From *Proc. Second International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society Press, Killington, VT, Oct. 1996.*

# Visual Interaction With Lifelike Characters

Matthew Turk
Vision Technology Group
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
mturk@microsoft.com

## Abstract

*This paper explores the use of fast, simple computer vision techniques to add compelling visual capabilities to social user interfaces. Social interfaces involve the user in natural dialog with animated, "lifelike" characters. However, current systems employ spoken language as the only input modality. Used effectively, vision can greatly enhance the user's experience interacting with these characters. In addition, vision can provide key information to help manage the dialog and to aid the speech recognition process.*

*We describe constraints imposed by the conversational environment and present a set of "interactive-time" vision routines that begin to support the user's expectations of a seeing character. A control structure is presented which chooses among the vision routines based on the current state of the character, the conversation, and the visual environment. These capabilities are beginning to be integrated into the Persona lifelike character project.*

## 1. Introduction

Human interactions with machines are inherently and unavoidably social. We respond to computers as if they were human, and the social and emotional aspects of that interaction is an important area of user interface research [1]. Social interfaces involve computer-generated characters which attempt to interact with people in natural ways. Current examples of these "lifelike characters" can understand human speech in limited domains and exhibit behavior that appears personable and intelligent. Such interfaces are more compelling in many situations than the more traditional techniques using dialog boxes, command lines, and stored presentations.

However, virtually all lifelike characters are currently blind, with no visual knowledge of the human participants or their environment. We are attempting to impart visual abilities to social interfaces so that the characters know if someone is there, how many people are there, where the participants are looking, what they are doing, etc. The integration of these capabilities will enable a much richer, more compelling experience for people interacting with lifelike characters—and with technology in general.

For the environment to be believable and compelling for the user, human interaction with computer-based characters must be similar to normal human-human interaction. Dialog is by nature interactive, requiring the response of the participants to be both meaningful and timely. As with other perceptual components (e.g., speech recognition and natural language understanding), vision must be reliable and fast, relative to the tasks at hand. These constraints characterize "interactive-time" vision [2] routines, which have the following properties:

- **Fast** This is defined by context; some visual events must be handled more rapidly than others. For example, interpreting user motion to control a pointing device needs to be done at a higher rate than interpreting a gesture to signal "goodbye".
- **Low latency** The total response time is more important than the processing rate (frames per second). Latency and speed requirements are constrained by the maximum acceptable delay in response to various visual events. This may vary among different scenarios (e.g., "power user" vs. entertainment application).
- **Task specific** Routines should take advantage of known constraints which simplify the processing, such as a non-moving camera, static background scene, or consistent lighting conditions. Enumerating

these conditions and choosing among task-specific solutions is often more efficient than implementing a general solution.

- **State dependent** The system state can be used to manage the task-specific routines. Preconditions are defined for each routine so that it is only invoked when applicable.
- **“Know when to say no”** Routines should have a confidence measure along with their output. They are then free to fail as long as they report failure or low confidence in their output.

We describe in this paper the use of such vision routines to add compelling visual capabilities to computer-based characters. Section 2 presents the context of the work, the Persona project. Section 3 describes the role of vision in this context. Section 4 presents a set of vision routines being developed, and Section 5 describes the integration of vision into the Persona system.

## 2. Background: the Persona project

Research in advanced human-computer interfaces is moving beyond the currently ubiquitous GUI/desktop metaphor towards the idea of computers as *assistants* rather than just *tools*. Lifelike computer characters serve as intelligent agents, interacting with and assisting the user in a dynamic, natural fashion. The social and emotional aspects of interaction are important elements in this user interface paradigm.

The Persona project, a multidisciplinary project originating in the User Interfaces Group at Microsoft Research, incorporates speech recognition, natural language understanding, reactive 3D animation, discourse management, and speech and sound output into conversational interfaces, enabling spoken conversation with computer agents. The goal of the project is to develop lifelike animated characters that interact with a user in a natural spoken dialog, building a rapport with the user. The group has developed a prototype system featuring a character named Peedy the Parrot [1], shown in Figure 1. Peedy is a conversational assistant who plays the role of a disk jockey, accepting user requests for, and playing, audio CDs. (A short video clip of Peedy in action can be viewed online at *www.research.microsoft.com/research/vision/mturk/Peedy.htm.*)

Peedy appears to be alert and intelligent, provided that the conversation stays within the domain of Peedy’s knowledge. He is expressive in his dialog, gestures, and facial expressions (e.g., falling asleep when he gets “bored”, putting a wing up to his hear and saying “What?” or “Huh?” if he doesn’t understand something). This dialog is enhanced by expressive sound effects and cinematic camera techniques (e.g., varying camera views of Peedy).

