Moving from GUIs to PUIs

Matthew Turk

November 30, 1998

Technical Report
MSR-TR-98-69

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
mturk@microsoft.com

Invited paper, *Symposium on Intelligent Information Media*,
Tokyo, Japan, December 1998

Moving From GUIs to PUIs

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. 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 components: 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 other people. Current computer interfaces, however, are primarily functional rather than social or natural, developed primarily for office productivity applications such as work 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 tools for a small number of tasks, computers will soon become part of the fabric of everyday life.

Table 1 shows the progression of major paradigms in human-computer interaction (HCI). At first, there was no significant abstraction between users (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).

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. Evolution of user interfaces

In recent years, people have been discussing post-WIMP [4] interfaces and interaction techniques, including such pursuits as desktop 3D graphics, multimodal 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.

2. 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 perceiving capabilities, along with social skills and conventions that we acquire at an early age. The ultimate interface is one which leverages 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* [6,7], or PUIs, will take advantage of both human and machine capabilities to sense, perceive, and reason. Some advantages of PUIs are:

- Moving beyond the current "glorified typewriter" GUI model, based on *commands* and *responses*, to a more natural, expressive model of dialog.
- Reducing the dependence on proximity that is required by keyboard and mouse systems.
- Transfer of natural social skills to the HCI makes learning the interface easy or unnecessary.
- Interfaces which extend to a wider range of users and tasks
- Interfaces that are user-centered, not device-centered.
- An emphasis on transparent and unobtrusive sensing.

Perceptual user interfaces should take advantage of people's 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 [6,7] is on integrating these various sub-disciplines at an early stage.

For example, the QuickSet system at OGI [8] 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 [9], applied to parsing video of a weather report.

Another example of tight integration between modalities is in the budding "speechreading" community [10,11]. These systems attempt to use both visual and auditory information to understand human speech – which is also what people do, especially in noisy environments.

The 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 take over from the GUI stronghold.

An objection to computer interfaces that are modeled after human-human interaction, and to anthropomorphic interfaces in general, is articulated by Shneiderman [12]. He emphasizes the importance of direct, comprehensible and predictable interfaces, giving users the feeling of accomplishment and responsibility. Without going into the complete argument here, we suggest that there are many situations where sophistication and power are preferred over complete predictability. 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 complete predictability is too limiting.

3. Vision Based Interfaces

Present-day computers are essentially deaf, dumb, and blind. Several people have (playfully) pointed out that the bathrooms in most airports are smarter than any computer one can buy, since they "know" 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 working on a 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 relevant 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 [13]).

In general, VBI 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. 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 next three sections describe VBI projects to quickly track a user's head and use this for both awareness and control (Section 4), (2) recognize a set of gestures in order to control virtual instruments (Section 5), and track the subject's body using an articulated kinematic model (Section 6).

4. 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 [14] 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.

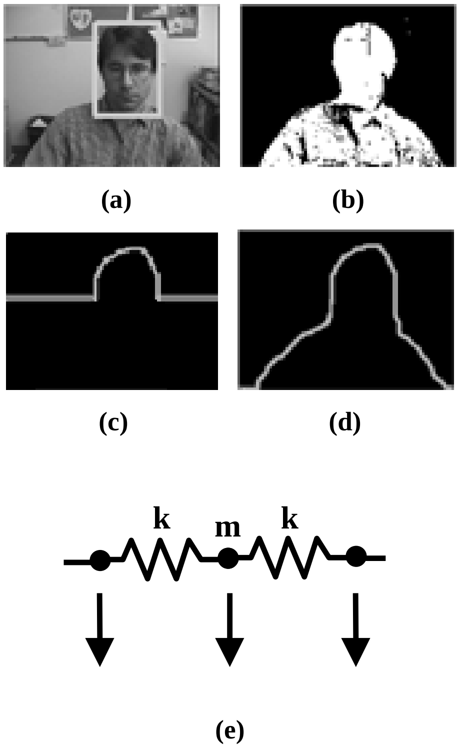

Figure 1. (a) Live video (with head location). (b) Foreground segmentation. (c) Early "draping" iteration. (d) Final "drape". (e) Draping simulates a point mass in each column, connected to its neighbors by springs.

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 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. Appearance-Based Gesture Recognition

Recognizing visual gestures may be useful for explicit control at a distance, adding context to a dialog, and monitoring human activity. We have developed a real-time, view-based gesture recognition system, in software only on a high-end PC, with the goal of enabling an interactive environment for children [15]. 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.

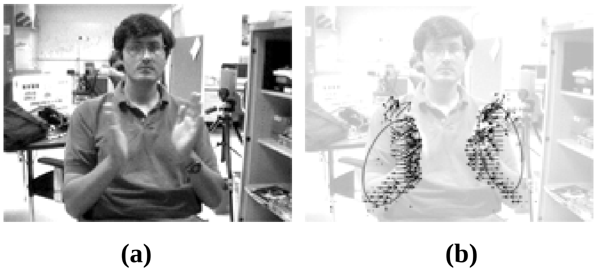

Figure 2. (a) Original image (b) Flow vectors and calculated flow blobs

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.

