Perceptual User Interfaces

Matthew Turk
Microsoft Research
One Microsoft Way, Redmond, WA 98052 USA
mturk@microsoft.com

Abstract

For some time, graphical user interfaces (GUIs) have been the dominant platform for human computer interaction. The GUI-based style of interaction has made computers simpler and easier to use, especially for office productivity applications where computers are used as tools to accomplish specific tasks. However, as the way we use computers changes and computing becomes more pervasive and ubiquitous, GUIs will not easily support the range of interactions necessary to meet users' needs. In order to accommodate a wider range of scenarios, tasks, users, and preferences, we need to move toward interfaces that are natural, intuitive, adaptive, and unobtrusive. The aim of a new focus in HCI, called Perceptual User Interfaces (PUIs), is to make human-computer interaction more like how people interact with each other and with the world. This paper describes the emerging PUI field and then reports on three PUI-motivated projects: computer vision-based techniques to visually perceive relevant information about the user.

1. Introduction

Recent research in the sociology and psychology of how people interact with technology indicates that interactions with computers and other communication technologies are fundamentally social and natural [1]. That is, people bring to their interactions with technology attitudes and behaviors similar to those which they exhibit in their interactions with one another. Current computer interfaces, however, are primarily functional rather than social, used mainly for office productivity applications such as word processing. Meanwhile, the world is becoming more and more "wired" – computers are on their way to being everywhere, mediating our everyday activities, our access to information, and our social interactions [2,3]. Rather than being used as isolated tools for a small number of tasks, computers will soon become part of the fabric of everyday life.

Table 1 shows one view of the progression of major paradigms in human-computer interaction (HCI). Historically, there was initially no significant abstraction between users (at that time only programmers) and machines – people "interacted" with computers by flipping switches or feeding a stack of punch cards for input, and reading LEDs or getting a hardcopy printout for output. Later, interaction was focused on a typewriter metaphor – command line interfaces became commonplace as interactive systems became available. For the past ten or fifteen years, the desktop metaphor has dominated the landscape – almost all interaction with computers is done through WIMP-based graphical interfaces (using windows, icons, menus, and pointing devices).

In recent years, people have been discussing post-WIMP [4] interfaces and interaction techniques, including such pursuits as desktop 3D graphics, multimodal interfaces, tangible interfaces, virtual reality and augmented reality. These arise from a need to support natural, flexible, efficient, and powerfully expressive interaction techniques that are easy to learn and use [5]. In addition, as computing becomes more pervasive, we will need to support a plethora of form factors, from workstations to handheld devices to wearable computers to invisible, ubiquitous systems. The GUI style of interaction, especially with its reliance on the keyboard and mouse, will not scale to fit future HCI needs.

The thesis of this paper is that the next major paradigm of HCI, the overarching abstraction between people and technology, should be the model of human-human interaction. *Perceptual user interfaces*, which seek to take advantage of both human and machine perceptual capabilities, must be developed to integrate in a meaningful way such relevant technologies as speech, vision, natural language, haptics, and reasoning, while seeking to understand more deeply the expectations, limitations, and possibilities of human perception and the semantic nature of human interactions.

Era	Paradigm	Implementation
1950s	None	Switches, wires, punched cards
1970s	Typewriter	Command-line interface
1980s	Desktop	GUI / WIMP
2000s	*Natural interaction*	*PUI (multimodal input and output)*

Table 1. The evolution of user interfaces

2. Social Interaction with Technology

In their book *The Media Equation*, Reeves and Nass [1] argue that people tend to equate media and real life. That is, in fact, the "media equation": *media = real life*. They performed a number of studies testing a broad range of social and natural experiences, with media taking the place of real people and places, and found that "individuals' interactions with computers, television, and new media are *fundamentally social and natural*, just like interactions in real life" [1, p. 5]. For example, people are polite to computers and display emotional reactions to technology.

