Internet Communication Using Real-Time Facial Expression Analysis and Synthesis

Naiwala P. Chandrasiri, Takeshi Naemura, Mitsuru Ishizuka, and Hiroshi Harashima
The University of Tokyo

István Barakonyi
Vienna University of Technology

Because popular technologies such as chat don't capture people's expressions or intonations, they fail to provide crucial emotional content. By modeling users' real-time facial expressions using a 3D agent, the proposed system adds emotional content and maintains user privacy.

How can technology support and invent new ways for geographically separate people to communicate and share experiences? The telephone is one of the most primeval methods, and is still widely used. As the Internet has evolved, however, novel communication methods have emerged. Among them are video telephony and videoconferencing technologies, including systems such as FreeWalk,[1] which supports small, casual group meetings in a virtual space with real-time video and audio of the participants. Far more popular, however, are real-time chat applications such as ICQ and MSN messenger, which let people exchange text- or voice-based messages.

Among the reasons that chat applications are more popular than real-time video applications is that many people are reluctant to show their faces to their communication partners, particularly in real time. People also tend to communicate more freely when they can hide their identity in informal communications. However, as the use of emoticons suggests, communication without nonverbal information such as facial expressions can be monotonous.

To address this, we have developed a system[2] that animates 3D facial agents based on real-time facial expression analysis techniques[3] and research on synthesizing facial expressions and text-to-speech capabilities.[4] (The "Related Work" sidebar discusses this research in more detail.) Our system combines visual, auditory, and primary interfaces to communicate one coherent multimodal chat experience. Users can represent themselves using agents they select from a group that we have predefined. When a user shows a particular expression while typing text, the 3D agent at the receiving end speaks the message aloud while it replays the recognized facial expression sequences and also augments the synthesized voice with appropriate emotional content. Because the visual data exchange is based on the MPEG-4 high-level Facial Animation Parameter for facial expressions (FAP 2), rather than real-time video, our method requires very low bandwidth. The "Web Extras" sidebar offers links to video files of our system at work.

Our system

Our system consists of three main modules: a real-time facial expression analysis component, which can calculate the MPEG-4 FAP 2; an affective 3D agent with facial expression synthesis and text-to-speech capabilities; and a communication module. We have implemented a prototype to explore attractive Internet communication methods, and have also experimented with using these new modalities in chat communication.

How it works

As Figure 1 (on p. 22) shows, our system captures the user's current face image using a head-mounted video camera and traditional GUI input, such as a keyboard and mouse. Several modules then process the flow of input data. The facial expression analysis module calculates the MPEG-4 FAP 2 for the user's current facial expression. Following recognition, the emotional voice generator inserts into the chat message special tags for the text-to-speech synthesis engine to artificially generate emotionally relevant intonation.[5] This adds another modality to the simple

1070-986X/04/$20.00 © 2004 IEEE

Published by the IEEE Computer Society

Related Work

Researchers have experimented with enhanced chat communication using talking heads,[1] but few consider real-time facial expressions as an input channel. Our research background includes facial animation, facial expression analysis, and affective agents.

Agents

Researchers have shown much interest in lifelike, animated agents with realistic behavior in recent years. The function and application domain of such agents is versatile; they've been used in various application domains such as virtual actors, personal interactive tutors, and presentation agents.[2] Our agents are 3D talking heads with synthetic speech, face, and behavior that represent the chat parties.

From the existing facial animation approaches, we chose the pseudomuscle-based concept with a face represented by a polygonal mesh. The pseudomuscle functions conform to the Facial Action Coding system's action units.[3] In addition to facial expressions, we use head and eye motion to construct complex behavioral patterns. Moreover, appropriately changing facial colors help emphasize the agent's emotional state. In our 3D agents, we didn't aim to achieve photorealism, but rather easy modification and animation—as well as reasonable speed and quality on today's average PC platform.

Facial animation

MPEG-4 is an international standard for multimedia communication that includes synthetic 3D objects such as the human face or body. With MPEG-4's Facial Animation Parameters (FAP), any 3D face models that comply with the standard can be animated. The high-level FAP 2 lets us synthesize various facial expressions with varying intensities—that is, we can blend two of the six primary facial expressions (joy, sadness, anger, surprise, disgust, and fear) with a given intensity.[4]

However, the inverse problem of extracting MPEG-4 low- and high-level FAPs from real images is much more problematic because the face is a highly deformable object. Furthermore, when it comes to recognizing facial expression intensities in real time, subtle failures in facial area segmentation and various noise sources—such as variations in lighting—cause crucial problems.

Facial expression analysis

Valente and colleagues[5] have summarized three general facial expression analysis and synthesis methods from the literature:

- feature-based techniques and animation rules;
- analysis-by-synthesis techniques and wireframe adaptation;[6] and
- view-based techniques and key-frame interpolation.

Our method falls into the third category and considers the whole facial image as a single pattern for analysis. Kimura and colleagues have reported a method to construct emotion space using a 2D elastic net model and Karhunen-Loeve expansion for facial images.[7] Their model is user-independent, however, and gives some undesirable results for unknown users. Ohba and colleagues also proposed a facial expression space (FES) that uses principal component analysis and considers the whole facial image as a single pattern.[8] FES is person-dependent and, in training mode, eigen vectors are calculated for specific users. Their input facial expression images are then mapped onto the eigen space. This work is the most similar to our own Personal Facial Expression Space (PFES) recognition method[9] in that both are person-dependent and consider the whole facial image as a single pattern. However, PFES is simpler, and easy to construct and reset for different environments.

