Head Gesture Recognition in Intelligent Interfaces: The Role of Context in Improving Recognition

Louis-Philippe Morency
MIT CSAIL
Cambridge, MA 02139, USA
lmorency@csail.mit.edu

Trevor Darrell
MIT CSAIL
Cambridge, MA 02139, USA
trevor@csail.mit.edu

ABSTRACT

Acknowledging an interruption with a nod of the head is a natural and intuitive communication gesture which can be performed without significantly disturbing a primary interface activity. In this paper we describe vision-based head gesture recognition techniques and their use for common user interface commands. We explore two prototype perceptual interface components which use detected head gestures for dialog box confirmation and document browsing, respectively. Tracking is performed using stereo-based alignment, and recognition proceeds using a trained discriminative classifier. An additional context learning component is described, which exploits interface context to obtain robust performance. User studies with prototype recognition components indicate quantitative and qualitative benefits of gesture-based confirmation over conventional alternatives.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Motion*; H.5.2 [**Information systems**]: User Interfaces

General Terms

Design, Experimentation, Algorithms, Human Factors, Performance

Keywords

Head gesture, Context-based recognition, nodding, nod recognition, multi-modal input, user study, IUI design

1. INTRODUCTION

When people interact naturally with each other, it is common to see indications of acknowledgment, agreement, or disinterest given with a simple head gesture. We conjecture that a human-computer interface that could perceive such gestures would enable less disruptive notification, as well as make other interface actions more efficient or easy to perform.

Modern computer interfaces often interrupt a user's primary activity with a notification about an event or condition, which may or may not be relevant to the main activity. Currently, a user must shift keyboard or mouse focus to attend to the notification, and use keyboard and mouse events to page through the displayed information and dismiss the notification before returning to the main activity. Requiring keyboard or mouse events to respond to notifications can clearly cause disruption to users during certain activities or tasks.

Recent advances in computer vision have led to efficient and robust head pose tracking systems, which can return the position and orientation of a user's head through automatic passive observation. Efficient methods for recognition of head gestures using discriminatively trained statistical classifiers have been proposed. In this work we use a robust real-time head tracking and gesture recognition system which was originally developed for interacting with conversational robots or animated characters [15]. We show here how detected head gestures can be used for interface commands under a traditional windows-based graphical user interface.

We explore two types of head gesture-based controls: dialog box acknowledgment/agreement, and document browsing. These components were chosen as they support the aspects of the notification scenario described above. The first allows a user to effectively accept or reject a dialog box or other notification window by nodding or shaking their head. The second component allows a user to page through a document using head nods.

Head gestures have been used in conversational speech interfaces [15] and context was shown to be a critical factor to obtain reliable recognition. Recognition based only on vision can be inaccurate or erratic; by learning a predictor of gesture likelihood from the current dialog state, robust performance was obtained. We develop an analogous context predictor here, but exploit cues from the window manager rather than a dialog model. Figure 1 presents our framework for context-based gesture recognition. We show that learning simple cues about interface state, including features from mouse activity and windows manager state, can similarly reduce erroneous detections by the gesture recognition system.

We conduct user studies with gesture-based recognition components. Results indicate that users feel acknowledgment and agreement are preferred modes of use, and quantitative evaluation of task performance are significantly im-

IUI'06, January 29–February 1, 2006, Sydney, Australia.

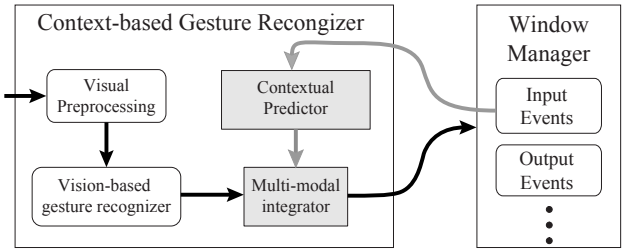

Figure 1: Framework for context-based gesture recognition. The contextual predictor translates contextual features into a likelihood measure, similar to the visual recognizer output. The multi-modal integrator fuses these visual and contextual likelihood measures.

proved under the gesture-based component. Users do not significantly prefer the gesture-based scrolling component, however, and task performance is equivalent. We conclude with a discussion of the applicability of head-gesture components for acknowledgment/agreement, and possible modifications to the scrolling component based on user feedback.