We have begun to work on a learning technique, involving decision trees and Hidden Markov Models, to learn gestures rather than explicitly model them. The results are very preliminary, but eventually it will be vital to more easily learn new gestures and to adapt to differences among users.

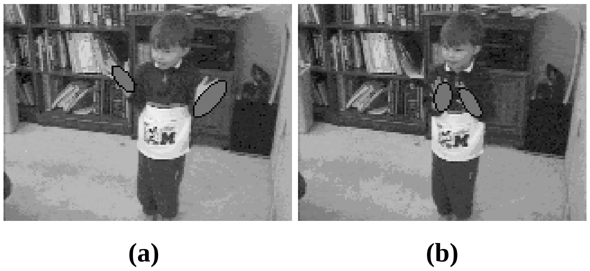

Figure 3. A user playing the virtual cymbals, with flow blobs overlaid

6. 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 [16] 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.

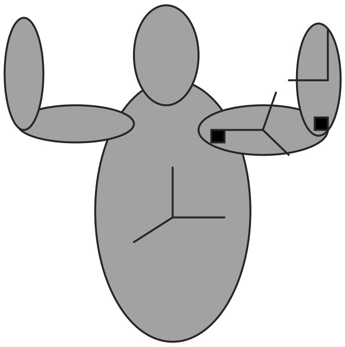

Figure 4. Articulated 3D blob body model

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.

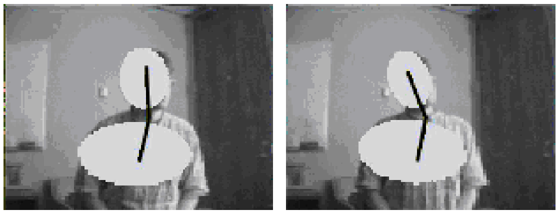

Figure 5. Tracking of connected head and torso blobs

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.

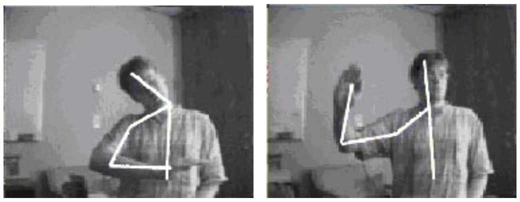

Figure 6. Tracking of head, torso, upper arm, and lower arm

7. Discussion

Sections 4, 5, and 6 are examples of projects which use computer vision techniques to monitor human activity in certain ways. Starting with a resurgence of face recognition research in the late 1980s, there are now many such “looking at people” research efforts as well as an international conference devoted to this area (see http://www-prima.imag.fr/FG99/).

Similarly, there are numerous research efforts in areas such as speech and sound recognition, multimodal user interfaces, haptic interfaces, 3D graphics and visualization, and other areas that are vital to the general goals of Perceptual User Interfaces. What has been missing is a significant overlap and integration among these areas, and also a significant interaction with sociologists and cognitive psychologists. The recent workshops devoted to PUIs [6,7] are an initial attempt to get together a mix of researchers in some of these areas who may not otherwise interact, and to generate enthusiasm for this field.

Early results are promising. As some of the component technologies become closer to commercial success (e.g., speech recognition and face recognition), there appears to be more and more interest in both academia and industry to see natural, intuitive, perceptual user interfaces develop.

Although there may not be a single, coherent interface platform to replace GUIs, the approach of perceptual user interfaces provides a model and a research direction that may prove to enable the future of how people interact with computers and technology.

Acknowledgements

I would like to thank Ross Cutler and Nebojsa Jojij for their contributions to this paper. Ross is largely responsible for the system described in Section 5. Nebojsa is primarily responsible for the systems described in Section 6. Thanks also to Kenji Mase for inviting me to give this talk.

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] M. Turk and Y. Takebayashi (eds.), Proceedings of the Workshop on Perceptual User Interfaces, Banff, Canada, October 1997.

[7] M. Turk (ed.), Proceedings of the Workshop on Perceptual User Interfaces, San Francisco, CA, November 1998. (http://research.microsoft.com/PUIWorkshop/)

[8] 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

[9] 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.

[10] D. Stork and M. Hennecke (eds.), *Speechreading by Humans and Machines: Models, Systems, and Applications*, Springer-Verlag, Berlin, 1996.

[11] C. Benoît and R. Campbell (eds.), *Proceedings of the Workshop on Audio-Visual Speech Processing*, Rhodes, Greece, September 1997.

[12] 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.

[13] 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.

[14] M. Turk, "Visual interaction with lifelike characters," *Proc. Second IEEE Conference on Face and Gesture Recognition*, Killington, VT, October 1996.

[15] 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.

[16] N. Jojic, M. Turk, and T. Huang, "Tracking articulated objects in stereo image sequences," submitted 1998.