

A Perceptual User Interface for Recognizing Head Gesture Acknowledgements

James W. Davis
Dept. Comp. and Info. Sci., Center for Cog. Sci.
Ohio State University
Columbus, OH 43210 USA
jwdavis@cis.ohio-state.edu

Serge Vaks
Dept. Comp. and Info. Sci.
Ohio State University
Columbus, OH 43210 USA
vaks@cis.ohio-state.edu

ABSTRACT

We present the design and implementation of a perceptual user interface for a responsive dialog-box agent that employs real-time computer vision to recognize user acknowledgements from head gestures (e.g., nod = YES). IBM PupilCam technology together with anthropometric head and face measures are used to first detect the location of the user's face. Salient facial features are then identified and tracked to compute the global 2-D motion direction of the head. For recognition, timings of natural gesture motion are incorporated into a state-space model. The interface is presented in the context of an enhanced text editor employing a perceptual dialog-box agent.

1. INTRODUCTION

Computers of today excel at mindlessly carrying out large numbers of repetitive calculations. But computers still remain deaf, dumb, and blind to the user. One of the next major steps in the advancement of computing devices is not only making them faster, but making them more interactive, responsive, and accessible to the user. To achieve natural human-computer interaction requires the use of the modalities that we ourselves use to communicate. *Perceptual user interfaces* combine natural human capabilities of communication, motor, cognitive, and perceptual skills with computer I/O devices, machine perception, and reasoning [28].

In this paper we describe a prototype perceptual user interface for a responsive dialog-box agent that receives mouse- and keyboard-free acknowledgements from the user by visually recognizing the user's intentional head gestures (e.g., nod=YES, shake=NO). Using computer vision, features of the face region are selected and tracked between consecutive frames and used to compute a global 2-D direction of head motion. We incorporate natural timings of gestural head motions into a Finite State Machine for recognition. The vision system uses the IBM PupilCam to detect the user's face and runs at 30Hz on a Pentium III 1Gz computer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PUI 2001 Orlando FL, USA

Copyright 2001 ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

The related work most relevant to recognizing head gestures for interfaces is presented in [29]. Recognition of gestures was achieved using continuous dynamic programming to compare sequences of rotation angles of the user's head (orientation angles were calculated from color-detected face and hair regions). The set of communicative gestures included YES, NO, HAVE DOUBTS, and SURPRISE (other non-communicative motions were also modeled, e.g., look-left). The recognition method was a cumulative distance match from dynamic programming that ignores the underlying temporal component. We assert that the temporal information (e.g., period of the head nod) is an important feature in perception. Another relevant work is by [27] that uses a robust 3-D face tracking system employing color, intensity templates, and point features for positioning a cursor on the computer monitor in real-time. Other related vision-based interfaces include an interactive playspace for children (The KidsRoom) [4, 5], and a reactive virtual exercise trainer [10].

We present a perceptual user interface that uses real-time computer vision to recognize YES/NO acknowledgements from head gestures. We begin with a description of the face detection method using the IBM PupilCam technology and anthropometry of the head and face (Sect. 2). Then we outline our approach to motion calculation using template-matching of selected facial features (Sect. 3). For recognizing the head gestures, we describe a timed Finite State Machine recognition model using the global motion direction of the head as input (Sect. 4). Given the importance of real-time operation, we discuss the relevant implementation details (Sect. 5). We then present an enhanced text editor application designed for the perceptual user interface (Sect. 6). We conclude with a summary of the research and discuss future work (Sect. 7).