2. BACKGROUND

Several authors have proposed face tracking for pointer or scrolling control and have reported successful user studies [20, 11]. In contrast to eye gaze [22], users seem to be able to maintain fine motor control of head gaze at or below the level needed to make fine pointing gestures[1]. However, many systems required users to manually initialize or reset tracking. These systems supported a direct manipulation style of interaction, and did not recognize distinct gestures.

There has been substantial research in hand/body gesture in for human-computer interaction. Lenman *et al.* explored the use of pie- and marking menus in hand gesture-based interaction[12]. Cohen *et al.* studied the issues involved in controlling computer applications via hand gestures composed of both static and dynamic symbols[5].

Several authors have considered head gesture in interactive dialog with a conversational agent. Cassell [2, 3] and Sidner [18, 19, 17], have developed rich models of multimodal output in the context of embodied natural language conversation, including multimodal representations of agreement and grounding gestures, but used mostly simple visual-based inputs like face detection. Nakano *et al.* analyzed eye gaze and head nods in computer–human conversation and found that their subjects were aware of the lack of conversational feedback from the Embodied Conversational Agent (ECA) [16]. They incorporated their results in an ECA that updated its dialogue state. Breazeal's work [1] on infantoid robots explored how the robot gazed at a person and responded to the person's gaze and prosodic contours in what might be called pre-conversational interactions. Davis and Vaks modeled head nods and head shakes using a timed finite state machine and suggested an application with on-screen embodied agent [6].

[1]Involuntary microsaccades are known to limit the accuracy of eye-gaze based tracking[8].

Recognition of head gestures has been demonstrated by tracking eye position over time. Kapoor and Picard presented a technique to recognize head nods and head shakes based on two Hidden Markov Models (HMMs) trained and tested using 2D coordinate results from an eye gaze tracker [9]. Kawato and Ohya suggested a technique for head gesture recognition using between eye templates [10]. When compared with eye gaze, head gaze can be more accurate when dealing with low resolution images and can be estimated over a larger range than eye gaze [13]. Fugie *et al.* also used HMMs to perform head nod recognition [7]. In their paper, they combined head gesture detection with prosodic recognition of Japanese spoken utterances to determine strongly positive, weak positive and negative responses to yes/no type utterances.

In this paper, we use a head gesture recognition approach based on head velocities and classified using a support vector machine (SVM). Our stereo-based head pose tracker is robust to strong illumination changes, automatically initializes without user intervention, and can re-initialize automatically if tracking is lost (which is rare) [14]. The SVM approach for head gesture recognition was shown to outperform previous techniques based on HMMs [15]. To our knowledge no previous work has investigated recognition-based head-gesture controls in the context of a conventional windows-based interface.

3. GESTURE-BASED INTERACTIONS

Head nods and head shakes are natural gestures commonly used during face-to-face interaction. Inspired by research with human-ECA interactions [19, 16], we propose using head gesture-based controls for two conventional windows interfaces: dialog boxes and document browsing.

Dialog boxes are special windows that are used by computer programs or by the operating system to display information to the user, or to get a response if needed [21]. We will focus our attention to two types of dialog boxes: *notification* dialog boxes and *question* dialog boxes.

Notification dialog boxes are one-button windows that show information from an application and wait for the user to acknowledge the information and click a confirmation button. During human-to-human interactions, the process of ensuring common understanding is called grounding [4]. Grounding is also present during interactions with embodied conversational agents, and human participants naturally head nod as a non-verbal feedback for grounding [16]. From these observations, we can expect human participants to naturally accept head nodding as a way to answer notification dialog boxes.

Question dialog boxes are two-button windows that display a question from the application and wait for positive or negative feedback from the user. This type of dialog box includes both confirmation and rejection buttons. If we look again at interactions that humans have with other humans or with embodied agents, head nods and head shakes are a natural way in many cultures to signify positive and negative feedback, so untrained users should be able to use these kinds of interfaces quite efficiently.