References

1. L.C. De Silva, "Putting a Face on the Chat Line," *Innovation*, vol. 2, no. 1, 2001, pp. 54-55.
2. S. Descamps, I. Barakonyi, and M. Ishizuka, "Making the Web Emotional: Authoring Multimodal Presentations Using a Synthetic 3D Agent," *Annual Conf. Computer-Human Interaction Special Interest Group (CHISIG) Ergonomics Soc. of Australia* (OZCHI), IEEE Press, 2001, pp. 25-30.
3. P. Ekman and W. Friesen, *Facial Action Coding System,* Consulting Psychologists Press, 1978.
4. M. Tekalp and J. Osterman, "Face and 2D Mesh Animation in MPEG- 4," *Signal Processing: Image Comm.,* special issue on MPEG-4, vol. 15, 2000, pp. 387-421.
5. S. Valente and J.L. Dugelay, "Face Tracking and Realistic Animations for Telecommunicant Clones," *IEEE MultiMedia,* vol. 7, no. 1, 2000, pp. 34-42.
6. C.S. Choi et al., "Analysis and Synthesis of Facial Image Sequences in Model-Based Image Coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 3, 1994, pp. 257-275.
7. M. Yachida and S. Kimura, "Facial Expression Recognition and Its Degree Estimation," *Proc. Computer Vision and Pattern Recognition* (CVPR'97), IEEE CS Press, 1997, pp. 295-300.
8. K. Ohba, et al., "Facial Expression Communication With FES," *Int'l Conf. Pattern Recognition*, IEEE CS Press, 1998, pp. 1378-1378.
9. N.P. Chandrasiri, T. Naemura, and H. Harashima, "Real-Time Facial Expression Recognition System with Applications to Facial Animation in MPEG-4," *Institute of Electronics, Information, and Comm. Eng. Trans. Information and Systems,* vol. E84-D, no. 8, 2001, pp. 1007-1017.

Web Extras

Video files that demonstrate some of our system capabilities are available at http://csdl.computer.org/comp/mags/mu/2004/02/u3toc.htm. The files include basic system features, such as the

- system initialization,
- real-time mapping of the primary facial expression image on to Personal Facial Expression Space (PFES),
- real-time analysis and synthesis of facial expressions based on PFES, and
- Internet communication using a 3D agent.

text information and helps emphasize the chat message's emotional content.

The agent action generator also processes the recognized MPEG-4 FAP 2; it decides which animation command to send to the user's agent and chooses the animation's type and intensity. These agent actions are immediately replayed by the user's local agent, providing some feedback on how he or she behaves during the chat conversation. The text tagger combines all available text modifications so that the agent action generator can animate the agent by synchronizing different modalities. The communication module then transmits the agent animation commands and tagged text to the chat party over the Internet. Because the channels are full duplex, the communication modules can simultaneously process messages from other chat parties and pass them to the chat parties' agents.

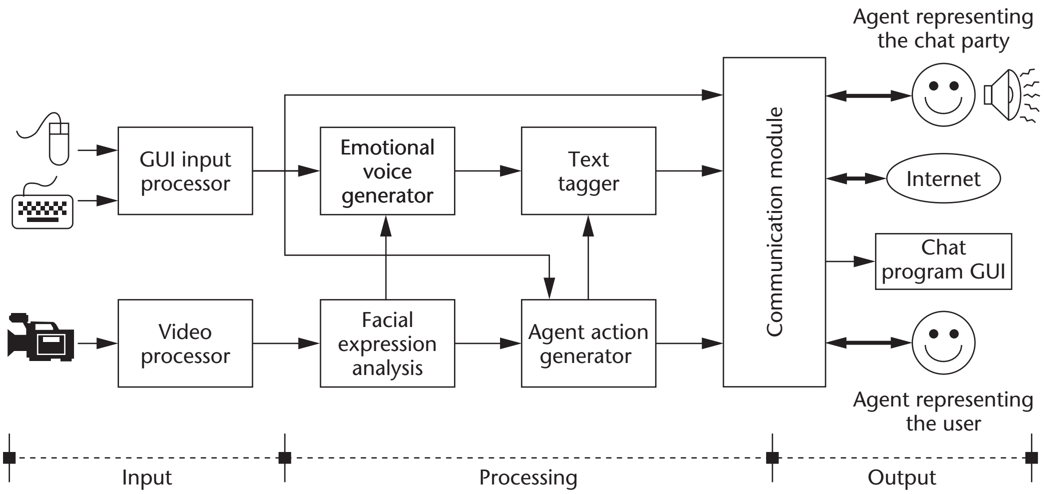
Facial expression analysis

Our main objective is to realize a reliable, real-time communication system. This is typically accomplished by analyzing facial expressions with a view-based technique. However, designing a view-based model is difficult because head pose and facial expressions are variable.[6] We therefore designed a small image-capturing device by attaching a complementary metal oxide semiconductor (CMOS) camera to a headphone, which lets us capture images in the same referential frame (eliminating the head pose problem). Users have already widely accepted headphone use, and a small add-on device such as our camera and its light mounting frame is unlikely to be too intrusive.

When extracting facial expression intensity in a real-time video, person-specific modeling is unavoidable because individuals display a considerable temporal and spatial variation in facial expressions and their apexes. To analyze complex facial expressions, a training database must contain images of all degrees of freedom. A simple way to create person-specific modeling is to use images of that person's primary facial expression apexes as references. When an unknown input facial image from a primary facial image sequence appears, the problem is reduced to constructing a model with generalization capability using known reference images and mapping the unknown input on it.

For person-specific modeling, we acquire reference images that are the apexes of neutral and primary facial expressions (Figure 2a). The system can acquire a reference facial image at different

Figure 1. System overview. After the system's modules process input data, the communication module transmits key information to the chat parties.