2. REAL-TIME FACE DETECTION

To recognize the target head gesture acknowledgements, we first must locate the face of the user in the image. Necessary requirements of the face detection algorithm are that it be real-time and robust to various lighting conditions, background noise, and skin color. The approach used in this research first locates the pupils in the image using the IBM PupilCam and simple image processing techniques. This method does not require any color models, motion, templates, strong geometric models, or examples (e.g., [6, 29, 8, 27, 12, 24, 26]), but instead is based on physiological properties of the eye. After the pupils are found, anthropometric head and face measurements are employed to select the

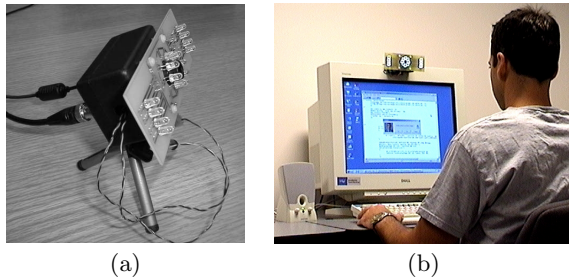


Figure 1: (a) IBM PupilCam. (b) PupilCam placed on monitor in front of computer user.

upper-face region in the image from which salient features are detected and used to compute the head motion.

2.1 PupilCam technology

The IBM PupilCam is a small, low-cost black-and-white video camera with two near-infrared time-multiplexed light sources that provides fast and robust pupil detection from images [18, 19]. The camera with its IR LEDs is shown in Fig. 1(a). We mount the PupilCam on top of a computer monitor (See Fig. 1(b)), with the expectation that future computers will embed cameras in such a position.

The center and side illuminators on the camera are synchronized to the even and odd video fields, respectively. The inner IR ring is sufficiently close the camera optical axis and generates a bright pupil image in the even field (similar to photographic “red-eye”), and the side illuminators generate a dark pupil image in the odd field with similar surrounding brightness. To identify the pupils of a person facing the camera/computer, a full-resolution video frame is first digitized (See Fig. 2(a)) and deinterlaced into the even and odd fields (See Fig. 2(b),(c)). Since the main graylevel intensity difference between these two fields is at the pupil locations (See Fig. 2(d)), image differencing the two fields followed by thresholding generates a binary pupil image. A region-growing algorithm produces the locations of the candidate pupil regions.

When the person’s head is stationary, the pupil regions are usually the only “bright” regions that appear in the difference image. But when the person’s head is in motion (as when nodding YES), image differencing produces several other regions due to the slight time delay between even and odd fields. To account for this type of image difference noise, we update the pupil locations with the two regions in the current image that are closest to the pupil locations from the previous image. The pupil distance is also checked to rule out any unlikely candidates. We additionally use a Kalman filter to attain better tracking results (each pupil region is filtered separately).

We are interested in computing the motion of the user’s head for recognizing gestures, but the motions of the pupil regions are not salient enough for recognition. As a person fixates at a particular location on the computer screen while nodding or shaking the head, the pupils remain fairly fixed to the attended location. But, as we will show, the computed pupil distance in the image can be used with anthropometric head and face measurements to estimate the face region from which we can track robust image features.

	Facial Dimensions (mm)			
	eu-eu	v-en	v-sn	pupil-se
Mean	151.1	121.3	164.3	33.5
SD	5.7	6.8	7.7	2.0
N	109	109	109	40

Table 1: Head and face anthropometry [11] for N males between the ages of 19–25. Female measurements are slightly smaller on average. An illustration of the listed dimensions is shown in Fig. 3.

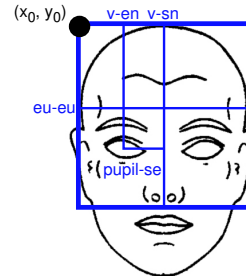


Figure 3: Illustration of anthropometric relations used to determine the upper-face region from the pupil distance.

2.2 Anthropometry

To determine the face region in the video imagery from the pupil locations, we rely on anthropometric data of average head and face proportions across several individuals. Since the mouth may be moving (saying YES or NO) during the gesturing of the head, we exclude the lower portion of the face and focus only on the upper region from the top of the head to the bottom of the nose.