These findings are not limited to a particular type of media nor to a particular type of person. Such interactions are not conscious – although people can bypass the media equation, it requires effort to do so and it is difficult to sustain. This makes sense, given the fact that, during millennia of human existence anything that appeared to be social was in fact a person. The social responses that evolved in this environment provide a powerful, built-in assumption that can explain social responses to technology – even when people know the responses are inappropriate.

This raises the issue of (although does not explicitly argue for) anthropomorphic interfaces, which are designed to appear intelligent by, for example, introducing a human-like voice or face in the user interface (e.g., [6]). Schneiderman [7, 8, 9] argues against anthropomorphic interfaces, emphasizing the importance of direct, comprehensible and predictable interfaces which give users a feeling of accomplishment and responsibility. In this view, adaptive, intelligent, and anthropomorphic interfaces are shallow and deceptive, and they preclude a clear mental model of what is possible and what will happen in response to user actions. Instead, users want a sense of direct control and predictability, with interfaces that support direct manipulation.

Wexelblat [10] questions this point of view and reports on a preliminary study that fails to support the anti-anthropomorphic argument. The experiment involved users performing tasks presented to them with different interfaces: a "standard" interface and an anthropomorphic interface. In general, the debate on anthropomorphic interfaces has engendered a great deal of (sometimes heated) discussion in recent years among interface designers and researchers. (As Wexelblat writes, "Don't anthropomorphize computers; they hate that!")

This debate may be somewhat of a red herring. When a computer is seen as a *tool* – e.g., a device used to produce a spreadsheet for data analysis – the anti-anthropomorphic argument is convincing. Users would not want a humanoid spreadsheet interface to be unpredictable when entering values or calculating sums, for example, or when moving cells to a different column. However, when computers are viewed as *media* or *collaborators* rather than as tools, anthropomorphic qualities may be quite appropriate. Tools and tasks that are expected to be predictable should be so – but as we move away from office productivity applications to more pervasive use of computers, it may well be that the requirements of predictability and direct manipulation are too limiting.

Nass and Reeves write about their initial intuitions:

> What seems most obvious is that media are *tools*, pieces of hardware, not players in social life. Like all other tools, it seems that media simply help people accomplish tasks, learn new information, or entertain themselves. People don't have social relationships with tools. [1, p. 6]

However, their experiments subsequently convinced them that these intuitions were wrong, and that people do not predominately view media as tools.

The growing convergence of computers and communications is a well-discussed trend [11,12]. As we move towards an infrastructure of computers mediating human tasks and human communications and away from the singular model of the computer as a tool, the anti-anthropomorphic argument becomes less relevant. The question becomes, how can we move beyond the current "glorified typewriter" model of human-computer interaction, based on commands and responses, to a more natural and expressive model of interaction with technology?

3. The Role of User Interfaces

The role of a user interface is to translate between application and user semantics. In other words, to translate user semantics to applications semantics using some combination of input modes, and to translate application semantics to user semantics using some combination of output modes. When people communication with one another, we have a rich set of modes to use – e.g., speech (including prosody), gesture, touch, non-speech sounds, and facial expression. Input modes and output modes are not necessarily distinct, mutually exclusive, and sequential; in real conversations they are tightly coupled. We interrupt one another, nod and shake our heads, look bored, say "uh-huh", and use other backchannels of communication.

To build interfaces that support understanding the semantics of the interaction, we must:

- model user semantics
- model application semantics
- model the context
- understand the constraints imposed by the technology
- understand the constraints imposed by models of human interaction

We also constantly deal with ambiguity in human-human interactions, resolving the ambiguity by either considering the context of the interaction or by active resolution (moving one's head to see better, asking "What?", or "Did you mean him or me?"). Alternatively, current human-computer interfaces try to eliminate ambiguity. To effectively model the semantics of the interaction we must support ambiguity at a deep level and not require a premature resolution of ambiguities.

