Available online at www.sciencedirect.com
ScienceDirect
Artificial Intelligence 171 (2007) 568–585
Artificial Intelligence
www.elsevier.com/locate/artint

Head gestures for perceptual interfaces: The role of context in improving recognition

Louis-Philippe Morency [a,*], Candace Sidner [b], Christopher Lee [c], Trevor Darrell [a]

[a] *MIT CSAIL, Cambridge, MA 02139, USA*
[b] *BAE Systems AIT, Burlington, MA 01803, USA*
[c] *Boston Dynamics, Waltham, MA 02139, USA*

Received 2 June 2006; received in revised form 16 March 2007; accepted 9 April 2007

Available online 19 April 2007

Abstract

Head pose and gesture offer several conversational grounding cues and are used extensively in face-to-face interaction among people. To accurately recognize visual feedback, humans often use contextual knowledge from previous and current events to anticipate when feedback is most likely to occur. In this paper we describe how contextual information can be used to predict visual feedback and improve recognition of head gestures in human–computer interfaces. Lexical, prosodic, timing, and gesture features can be used to predict a user's visual feedback during conversational dialog with a robotic or virtual agent. In non-conversational interfaces, context features based on user–interface system events can improve detection of head gestures for dialog box confirmation or document browsing. Our user study with prototype gesture-based components indicate quantitative and qualitative benefits of gesture-based confirmation over conventional alternatives. Using a discriminative approach to contextual prediction and multimodal integration, performance of head gesture detection was improved with context features even when the topic of the test set was significantly different than the training set.

Keywords: Visual feedback; Head gesture recognition; Contextual information; Context-based recognition

1. Introduction

During face-to-face conversation, people use visual feedback to communicate relevant information and to synchronize rhythm between participants. When people interact naturally with each other, it is common to see indications of acknowledgment, agreement, or disinterest given with a simple head gesture. Nonverbal feedback includes head nodding and its use (i.e., interpretation) for visual grounding, turn-taking and answering yes/no questions. When recognizing and interpreting such visual feedback, people use more than their visual perception: knowledge about the current topic and expectations from previous utterances help guide recognition of nonverbal cues. Our goal is to

* Corresponding author.
E-mail addresses: lmorency@csail.mit.edu (L.-P. Morency), candy.sidner@baesystems.com (C. Sidner), clee@bostondynamic.com (C. Lee), trevor@csail.mit.edu (T. Darrell).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.04.003

equip computer interfaces with the ability to similarly perceive visual feedback gestures, and to exploit contextual information from the current interaction state when performing visual feedback recognition.

Recent advances in computer vision have led to efficient head pose tracking systems, which can return the position and orientation of a user's head through automatic passive observation, as well as methods for recognition of head gestures using discriminatively trained statistical classifiers. We show how detected head gestures can be used for visual feedback both in a conversational dialog interaction and in interaction with a traditional windows-based graphical user interface. In both cases the use of interaction context proves critical to system performance.

When interacting with a computer in a conversational setting, dialog state can provide useful context for recognition. In the last decade, many embodied conversational agents (ECAs) have been developed for face-to-face interaction, using both physical robots and virtual avatars. A key component of these systems is the dialog manager, usually consisting of a history of the past events, current discourse moves, and an agenda of intended or likely actions. The dialog manager uses contextual information to decide which verbal or nonverbal action the agent should perform next (i.e., context-based synthesis). Contextual information has proven useful for aiding speech recognition: in [17], a speech recognizer's grammar changes dynamically depending on the agent's previous action or utterance. In a similar fashion, we have developed a context-based visual recognition module that builds upon the contextual information available in the dialog manager to improve performance of visual feedback recognition (Fig. 1).

In a non-conversational interface, head gestures can also be useful for interacting with interface elements. A perceptive interface sensitive to head gesture and its meaning can lead to more natural notification and navigation interactions. 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. We explore two types of head gesture-based window 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.

We present a prediction framework for incorporating context with vision-based head gesture recognition. Contextual features are derived from the utterances of an ECA or the event state of a traditional user interface. Fig. 2 presents our framework for context-based gesture recognition. Our framework allows us to predict, for example, that in certain contexts a glance is not likely whereas a head shake or nod is (as in Fig. 1), or that a head nod is not likely and a head nod misperceived by the vision system can be ignored. The use of dialog or interface context for visual gesture recognition has, to our knowledge, not been explored before for conversational or windows-based interaction.

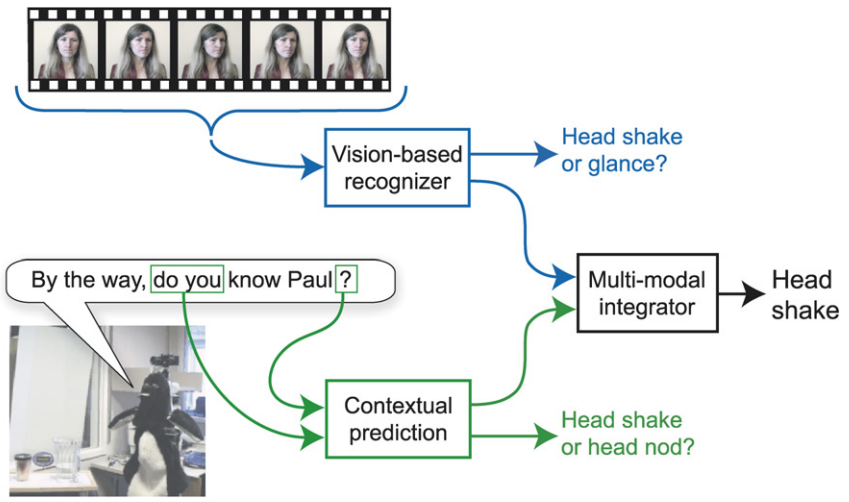

Fig. 1. Contextual recognition of head gestures during face-to-face interaction with a conversational robot. In this scenario, contextual information from the robot's spoken utterance helps disambiguating the listener's visual gesture.