The relevant measurements to locate the upper-face region from the pupil distance can be determined using the data provided in [11]. Actual data measurements for this process are presented in Table 1, and are illustrated in Fig. 3. Letting pd be the calculated pixel distance between the detected right and left pupils pr and pl , we use ratios of the mean values in Table 1 to compute the expected size (width, height) and anchor position (x_0, y_0) of the rectangle enclosing the upper facial region in the deinterlaced image:

$$\text{width} = \frac{\text{eu-eu}}{\text{pupil-se}} \cdot \frac{pd}{2} = 2.26pd \quad (1)$$

$$\text{height} = \frac{\text{v-sn}}{\text{pupil-se}} \cdot \frac{pd}{4} = 1.23pd \quad (2)$$

$$x_0 = pr.x + \frac{pd}{2} - \frac{\text{width}}{2} = pr.x - 0.63pd \quad (3)$$

$$y_0 = pr.y - \frac{\text{v-en}}{\text{pupil-se}} \cdot \frac{pd}{4} = pr.y - 0.91pd \quad (4)$$

Any in-plane rotation of the head is first removed before the rectangle coordinates are calculated. This anthropometric pupil method produces good results, as shown in Fig. 4, and requires very little computational power. Within this extracted face region, we next identify robust facial features for computing the motion of the head to recognize the gestures.

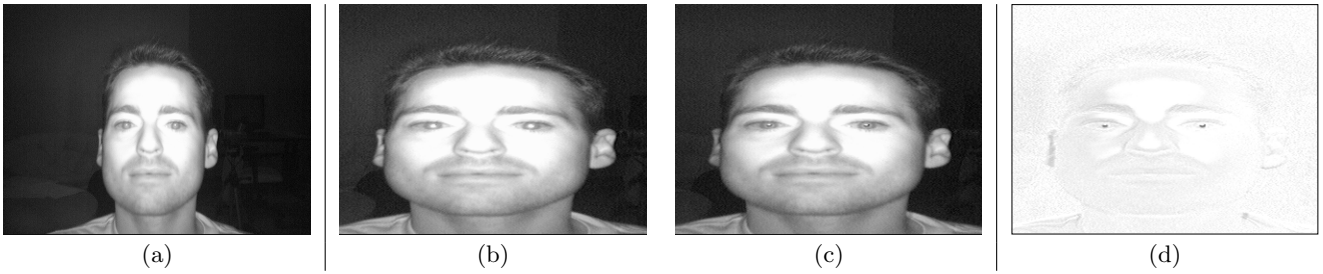


Figure 2: Detecting pupils. (a) PupilCam image. (b) Even video field. (c) Odd video field. (d) Image difference of even and odd fields highlighting the pupils (reverse grayscale image).

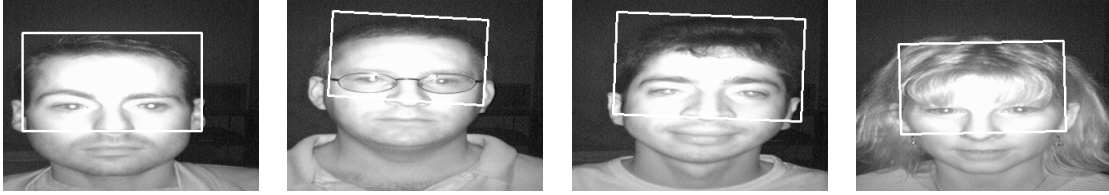


Figure 4: Results of anthropometric upper-face detection with four individuals.

3. MOTION CALCULATION

Our next stage computes the global 2-D image motion of the head for recognizing gestures. The motion is calculated by tracking a set of facial features between consecutive images digitized at 30Hz. For the remaining discussion, we treat the *even* video field as the working image.

3.1 Features detection

Within the selected face region, we identify and rank several salient features to be used for tracking the global motion of the head. Several corner detectors or interest operators could be used to identify the features [7, 21, 14, 17], but we opted for a fast detection method derived from the determinant of the Hessian matrix around a small neighborhood of graylevel pixel values.