Understanding and communicating semantics is not just an issue of knowledge representation, but also of interaction techniques. The use of a keyboard, mouse, and monitor in the GUI paradigm limits the interaction to a particular set of actions – typing, pointing, clicking, etc. This in turn limits the semantic expression of the interface.

The ideal user interface is one that imposes little or no cognitive load on the user, so that the user's intent is communicated to the system without an explicit translation on the user's part into the application semantics and a mapping to the system interaction techniques. As the nature of computing changes from the predominantly desktop office productivity scenario toward more ubiquitous computing environments, with a plethora of form factors and reasons to interact with technology, the need increases for a paradigm of human-computer interaction that is less constraining, more compelling to the non-technical elite, and more natural and expressive than current GUI-based interaction. An understanding of interaction semantics and the ability to deal with ambiguity are vital to meet these criteria. This may help pave the way for the next major paradigm of how people interact with technology – perceptual interfaces modeled after natural human interaction.

4. Perceptual User Interfaces

The most natural human interaction techniques are those which we use with other people and with the world around us – that is, those that take advantage of our natural sensing and perception capabilities, along with social skills and conventions that we acquire at an early age. We would like to leverage these natural abilities, as well as our tendency to interact with technology in a social manner, to model human-computer interaction after human-human interaction. Such *perceptual user interfaces* [13,14], or PUIs, will take advantage of both human and machine capabilities to sense, perceive, and reason. Perceptual user interfaces may be defined as:

> *Highly interactive, multimodal interfaces modeled after natural human-to-human interaction, with the goal of enabling people to interact with technology in a similar fashion to how they interact with each other and with the physical world.*

The perceptual nature of these interfaces must be bidirectional – i.e., both taking advantage of machine perception of its environment (especially hearing, seeing, and modeling people who are interacting with it), and leveraging human perceptual capabilities to most effectively communicate to people (through, for example, images, video, and sound). When there is sensing involved, it should be transparent and unobtrusive – users should not be required to don awkward or limiting devices in order to communicate. Such systems will serve to reduce the dependence on proximity that is required by keyboard and mouse systems. They will enable people to transfer their natural social skills to their interactions with technology, reducing the cognitive load and training requirements of the user. Such interfaces will extend to a wider range of users and tasks than traditional GUI systems, since a semantic representation of the interaction can be rendered appropriately by each device or environment. Perceptual interfaces will also leverage the human ability to do and perceive multiple things at once, something that current interfaces do not do well.

Perceptual user interfaces should take advantage of human perceptual capabilities in order to present information and context in meaningful and natural ways. So we need to further understand human vision, auditory perception, conversational conventions, haptic capabilities, etc. Similarly, PUIs should take advantage of advances in computer vision, speech and sound recognition, machine learning, and natural language understanding, to understand and disambiguate natural human communication mechanisms.

These are not simple tasks, but progress is being made in all these areas in various research laboratories worldwide. A major emphasis in the growing PUI community [13,14] is on integrating these various sub-disciplines at an early stage. For example, the QuickSet system at OGI [15] is an architecture for multimodal integration, and is used for integrating speech and (pen) gesture as users create and control military simulations. Another system for integrating speech and (visual) gesture is described in [16], applied to parsing video of a weather report. Another example of tight integration between modalities is in the budding "speechreading" community [17,18]. These systems attempt to use both visual and auditory information to understand human speech – which is also what people do, especially in noisy environments.

One main reason that GUIs became so popular is that they were introduced as application-independent *platforms*. Because of this, developers could build applications on top of a consistent event-based architecture, using a common toolkit of widgets with a consistent look and feel. This model provided users with a relatively consistent mental model of interaction with applications. Can PUIs provide a similar

platform for development? Are there perceptual and social equivalents to atomic GUI events such as mouse clicks and keyboard events? (For example, an event that a person entered the scene, a user is looking at the monitor or nodding his head.) These and other questions need to be address more thoroughly by the nascent PUI community before this new paradigm can have a chance to dislodge the GUI paradigm.