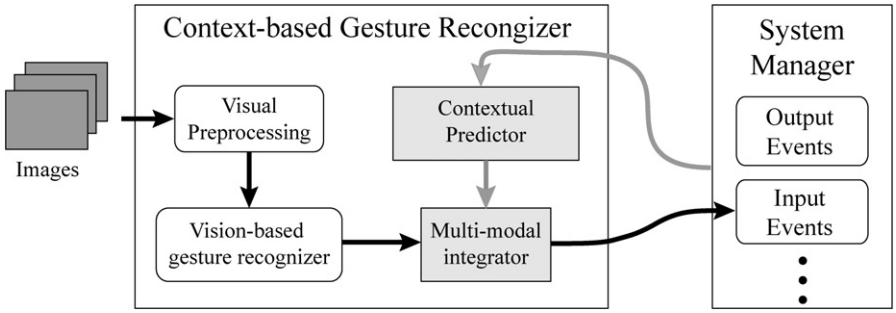

Fig. 2. Framework for context-based gesture recognition. The contextual predictor translates contextual features into a likelihood measure, similar to the visual recognizer output. The multi-modal integrator fuses these visual and contextual likelihood measures. The system manager is a generalization of the dialog manager (conversational interactions) and the window manager (window system interactions).

In the following sections we describe the contextual information available in conversational dialog systems and traditional window interfaces, our approach to context-based gesture recognition based on a discriminative classifier cascade, and our experiments in each domain.

2. Related work

There has been considerable work on gestures with ECAs. Bickmore and Cassell developed an ECA that exhibited many gestural capabilities to accompany spoken conversation and could interpret spoken utterances from human users [2]. Sidner et al. have investigated how people interact with a humanoid robot [25]. They found that more than half their participants naturally nodded at the robot's conversational contributions even though the robot could not interpret head nods. 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 ECA [23]; they incorporated their results in an ECA that updated its dialog state. Numerous other ECAs (e.g. [4,32]) are exploring aspects of gestural behavior in human-ECA interactions. Physically embodied ECAs—for example, ARMAR II [9,10] and Leo [3]—have also begun to incorporate the ability to perform articulated body tracking and recognize human gestures. Matsusaka et al. uses head pose to determine who is speaking in three party conversations [19].

Several authors have proposed face tracking for pointer or scrolling control and have reported successful user studies [16,31]. In contrast to eye gaze [34], 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 for human–computer interaction. Lenman et al. explored the use of pie- and marking menus in hand gesture-based interaction [18]. Cohen et al. studied the issues involved in controlling computer applications via hand gestures composed of both static and dynamic symbols [7].

Head pose and gesture offer several key conversational grounding cues and are used extensively in face-to-face interaction among people. Stiefelhagen developed several systems for tracking face pose in meeting rooms and has shown that face pose is very useful for predicting turn-taking [28]. Takemae et al. also examined face pose in conversation and showed that if tracked accurately, face pose is useful in creating a video summary of a meeting [29]. Siracusa et al. developed a system that uses head pose tracking to interpret who was talking to who in conversational setting [27]. The position and orientation of the head can be used to estimate head gaze which is a good estimate of a person's attention.

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 [14]. Kawato and Ohya suggested a technique for head gesture recognition using between eye templates [15]. When compared with eye gaze, head gaze can be more accurate

[1] Involuntary microsaccades are known to limit the accuracy of eye-gaze based tracking [12].

when dealing with low resolution images and can be estimated over a larger range than eye gaze [21]. Fujie et al. also used HMMs to perform head nod recognition [11]. In their paper, they combined head gesture detection with prosodic low-level features computed from Japanese spoken utterances to determine strongly positive, weak positive and negative responses to yes/no type utterances.

Context has been previously used in computer vision to disambiguate recognition of individual objects given the current overall scene category [30]. While some systems [3,23] have incorporated tracking of fine motion actions or visual gesture, none have included top–down dialog context as part of the visual recognition process. In contrast to the idea of fusing multiple modalities from the human participant to improve recognition (e.g., Kaiser et al. work on multi-modal interaction in augmented and virtual reality [13]), our approach takes its contextual information directly from the interactive agent. To our knowledge no previous work has explored the use of dialog or window manager state as context for visual recognition of interaction gestures.

3. Context in conversational interaction

Reliable recognition of nodding gestures is essential for face-to-face conversation. Human speakers, even when speaking to an ECA, nod without accompanying phrases, such as "yes, uh-huh" or "ok" [1,26]. In particular, knowledge of their own dialog contribution allows people to predict that a gesture (e.g., a nod) would be interpreted as meaningful (e.g., as an acknowledgment) if it occurred at the current time in the dialog. When a gesture occurs, the recognition and meaning of the gesture is enhanced due to this dialog context prediction. Thus recognition is enabled by the meaningfulness of a gesture in dialog.

However, because the contextual dialog information is subject to variability when modeled by a computational entity, it cannot be taken as ground truth. Instead features from the dialog that predict a certain meaning (e.g., acknowledgment) are also subject to recognition prediction. Hence in the work reported here, recognition of dialog features and recognition of visual features are interdependent processes.

For reliable recognition of head gestures, people use knowledge about the current dialog during face-to-face conversational interactions to anticipate visual feedback from their interlocutor. Our goal is to equip computer interfaces with the ability to similarly perceive visual feedback gestures. As depicted in Fig. 1, knowledge of an ECA's spoken utterance can help predict which visual feedback is most likely.

We can use a conversational agent's knowledge about the current dialog to improve recognition of visual feedback (i.e., head gestures). Fig. 3 shows a simplified architecture which captures aspects common to several different systems [23,24]. The dialog manager merges information from the input devices with the history and the discourse model. The dialog manager contains two main sub-components, an agenda and a history: the agenda keeps a list of all the possible actions the agent and the user (i.e., human participant) can do next. This list is updated by the dialog manager based on its discourse model (prior knowledge) and on the history. Dialog managers generally exploit contextual information to produce output for the speech and gesture synthesizer, and we can use similar cues to predict when visual feedback gestures from the user will be likely.

We extract information from the dialog manager rather than directly access internal ECA state. Our proposed method extracts contextual features from the messages sent to the audio and gesture synthesizers. This strategy allows us to extract dialog context without any knowledge of the internal representation and so, our method can be applied to most ECA architectures.

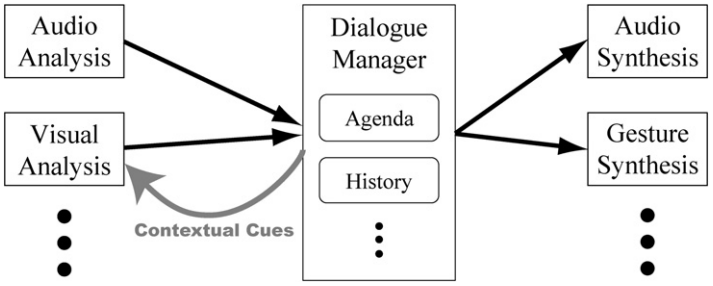

Fig. 3. Simplified architecture for embodied conversational agent. Our method integrates contextual information from the dialog manager inside the visual analysis module.