The Hessian matrix H is an $n \times n$ matrix whose (i, j) -th entry is a function of the second-order partial derivative $\frac{\partial^2 I}{\partial x_i \partial x_j}$. The determinant of H for a 2-D image I around a local neighborhood is given by

$$D = \det \begin{vmatrix} \frac{\partial^2 I}{\partial x \partial x} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y \partial y} \end{vmatrix} = I_{xx}I_{yy} - I_{xy}^2 \quad (5)$$

Extrema from the determinant of the Hessian matrix at a critical point can be calculated [1], with relative *minimums* having $D > 0$ and $I_{xx} > 0$, and *maximums* having $D > 0$ and $I_{xx} < 0$. Pixels having large determinants produce good features to track without “aperture” problems [2]. Given the fairly fixed distance of people to the camera, we found 5×5 second-derivative Sobel masks to work well.

Small glints are sometimes produced on the face and hair due to the close proximity of the PupilCam IR light sources. These maximum-point secularities do not move with the motion of the head. Therefore we retain only those Hessian values that signal a minimum point ($D > 0$ and $I_{xx} > 0$). We compute the determinants across the detected upper-face region (at every 7th row and column to reduce the number

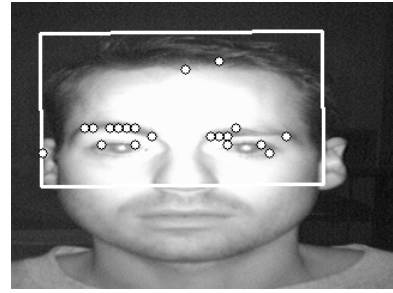


Figure 5: Feature detection. The top 20 features for tracking using Eqn. 5 (relative minimums of determinant extrema).

of pixels examined, and exclude the pupil regions) and sort the minimum extrema in descending order of their determinant value. The top 20 points (largest determinants) are selected for tracking (See Fig. 5). The determinant features are re-computed for each new video frame to allow for accretion/deletion of features (e.g., user turns away from camera) and for fast error recovery.

3.2 Fast motion computation

To compute the 2-D motion of the head from the selected image features, we extract a small template around each of the feature points in the current video frame and use template matching in the previous frame to find the best match for each template. For each of the facial features, we extract a 5×5 local neighborhood of graylevels around the center point (x_c, y_c) as the template and perform a sum-of-squared difference (SSD) within a 10×10 search region in the previous video frame to determine the motion (dx, dy)

for each feature:

$$(dx, dy) \leftarrow \underset{i=-2, j=-2}{\operatorname{argmin}} \left\{ \sum_{i=-2}^2 \sum_{j=-2}^2 [I_t(x_c + i, y_c + j) - I_{t-1}(x_c + i - dx, y_c + j - dy)]^2 \right\} \quad (6)$$

The resulting vertical motion component dy is then doubled to account for the missing horizontal lines in the deinterlaced image (even field). Any dx or dy value ≤ 1 pixel is then set to zero to compensate for any noise. If at least 50 percent of the selected features produce motion vectors, then we construct a motion orientation histogram (quantized to 45 degrees) to select the global orientation θ as the maximum peak in the histogram. The angle θ is assigned a label of U (up motion) if $\theta = 90^\circ$, D (down motion) if $\theta = 270^\circ$, L (left motion) if $\theta = 0^\circ$, R (right motion) if $\theta = 180^\circ$, or NULL otherwise. The temporal behavior of θ is used for recognizing the head gestures.

4. RECOGNIZING NATURAL HEAD GESTURES

The categories of natural head gesture in which we are interested for this work include the common and natural *nodding* and *shaking* motions of the head for relating intentions of YES and NO acknowledgements, respectively. Interestingly, in Bulgaria the gesture meanings are reversed, where one signals NO by nodding up-and-down and YES by shaking back-and-forth. These movements can also refer to different meanings based on the context of the interaction. For example, nodding by a listener in a conversational setting may correspond to “I understand, please continue”. Our perceptual dialog-box domain sets the interaction context to [nod = YES, shake = NO] acknowledgement responses of the user.

4.1 Gestural timings

We designed our system to use the *natural* temporal regularity of communicative head motion, rather than insisting the user to perform slow, exaggerated (un-natural) movements with several repetitions. The natural period of motion for common nod and shake gestures must have some temporal regularity (related to the physical properties of head movement and to communicative salience) if we ourselves are to be able to perceptually understand and recognize them.