The next section describes a few projects in our lab which emphasize one aspect of perceptual interfaces – using computer vision techniques to visually perceiving relevant aspects of the user.

5. Vision Based Interfaces

Present-day computers are essentially deaf, dumb, and blind. Several people have pointed out that the bathrooms in most airports are smarter than any computer one can buy, since the bathroom "knows" when a person is using the sink or toilet. Computers, on the other hand, tend to ask us questions when we're not there (and wait 16 hours for an answer) and decide to do irrelevant (but CPU-intensive) work when we're frantically working on an overdue document.

Vision is clearly an important element of human-human communication. Although we can communicate without it, people still tend to spend endless hours travelling in order to meet face to face. Why? Because there is a richness of communication that cannot be matched using only voice or text. Body language such as facial expressions, silent nods and other gestures add personality, trust, and important information in human-to-human dialog. We expect it can do the same in human-computer interaction.

Vision based interfaces (VBI) is a subfield of perceptual user interfaces which concentrates on developing visual awareness of people. VBI seeks to answer questions such as:

- Is anyone there?
- Where are they?
- Who are they?
- What are the subject's movements?
- What are his facial expressions?
- Are his lips moving?
- What gestures is he making??

These questions can be answered by implementing computer vision algorithms to locate and identify individuals, track human body motions, model the head and face, track facial features, interpret human motion and actions. (For a taxonomy and discussion of movement, action, and activity, see [19]).

VBI (and, in general, PUIs) can be categorized into two aspects: *control* and *awareness*. Control is explicit communication to the system – e.g., put *that* object *there*. Awareness, picking up information about the subject without an explicit attempt to communicate, gives *context* to an application (or to a PUI). The system may or may not change its behavior based on this information. For example, a system may decide to stop all unnecessary background processes when it sees me enter the room – not because of an explicit command I issues, but because of a change in its context. Current computer interfaces have little or no concept of awareness. While many research efforts emphasize VBI for control, it is likely that VBI for awareness will be more useful in the long run.

The remainder of this section describes VBI projects to quickly track a user's head and use this for both awareness and control (Section 5.1), recognize a set of gestures in order to control virtual instruments (Section 5.2), and track the subject's body using an articulated kinematic model (Section 5.3).

5.1. Fast, Simple Head Tracking

In this section we present a simple but fast technique to track a user sitting at a workstation, locate his head, and use this information for subsequent gesture and pose analysis (see [20] for more details). The technique is appropriate when there is a static background and a single user – a common scenario.

First a representation of the background is acquired, by capturing several frames and calculating the color mean and covariance matrix at every pixel. Then, as live video proceeds, incoming images are compared with the background model and pixels that are significantly different from the background are labeled as "foreground", as in Figure 1(b). In the next step, a flexible "drape" is lowered from the top of the image until it smoothly rests on the foreground pixels. The "draping" simulates a row of point masses, connected to each neighbor by a spring – gravity pulls the drape down, and foreground pixels collectively push the drape up (see Figure 1(e)). A reasonable amount of noise and holes in the segmented image is acceptable, since the drape is insensitive to isolated noise. After several iterations, the drape rests on the foreground pixels, providing a simple (but fast) outline of the user, as in Figure 1(d).