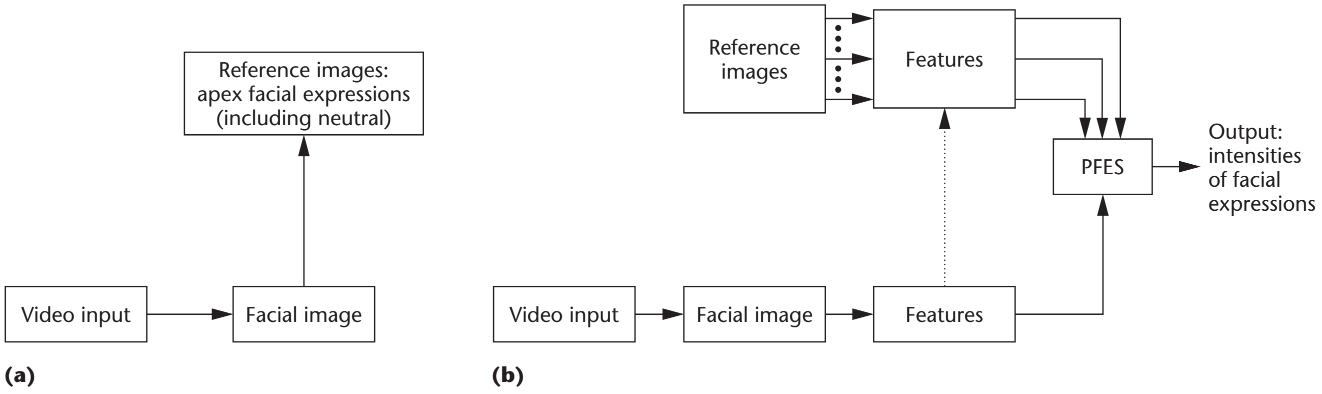

Reference images: apex facial expressions (including neutral)
Video input
Facial image
(a)
Reference images
Features
PFES
Output: intensities of facial expressions
Video input
Facial image
Features
(b)

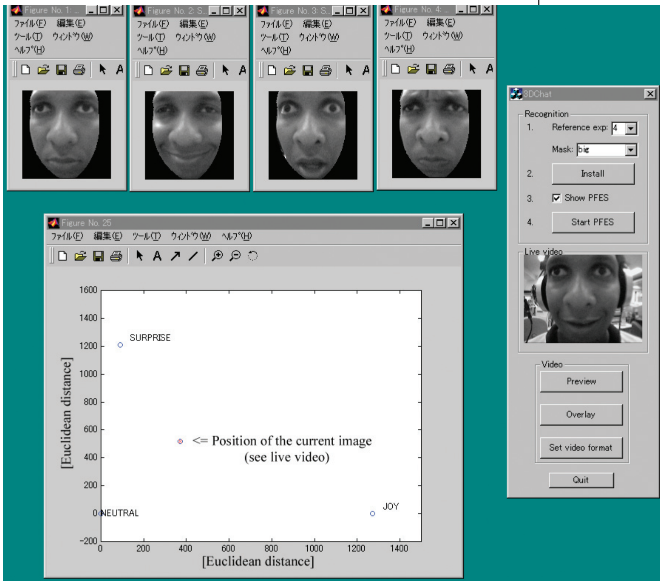
Figure 2. Analysis of primary facial expressions. (a) Acquiring reference facial images. (b) MPEG-4's FAP 2 calculations.

levels. At the lowest level, the system asks the user to show primary facial expression apexes in the initialization step. This is the level our system currently operates at. At the mid-level, the system can analyze facial expressions in a user-independent mode, storing apex facial expression images while the camera tracks the user's face. The user or system operator can then select from among the apex facial expression images by, for example, double clicking on candidate images. At the highest level, acquiring apex facial expression images is fully automated.

Yacoob and colleagues explored facial expression recognition capabilities using optical flow calculation.[7] The resulting model suggests that there is a time interval when no motion is detected at a facial expression's apex. They also suggest that, in an online environment, it's reasonable to assume that people express emotions for brief portions of time only, so the system could categorize the relatively long periods of no motion as neutral. We might use such information in the future for extracting reference facial images.

Our Personal Facial Expression Space (PFES) method[3] recognizes person-specific, primary facial expression image sequences by considering both temporal and spatial characteristics. It also views the whole facial image as a single pattern, without block segmentation. For model features, PFES uses an expression vector representing the changes in the low global frequency coefficient (the discrete cosine transform) relative to a person's neutral facial images. These features are robust and sufficient for real-time processing. The system constructs the PFES through multidimensional scaling, using its generalization capability to map an unknown input image relative to known reference images (see Figure 2b). Because the system possesses linear mapping characteristics, it can easily calculate the expression intensities of an input facial image that contains a mix of two primary expression categories based on its location on the PFES; this also lets the system calculate the MPEG-4 FAP 2.

In our experiments, we use three facial expression categories—joy, surprise, and anger—because they're relatively easy for users to register. Figure 3 shows a screen shot of a facial expression image's real-time mapping onto the PFES. The system interactively registers the top images, which are reference apex facial expressions. To improve the recognition results, we prepared two image masks based on an average face contour. By changing the camera settings and the image mask type, the system can easily eliminate the background image, which constrains



<= Position of the current image (see live video)

1. Reference exp: 4

Mask: big

3. ☑ Show PFES

Figure 3. Real-time facial expression analysis module. The setup window (right) shows the live video, while the top window shows the registered reference images: neutral, joy, surprise, and anger. The bottom window shows the facial image's current position in Personal Facial Expression Space (PFES).