In a preliminary study, we video-taped 9 individuals (fluent English speakers) naturally performing the nod and shake head gestures, digitized the video, and clocked the temporal intervals of the *up*, *down*, *left*, and *right* motion components for two to four cycles of each gesture type for each person. The timings for up and down movements were nearly equivalent, as were the timings for left and right motions. In Fig. 6 we show the resulting histograms of time intervals for the up-down and left-right component motions. From these histograms, we can identify the temporal interval regularity associated with each portion of a typical head gesture. We represent each histogram as a Normal distribution with ($\mu_{du} = 0.1773$ sec., $\sigma_{du} = 0.0447$) and ($\mu_{lr} = 0.2023$ sec., $\sigma_{lr} = 0.0632$).

4.2 Timed Finite State Machine model

We model each head gesture as a set of motion states, each having an associated temporal duration. There can be one-to-several cycles of certain timings for head gestures [3],

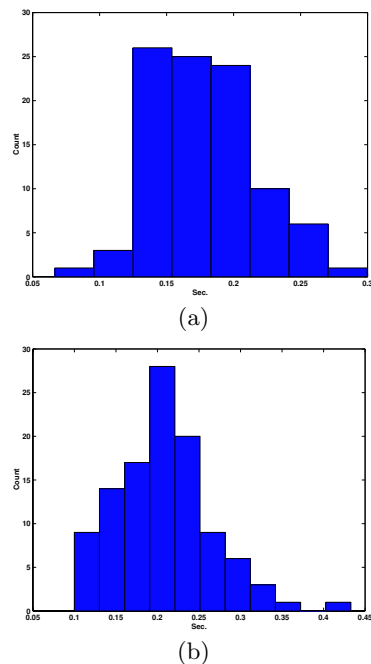


Figure 6: Histograms of manually determined vertical and horizontal motion timings for head nods and shakes of nine people. (a) Histogram of temporal intervals for moving up and moving down (nod). (b) Histogram for moving left and moving right (shake).

hence our model is constructed to identify one gesture cycle at a time (multiple gesture cycles sequentially trigger the model).

We model each head gesture with a timed Finite State Machine (TFSM). An example for the down-up head nod TFSM is shown in Fig. 7. This model contains four states (down motion, up motion, match, fail) with limited state transitions. The calculated temporal interval in each motion state (Δt_D , Δt_U) is retained and delivered to the final match state. Using the temporal histograms in Fig. 6, those motion traces reaching the match state are then verified to have temporal durations above $1/\text{FPS}$ and within $\mu_{du} \pm 3.0\sigma_{du}$ for each state (*Down*, *Up*). If the temporal intervals are found acceptable, then a match is announced. To account for minor errors in the motion calculation, we allow for at most one noise motion (or no motion) in the initial state; these errors usually occur at the state transitions. Because there are no fixed starting directions for the gestures (e.g., some people start a shake from the left, others from the right), we need a total of four TFSMs (down-up, up-down, left-right, and right-left) to recognize the nod and shake gestures from one or several cycles. We opted for the simpler TFSM model rather than the more extensive HMM approach because of the limited structure and timings associated with the gestures.

We examined the approach with video sequences of two people performing several natural head nods and shakes, along with random head motions. Temporal traces of the computed head motion direction θ for Person-1 and Person-2 are shown in Fig 8(a),(b) with frames marked where the

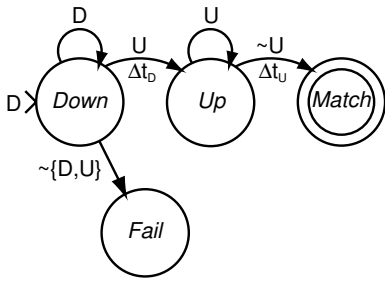


Figure 7: Timed Finite State Machine recognition model for a down-up head nod. The symbol $\sim X$ refers to any motion other than X .

system found a gesture match. The method found all the intended gestures cycles except for a few individual cycles that had multiple noise frames. In Fig. 8(c), we show the results when random head motion of a person is given to the system. No gesture matches were found in this example. Since only one gesture cycle is required to signal an acknowledgement, the method is acceptable for our application. We plan to test the recognition system across several individuals to report more fully on the sensitivity of the approach.