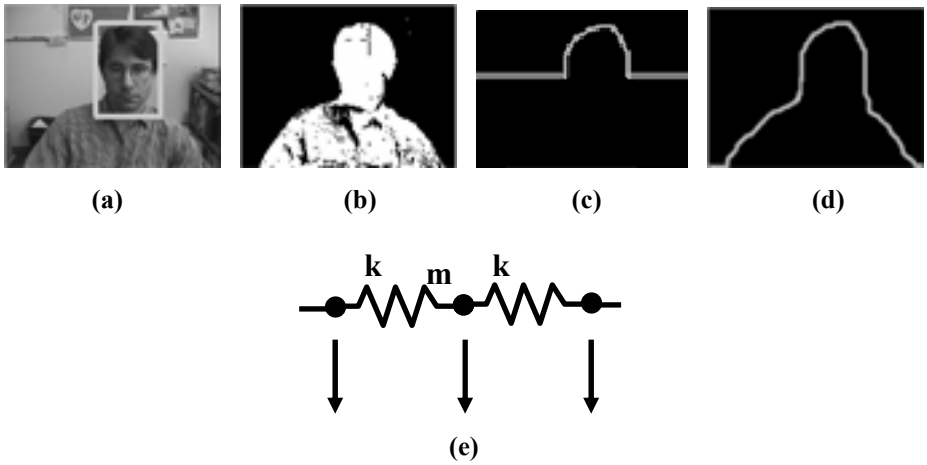

Figure 1. (a) Live video (with head location). (b) Foreground segmentation. (c) Early "draping" iteration. (d) Final "drape". (e) Draping simulates a point mass in each column, connected to its neighbors by springs.

Once the user outline ("drape") settles, it is used to locate the user's head – Figure 1(a) shows the head location superimposed on the live video. All this is done at frame rate in software on a standard, low-end PC. The head location can then be used for further processing. For example, we detect the "yes" and "no" gestures (nodding and shaking the head) by looking for alternating horizontal or vertical patterns of coarse optical flow within the head box. Another use of the head position is to match head subimages with a stored set, taken while looking in different directions. This is used to drive a game of Tic-Tac-Toe, where the head direction controls the positioning of the user's X.

Finally, the shape of the drape (Figure 1(d)) is used to recognize among a small number of poses, based on the outline of the user. Although limited to the user outline, this can be used for several purposes – for example, to recognize that there is a user sitting in front of the machine, or to play a simple visual game such as Simon Says.

5.2. Appearance-Based Gesture Recognition

Recognizing visual gestures may be useful for explicit control at a distance, adding context to a conversation, and monitoring human activity. We have developed a real-time, view-based gesture recognition system, in software only on a standard PC, with the goal of enabling an interactive environment for children [21]. The initial prototype system reacts to the user's gestures by making sounds (e.g., playing virtual bongo drums) and displaying animations (e.g., a bird flapping its wings along with the user).

The algorithm first calculates dense optical flow by minimizing the sum of absolute differences (SAD) to calculate disparity. Assuming the background is relatively static, we can limit the optical flow computation time by only computing the flow for pixels that appear to move. So we first do simple three-frame motion

detection, then calculate flow at the locations of significant motion. Once the flow is calculated, it is segmented by a clustering algorithm into 2D elliptical "motion blobs." See Figure 2 for an example of the segmented flow and the calculated flow blobs. Since we are primarily interested in the few dominant motions, these blobs (and their associated statistics) are sufficient for subsequent recognition.

(a) **(b)**

Figure 2. (a) Original image (b) Flow vectors and calculated flow blobs

After calculating the flow blobs, we use a rule-based technique to identify an action. The action rules use the following information about the motion blobs: the number of blobs, the direction and magnitude of motion within the blobs, the relative motion between blobs, the relative size of the blobs, and the relative positions of the blobs. Six actions – waving, clapping, jumping, drumming, flapping, and marching – are currently recognized. Once the motion is recognized, the system estimates relevant parameters (e.g., the tempo of hand waving) until the action ceases. Figure 3 shows two frames from a sequence of a child playing the "virtual cymbals."

Informal user testing of this system is promising. Participants found it to be fun, intuitive, and compelling. The immediate feedback of the musical sounds and animated characters that respond to recognized gestures is engaging, especially for children. An interesting anecdote is that the child shown in Figure 3, after playing with this system in the lab, went home and immediately tried to do the same thing with his parents' computer.