We highlight four types of contextual features easily available from the dialog manager: lexical features, prosody and punctuation features, timing information, and gesture displays.

Lexical features Lexical features are computed from the words said by the embodied agent. By analyzing the word content of the current or next utterance, one should be able to anticipate and distinguish certain visual feedback gestures. For example, if the current spoken utterance started with "Do you", the interlocutor will most likely answer using affirmation or negation. In this case, visual feedback in the form of a head nod or a head shake is likely and would be interpreted accordingly. On the other hand, if the current spoken utterance started with "What", then it is less likely to see the listener head shake or head nod—other visual feedback gestures (e.g., pointing) are more likely.

Prosody and punctuation Prosody can also be an important cue to predict gesture displays. We use punctuation features output by the dialog system as a proxy for prosody cues. Punctuation features modify how the text-to-speech engine will pronounce an utterance. Punctuation features can be seen as a substitute for more complex prosodic processing that are not yet available from most speech synthesizers. A comma in the middle of a sentence will produce a short pause, which will most likely trigger some feedback from the listener. A question mark at the end of the sentence represents a question that should be answered by the listener. When merged with lexical features, the punctuation features can help recognize situations (e.g., yes/no questions) where the listener will most likely use head gestures to answer.

Timing Timing is an important part of spoken language and information about when a specific word is spoken or when a sentence ends is critical. This information can aid the ECA to anticipate visual grounding feedback. People naturally give visual feedback (e.g., head nods) during pauses of the speaker as well as just before the pause occurs. In natural language processing (NLP), lexical and syntactic features are predominant but for face-to-face interaction with an ECA, timing is also an important feature.

Gesture display Synthesized gestures are a key capability of ECAs and they can also be leveraged as context cues for gesture interpretation. As described in [5], visual feedback synthesis can improve the engagement of the user with the ECA. The gestures expressed by the ECA influence the type of visual feedback from the human participant. For example, if the agent makes a deictic pointing gesture, the user is more likely to look at the location that the ECA is pointing to.

The dialog manager sends the next spoken utterance, a time stamp and an approximated duration to the visual analysis module. The next spoken utterance contains the words, punctuation, and gesture tags used to generate the ECA's actions. The utterance information is processed to extract the lexical, punctuation, timing, and gesture features. Approximate duration of utterances is generally computed by speech synthesizers and made available in the synthesizer API.

4. Context in window system interaction

We investigate the use of context-based visual feedback recognition to interact with conventional window system components. Dialog boxes are well-known 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 [33]. We 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 [6]. Grounding is also present during interactions with embodied conversational agents, and human participants naturally head nod as a non-verbal feedback for grounding [23]. 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 multiple 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 propose 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.

With a window system interface, there are several sources of potential errors with a gesture-based recognition system. We use contextual features that distinguish between visually confounded states and reduce false positives that happen during interaction with conventional input devices. Contextual features should be easily computed using pre-existing information in the user interface system.

For traditional, windows-based human–computer interfaces, interaction context is defined by the event state of the user interface system. We highlight two types of contextual features easily available from the window manager: input device events and display events.

Input device events Recent events from a conventional input device like keyboard or mouse can help to distinguish voluntary head gestures. 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 difficulty, especially for users who are not aware of the tracking system.

Display events Knowledge from what is displayed on screen can help predicting when user's input is more likely. A simple example of such a contextual feature is the time since a dialog box was displayed. This contextual feature can help head gesture recognition because user's input is most likely after such an event but also because people answer a dialog box usually after reading its content (∼2.5 second average delay in our experiments). So the time since the display event can be as important as the event itself.

These contextual features can be easily computed by listening to the input and output events sent inside the message dispatching loop of the application or operating system (see Fig. 2).

5. Context-based gesture recognition

We use a two-stage discriminative classification scheme to integrate interaction context with visual observations and detect gestures. A two-stage scheme allows us the freedom to train context prediction and visual gesture recognition components separately, potentially using corpora collected at different times. Fig. 2 depicts our complete system.

In the contextual predictor, we learn a measure of the likelihood of certain visual gestures given the current contextual feature values using a multi-class Support Vector Machine (SVM) [8]. 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:

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

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

In the multi-modal integrator, we merge context-based predictions with observations from a vision-based head gesture recognizer. We adopted a late fusion approach because data acquisition for the contextual predictor is greatly simplified with this approach, and initial experiments suggested performance was equivalent to an early, single-stage integration scheme. Most recorded interactions between human participants and conversational robots do not include estimated head position; a late fusion framework gives us the opportunity to train the contextual predictor on a larger data set of linguistic features.

Our integration component takes as input the margins from the contextual predictor described earlier in this section and the visual observations from the vision-based head gesture recognizer, and recognizes whether a head gesture has been expressed by the human participant using a second multi-class SVM. The output from the integrator is further sent to the dialog manager or the window manager so it can be used to decide the next action of the ECA or to trigger the perceptive window interface.

Since the multi-class SVM returns a measurement at each frame, to recognize a gesture over a subsequence of frames, a simple smoothing technique is applied to all measurements inside the subsequence window. If the integrator triggers at least k times inside the time window, the subsequence is tagged as a gesture. The smoothing parameter k is set empirically to optimize the baseline technique: the vision-only technique. As future work, we are planning to use a sequence classifier (e.g., Conditional Random Fields [20]) for integration instead of a multi-class SVM, so that the gesture dynamic is automatically modeled, and no smoothing step needed.

In the vision-based gesture recognizer, we compute likelihood measurements of head gestures using a two-step process: 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 by [22]. We found this tracker was able to track subtle movements of the head for a long 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 the second step of the vision-based gesture recognition (before integration of context features), 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 18 Hz). We transform the velocity signal into a frequency-based feature by applying a windowed Fast-Fourier Transform (FFT) to each dimension of the velocity independently using a 32-sample, 1-second window. The multi-class SVM was trained on this input using an RBF kernel.

6. Conversational experiments

The following experiment demonstrates how contextual features inferred from an agent's multi-modal dialog can improve head nod and head shake recognition. We compare the performance of the vision-only recognizer with context-only prediction and with multi-modal integration.

6.1. Experimental setup

For this experiment, a first data set was used to train the contextual predictor and the multi-modal integrator, while a second data set with a different topic was used to evaluate head gesture recognition performance. In the training data set, a robot interacted with a participant by demonstrating its own abilities and characteristics. This data set, called Self, contains 7 interactions. The test data set, called iGlass, consists of nine interactions of the robot describing an invention called iGlassware ($\sim$340 utterances).

