J. Smith, H. Vilhjálmsson and H. Yan, Shared reality: physical collaboration with a virtual peer, in: *Proceedings of CHI*, 2000.
- [12] Chant Inc., *Chant Software Home* (accessed 22 December, 2009). <http://www.chant.net/default.aspx?doc=sitemap.htm>.
- [13] K. M. Colby, *Artificial Paranoia*, Pergamon Press, New York, 1975.
- [14] L. Dutreuve, A. Meyer and S. Bouakaz, Easy acquisition and real-time animation of facial wrinkles, *Comput. Anim. Virtual Worlds* **22** (2011), 169–176.
- [15] C. Donner and H. Jensen, Light diffusion in multi-layered translucent materials, in: *ACM SIGGRAPH Papers*, 2005.
- [16] P. Ekman, Universals and cultural differences in facial expressions of emotion, in: J. Cole (ed.), *Nebraska Symposium on Motivation*, vol. 19, pp. 207–282, University of Nebraska Press, 1972.
- [17] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press, Palo Alto, CA, 1978.
- [18] C. V. Emgu (2009). <http://sourceforge.net/projects/emgucv/>.
- [19] E. d’Eon and D. Luebke. Advanced techniques for realistic real-time skin rendering, in: *GPU Gems 3*. H. Nguyen (Ed.) Chapter 14, Addison Wesley, Reading, 2007.
- [20] FaceGen, *FaceGen Modeller: 3D Face Generator* (accessed 24 September, 2012). <http://www.facegen.com/modeller.htm>.
- [21] FBX, *Autodesk FBX: 3D Data Interchange Technology* (accessed 24 September, 2012). <http://usa.autodesk.com/fbx>.
- [22] L. Foner, *What’s an Agent, Anyway? A Sociological Case Study*, Agents Memo 93–01, Agents Group. MIT Media Lab, 1993.

- [23] A. J. Gonzalez and D. D. Dankel, *The Engineering of Knowledge-Based Systems Theory and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [24] A. Gonzalez, B. Stensrud and G. Barrett, Formalizing context-based reasoning: a modeling paradigm for representing tactical human behavior, *Int. J. Intell. Syst.* **23** (2008), 822–847.
- [25] G. Güzeldere and S. Franchi, Dialogues with colorful personalities of early AI, *Stanford Hum. Rev.* **4** (1995), 161–169.
- [26] S. Harabagiu, M. Pasca and S. Maiorano, Experiments with open-domain textual question answering, in: *Proceedings of the COLING-2000*, 2000.
- [27] V. C. Hung, Robust dialog management through a context-centric architecture, Doctoral dissertation, Department of Electrical Engineering and Computer Science, University of Central Florida, August 2010.
- [28] A. Hunt and S. McGlashan (eds.), *Speech Recognition Grammar Specification Version 1.0: W3C Recommendation 16 March 2004* (accessed 28 February, 2012). <http://www.w3.org/TR/speech-grammar/>.
- [29] H. Jensen, *Subsurface Scattering* (2005). <http://graphics.ucsd.edu/~henrik/images/subsurf.html>.
- [30] T. Kanade, J. F. Cohn and Y. Tian, Comprehensive database for facial expression analysis, in: *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–53, 2000.
- [31] T. Kaneko, T. Takahei, M. Inami, N. Kawakami, Y. Yanagida, T. Maeda and S. Tachi, Detailed shape representation with parallax mapping, in: *Proceedings of ICAT 2001*, pp. 205–208, 2001.
- [32] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsela and D. Piepol, Building interactive virtual humans for training environments, in: *ITSEC'07*, 2007.
- [33] C. Lee, C. Sidner and C. Kidd, Engagement during dialogues with robots, in: *AAAI 2005 Spring Symposia*, 2005.
- [34] L. Levin, O. Glickman, Y. Qu, D. Gates, A. Lavie, C. P. Rose, C. Van Ess-Dykema and A. Waibel, Using context in machine translation of spoken language, in: *Proceedings of Theoretical and Methodological Issues in Machine Translation (TMI-95)*, 1995.
- [35] K. Lucas, G. Michael and P. Frederic, Motion graphs, in: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 2002.
- [36] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression, in: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 94–101, 2010.
- [37] B. MacWhinney, J. M. Keenan and P. Reinke, The role of arousal in memory for conversation *Memory Cogn.* **10** (1982), 308–317.
- [38] M. L. Mauldin, ChatterBots, TinyMuds, and the Turing test: entering the Loebner Prize competition, in: *Proceedings of the 12th National Conference on Artificial Intelligence*, vol. 1, 1994.