(a) **(b)**

Figure 3. A user playing the virtual cymbals, with flow blobs overlaid

5.3. Full Body Tracking

To interpret human activity, we need to track and model the body as a 3D articulated structure. We have developed a system [22] which uses disparity maps from a stereo pair of cameras to model and track articulated 3D blobs which represent the major portions of the upper body: torso, lower arms, upper arms, and head. Each blob is modeled as a 3D gaussian distribution, shown schematically in Figure 4. The pixels of the disparity image are classified into their corresponding blobs, and missing data created by self-occlusions is properly filled in. The model statistics are then re-computed, and an extended kalman filter is used in tracking to enforce the articulation constraints of the human body parts.

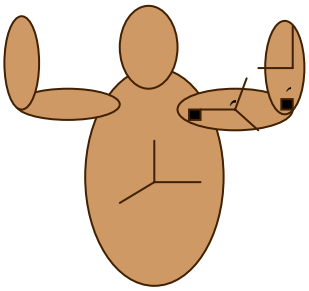

Figure 4. Articulated 3D blob body model

After an initialization step in which the user participates with the system to assign blob models to different body parts, the statistical parameters of the blobs are calculated and tracked. In one set of experiments, we used a simple two-part model consisting of head and torso blobs. Two images from a tracking sequence are shown in Figure 5.

Figure 5. Tracking of connected head and torso blobs

In another set of experiments, we used a four-part articulated structure consisting of the head, torso, lower arm and upper arm, as shown in Figure 6. Detecting and properly handling occlusions is the most difficult challenge for this sort of tracking. The figure shows tracking in the presence of occlusion. Running on a 233 MHz Pentium II system, the unoptimized tracking runs at 10-15 Hz.

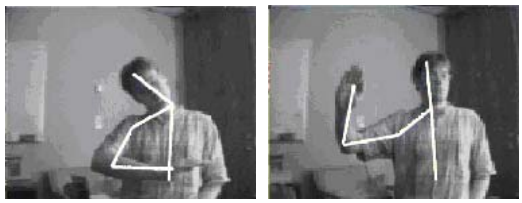

Figure 6. Tracking of head, torso, upper arm, and lower arm

6. Summary and Critical Issues

People treat media – including computers, and technology in general – in ways that suggest a social relationship with the media. Perceptual user interfaces, modeled after human-to-human interaction and interaction with the physical world, may enable people to interact with technology in ways that are natural, efficient, and easy to learn. A semantic understanding of application and user semantics, which is critical to achieving perceptual interfaces, will enable a single specification of the interface to migrate among a diverse set of users, applications, and environments.

Perceptual interfaces do not necessarily imply anthropomorphic interfaces, although the jury is still out as to the utility of interfaces that take on human-like characteristics. It is likely that, as computers are seen

less as tools for specific tasks and more as part of our communication and information infrastructure, combining perceptual interfaces with anthropomorphic characteristics will become commonplace.

Although the component areas (such as speech, language, and vision) are well researched, the community of researchers devoted to integrating these areas into perceptual interfaces is small – but growing. Some of the critical issues that need to be addressed in the early stages of this pursuit include:

- What are the most relevant and useful perceptual modalities?
- What are the implications for usability testing – how can these systems be sufficiently tested?
- How accurate, robust, and integrated must machine perceptual capabilities be to be useful in a perceptual interface?
- What are the compelling tasks ("killer apps") that will demand such interfaces, if any?
- Can (and should) perceptual interfaces be introduced in an evolutionary way in order to build on the current GUI infrastructure, or is this fundamentally a break from current systems and applications?

The research agenda for perceptual user interfaces must include both (1) development of individual components, such as speech recognition and synthesis, visual recognition and tracking, and user modeling, along with (2) integration of these components. A deeper semantic understanding and representation of human-computer interaction will have to be developed, along with methods to map from the semantic representation to particular devices and environments. In short, there is much work to be done. But the expected benefits are immense.

Acknowledgements

Thanks to Ross Cutler and Nebojsa Jojic for their contributions to this paper. Ross is largely responsible for the system described in Section 5.2. Nebojsa is primarily responsible for the system described in Section 5.3.

References

[1] B. Reeves and C. Nass, *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*, Cambridge University Press, September 1996.