An interesting characteristic of notification and question dialog boxes is that quite often they appear while the user is performing a different task. For example, some email clients will notify the user of new email arrivals using a dialog box saying "You've got mail!". Another example is operating

systems and applications that question the user about installing software updates. In both cases, the user may already be working on another task such as reading emails or browsing a document, and want to answer the dialog box without changing focus. Answering a dialog box using a head gesture makes it possible for users to keep keyboard and mouse focus undisturbed.

Based on our observations, we hypothesize that head gestures are a natural and efficient way to respond to dialog boxes, especially when the user is already performing a different task. We suggest a gesture-based interface design where notification dialog boxes can be acknowledged by head nodding and question dialog boxes can be answered by head nods or head shakes.

Similarly, people use head nods as a grounding cue when listening to information from another person. We conjecture that reading may be similar to listening to information, and that people may find it natural to use head nod gestures to turn pages. We design a prototype gesture-based page-forward control to browse a document, and evaluate it in a user study as described below.

4. CONTEXTUAL FEAUTURES

There are several sources of potential errors with a gesture based recognition system. For example when people search for their cursor on the screen, they perform fast short movements similar to head nods or head shakes, and when people switch attention between the screen and keyboard to place their fingers on the right keys, the resulting motion can appear like a head nod. These types of false positives can cause trouble, especially for users who are not aware of the tracking system.

In research on interactions with ECAs, it has been shown that contextual information about dialog state is a productive way to reduce false positives [15]. A context-based recognition framework can exploit several cues to determine whether a particular gesture is more or less likely in a given situation. To apply the idea of context-based recognition to non-embodied interfaces, i.e. windows-based interfaces, here we define a new set of contextual features based on window manager state.

We want to find contextual features that will reduce false positives that happen during interaction with conventional input devices, and contextual features that can be easily computed using pre-existing information. For our initial prototype we selected two contextual features: f_d and f_m, defined as the time since a dialog box appeared and time since the last mouse event and respectively. These features can be easily computed by listening to the input and display events sent inside the message dispatching loop of the application or operating system (see Figure 1). We compute the dialog box feature f_d as

$$f_d(t) = \begin{cases} C_d & \text{if no dialog box was shown} \\ t - t_d & \text{otherwise} \end{cases}$$

where t_d is the time-stamp of the last dialog box appearance and C_d is default value if no dialog box was previously shown. The same way, we compute the mouse feature f_m as

$$f_m(t) = \begin{cases} C_m & \text{if no mouse event happened} \\ t - t_m & \text{otherwise} \end{cases}$$

where t_m is the time-stamp of the last mouse event and C_m is default value if no mouse event happened recently. In our experiments, C_d and C_m were set to 20.

The contextual features are evaluated at the same rate as the vision-based gesture recognizer (about 18Hz).

5. CONTEXTUAL PREDICTION

We wish to learn a measure of likelihood for a gesture given only the contextual features described in the previous section. This measure will later be integrated with the measure from our vision-based head gesture recognizer to produce the final decision of our context-based gesture recognizer (see Figure 1).

The measure of likelihood is taken to be the distance to a separating surface of a multi-class Support Vector Machine (SVM) classifier that predicts the gesture based on contextual features only. The SVM classifier learns a separating function whose distance $m(x)$ to training labels is maximized. The margin $m(x)$ of the feature vector x, created from the concatenation of the contextual features, can easily be computed given the learned set of support vectors x_i, the associated set of labels y_i and weights w_i, and the bias b:

$$m(x) = \sum_{i=1}^{l} y_i w_i K(x_i, x) + b \tag{1}$$

where l is the number of support vectors and $K(x_i, x)$ is the kernel function. In our experiments, we used a radial basis function (RBF) kernel:

$$K(x_i, x) = e^{-\gamma \|x_i - x\|^2} \tag{2}$$

where γ is the kernel smoothing parameter learned automatically using cross-validation on our training set. After training the multi-class SVM, we can easily compute a margin for each class and use this scalar value as a prediction for each visual gesture.