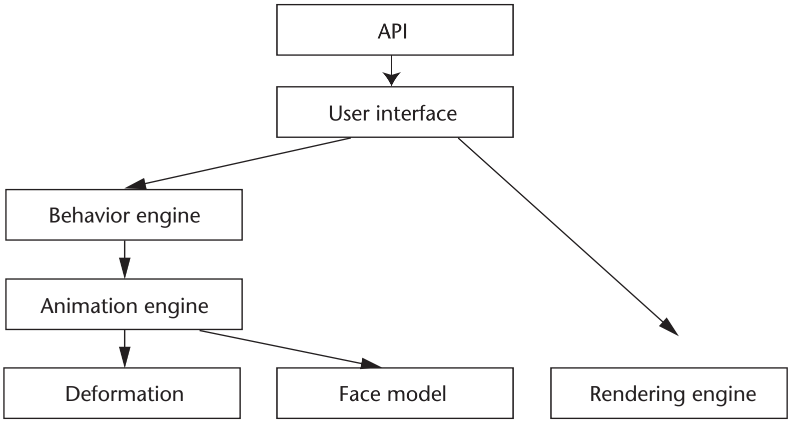


Figure 4. Internal agent structure. The multilayer structure lets the system interact with each level independently.

recognition to the facial area. The system maps the actual input image (right side of Figure 3) onto the PFES (bottom of Figure 3). The MPEG-4 facial animation framework lets systems animate two primary facial expressions simultaneously, with amplitudes defined for each. To achieve MPEG-4 compatibility, we use a 2D PFES, which changes dynamically based on the input and reference images' expression vectors.

3D facial agent

To synthesize believable 3D facial agents, we must model facial anatomy in terms of both structure and function, which involves facial expression synthesis and lip synchronization for text-to-speech synthesis. Synthesizing and animating human faces is a complex procedure with several abstraction levels. We use a multilayer structure that treats levels independently, rather than monolithically (see Figure 4).

The object at the lowest level represents the rendering engine. In addition to rendering the facial mesh's vertices and facets, the rendering engine calculates a smooth skin surface while preserving creases and wrinkles with a shading algorithm. The face model and deformation components contain the face's geometric data and the pseudomuscles for facial expression composition.

The animation engine animates the face at an atomic level (contracting muscles, for example, or moving the eye to a certain position), and provides a smooth, natural transition between movements. We built the behavior engine on top of these primitives; it deals with higher level units like facial expressions at a certain intensity, lip shapes for the speech, or maintaining or breaking eye contact. Moreover, the behavior engine brings the synthetic head to life by automatically generating motion to support speech and give a lifelike impression using natural eye gaze, eye blinks, mouth watering, and so on. However, the system gives priority to the chat partner's animation parameters over automatically generated movements.

The architecture's top-level object is the API, which provides a standardized communication interface with the outside world. This is the service access point for other applications, which in our case is the communication module that delivers text and parameters from the transmitter.

The facial agent component can either be reused inside another application's code or scripted as an individual component. The agent can both react to explicit gesture commands (such as "make a nod" or "show a happy face") and generate natural movements to provide a dynamic, realistic appearance. These automatically generated movements include head rocking, gazing around when idle, turning toward the listener when speaking, and turning away when finishing an utterance. Our work on this is inspired by existing research,[8,9] which are extending with our own studies.

We synthesize primary facial expressions based on the Facial Action Coding system's action units.[10] In Figure 5a, for example, the system synthesizes a surprised expression by activating four different action units. Once primary facial expression images are in the library, the system can synthesize mixed primary expressions of different intensities by interpolating between their facial muscle contraction values. We automatically synchronize lip movement to synthesized speech using the look-ahead coarticulation model.[11]

Figure 5b shows examples of visemes corresponding to phonemes. The agent blends facial expressions with lip shapes coming from the synthesized speech, while preserving a natural-looking face. Behavioral actions are composed of animation channels like facial expression, head and eye movement, and facial color changes (such as blushing or pallor). Figure 6 shows an example of 3D facial agents and their color manipulation.

The agent can also interpret MPEG-4 FAP 2 parameters as input commands to generate appropriate facial expressions. The mesh representing the face is deformable; users can choose from predefined faces at any time during a chat session to change their appearance. To synthesize speech, the agent uses a text-to-speech (TTS) engine, and users can choose various TTS voices to represent sex, age, and personality. Moreover, the agent can insert special tags into the text to be uttered to communicate emotional content with varying intonation.[12] Currently, our system uses Microsoft Speech API 5.x for speech synthesis, so all TTS engines conforming to this interface can be used with the agent. The users' text and the tags related to facial expressions and emotional voice are sent to the other party over the text channel.

As an example, we'll use a basic command that makes the agent say a sentence without modifying the facial expression:

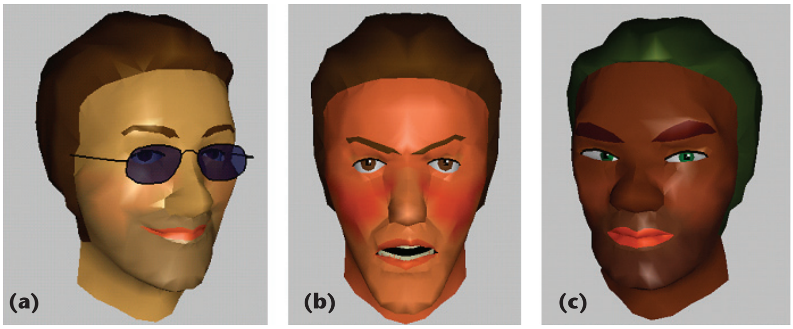


Figure 5. 3D facial agents and facial color manipulation.

```
Agent.Speak(“How are you?”)
```

In the following tagged speech strings, the agent again says “How are you?” but with a happy expression (joy in MPEG-4 FAP 2) and “Oh my god!” with a surprised facial expression:

```
Agent.Speak(“\act=’happiness’
intensity=90 scale=120\ How are
you?”)

Agent.Speak(“\act=’surprise’
intensity=95 scale=70\ Oh my God!”)
```

Tag syntax follows `act=’name’ intensity=i scale=s.`, where *name* stands for the name of the facial animation action (“happy,” “surprise,” and so on), *i* sets its intensity, and *s* scales the action's default duration. The agent action generator automatically generates these tags based on the user's current facial expression when he or she presses a dedicated key on the keyboard. This gives users control over their agents while chatting, and acts as a “capture expression” action in the user interface. By inserting special tags such as pitch, speed, and volume into the text to be uttered, the agent can communicate emotional content with varying intonation, which is more natural.[12] The emotional voice generator automatically inserts these TTS parameters. So, for example, to add a happy tone—typically faster, louder, and a bit higher than neutral—we use the following command:

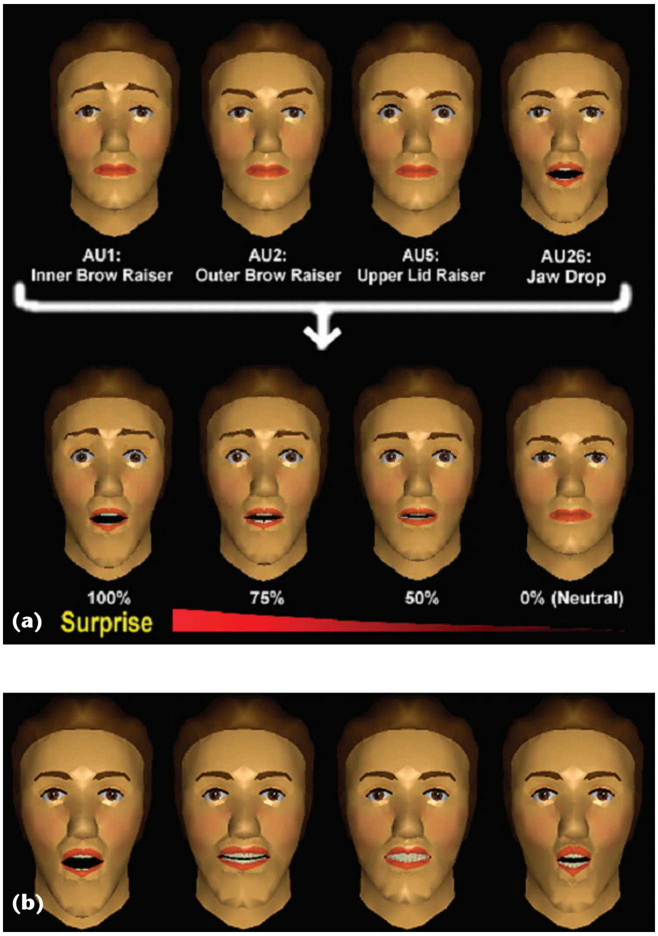
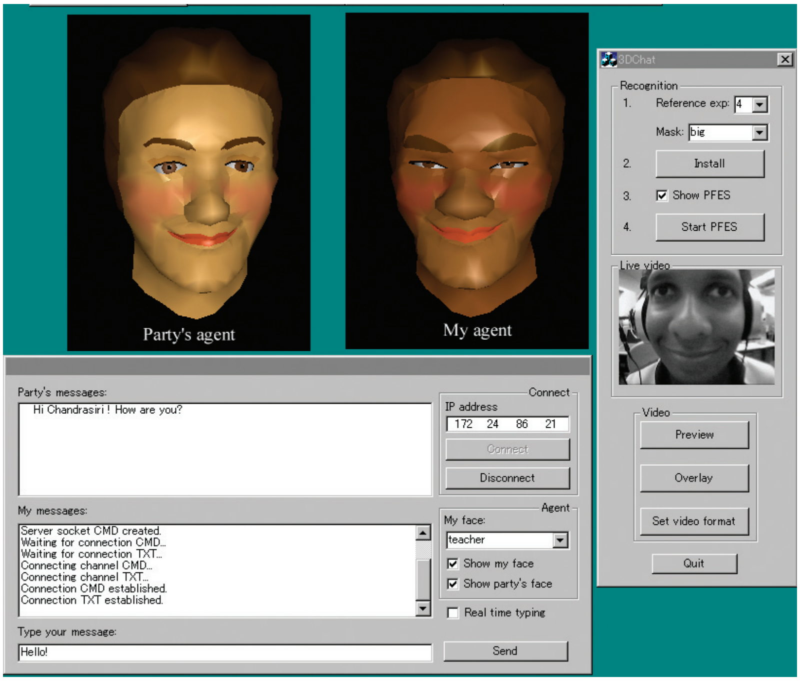


Figure 6. Facial expressions and visemes. (a) The system synthesizes a surprised expression from four different action units. (b) The agent blends facial expressions with lip shapes.

```
Agent.Speak(“<rate
speed=’0’><volume
level=’90’><pitch middle=’4’>
Hello!
How are you doing?</pitch>
</volume></rate>”)
```

The text tagger combines all text modifications available, including facial expressions and intonation, before sending the text to the remote partner.



1. Reference exp: 4
   Mask: big
2. Install
3. ☑ Show PFES
4. Start PFES

Party's messages:

Hi Chandrasiri ! How are you?

Connect

IP address: 172 24 86 21

My messages:

```
Server socket CMD created.
Waiting for connection CMD...
Waiting for connection TXT...
Connecting channel CMD...
Connecting channel TXT...
Connection CMD established.
Connection TXT established.
```

Agent

My face: teacher

☑ Show my face

☑ Show party's face

☐ Real time typing

Type your message: Hello!

Figure 7. A chat session. When real-time typing mode is active, chat partners see each others' real-time expressions and whatever is typed as it's typed.

Communication module

The Internet communication engine's design follows the client–server model, allowing for several chat parties. However, we tested our current system with only two parties, where—depending on the situation—both parties can be either a server or a client. Between server and client, the system maintains two communication channels: text and command channels. The text channel transmits messages typed by the chat party and tagged with agent animation information; the command channel handles commands such as synchronizing the two systems if command message losses are possible, setting the chat parties' facial appearances, or choosing between the chat modes.