During each interaction, we recorded the results of the vision-based head gesture recognizer (described in Section 5) as well as the contextual cues (spoken utterances with start time and duration) from the dialog manager. These contextual cues were later automatically processed to create contextual features necessary for the contextual predictor. The details of the contextual feature computation is discussed in the following subsection.

For both data sets, human participants were video recorded while interacting with the robot (see Fig. 4). The vision-based head tracking and head gesture recognition were run online at approximately 18 Hz. The robot's conversational model, based on COLLAGEN [24], determined the next activity on the agenda using a predefined set of engagement

Fig. 4. Mel, an interactive robot, can present the iGlassware demo (table and copper cup on its right) or talk about its own dialog and sensorimotor abilities.

rules, originally based on results from a human–human interaction study [25]. Each interaction lasted between 2 and 5 minutes. For ground truth, we hand labeled each video sequence to determine exactly when the participant nodded or shook his/her head. A total of 274 head nods and 14 head shakes were naturally performed by the participants while interacting with the robot.

6.2. Contextual features

The robot's spoken utterances were automatically processed to compute contextual features. We use four types of contextual features easily available from the dialog manager: lexical features, prosody and punctuation features, timing information, and gesture displays. In our implementation, the lexical feature relies on extracted bigrams (pairs of words that occur in close proximity to each other, and in particular order) since they can efficiently be computed given the transcript of the utterance.

While a range of bigrams may be relevant to gesture context prediction, we currently focus on the single phrase "do you", as we observed it was an efficient predictor of a yes/no question in many of our training dialogs. Other bigram features will probably be useful as well (for example, "have you, will you, did you"), and could be learned using a feature selection algorithm from a set of candidate bigram features.

We extract bigrams from the utterance and set the following binary feature:

$$f_{\text{"do you"}} = \begin{cases} 1 & \text{if bigram "do you" is present} \\ 0 & \text{if bigram "do you" is not present} \end{cases}$$

The punctuation feature and gesture feature are coded similarly:

$$f_{?} = \begin{cases} 1 & \text{if the sentence ends with "?"} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{\text{look\_left}} = \begin{cases} 1 & \text{if a "look left" gesture happened during the utterance} \\ 0 & \text{otherwise} \end{cases}$$

The timing contextual feature f_t represents proximity to the end of the utterance. The intuition is that verbal and non-verbal feedback are most likely at pauses and also just before the pause occurs. This feature can easily be computed given only two values: t_0, the utterance start-time, and δ_t, the estimated duration of the utterance. Given these two values for the current utterance, we can estimate f_t at time t using:

$$f_t(t) = \begin{cases} 1 - \left|\frac{t-t_0}{\delta_t}\right| & \text{if } t \leqslant t_0 + \delta_t \\ 0 & \text{if } t > t_0 + \delta_t \end{cases}$$

We selected our features so that they are topic independent. This means that we should be able to learn how to predict head gesture from a small set of interactions and then use this knowledge on a new set of interactions with a different

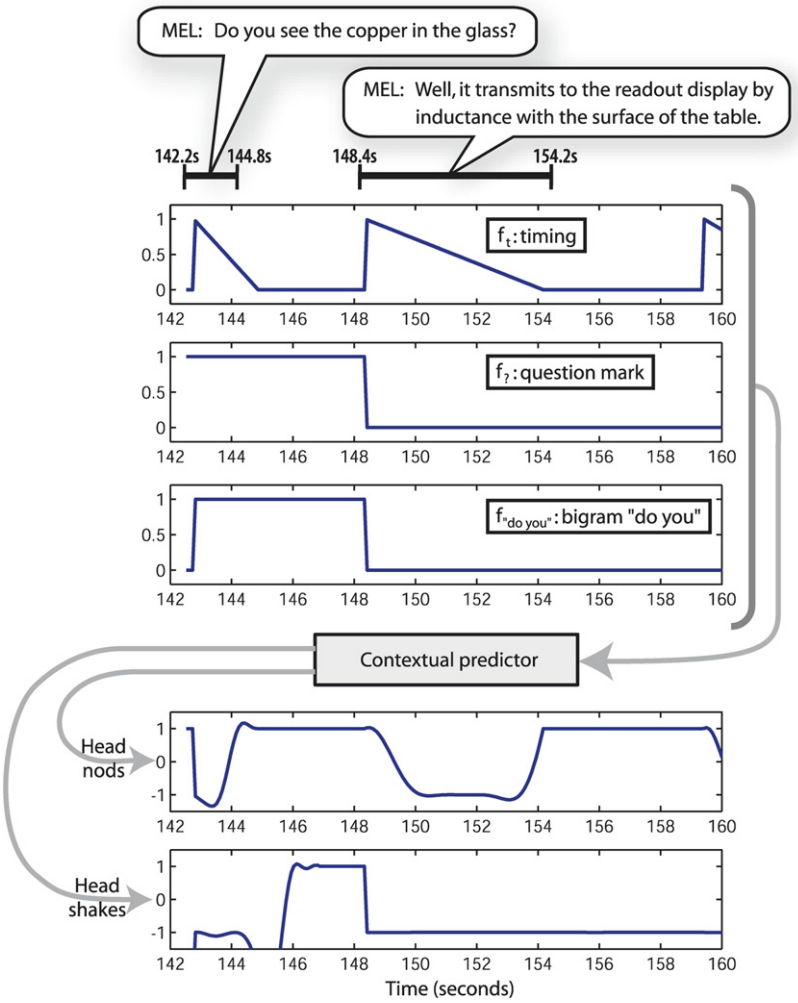

Fig. 5. Prediction of head nods and head shakes based on 3 contextual features: (1) distance to end-of-utterance when ECA is speaking, (2) type of utterance and (3) lexical bigram feature. We can see that the contextual predictor learned that head nods should happen near or at the end of an utterance or during a pause while head shakes are most likely at the end of a question.

topic discussed by the human participant and the ECA. However, different classes of dialogs might have different key features, and ultimately these should be learned using a feature selection algorithm (this is a topic of future work).

The contextual features are evaluated for every frame acquired by the visual analysis module (about 18 Hz). The lexical, punctuation and gesture features are evaluated based on the current spoken utterance. The effect of an utterance starts when it starts to be spoken and ends after the pause following the utterance. The top three graphs of Fig. 5 show how two sample utterances will be coded for the bigram "do you", the question mark and the timing feature.