- [39] M. L. Mauldin, Going under cover: passing as human, in: *Parsing the Turing Test – Philosophical and Methodological Issues in the Quest for the Thinking Computer*, R. Epstein, G. Roberts and G. Beber (Eds.), pp. 413–429, Springer, 2008.
- [40] Maya, *Autodesk Maya: 3D Modeling and Animation Software* (accessed 24 September, 2012). Autodesk: <http://usa.autodesk.com/maya/>.
- [41] I. McCowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner and H. Bourlard, *On the Use of Information Retrieval Measures for Speech Recognition Evaluation*, Research Report, INDIAP Research Institute, 2005.

- [42] Microsoft MSDN, *Microsoft Speech API (SAPI) 5.3* (accessed 22 December, 2009). [http://msdn.microsoft.com/en-us/library/ms723627\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms723627(VS.85).aspx).
- [43] R. J. Mooney, Learning language from perceptual context: a challenge problem for AI, in: *Proceedings of the 2006 AAAI Fellows Symposium*, 2006.
- [44] M. Mori, The Uncanny Valley, in K. F. MacDorman and T. Minato (trans.), *Energy*, 7 (1970), 33–35 [in Japanese].
- [45] MotionBuilder, *Autodesk MotionBuilder: 3D Character Animation for Virtual Production* (accessed 24 September, 2012). <http://usa.autodesk.com/adsk/servlet/pc/index?id=13581855&siteID=123112>.
- [46] National Science Foundation, Industry & University Cooperative Research Program (I/UCRC) (2008) (accessed 23 November, 2009). <http://www.nsf.gov/eng/iip/iucr/>.
- [47] *Normal Map Filter, NVIDIA Texture Tools for Adobe Photoshop* (2011). <http://developer.nvidia.com/content/nvidia-texture-tools-adobe-photoshop>.
- [48] C. Oat, Animated wrinkle maps, in: *ACM SIGGRAPH 2007 Courses*, New York, NY, pp. 33–37, 2007.
- [49] *Oblivion, The Elder Scrolls: Oblivion* (accessed 24 September, 2012). <http://www.elderscrolls.com/oblivion>.
- [50] OpenCV, *OpenCV 2.0 C Reference* (accessed 22 December, 2009). <http://opencv.willowgarage.com/documentation/index.html>.
- [51] Ogre3D, *Object-Oriented Graphics Rendering Engine (OGRE)* (accessed 24 September, 2012). OGRE: <http://www.ogre3d.org>.
- [52] R. Porzel and M. Strube, Towards context-dependent natural language processing in computational linguistics for the new millennium: divergence or synergy, in: *Proceedings of the International Symposium*, pp. 21–22, 2002.
- [53] R. Porzel, H. Zorn, B. Loos and R. Malaka, Towards a separation of pragmatic knowledge and contextual information, in: *ECAI-06 Workshop on Contexts and Ontologies*, 2006.
- [54] A. Safonova and J. K. Hodgins, Construction and optimal search of interpolated motion graphs, in: *ACM SIGGRAPH 2007 Papers*, 2007.
- [55] G. Sanders and J. Scholtz, Measurement and evaluation of embodied conversation agents, *Embodied Conversational Agents* (2000), 346–373.
- [56] D. R. Traum, A. Roque, A. Leuski, P. Georgiou, J. Gerten and B. Martinovski, Hassan: a virtual human for tactical questioning, in: *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, 2007, 2008.
- [57] A. M. Turing, Computing machinery and intelligence, *Mind* 59 (1950), 433–460.
- [58] Vicon, *Vicon: FX Motion Capture Camera* (accessed 24 September, 2012). <http://www.vicon.com/company/releases/041007.htm>.
- [59] M. A. Walker, D. J. Litman, C. A. Kamm and A. Abella, PARADISE: a framework for evaluating spoken dialogue agents, in: *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, pp. 271–280, 1997.
- [60] R. S. Wallace, The anatomy of A.L.I.C.E., in: *Parsing the Turing Test*, pp. 181–210, Springer, Netherlands, 2008.
- [61] J. Weizenbaum, ELIZA – a computer program for the study of natural language communication between man and machine, *Commun. ACM*, 9 (1966), 36–45.
- [62] D. William and L. Andrew, Variance shadow maps, in: *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, 2006.
- [63] K. A. Zechner, Minimizing word error rate in textual summaries of spoken language, in: *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pp. 186–193, Morgan Kaufmann Publishers, Seattle, WA, 2000.