Experimental results

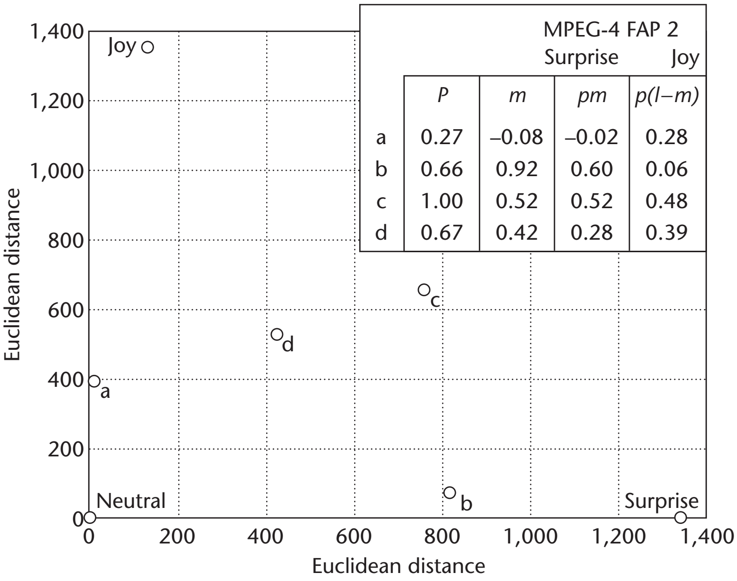
Figure 7 shows a screen shot of a chat session. The two agents at the top of the figure represent two different chat parties. While the local user's agent is pronouncing what the remote user types, the text messages appear in the chat window (bottom). At the same time, the local user can type and send messages. At any point in the chat system, users can

- choose their own facial representation,
- show/hide the local or remote user's agent, or
- turn real-time typing off or on.

If the real-time typing mode is turned on, the remote party can see the user's facial expression and what he or she types letter by letter. The user can also select an agent from a set of predefined agents. The figure's right side shows the setup and live video window, which initializes the facial expression analysis component.

Figures 8 and 9 show the process of calculating MPEG-4 FAP 2 from facial expression images and synthesizing the agent's facial expressions accordingly. The input face's trajectory on the PFES can be calculated in real time—12 frames per second on a Matlab program, running on a PC with a 1-GHz Athlon processor). Input images (left column in Figure 9) are mapped onto the PFES; the system then calculates the MPEG-4 FAP 2. These parameters control synthesized 3D agents (right column in Figure 9). Extracted intensities for joy and surprise are 0.48 and 0.52 for Figure 9c, indicating a 48/52 mixed expression from these two components.

As Figure 10 (p. 28) shows, despite the fact that we use only two parameters to synthesize the agent, we can still obtain convincing



MPEG-4 FAP 2

	Surprise		Joy	
	P	*m*	*pm*	*p*(*l*–*m*)
a	0.27	–0.08	–0.02	0.28
b	0.66	0.92	0.60	0.06
c	1.00	0.52	0.52	0.48
d	0.67	0.42	0.28	0.39

Figure 8. Facial expression analysis and synthesis. Based on input images (see Figure 9, left column) and their locations on the joy–surprise Personal Facial Expression Space, the system calculates the MPEG-4 FAP 2 (top right) and synthesizes 3D agents (Figure 9, right column) based on the results.

results. While the user is typing a sentence, the system analyzes his or her facial expressions; if the user presses a dedicated key on the keyboard, the system inserts his or her facial expression and its intensity at that time into the text as an animation tag (one emotion per message segment). The user can thus control the agent's animation while chatting.

Tests

To assess our communication system's effectiveness, we carried out subjective evaluations involving 30 graduate and undergraduate students at the University of Tokyo. We randomly divided the students into two separate groups of 15. One group experienced the enhanced chat system with full capabilities (Figure 7). For maintaining a controlled condition, the other group experienced a chat system with a chat dialog box only (the bottom window in Figure 7). The subjects had to communicate with a single partner for 15 minutes; we then asked them to fill out a questionnaire (written in both English and Japanese). We asked participants to rate the system on interactivity, attractiveness, naturalness, user interface, and overall effectiveness. We also asked them to rate how effective the system was in communicating closeness and their partner's emotions—so that we could assess how the system affected participants' impressions of their chat partner—as well as how well they felt it transmitted their own emotions. Subjects had to score each item on a scale from 1 to 5, with 1 corresponding to bad and 5 to excellent.

Figure 11a (next page) shows the results. The system's attractiveness score was significantly higher than the controlled condition (with a t-test, where p = 0.001). This suggests that the system's new functions—including facial agents—were attractive to users. Our system's user interface also tended to score higher (p = 0.01), as did its naturalness and overall effectiveness on average. However, there was no significant difference between the two systems' average interactivity scores. One reason for this might be that, due to TTS restrictions, the agent requires a full sentence to speak smoothly, which interrupts interactivity as users typically must wait while their chat partner finishes typing a complete sentence. We plan to enhance this interaction through further research.