However clearly lacking in Peedy, as in most current efforts developing social interfaces, is visual interaction. Peedy is effectively blind and knows nothing about the user except what may be inferred from the spoken voice. In typical human-to-human interaction, visual cues are important to provide context, emphasis, and specific detail [3]—and a participant’s reactions to such cues are important to provide feedback and a feeling of natural interaction. Adding visual capabilities to Peedy and other lifelike characters will therefore enrich the man-machine interaction.

**Figure 1. Peedy**

## 3. Vision in the loop

Not only is the user’s experience enhanced by the impression of a seeing lifelike character, but simple visual capabilities can also be used to augment other modalities or behaviors of the character. Vision can help to guide or disambiguate the speech recognition process at various levels of precision, from coarse to fine. It can provide important information to the discourse management system such as emphasis on certain words or the user’s puzzled expression. Vision can also provide useful information for general character behavior, for example, the location or identity of the user.

The speech recognition problem becomes much more difficult in noisy environments. Vision can help to spatially locate the speaker and subsequently provide a better input signal by guiding a phased array of microphones. Vision can also provide cues to speech such as which participant is currently speaking, when the speaker's lips are moving, and some degree of lip reading or phoneme disambiguation.

In the Persona project, the discourse manager triggers an appropriate reaction from the character given the current dialog state and event. Vision can provide information used to change state (e.g., the user has gone away, the user is tall), or set a condition for a behavior (e.g., the "Hey, look at me!" behavior). It can also provide important cues to communication by recognizing various semantic and syntactic gestures and facial expressions.

Vision can also serve as the primary input for certain character actions, such as visual tracking or game playing. A simple game of Simon Says will convince many a user that the animated character can really see!

It is important to note that, in the social interface scenario, visual capabilities are not independent, isolated tasks, but elements of a tightly integrated system. Vision shares performance and time constraints with other modalities. Just as importantly, the visual capabilities do not have to exceed those needed to service the user. Even if comprehensive, instantaneous measurements of the users' positions, gestures, identities, and facial expressions were available, only some of the information would be useful to the agent some of the time.

With this in mind, we seek to use computer vision techniques to satisfy a basic level of visual competence, one that gives the user the impression of a seeing character without demanding high precision or general visual capabilities. We also seek to develop software-only solutions that will run on inexpensive, commonly available platforms. We believe that building systems that work, albeit in limited domains, and experimenting with them extensively, will lead to faster development of useful vision-based HCI systems.

## 4. Vision routines

Because vision is considered in the context of the conversational character's behavior, we are developing algorithms to answer character-centered questions such as the following:

- Is someone there?
- How many people are there?

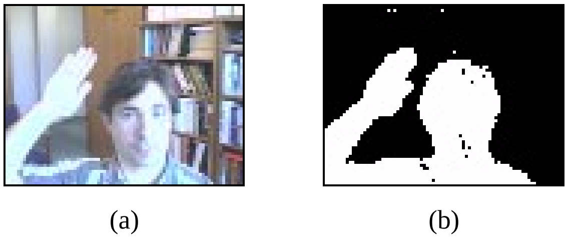

**Figure 2. (a) Input (b) Foreground mask**

- Where is the speaker?
- Where is the speaker looking (towards or away from the character/monitor)?
- Are the speaker's lips moving?
- Who is speaking?
- Is this the same person as before (yesterday, an hour ago, a minute ago, a question ago)?
- How are the speaker's arms and hands positioned?

In addition, we are developing a vision infrastructure to enable a simple game of Simon Says. The following sections briefly describe specific techniques, including segmentation, silhouette tracking, and head tracking, which begin to provide these capabilities. The assumptions or preconditions of each are mentioned—these describe the constraints under which the routine should return something useful.

### 4.1. User segmentation #1 – color and motion

This routine produces a user/background confidence map based on color and motion measurements. The map may then be thresholded and used as a (possibly noisy) binary foreground/background segmentation.