[2] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, and D. Robbins, "The New EasyLiving Project at Microsoft Research," *Proc. Joint DARPA/NIST Smart Spaces Workshop*, Gaithersburg, Maryland, July 30-31, 1998.

[3] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, September 1991, pp. 94-104.

[4] A. van Dam, "Post-WIMP user interfaces," *Communications of the ACM,* Vol. 40, No. 2, Pages 63-67, Feb. 1997.

[5] S. Oviatt and W. Wahlster (eds.), *Human-Computer Interaction* (Special Issue on Multimodal Interfaces), Lawrence Erlbaum Associates, Volume 12, Numbers 1 & 2, 1997.

[6] K. Waters, J. Rehg, M. Loughlin, S. B. Kang, and D. Terzopoulos, "Visual sensing of humans for active public interfaces," Technical Report CRL 96/5, DEC Cambridge Research Lab, March 1996.

[7] B. Shneiderman, "Direct Manipulation for Comprehensible, Predictable, and Controllable User Interfaces," *Proceedings of IUI97, 1997 International Conference on Intelligent User Interfaces*, Orlando, FL, January 6-9, 1997, pp. 33-39.

[8] B. Shneiderman, "A nonanthropomorphic style guide: overcoming the humpty dumpty syndrome," The Computing Teacher, 16(7), (1989) 5.

[9] B. Shneiderman, "Beyond intelligent machines: just do it!" *IEEE Software,* vol. 10, 1, Jan 1993, pp. 100-103.

[10] A. Wexelblat, "Don't Make That Face: A Report on Anthropomorphizing an Interface," in *Intelligent Environments*, Coen (ed.), AAAI Technical Report SS-98-02, AAAI Press, 1998.

[11] J. Straubhaar and R. LaRose, *Communication Media in the Information Society*. Belmont, CA: Wadsworth, 1997.

[12] Negroponte, N.. *Being Digital.* New York: Vintage Books, 1995.

[13] M. Turk and Y. Takebayashi (eds.), Proceedings of the Workshop on Perceptual User Interfaces, Banff, Canada, October 1997.
[14] M. Turk (ed.), Proceedings of the Workshop on Perceptual User Interfaces, San Francisco, CA, November 1998. (http://research.microsoft.com/PUIWorkshop/).
[15] P. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "QuickSet: Multimodal interaction for distributed applications," *Proceedings of the Fifth Annual International Multimodal Conference,* ACM Press: New York. November, 1997.
[16] I. Poddar, Y. Sethi, E. Ozyildiz, and R. Sharma, "Toward natural speech/gesture HCI: a case study of weather narration," *Proc. PUI'98 Workshop*, November 1998.
[17] D. Stork and M. Hennecke (eds.), *Speechreading by Humans and Machines: Models, Systems, and Applications*, Springer-Verlag, Berlin, 1996.
[18] C. Beno✕t and R. Campbell (eds.), *Proceedings of the Workshop on Audio-Visual Speech Processing*, Rhodes, Greece, September 1997.
[19] A. Bobick, "Movement, Activity, and Action: The Role of Knowledge in the Perception of Motion," *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, London, England, February 1997.
[20] M. Turk, "Visual interaction with lifelike characters," *Proc. Second IEEE Conference on Face and Gesture Recognition*, Killington, VT, October 1996.
[21] R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition," *Proc. Third IEEE Conference on Face and Gesture Recognition*, Nara, Japan, April 1998.
[22] N. Jojic, M. Turk, and T. Huang, "Tracking articulated objects in stereo image sequences," submitted 1998.

Biography

Matthew Turk is a founding member of the Vision Technology Group at Microsoft Research in Redmond, Washington. He worked on vision for mobile robots in the mid 1980s and has been working in various aspects of vision based interfaces since his PhD work at the MIT Media Laboratory in 1991. His research interests include perceptual user interfaces, gesture recognition, visual tracking, and real-time vision.