We trained the contextual predictor using a subset of twelve participants from our user study described in Section 7. Positive and negative samples were selected from this data set based on manual transcription of head nods and head shakes.

6. MULTI-MODAL INTEGRATION AND RECOGNITION

Having described how we anticipate a listener's visual feedback based on contextual information, we now integrate these predictions with observations from a vision-based head gesture recognizer. We will first describe the visual recognizer used during our experiments and then describe integration of contextual predictions.

6.1 Vision-based Head Gesture Recognition

We use a two-step process to recognize head gestures: we first track head position and rotation, and then use a computed head velocity feature vector to recognize head gestures. We use a head tracking framework that merges differential tracking with view-based tracking based on the system described in [14]. We found this tracker was able to track subtle movements of the head for extended periods of time. While the tracker recovers the full 3-D position and velocity of the head, we found features based on angular velocities were sufficient for gesture recognition.

For vision-based gesture recognition (without dialog context), we trained a multi-class SVM with two different classes: head nods and head shakes. The head pose tracker outputs a head rotation velocity vector at each time step (sampled at approximately 18Hz). We transform the velocity signal into a frequency-based feature by applying a windowed FFT to each dimension of the velocity independently. We resample the velocity vector to have 32 samples per second. This transforms the time-based signal into an instantaneous frequency feature vector more appropriate for discriminative training. The multi-class SVM was trained using the RBF kernel described in Equation 2.

We decided to adopt this discriminative approach for our visual head gesture recognition based on our previous experiments [15]. However, other classification schemes could also fit into our context-based recognition framework; all that we require for the multi-modal context fusion described below is that the vision-based head gesture recognizer return a single detection per head gesture. These detections are margins computed directly from the output of the multi-class SVM using Equation 1.

6.2 Multi-modal Integrator

To recognize visual gestures in the context of the current interaction state, we fuse the output of the context predictor with the output of visual head gesture recognizer (see Figure 1).

Our integration component takes as input the margins from the contextual predictor (see Section 5) and the visual observations from the vision-based head gesture recognizer (Section 6.1), and recognizes if a head gesture has been expressed by the human participant. The output from the integrator is further sent to the application so it can be interpreted by one of the two new user interfaces described in Section 3.

We use a multi-class SVM for the integrator since previous experiences show better performance than a linear classifier or simple thresholding [15]. One advantage of our context-based recognition framework is that the integrator can be trained on a smaller data set than the contextual predictor. In our experiment, we trained the integrator on a subset of five participants since the results from the head pose tracker were logged only for these participants.

7. EXPERIMENTAL STUDY

The physical setup consists of a desk with a 21" screen, a keyboard and a mouse. A stereo camera was installed on top of the screen to track the head gaze and recognize head gestures (see Figure 2). This camera was connected to a laptop that ran the recognition system described in Section 6. The recognition system sends recognition results to the main application, which is displayed on the desktop screen in a normal fashion. No feedback about the recognition results is shown on this screen.

We designed our experimental system to evaluate the two gesture-based widgets described in Section 3: dialog box answering and document browsing. The main experiment consisted of two tasks: (1) reading a short text and (2) answering three related questions. Both tasks were performed under three different experimental interaction phases: conventional input only, head gesture input only and user-selected input method. For each interaction, the text and questions were different. During both tasks, dialog boxes would appear at different times asking a question or stating new information.

Figure 2: Experimental setup. A stereo camera is placed on top of the screen to track the head position and orientation.

The reading task was designed to replicate a situation where a person reads an informal text (3 pages) using a document viewer like Adobe Acrobat Reader. At startup, our main application connects to Acrobat Reader, using Component Object Model (COM) technology, displays the Portable Document File (PDF) and waits for the user input. When the participant reached the end of the document, he/she was instructed to close Acrobat Reader and automatically the window for the second task would start. The document browsing widget was tested during this task.

The writing task was designed to emulate an email writing process. The interface was similar to most email clients and included the conventional fields: "To:", "CC:", "Subject:" and the email body. A "Send" button was placed in the top left corner. The questions were already typed inside the email as if the participant was replying to a previous email.