As in each of the segmentation routines, the threshold is not fixed but is a function of the background and information from previous frames. A stationary camera, fixed background, and fixed lighting are assumed. A background image (or set of background images) is assumed to be available ahead of time (Section 4.4).

For each pixel in the input image, the output image value is a combination of calculations based on a color description of the pixel's local area (as distinguished from the color of the background) and local motion, defined by simple temporal filtering. Both measures are fast though noisy; their combination improves the accuracy. Figure 2 shows a single frame and its thresholded output..

### 4.2. User segmentation #2 – camera motion

As in User segmentation #1, this routine produces a user/background confidence map, but allows for some movement of the camera. A fixed background and fixed lighting are assumed, but the camera may move somewhat. This is useful in typical office settings, for example, where the camera may be placed on top of a monitor which may shake or sway or be turned on its swivel.

A translation of the image is assumed, and is estimated by local correlation of small patches at several points in the image. These locations are chosen to be unlikely foreground areas. The maximum translation is around eight percent of the frame size.

### 4.3. User segmentation #3 – background motion

This routine produces a user/background confidence map, allowing for some background motion. Fixed lighting conditions are assumed.

Ignoring the background motion is simply a matter of only looking for motion near the previous frame's motion. This is effective in eliminating spurious motion that is not too close to the user in the camera's field of view.

### 4.4. Background (re)calculation

This routine calculates the background information, used in producing the user/background segmentation. The background may either be calculated as a separate step when no users are present, or "on the fly" when users may be entering, leaving, and moving in the scene. The first scenario requires an explicit setup step or cooperation with the user.

To produce the full background model, multiple background images are taken, and color covariances are calculated for each pixel, similar to [4]. Recalculating the background on the fly involves an iterative procedure over several frames (perhaps spaced over many seconds), where the background is tentatively calculated for areas of the scene in which there is no significant motion (based on temporal characteristics only). As the tentative background model is built, elusive areas (e.g., where a user never leaves) are blacked out and will be defined, until further refined, as foreground areas.

### 4.5. "Draping"

This routine produces a "head and shoulders" silhouette from the user/background segmentation map. It is similar to draping a 1D sheet over the segmentation map from the top, where the sheet can stretch to an approximate fit over the foreground user. A valid user/background segmentation is required.

A procedure is used that is similar in spirit to deformable snakes, but simpler and fast. The 1D "sheet" is defined by point masses at each column of the image, attached by springs. Gravity pulls the sheet down over the thresholded foreground object (assumed to be the user), which holds the sheet in place. Figure 3 shows an example of the draped sheet. The output of the routine is a 1D array of values representing the height of the sheet at each column. This operation runs at frame rate on small (160x120) images, including the segmentation step. The output of this routine is used in coarse pose and gesture recognition (Section 4.9).

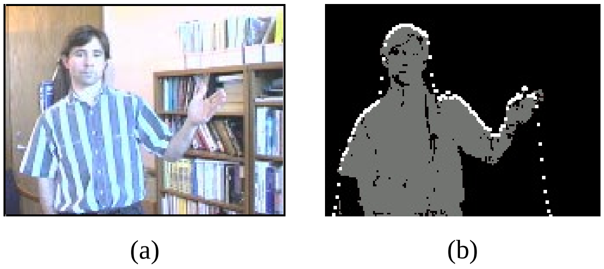
(a) (b)

**Figure 3. (a) Input. (b) Draped segmentation. The dots represent point masses, connected to their neighbors by springs.**

### 4.6. Head tracking

The user's 2D head position is tracked frame by frame, along with a rough estimate of distance from the camera. A valid user/background segmentation is necessary for the initial head location, and a previous head position estimate is needed for tracking.

The head location is accomplished by projecting the foreground segmentation onto vertical and horizontal histograms, limited in area by the previous head position (if available). Analyzing these histograms reveals a reasonable estimate of 2D position and width of the head. Depth is estimated roughly by assuming a single head width for all users.

In another head tracking technique, the user/background segmentation is not necessary, and the head is located and tracked using multiple scale, low-resolution eigenfaces [5].

### 4.7. Head counting

This routine estimates the number of people in the scene by counting heads. It assumes that the heads are separated in the camera's 2D view, that they are at approximately the same height and distance away, and that arms and hands are not raised. A valid user/background segmentation is required.

The "head and shoulders silhouette" is used to count heads, by looking for significant "humps' in the silhouette. It is reasonably reliable for three or fewer heads.