5. IMPLEMENTATION EXTRAS

The described vision system runs at 30Hz on a 1GHz Pentium III computer. Additional hardware includes the IBM PupilCam and a Matrox Meteor II frame grabber.

To increase the speed and efficiency of the implementation, we used the Intel Image Processing Library (IPL) and OpenCV library [13]. The main advantage to using the Intel libraries is that faster image processing is accessible on standard PC-based platforms rather than on specialty systems or costly workstations. The libraries enabled fast calculation of the following compute-intensive image operations:

1. Deinterlacing the 640×480 digitized PupilCam image within a center 320×480 region-of-interest.
2. Image differencing the even-odd video fields and dilation.
3. Calculating the determinant of the Hessian matrix (Eqn. 5) for pixels throughout the face region (using 5×5 second-derivative Sobel operators).

6. PERCEPTUAL AGENT-BASED DIALOG-BOX APPLICATION

The perceptual user interface application for the described computer vision technology is a responsive dialog-box agent. The agent is a “talking head” that resides in a standard Windows dialog-box that appears when the program requires a yes/no acknowledgement from the user, such as “Do you really want to quit?” (See Fig. 9). People interacting with a face attribute a personality to it, and make the computer interface more “human-like” [25]. The character represents the computer side of the interaction, speaking words normally only displayed in the dialog box. The agent character has natural movement and verbally asks the user the prompted request when the dialog-box first appears. Movie

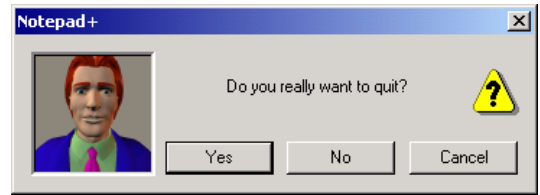


Figure 9: Perceptual dialog-box agent.

clips of the agent (embedded in the dialog-box) were created using Poser [22] and synced with audio files using Mimic [16].

Using the gesture recognition system, the agent determines if the person has gestured with their head a YES or NO response and then performs the appropriate action. Any gesturing when the dialog-box is not displayed is ignored (contextual recognition). To handle situations when the vision system may not recognize the gestures, the interaction defaults to the standard mouse and keyboard interface (available at all times).

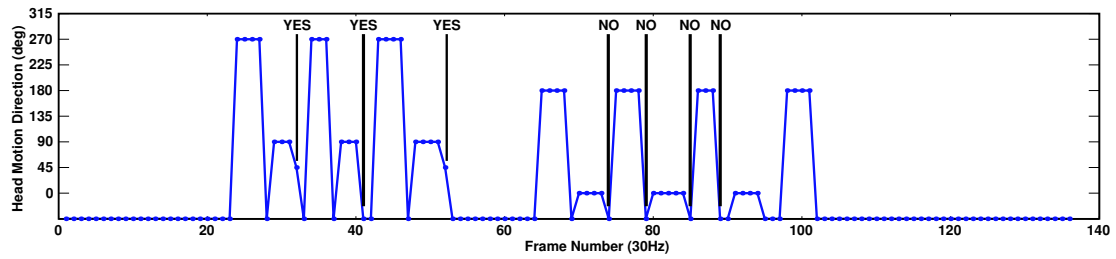
We are currently developing the prototype application fashioned after the Windows Notepad text editor to incorporate and test the agent dialog-box interface. The program, Notepad+, behaves the same as the standard Notepad application except that certain user commands trigger the agent-based dialog-box overriding the standard Windows counterpart. Application events that trigger the agent include **Save**, **Quit**, **Print**, **Close-document**, and **Delete-text-block**. A picture of a person experimenting with the system is shown in Fig. 1(b).