The dialog boxes appearing during both tasks were designed to replicate reminders sent by a calendar application (i.e. Microsoft Outlook), alerts sent by an email client, and questions asked during impromptu moments about software updates or assistant help. Between 4 to 8 dialog boxes would appear during each experiment. The position and text displayed on the dialog box changed between appearances. Participants were asked to answer each dialog box that appeared on the screen. Two types of dialog boxes were displayed: one "OK" button and two "Yes/No" buttons.

Both tasks were repeated three times with three different experimental interaction phases. During the first interaction, the participants were asked to use the mouse or the keyboard to browse the PDF document, answer all dialog boxes and reply to the email. This interaction phase was used as a baseline where participants were introduced to both tasks and they could remember how it feels to interact with conventional input devices.

Between the first and second interaction, a short tutorial about head gestures for user interface was performed where participants practiced the new techniques for dialog box an-

swering and document browsing as described in Section 3. Participants were free to practice it as long as they wanted but most participants were ready to start the second phase after one minute.

During the second phase, participants were asked to browse the PDF document and answer dialog boxes using head nods and head shakes. During the email task, participants had to use the keyboard for typing and could use the mouse for navigating in the email but they were asked to answer any dialog box with a head gesture. This interaction phase was designed to introduce participants to gesture-based widgets.

During the third phase of the experiment, participants were told that they could use any input technique to perform the browsing and email tasks. This interaction was designed so that participants could freely choose between keyboard, mouse or head gestures. In contrast to the previous two phases, this phase should give us an indication of which interaction technique or combination of technique is preferred.

This phase was also designed to compare the accuracy of the head recognizer with the judgement of human observer. For this reason, during this third phase of the experiment a human observer was recognizing intentional head nods from each participant, in a "Wizard of Oz" manner. The vision-based head gesture recognizer was still running during this phase and its results were logged for later comparison.

The study was a within-subject design, where each participant performed more then one interaction phase. A total of 19 people participated in our experiment. All participants were accustomed to use the keyboard and mouse as their main input devices and none of them had used head gesture in a user interface before. Twelve participants completed the first two conditions and only seven participants completed all three conditions. Each condition took 2-3 minutes to complete on average. All participants completed a short questionnaire about their experience and preference at the end of the experiment.

The short questionnaire contained two sets of questions where participants were asked to compare keyboard, mouse and head gestures. The first set of questions was about document browsing while the second set was about dialog box answering. Both sets had the same structure: 2 questions about efficiency and natural interaction followed by a section for general comments. Each question asked the participant to grade all three types of user interfaces (keyboard, mouse and head gesture) from 1 to 5 where 5 is the highest score. The first question asked participants to grade input techniques on how efficient the technique was. The second question asked participants to grade input techniques on how natural the technique was.

8. RESULTS AND DISCUSSION

In this section, we first present the results of the user study, then discuss its implication and finally present results about context-based recognition.

We analyzed the choices each participants made during the third phase of the experiment. During this part of the experiment, the participant was free to decide which input device to use. Figure 3 shows how participants decided to answer dialog boxes and browse documents. For the dialog boxes, 60.4% of the time they used a head gesture to answer the dialog box while using mouse and keyboard only 20.9% and 18.6% respectively. For document browsing, 31.2% of the time they used a head gesture to answer the dialog box while using mouse and keyboard only 22.9% and 45.8% respectively.

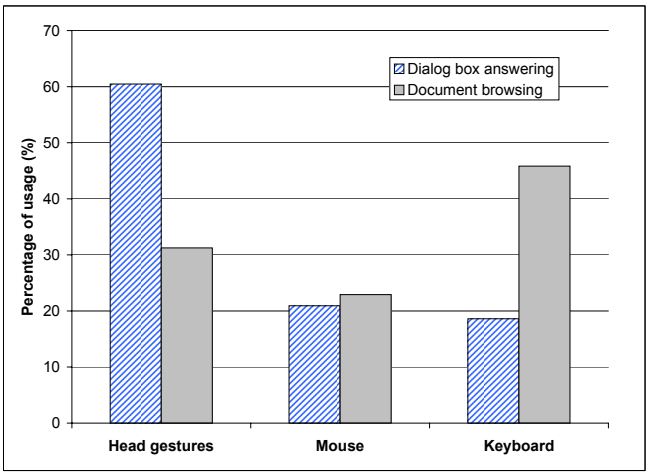

Figure 3: Preferred choices for input technique during third phase of the experiment.