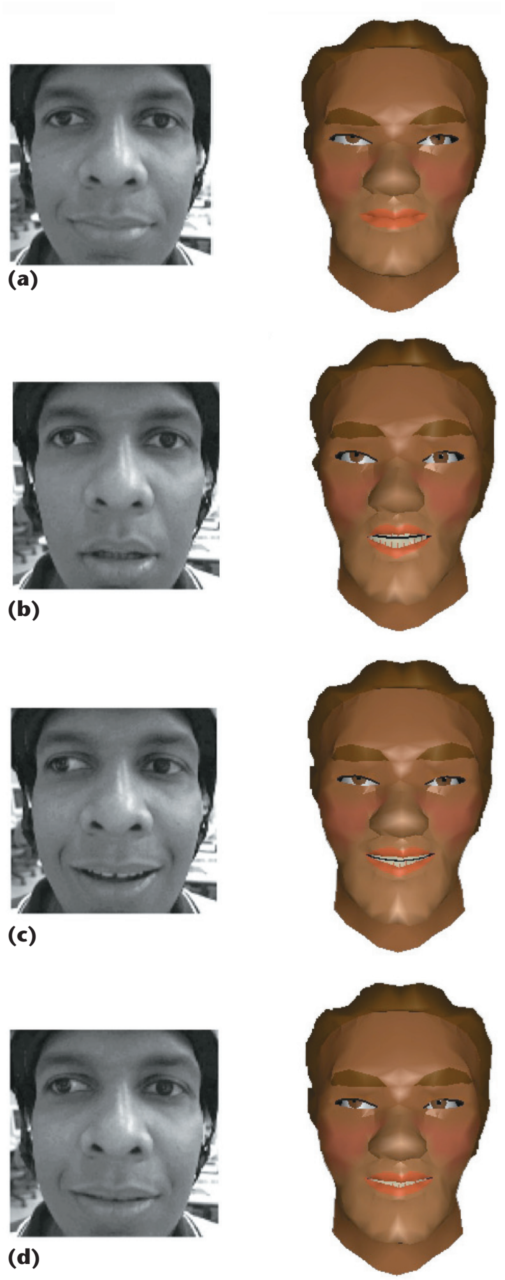
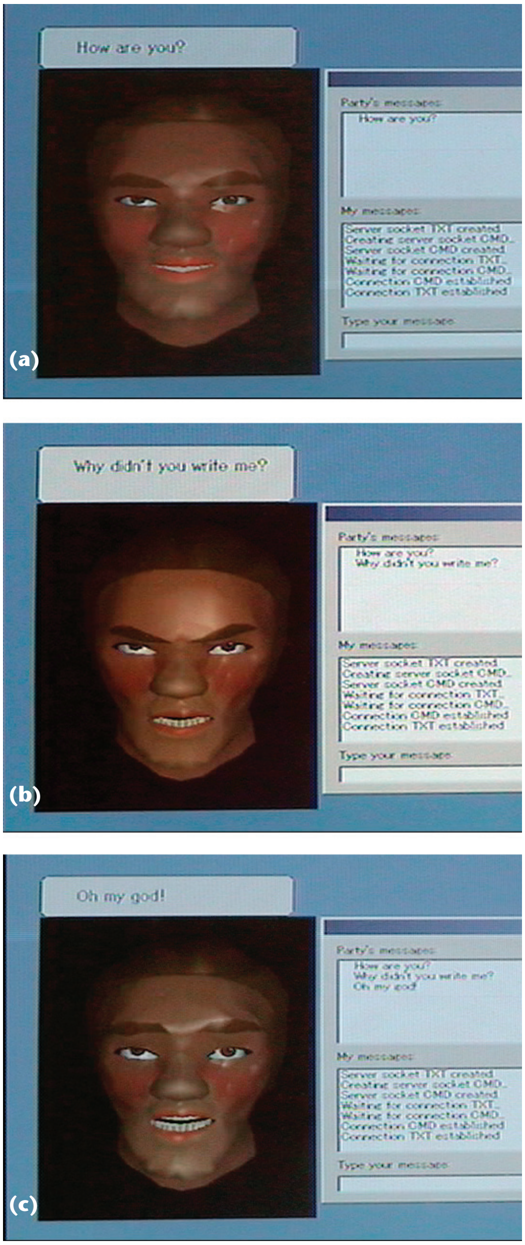


Figure 9. Input images (left) are mapped onto the Personal Facial Expression Space in Figure 8 to produce synthesized 3D agents (right).

Figure 11b shows users' impressions of their chat partners. Users of our system tended to have a higher understanding of their partners' emotions, compared with those who used the controlled system (p = 0.01). Our system also scored higher on feelings of closeness and on transmitting the user's own emotions. Ultimately, our system scored the same or higher than the controlled condition in all cases.

Applications and future work

Potential applications of our system include virtual teleconferencing, entertainment, computer games (such as Internet role-playing games), human-to-human communication training, and distance learning. In addition to work on minimizing interaction delays, our future research



(a)

How are you?

Party's messages:

How are you?

My messages:

Server socket TXT created
Creating server socket CMD...
Server socket CMD created
Waiting for connection TXT...
Waiting for connection CMD...
Connection CMD established
Connection TXT established

Type your message:

(b)

Why didn't you write me?

Party's messages:

How are you?
Why didn't you write me?

My messages:

Server socket TXT created
Creating server socket CMD...
Server socket CMD created
Waiting for connection TXT...
Waiting for connection CMD...
Connection CMD established
Connection TXT established

Type your message:

(c)

Oh my god!

Party's messages:

How are you?
Why didn't you write me?
Oh my god!

My messages:

Server socket TXT created
Creating server socket CMD...
Server socket CMD created
Waiting for connection TXT...
Waiting for connection CMD...
Connection CMD established
Connection TXT established

Type your message:

Figure 10. An active agent. The agent is speaking sentences typed by a user conveying various emotions. (a) "How are you?" (joy), (b)"Why didn't you write me?" (anger), and (c) "Oh my God!" (surprise).