A total of 236 utterances were processed to train the multi-class SVM used by our contextual predictor. Positive and negative samples were selected from the same data set based on manual transcription of head nods and head shakes. Test data was withheld during evaluation in all experiments in this paper (i.e. no subjects from the training data were used during testing).

Fig. 5 displays the output of each class of our contextual predictor for a sample dialog segment between the robot and a human participant. Positive margins represent a high likelihood for the gesture. It is noteworthy that the contextual predictor automatically learned that head nods are more likely to occur around the end of an utterance or during a pause, while head shakes are most likely to occur after the completion of an utterance. It also learned that head shakes are directly correlated with the type of utterance (a head shake will most likely follow a question), and

that head nods can happen at the end of a question (i.e., to represent an affirmative answer) and can also happen at the end of a normal statement (i.e., to ground the spoken utterance).

6.3. Performance

Our hypothesis was that the inclusion of contextual information within the head gesture recognizer would increase the number of recognized head nods while reducing the number of false detections. We tested three different configurations: (1) using the vision-only approach, (2) using only the contextual information as input (contextual predictor), and (3) combining the contextual information with the results of the visual approach (multi-modal integration).

Fig. 6 shows the head nod recognition results for a sample dialog. When only vision is used for recognition, the algorithm makes a mistake at approximately $t = 101$ s by detecting a false head nod; visual grounding is less likely during the middle of an utterance. By incorporating contextual information, our context-based gesture recognition algorithm is able to reduce the number of false positives.

Fig. 7 shows head nod detection results for all 9 subjects used during testing. The ROC curves present the detection performance for each recognition algorithm when varying the detection threshold. The area under the curve for each techniques are 0.9579 for the vision only, 0.8386 for the predictor and 0.9769 for the integrator. Pairwise comparisons

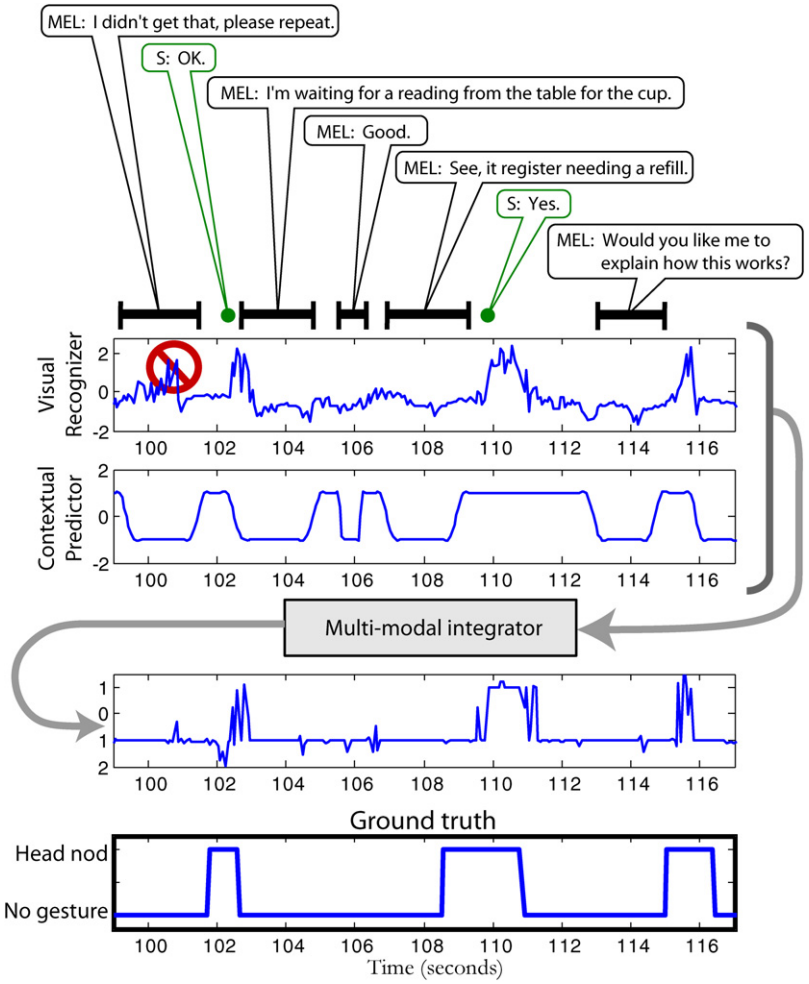

Fig. 6. Context-based head nod recognition results for a sample dialog. The last graph displays the ground truth. We can observe at around 101 seconds (circled and crossed in the top graph) that the contextual information attenuates the effect of the false positive detection from the visual recognizer.

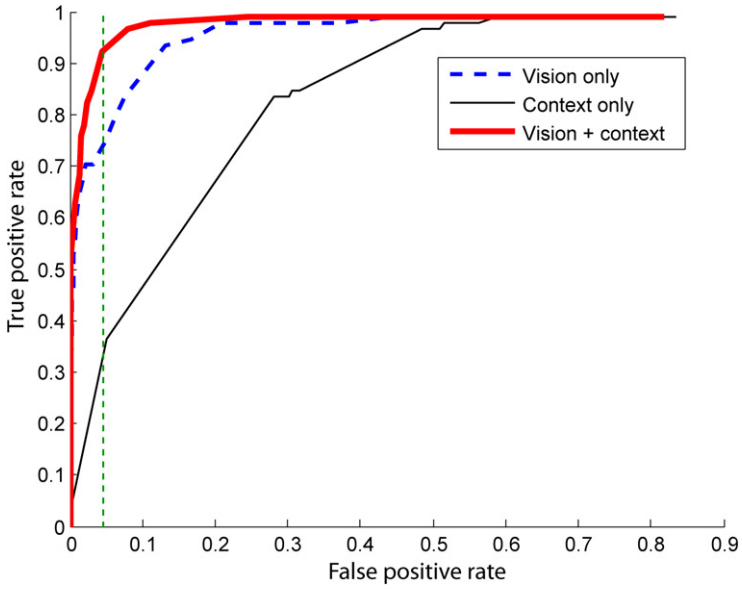

Fig. 7. Head nod recognition curves when varying the detection threshold. For a fixed false positive rate of 0.0409 (operating point), the context-based approach improves head nod recognition from 72.5% (vision only) to 90.4%.