7. SUMMARY AND FUTURE WORK

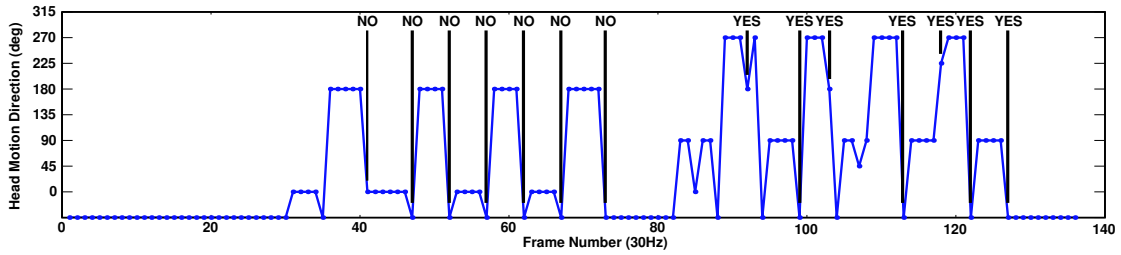
In this paper we presented the design and implementation of a prototype perceptual user interface for a responsive dialog-box agent. The method incorporates real-time computer vision techniques to recognize user acknowledgements from natural head gestures (nod=YES, shake=NO). IBM PupilCam technology together with anthropometric head and face measurements are first used to detect the location of the user’s face. Salient facial features are then identified and tracked between frames to compute the global 2-D motion direction of the head. A Finite State Machine incorporating the natural timings of the computed head motions was employed for recognition. An enhanced text editor application using the perceptual dialog-box agent was also described.

Immediate future work includes further gesture analysis and testing of the system with several users in a more probabilistic TFSM framework. Our next step is to add a speech recognition module to create a multi-modal interface. A preliminary system using the IBM ViaVoice SDK is currently under development. We are additionally considering adding face recognition capabilities to personalize the verbal queries of the agent. Also of interest is a cultural study on the similarities and differences in head gestures (See [20] for an interesting cross-cultural study on hand gestures).

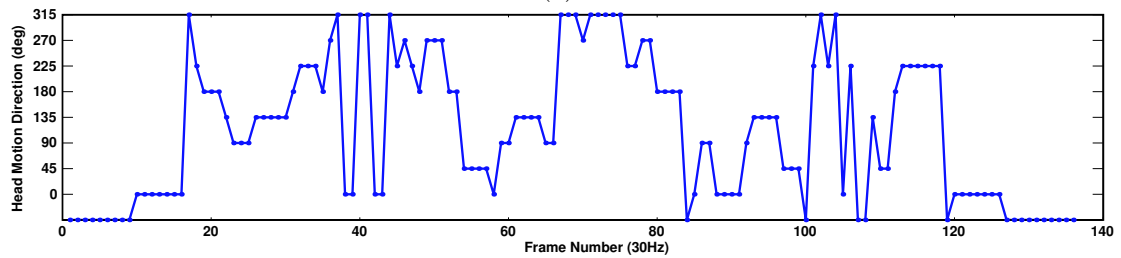
It is conceivable to extend the main theme of this research to include other domains such as smart kiosks and interactive displays [23, 9, 15], but we envision a more immediate future with computers that embrace perceptual user interfaces.



(a)



(b)



(c)

Figure 8: Gesture recognition for YES and NO head motions. (a) Temporal trace of the head motion directions for person-1. (b) Temporal trace of the head motion directions for person-2. (c) Trace of random head motion direction with no gestures signaled.

8. ACKNOWLEDGEMENTS

We thank Myron Flickner and Dave Koons at IBM Almaden for providing us with a PupilCam for this research. We also thank Gary Bradski and Vadim Pisarevsky at Intel for assistance with the OpenCV libraries. We lastly acknowledge support from Intel for this research.