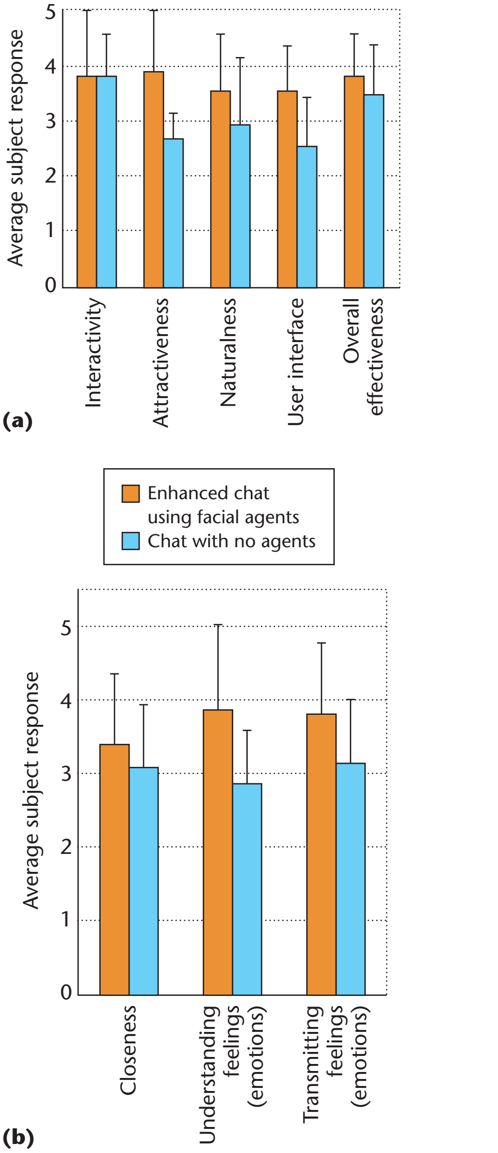


Figure 11. Results of subjective evaluation of our enhanced-chat system compared to a chat system without agents. The y axis indicates average subject response on a scale of 1 (bad) to 5 (excellent). (a) Evaluation of the communication system itself, and (b) evaluations of chat partners.

will include work on adding inputs, such as voice and voice-to-viseme generation, to give users multiple communication inputs. **MM**

Acknowledgments

We thank Sylvain Descamps for providing us with constructive comments and criticism, and Kiyoharu Aizawa for giving us valuable comments.

References

1. H.Nakanishi et al., "Freewalk: 3D Virtual Space for Casual Meetings," *IEEE MultiMedia*, vol. 6, no. 2, 1999, pp. 20-28.
2. N.P. Chandrasiri et al., "Communication over the Internet Using a 3D Agent with Real-Time Facial Expression Analysis, Synthesis, and Text-to-Speech Capabilities," *Proc. IEEE Int'l Conf. Comm. Systems* (ICCS 2002), Springer Verlag (CD ROM), 2002, paper no. 479.
3. N.P. Chandrasiri, T. Naemura, and H.Harashima, "Real Time Facial Expression Recognition System with Applications to Facial Animation in MPEG-4,"

Institute Electronics, Information, and Comm. Eng. Trans. Information and Systems, vol. E84-D, no. 8, 2001, pp. 1007-1017.

4. I. Barakonyi and M. Ishizuka, "A 3D Agent with Synthetic Face and Semiautonomous Behavior for Multimodal Presentations," *Proc. Multimedia Technology and Applications Conf.* (MTAC 2001), IEEE CS Press, 2001, pp. 21-25.
5. S. Descamps, I. Barakonyi, and M. Ishizuka, "Making the Web Emotional: Authoring Multimodal Presentations Using a Synthetic 3D Agent," *Annual Conf. Computer-Human Interaction Special Interest Group (CHISIG) Ergonomics Soc. of Australia* (OZCHI), IEEE Press, 2001, pp. 25-30.
6. S. Valente and J.L. Dugelay, "Face Tracking and Realistic Animations for Telecommunicant Clones," *IEEE MultiMedia,* vol. 7, no. 1, 2000, pp. 34-42.
7. Y. Yacoob and L S. Davis, "Recognizing Human Facial Expressions from Long Image Sequences Using Optical Flow," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, 1996, pp. 636-642.
8. J. Cassell et al., "Animated Conversation: Rule-based Generation of Facial Expression, Gesture, and Spoken Intonation for Multiple Conversational Agents," *Proc. SIGGRAPH'94*, ACM Press, 1994, pp. 413-420.
9. C. Pelachaud, N. Badler, and M. Steedman, "Generating Facial Expressions for Speech," *Cognitive Science*, vol. 20, no. 1, 1996, pp. 1-46.
10. P. Ekman and W. Friesen, *Facial Action Coding System*, Consulting Psychologists Press, 1978.
11. M. Cohen and D. Massaro, "Modeling Coarticulation in Synthetic Visual Speech," *Computer Animation '93*, M. Thalmann and D. Thalmann, eds., Springer Verlag, 1993.
12. J. Cahn, "The Generation of Affect in Synthesized Speech," *J. American Voice I/O Society*, MIT Press, vol. 8, 1990, pp. 1-19.

Naiwala Pathirannehelage Chandrasiri is a research associate in the Intelligent Modelling Laboratory at the University of Tokyo. His research interests include image processing and coding, man-machine interfaces, and human communication engineering. He received his BEng and MEng in information engineering from the Toyohashi University of Technology, Japan, and a PhD in information and communication engineering from the University of Tokyo in 2001.

István Barakonyi is a PhD student at the Vienna University of Technology. His research interests include affective animated agents, facial animation, multimodal interaction, and mixed reality applications. He received his MSc in software engineering at the Budapest University of Technology and Economics in 2000.

Takeshi Naemura is an associate professor in the Interfaculty Initiative in Information Studies at the University of Tokyo. His research interests include image-based rendering, mixed reality, interactive display system, and augmented spatial communications. Naemura has a BE, ME, and PhD in electronic engineering from the University of Tokyo. He is a member of the IEEE.

Mitsuru Ishizuka is a professor in the Department of Information and Communication Engineering at the University of Tokyo. He earned his BS, MS, and PhD degrees from the University of Tokyo. His research interests include artificial intelligence, Internet and WWW intelligence, and multimodal lifelike agents. He is a member of several professional organizations, including the IEEE, AAAI, IEICE Japan, IPS Japan, and the Japanese Society for AI.

Hiroshi Harashima is a professor in the Interfaculty Initiative in Information Studies at the University of Tokyo. His research interests include communication theory, image coding and processing, and human communication engineering. He has a BE, ME, and PhD in electrical engineering from the University of Tokyo. He is the president of the Virtual Reality Society of Japan and an IEEE member.