Using a standard analysis of variance (ANOVA) on all 7 subjects who participated to the third phase, results on the dialog box answering widget showed a significant difference among the means of the 3 input techniques: $p = 0.060$. Pairwise comparisons show a significant difference for pairs gesture-mouse and gesture-keyboard, with respectively $p = 0.050$ and $p = 0.083$, while the pair mouse-keyboard shown no significant difference: $p = .45$. Pairwise comparisons for the document browsing show no significant difference between all pair, with $p = 0.362$, $p = 0.244$, and $p < 0.243$ for gesture-mouse, gesture-keyboard, and mouse-keyboard respectively.

We compared the results from vision-based head gesture recognizer with the "Wizard of Oz" results on three participants. The vision-based system recognized 91% of the head nods with a false positive rate of 0.1. This result shows that a vision-only approach can recognize intentional head gestures but suggests the use of contextual information to reach a lower false positive rate.

We also measured the qualitative results from the questionnaire. Figure 4 shows how 19 participants scored each input device for efficiency and natural feeling when interacting with dialog boxes. The average scores for efficiency were 3.6, 3.5 and 4.2, for keyboard, mouse and head gestures respectively. In the case of natural feeling, the average scores were 3.4, 3.7 and 4.2.

Figure 5 shows how 19 participants scored each input device for efficiency and natural feeling for document browsing. The average scores for efficiency were 4.3, 3.8 and 2.6, for keyboard, mouse and head gestures respectively. In the case of natural feeling, the average scores were 4.1, 3.9 and 2.7.

One important fact when analyzing this data is that our participants were already trained to use mouse and keyboard. This previous training affected their choices. The results from Figures 3 and 4 suggest that head gestures are perceived as a natural and efficient way to answer and acknowledge dialog boxes. Participants didn't seem to ap-

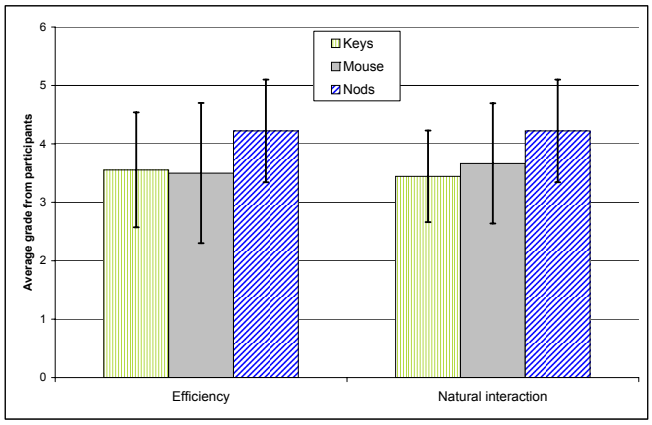

Figure 4: Survey results for dialog box task. All 19 participants graded the naturalness and efficiency of interaction on a scale of 1 to 5, 5 meaning best.

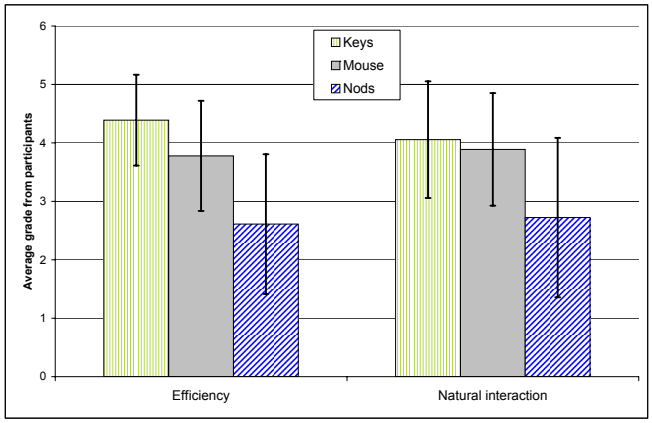

Figure 5: Survey results for document browsing task. All 19 participants graded the naturalness and efficiency of interaction on a scale of 1 to 5, 5 meaning best.