9. REFERENCES

- [1] H. Anton. *Calculus with Analytic Geometry*. John Wiley & Sons, New York, 1988.
- [2] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 15(6):602–605, 1993.
- [3] R. Birdwhistell. *Kinesics and context: Essays on Body Motion Communication*. Univ. Penn. Press, Philadelphia, PA, 1970.
- [4] A. Bobick, J. Davis, and S. Intille. The KidsRoom: an example application using a deep perceptual interface. In *Proc. Wkshp. Percept. User Interfaces*, pages 1–4, 1997.
- [5] A. Bobick, S. Intille, J. Davis, F. Baird, L. Campbell, Y. Ivanov, C. Pinhanez, A. Schutte, and A. Wilson. The KidsRoom: action recognition in an interactive story environment. *Presence: Teleoperators and Virtual Environments*, 8(4):367–391, 1999.
- [6] G. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proc. Wkshp. Applications of Comp. Vis.*, pages 214–219. IEEE, 1998.
- [7] J. Cooper, S. Venkatesh, and L. Kitchen. Early jump-out corner detectors. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 15(8):823–828, 1993.
- [8] J. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proc. Comp. Vis. and Pattern Rec.*, pages 640–645. IEEE, 1997.
- [9] T. Darrell, G. Gordon, M. Marville, and J. Woodfill. Robust visual person tracking for interactive displays. In *Proc. Wkshp. Percept. User Interfaces*, pages 91–94, 1998.
- [10] J. Davis and A. Bobick. Virtual PAT: a virtual personal aerobics trainer. In *Proc. Wkshp. Percept. User Interfaces*, pages 13–18, 1998.
- [11] L. Farkas, editor. *Anthropometry of the Head and Face*. Raven Press, New York, 1994.
- [12] S. Gokturk, J. Bouguet, and R. Grzeszczuk. A data-driven model for monocular face tracking. In *Proc. Int. Conf. Comp. Vis.*, pages 701–708. IEEE, 2001.
- [13] Intel Open Source Computer Vision Library, www.intel.com/research/mrl/research/opencv.
- [14] L. Kitchen and A. Rosenfeld. Gray level corner detection. *Patt. Recogn. Lett.*, 1:95–102, 1982.
- [15] D. Lyons, D. Pelletier, and D. Knapp. Multimodal interactive advertising. In *Proc. Wkshp. Percept. User Interfaces*, pages 83–86, 1998.
- [16] Mimic. Lipsinc, Inc.
- [17] H. Moravec. Towards automatic visual obstacle avoidance. In *Proc. Int. Joint Conf. Artif. Intell.*, 1977.
- [18] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Real-time detection of eyes and faces. In *Proc. Wkshp. Percept. User Interfaces*, pages 117–120, 1998.
- [19] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Comp.*, 18(4):331–335, 2000.
- [20] D. Morris, P. Collett, and P. Marsh. *Gestures: Their Origins and Distribution*. Stein and Day, New York, 1979.
- [21] A. Nobel. Finding corners. *Image and Vision Comp.*, 6:121–128, 1988.
- [22] Poser. Curious Labs, Inc.
- [23] J. Rehg, M. Loughlin, and K. Waters. Vision for a smart kiosk. In *Proc. Comp. Vis. and Pattern Rec.*, pages 690–696. IEEE, 1997.
- [24] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 20(1):23–38, 1998.
- [25] L. Sproull, M. Subramani, S. Kiesler, J. Walker, and K. Waters. When the interface is a face. *Human-Computer Interaction*, 11:97–124, 1996.
- [26] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 20(1):39–51, 1998.
- [27] K. Toyama. Look, ma – no hands! Hands free cursor control with real-time 3D face tracking. In *Proc. Wkshp. Percept. User Interfaces*, pages 49–54, 1998.
- [28] M. Turk and G. Robertson. Perceptual user interfaces. *C-ACM*, 43(3):33–34, 2000.
- [29] H. Wu, T. Shioyama, and H. Kobayashi. Spotting recognition of head gestures from color image series. In *Proc. Int. Conf. Pat. Rec.*, pages 83–85. IEEE, 1998.