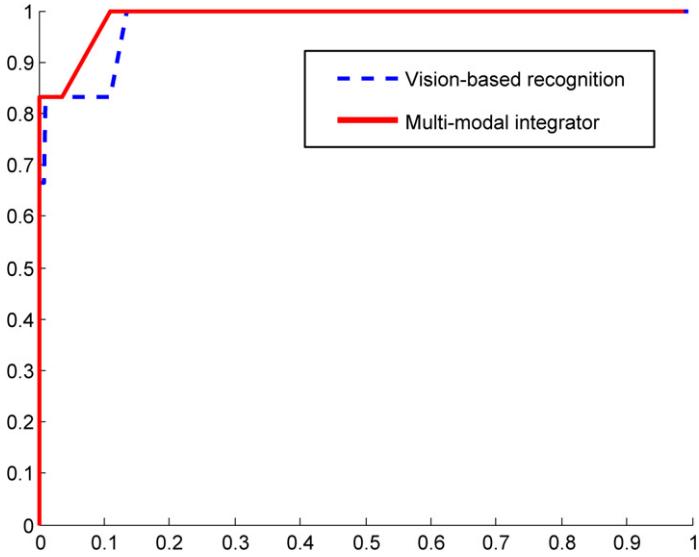

Fig. 8. Head shake recognition curves when varying the detection threshold.

over all tested subjects show a significant difference between all pairs, with $p = 0.0088$, $p = 0.0446$, and $p < 0.001$ for vision-predictor, vision-integrator, and predictor-integrator respectively.

The true positive rate is computed as the ratio between the number of correctly detected gestures and the total number of gestures. Similarly, the false positive rate is defined by the ratio of the number of falsely detected non-gestures and the total number of non-gestures. Since non-gesture sequences were much longer than gestures sequences, we subdivided the non-gestures sequences into subsequences so that the gesture and non-gesture sequences are of similar sizes. The window size was set to 0.33 seconds, the size of the smallest ground truth gesture. The iGlass dataset is comprised of 4272 non-gesture subsequences, 91 head nods and 6 head shakes. For this experiment, the smoothing parameter optimizing the vision-only approach (i.e. baseline approach) was $k = 4$.

During our experiments, the average false positive rate from the vision-based system was 0.0409 (i.e. operating point) and the recognition performance was 72.5%. For the same false positive rate, the multi-modal integrator recognized on average 90.4% of the head nods while the contextual predictor recognized only 29.2%. Using a standard analysis of variance (ANOVA) on all the subjects, results on the head nod detection task showed a significant difference among the means of the 3 methods of detection: $F = 25.41$, $p < 0.001$, $d = 0.688$. Pairwise comparisons show a significant difference between all pairs, with $p = 0.0027$, $p = 0.0183$, and $p < 0.001$ for vision-predictor, vision-integrator, and predictor-integrator respectively.

Fig. 8 shows head shake detection results for each recognition algorithm when varying the detection threshold. The areas under the curve for each techniques are 0.9780 for the vision only, 0.4961 for the predictor and 0.9872 for the integrator.

7. Window system experiments

In this section, we first describe a user study evaluating two gesture-based widgets described in Section 4: dialog box answering and document browsing. We then describe how we compute contextual features from the windows manager. Finally, we present the result from the user study as well as a comparison between the vision-only recognizer and the context-based recognizer.

7.1. Experimental setup

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 appeared at different times asking a question or stating new information.

The reading task was designed to replicate a situation where a person reads an informal text ($\sim$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 second 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.

During both tasks, dialog boxes appeared at random intervals and positions. These dialog boxes were designed to replicate the reminders sent by a calendar application (i.e., Microsoft Outlook) or alerts sent by an email client. Between 4 and 8 dialog boxes appeared during each experiment and participants were asked to answer each of them. Two types of dialog boxes were used: one "OK" button and two "Yes/No" buttons.

Reading and question answering 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 felt to interact with conventional input devices.

Between the first and second interaction, a short tutorial about head gestures for user interfaces was performed where participants practiced the new techniques for dialog box answering and document browsing as described in Section 4. 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,

Fig. 9. Window system experimental setup. A stereo camera is placed on top of the screen to track the head position and orientation.

mouse or head gestures. In contrast to the previous two phases, this phase provided an indication of which interaction technique or combination of technique was preferred.

This phase was also designed to compare the accuracy of the head recognizer with the judgment of a 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-subjects design, where each participant performed more than one interaction phase. A total of 19 people participated in our experiment. All participants were accustomed to using 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.

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 Fig. 9). This camera was connected to a laptop that ran the recognition system described in Section 5. 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.

7.2. Contextual features

For the non-conversational interface experiments, we exploit two features of interface state commonly available through a system event dispatch mechanism: dialog box display and mouse motion. The time from the most recent dialog box display or mouse motion or button press event is computed, and provided as input to a context prediction module. Such window event features can significantly reduce potential false positives that would otherwise result based on detection from the visual modality alone.

We defined two contextual features based on window system event state: f_d and f_m, defined as the time since a dialog box appeared and time since the last mouse event 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 Fig. 2). 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 18 Hz).

7.3. Results and discussion

To obtain a qualitative analysis of our user study, we looked at the choices each participant made during the third phase of the non-conversational experiment. During this part of the experiment, the participant was free to decide which input device to use. Fig. 10 shows how participants decided to answer dialog boxes and browse documents. We calculated average use over the whole group of 7 participants. 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 22.9% and 45.8% respectively.

Using a standard analysis of variance (ANOVA) on all 7 subjects who participated in the third phase, results on the dialog box answering widget showed a marginally significant difference among the means of the 3 input techniques: $p = 0.060$. Pairwise comparisons show a marginally 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: $p = 0.45$. Pairwise comparisons for the document browsing show no significant difference between all pairs, with $p = 0.362$, $p = 0.244$, and $p = 0.243$ for gesture–mouse, gesture–keyboard, and mouse–keyboard respectively.

We also measured qualitative results from the questionnaire. Fig. 11 shows how 19 participants scored each input device for efficiency and natural feeling when interacting with dialog boxes. The average scores for efficiency were

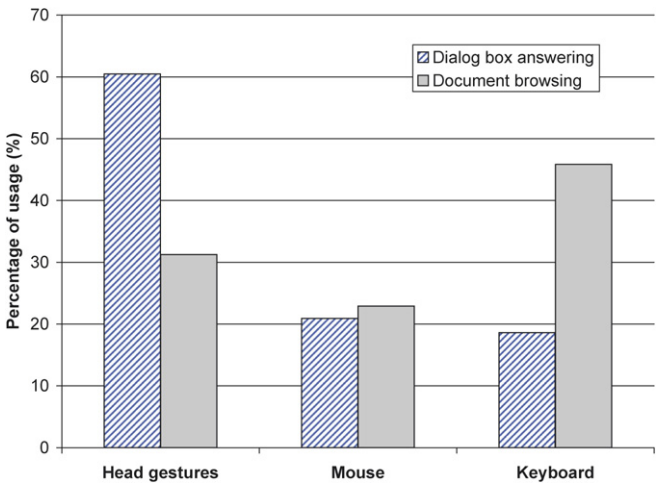

Fig. 10. Preferred choices for input technique during third phase of the experiment.