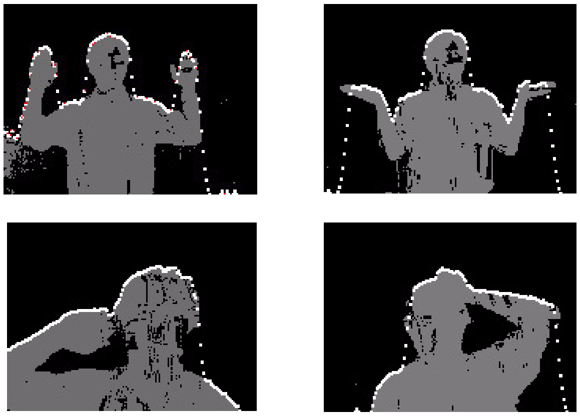

Figure 4. Typical classifiable poses with "draping"

### 4.8. Moving lips

This routine describe the lips as moving or not moving during a short period of time, during which the head is otherwise not moving.

Given a head location, the procedure measures variance, in the color images, of the approximate mouth area over several frames, and declares the mouth to be moving if this measure is above a threshold. The head must be stationary for the short duration. To verify this, a similar measure is done at different locations (the forehead area and just outside the face on either side), and the confidence of the output is only high if these other areas report little change.

### 4.9. Pose recognition

Simple pose ("static gesture") recognition is based on the "head and shoulder silhouette" (Section 4.5). A pose definition stage is performed with the user's cooperation, and each pose is normalized and stored as a 1D array. Poses are subsequently classified via normalized correlation with each frame's silhouette.

The range of discernible poses is limited using this representation, but it is very fast and can be calculated for every frame. Some typical poses are shown in Figure 4.

### 4.10. Gesture recognition

Simple gesture recognition is achieved by constructing a state space represented by a directed graph, where nodes correspond to states and arcs correspond to recognized poses and time events. A sequence of poses over time defines a path through the state space, and particular nodes indicate a recognized gesture. At any given time, several tokens may be moving through the state space.

This representation of gesture is intended for coarse body/arm/head gestures rather than fine gestures such as sign language. In keeping with the spirit of the real-time dialog-based interaction, coarse gestures are sufficient for our present purposes.

### 4.11. Object color definition/classification

This routine defines a general color signature associated with a user, determined by clothing, skin, and hair colors under fixed lighting conditions. This signature is subsequently used to verify that a user is likely to be the same person as in an earlier interaction. A valid user/background segmentation is required.

After a valid segmentation, the foreground pixels are used to define a concise color descriptor, based on color moments [6], which encodes the dominant colors in the area. The descriptor is calculated to be significantly different from color descriptors of the background.

### 4.12. Coarse gaze determination

Is the user looking at the character (the screen), or away? This routine requires that the camera is positioned close to the monitor, so that looking at the animated character can be approximated by looking at the camera.

Given a valid head location, the eyes are located using a local correlation (with "eigeneyes"), and then classified as looking "straight ahead", "left", or "right", as in [2].

## 5. Integration

The routines described in the preceding section are a sample of our ongoing development of visual routines, and a small subset of the routines that may be useful in general interaction with a lifelike character. In this section we briefly describe the control structure that enables the character's dialog manager to use the fast, simple vision routines effectively in a conversational environment.

The diagram in Figure 5 represents the modules and communication paths among the various components of the Persona system. Without the vision component, there is a unidirectional flow of control, from spoken input to graphics, speech, and audio output. At the core of the Dialog module is a state machine which enumerates the character's possible states and state transitions.

The vision system monitors the character state and maintains its own internal state relating to visual tasks. At any given point in time, these states may fulfill one or more or the vision routines' preconditions, at which point the corresponding routines are invoked. Valid output information is flagged and available to both the speech recognition system and the Dialog module.