preciate as much the head gesture for document browsing. Some participants stated in their questionnaire that they wanted to have a more precise control over the scrolling of PDF documents. Since the head gesture paradigm only offered control at the page level, we think that this paradigm would apply better to a slide-show application like PowerPoint.

An interesting fact that came from our post-analysis of the user study is that some participants performed head shakes at the notification dialog box (the dialog box with only "OK"). This probably means that they didn't want to be disturbed at that specific moment and expressed their disapproval by a head shake.

To analyze the performance of the context-based head gesture recognizer described in Section 6, we manually annotated 12 interaction sequences for head gestures so that we have a ground truth. From this dataset of 79 minutes of interaction, 269 head nods and 121 head shakes were labeled as ground truth.

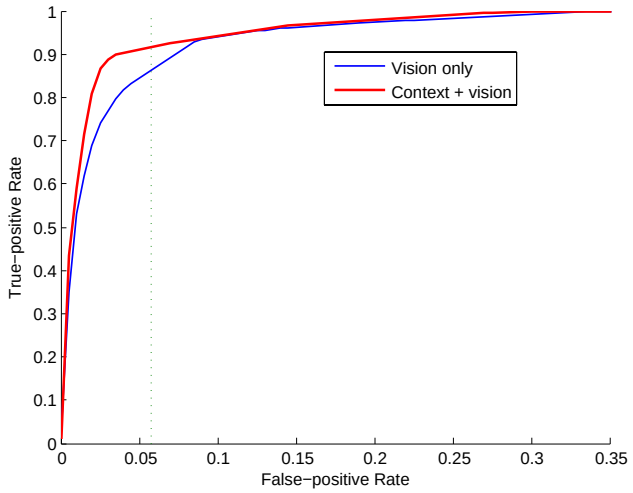

Figure 6: Average ROC curves for head nods recognition. For a fixed false positive rate of 0.058 (operational point), the context-based approach improves head nod recognition from 85.3% (vision only) to 91.0%.

Figure 6 shows head nod detection results for the vision-only technique and the context-based recognizer. The ROC curves present the detection performance of each recognition algorithm when varying the detection threshold. We computed the true positive rate using the following ratio:

$$\text{True positive rate} = \frac{\text{Number of detected gestures}}{\text{Total number of ground truth gestures}}$$

A head gesture is tagged as detected if the detector triggered at least once during a time window around the gesture. The time window starts when the gesture starts and ends k seconds after the gesture. The parameter k was empirically set to the maximum delay of the vision-based head gesture recognizer (1.0 second). The false positive rate is computed at a frame level:

$$\text{False positive rate} = \frac{\text{Number of falsely detected frames}}{\text{Total number of non-gesture frames}}$$

A frame is tagged as falsely detected if the head gesture recognizer triggers and if this frame is outside any time window of a ground truth head gesture. The denominator is the total of frames outside any time window.

During our experiments, the detection threshold for head nod recognition was set to 0 which represents for the vision-based system an average false positive rate of 0.058 and a recognition performance of 85.3%. For the same false positive, the context-based approach recognized on average 91.0% of the head nods. A paired t-test analysis over all tested subject returns a one-tail p-value of 0.070. Figure 6 shows that adding contextual information to the recognition framework does reduce significantly the number of false positives.