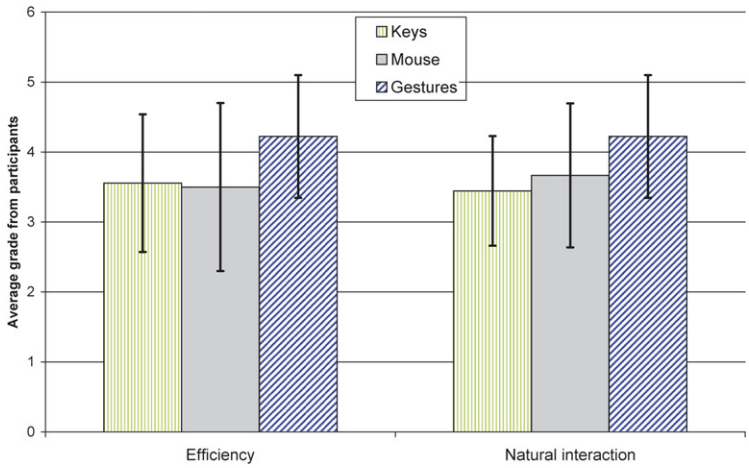

Fig. 11. Survey results for dialog box task. All 19 participants graded the naturalness and efficiency of interaction on a scale of 1 to 5, 5 meaning best.

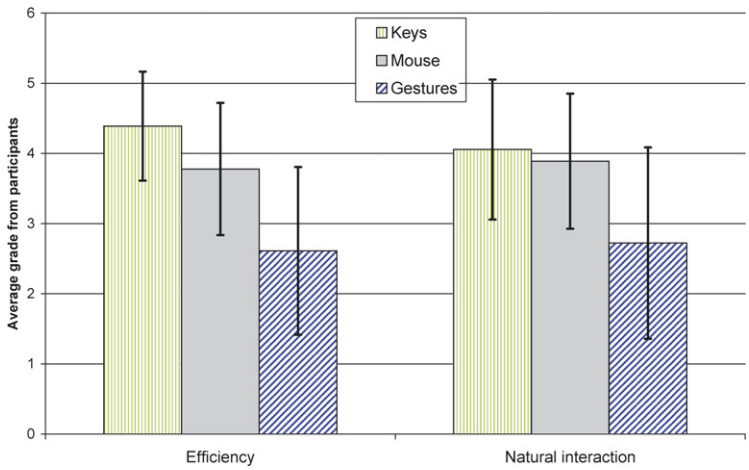

Fig. 12. Survey results for document browsing task. All 19 participants graded the naturalness and efficiency of interaction on a scale of 1 to 5, 5 meaning best.

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.

Fig. 12 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 Figs. 10 and 11 suggest that head gestures are perceived as a natural and efficient way to answer and acknowledge dialog boxes. Participants did not seem to appreciate head gestures as much 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 gesture could have been used to indicate 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 5, we manually annotated 12 interaction sequences for head gestures so that we have a ground truth. From this dataset of 79 minutes

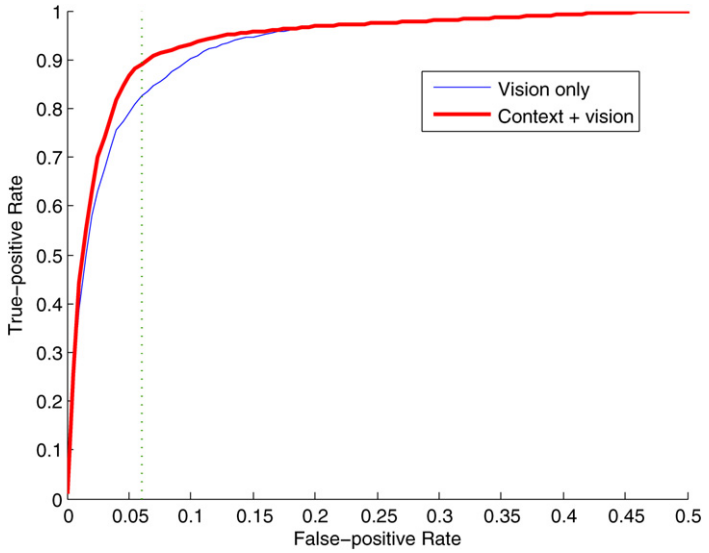

Fig. 13. Average ROC curves for head nods recognition. For a fixed false positive rate of 0.058 (operating point), the context-based approach improves head nod recognition from 80.8% (vision only) to 88.5%.

of interaction, 269 head nods and 121 head shakes were labeled as ground truth. Positive and negative samples were selected from this data set. The evaluation of the context-based gesture recognizer was done using a leave-one-out approach where eleven participants are used to train the contextual predictor and multi-modal integrator and one participant is kept for testing. Fig. 13 shows the average head nod detection results for the vision-only technique and the context-based recognizer. The ROC curves were computed using the same technique described in Section 6.3. For this experiment, the smoothing parameter optimizing the vision-only approach (i.e. baseline approach) was $k = 1$.

During our experiments, the average false positive rate from the vision-based system was 0.058 (i.e. operational point) and the recognition performance was 80.9%. For the same false positive rate, the context-based approach recognized on average 88.5% of the head nods. A paired t-test analysis over all tested subject returns a one-tail p-value of 0.0593. Fig. 13 shows that adding contextual information to the recognition framework does reduce significantly the number of false positives.

8. Conclusion and future work

Our results show that contextual information can improve user gesture recognition for interactions with embodied conversational agents and interactions with a window system. We presented a prediction framework that extracts knowledge from the spoken dialog of an embodied agent or from the user–interface system events to predict which head gesture is most likely. By using simple lexical, punctuation, gesture and timing context features, we were able to improve the recognition rate of the vision-only head gesture recognizer from 73% to 90% for head nods and from 83% to 98% for head shakes. Similar improvements were shown when performing context-based recognition during window system interactions. Our user study indicated quantitative and qualitative benefits of gesture-based confirmation over conventional alternatives for dialog boxes in GUI interfaces, 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, and to incorporate general feature selection to our prediction framework so that a wide range of potential context features can be considered and the optimal set determined from a training corpus.