On a higher level, the lifelike character may have knowledge of certain visual abilities (e.g., to support a

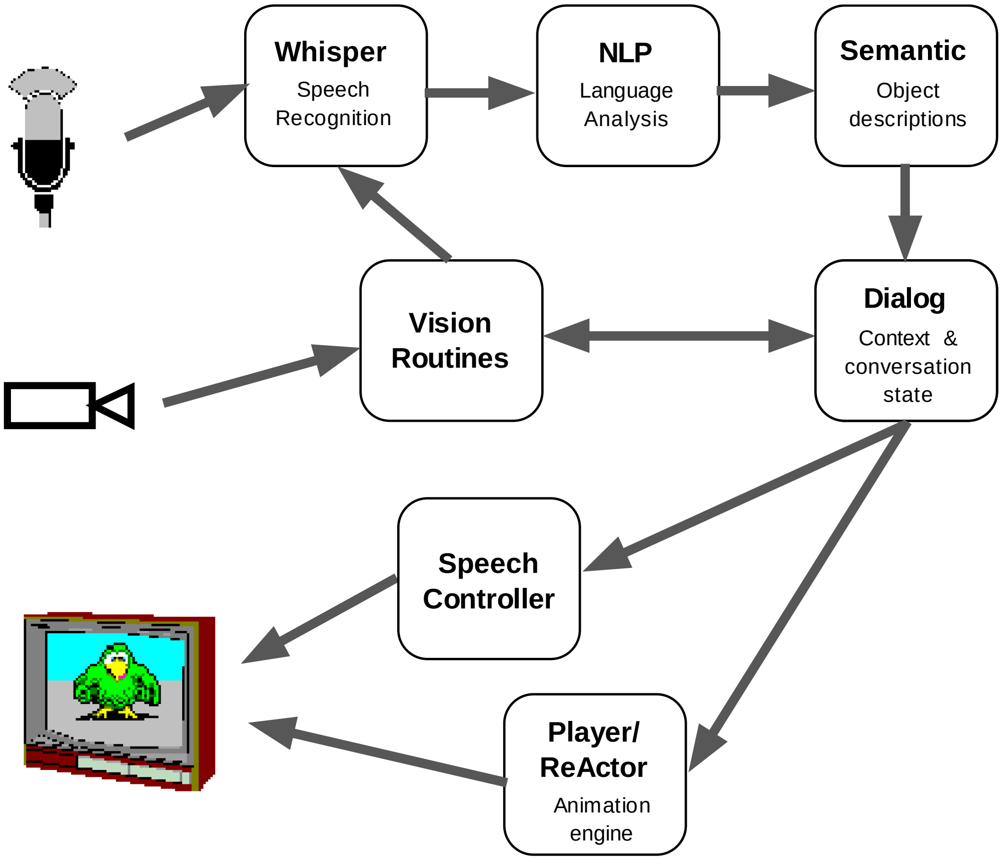

**Figure 5. Persona system overview**

game of “Simon Says”). The Dialog module can explicitly require the vision system to begin supporting such agreed upon behaviors.

There is currently no explicit representation of the speed of vision routines. However, this would be very useful so that the Dialog module can estimate how long it may have to wait to satisfy a particular query, and choose whether or not to abort.

## 6. Conclusion

Social interfaces are already part of consumer multimedia products. More and more CD-ROM titles have animated characters to provide help, lead the user through the story, or make regular GUI interactions more friendly. These are currently simple, deterministic, 2D animated characters. The next generation of social interfaces will include lifelike 3D characters that provide a more intuitive mode of user interaction. Vision is an important part of this interaction, as the appearance of sight adds to the realism of the character and therefore the richness of the user’s experience.

We have presented a set of simple “interactive-time” vision routines that serve as a starting point for visual interaction with lifelike characters. These have been implemented in a desktop Pentium-based computer and are currently being integrated into the Peedy system.

## References

[1] G. Ball et al., “Lifelike computer characters”, in Jeffrey Bradshaw (ed.), *Software Agents*, M.I.T. Press, 1995

[2] M. Turk, *Interactive-Time Vision: Face Recognition as a Visual Behavior*, Ph.D. Thesis, MIT Media Lab, Sept. 1991.

[3] J. Cassell et al., “Modeling the interaction between speech and gesture,” *Proc. 16th Annual Conference of the Cognitive Science Society*, Georgia Institute of Technology, Atlanta, 1994.

[4] C. Wren et al., “Pfinder: Real-time tracking of the human body,” *SPIE Photonics East 1995*, Vol. 2615, pp. 89-98.

[5] M. Turk and A. Pentland, “Eigenfaces for Recognition”, *Proc. CVPR-91*, pp. 1-2, 1991.

[6] M. Stricker and M. Orengo, “Similarity of color images,” in *Storage and Retrieval for Image and Video Databases III*, Wayne Niblack, Ramesh C. Jain, Eds., Proc. SPIE 2420, pp. 381-392, 1995.