9. CONCLUSION AND FUTURE WORK

We developed vision-based head gesture interface components to allow simple user interface interactions without disrupting keyboard and mouse focus for a primary activity. We explored two prototype perceptual interface components which use detected head gestures for dialog box confirmation and document browsing, respectively. Tracking was performed using stereo-based alignment, with gesture recognition performed using a SVM-based classifier. A context learning component exploited interface state to improve performance. User studies with prototype recognition components indicated quantitative and qualitative benefits of gesture-based confirmation over conventional alternatives, but did not show support for the prototype document browsing interface. As future work, we plan to experiment with a richer set of contextual cues including those based on eye gaze, and to study new gesture-based interactions like slide show navigation.

10. REFERENCES

[1] C. Breazeal and L. Aryananda. Recognizing affective intent in robot directed speech. *Autonomous Robots*, 12(1):83–104, 2002.

[2] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjalmsson, and H. Yan. Embodiment in conversational interfaces: Rea. In *Proceedings of the CHI'99 Conference*, pages 520–527, Pittsburgh, PA, 1999.

[3] J. Cassell, T. Bickmore, H. Vilhjalmsson, and H. Yan. A relational agent: A model and implementation of building user trust. In *Proceedings of the CHI'01 Conference*, pages 396–403, Seattle, WA, 2001.

[4] H. Clark and E. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

[5] C. J. Cohen, G. J. Beach, and G. Foulk. A basic hand gesture control system for pc applications. In *Proceedings. 30th Applied Imagery Pattern Recognition Workshop (AIPR'01)*, pages 74–79, 2001.

[6] J. Davis and S. Vaks. A perceptual user interface for recognizing head gesture acknowledgements. In *ACM Workshop on Perceptual User Interfaces*, pages 15–16, November 2001.

[7] S. Fujie, Y. Ejiri, K. Nakajima, Y. Matsusaka, and T. Kobayashi. A conversation robot using head gesture recognition as para-linguistic information. In *Proceedings of 13th IEEE International Workshop on Robot and Human Communication, RO-MAN 2004*, pages 159–164, September 2004.

[8] R. Jacob. *Eye tracking in advanced interface design*, pages 258–288. Oxford University Press, 1995.

[9] A. Kapoor and R. Picard. A real-time head nod and shake detector. In *Proceedings from the Workshop on Perspective User Interfaces*, November 2001.

[10] S. Kawato and J. Ohya. Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes. In *Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 40–45, 2000.

[11] R. Kjeldsen. Head gestures for computer control. In *Proc. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 62–67, 2001.

[12] S. Lenman, L. Bretzer, and B. Thuresson. Computer vision based hand gesture interfaces for human-computer interaction. Technical Report CID-172, Center for User Oriented IT Design, June 2002.

[13] L.-P. Morency, A. Rahimi, N. Checka, and T. Darrell. Fast stereo-based head tracking for interactive environment. In *Proceedings of the Int. Conference on Automatic Face and Gesture Recognition*, pages 375–380, 2002.

[14] L.-P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance model. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 803–810, 2003.

[15] L.-P. Morency, C. Sidner, C. Lee, and T. Darrell. Contextual recognition of head gestures. In *Proceedings of the International Conference on Multi-modal Interfaces*, October 2005.

[16] Nakano, Reinstein, Stocky, and J. Cassell. Towards a model of face-to-face grounding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003.

[17] Rich, Sidner, and N. Lesh. Collagen: Applying collaborative discourse theory to human–computer interaction. *AI Magazine, Special Issue on Intelligent User Interfaces*, 22(4):15–25, 2001.

[18] Sidner, Kidd, Lee, and N. Lesh. Where to look: A study of human–robot engagement. In *Proceedings of Intelligent User Interfaces*, Portugal, 2004.

[19] Sidner, Lee, and N. Lesh. Engagement when looking: Behaviors for robots when collaborating with people. In *Diabruck: Proceedings of the 7th workshop on the Semantic and Pragmatics of Dialogue*, pages 123–130, University of Saarland, 2003. I. Kruiff-Korbayova and C. Kosny (eds.).

[20] K. Toyama. Look,ma - no hands!hands-free cursor control with real-time 3D face tracking. In *PUI98*, 1998.

[21] Wikipedia. *Wikipedia encyclopedia*. http://en.wikipedia.org/wiki/Dialog_box.

[22] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *CHI99*, pages 246–253, 1999.