Acknowledgements

The contributions of Sidner and Lee to this work were supported by Mitsubishi Electric Research Labs, Cambridge, MA 02139. The authors thank Charles Rich for his support in the use of Collagen in this effort.

References

[1] T. Bickmore, Towards the design of multimodal interfaces for handheld conversational characters, in: Proceedings of the CHI Extended Abstracts on Human factors in Computing Systems Conference, ACM Press, 2002, pp. 788–789.

[2] T. Bickmore, J. Cassell, Social dialogue with embodied conversational agents, in: J. van Kuppevelt, L. Dybkjaer, N. Bernsen (Eds.), Natural, Intelligent and Effective Interaction with Multimodal Dialogue Systems, Kluwer Academic, 2004.

[3] C. Breazeal, G. Hoffman, A. Lockerd, Teaching and working with robots as a collaboration, in: The Third International Conference on Autonomous Agents and Multi-Agent Systems AAMAS, 2004, ACM Press, July 2004, pp. 1028–1035.

[4] B. De Carolis, C. Pelachaud, I. Poggi, F. de Rosis, Behavior planning for a reflexive agent, in: Proceedings of the International Joint Conference on Artificial Intelligence, Seattle, Morgan Kaufmann Publishers, September 2001, pp. 1059–1064.

[5] J. Cassell, K.R. Thorisson, The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents, Applied Artificial Intelligence (1999) 519–538.

[6] H.H. Clark, E.F. Schaefer, Contributing to discourse, Cognitive Science 13 (1989) 259–294.

[7] C.J. Cohen, G.J. Beach, G. Foulk, A basic hand gesture control system for PC applications, in: Proceedings 30th Applied Imagery Pattern Recognition Workshop (AIPR'01), John Wiley & Sons, 2001, pp. 74–79.

[8] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[9] R. Dillman, R. Becher, P. Steinhaus, ARMAR II—a learning and cooperative multimodal humanoid robot system, International Journal of Humanoid Robotics 1 (1) (2004) 143–155.

[10] R. Dillman, M. Ehrenmann, P. Steinhaus, O. Rogalla, R. Zoellner, Human friendly programming of humanoid robots—the German Collaborative Research Center, in: The Third IARP International Workshop on Humanoid and Human-Friendly Robotics, Tsukuba Research Centre, Japan, December 2002.

[11] S. Fujie, Y. Ejiri, K. Nakajima, Y. Matsusaka, 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, September 2004, pp. 159–164.

[12] R.J.K. Jacob, Eye tracking in advanced interface design, in: Advanced Interface Design and Virtual Environments, Oxford University Press, 1995, pp. 258–288.

[13] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, S. Feiner, Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality, in: Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI 2003), Vancouver, BC, Canada, ACM Press, November 2003, pp. 12–19.

[14] A. Kapoor, R. Picard, A real-time head nod and shake detector, in: Proceedings from the Workshop on Perspective User Interfaces, ACM Press, November 2001, pp. 1–5.

[15] S. Kawato, 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, 2000, pp. 40–45.

[16] 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, 2001, pp. 62–67.

[17] O. Lemon, A. Gruenstein, S. Peters, Collaborative activities and multi-tasking in dialogue systems, Traitement Automatique des Langues (TAL) 43 (2) (2002) 131–154. Special issue on dialogue.

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

[19] Y. Matsusaka, T. Tojo, T. Kobayashi, Conversation robot participating in group conversation, IEICE Transaction of Information and System E86-D (1) (January 2003) 26–36.

[20] L.-P. Morency, A. Quattoni, T. Darrell, Latent-dynamic discriminative models for continuous gesture recognition, Technical Report MIT-CSAIL-TR-2007-002, CSAIL Technical report, January 2007.

[21] L.-P. Morency, A. Rahimi, N. Checka, T. Darrell, Fast stereo-based head tracking for interactive environment, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2002, pp. 375–380.

[22] L.-P. Morency, A. Rahimi, T. Darrell, Adaptive view-based appearance model, in: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 803–810.

[23] Y. Nakano, G. Reinstein, T. Stocky, 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, pp. 553–561.

[24] C. Rich, C. Sidner, N. Lesh, Collagen: Applying collaborative discourse theory to human–computer interaction, AI Magazine 22 (4) (2001) 15–25. Special issue on intelligent user interfaces.

[25] C. Sidner, C. Lee, C.D. Kidd, N. Lesh, C. Rich, Explorations in engagement for humans and robots, Artificial Intelligence 166 (1–2) (August 2005) 140–164.

[26] C. Sidner, C. Lee, L.-P. Morency, C. Forlines, The effect of head-nod recognition in human–robot conversation, in: Proceedings of the 1st Annual Conference on Human–Robot Interaction, ACM Press, March 2006, pp. 290–296.

[27] M. Siracusa, L.-P. Morency, K. Wilson, J. Fisher, T. Darrell, Haptics and biometrics: A multimodal approach for determining speaker location and focus, in: Proceedings of the 5th International Conference on Multimodal Interfaces, ACM Press, November 2003, pp. 77–80.

[28] R. Stiefelhagen, Tracking focus of attention in meetings, in: Proceedings of International Conference on Multimodal Interfaces, Pittsburgh, PA, ACM Press, 2002, pp. 273–280.

[29] Y. Takemae, K. Otsuka, N. Mukaua, Impact of video editing based on participants' gaze in multiparty conversation, in: Proceedings of the CHI Extended Abstracts on Human factors in Computing Systems Conference, ACM Press, April 2004, pp. 1333–1336.

[30] A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin, Context-based vision system for place and object recognition, in: IEEE Intl. Conference on Computer Vision (ICCV), Nice, France, October 2003, pp. 273–280.

[31] K. Toyama, Look, Ma—No Hands! Hands-free cursor control with real-time 3D face tracking, in: Proceedings from the Workshop on Perspective User Interfaces, ACM Press, 1998, pp. 49–54.
[32] D. Traum, J. Rickel, Embodied agents for multi-party dialogue in immersive virtual world, in: Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), July 2002, pp. 766–773.
[33] Wikipedia, Wikipedia encyclopedia, http://en.wikipedia.org/wiki/Dialog_box, 2002.
[34] S. Zhai, C. Morimoto, S. Ihde, Manual and gaze input cascaded (magic) pointing, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, ACM Press, 1999, pp. 246–253.