# 1

# Dialogue Context for Visual Feedback Recognition

**Louis-Philippe Morency[†], Candace Sidner[⋆] and Trevor Darrell[†]**

**[†]MIT Computer Sciences and Artificial Intelligence Laboratory**
**[⋆]BAE Systems AIT**

*Head pose and gesture offer several key conversational grounding cues and are used extensively in face-to-face interaction among people. When recognizing visual feedback, people use more than their visual perception. Knowledge about the current topic and expectations from previous utterances help guide our visual perception in recognizing nonverbal cues. In this chapter, we investigate how dialogue context from an embodied conversational agent (ECA) can improve visual recognition of user gestures. We present a recognition framework which (1) extracts contextual features from an ECA's dialogue manager, (2) computes a prediction of head nod and head shakes, and (3) integrates the contextual predictions with the visual observation of a vision-based head gesture recognizer. We found a subset of lexical, prosodic, timing and gesture features that are easily available in most ECA architectures and can be used to learn how to predict user feedback. Using a discriminative approach to contextual prediction and multi-modal integration, we were able to improve the performance of head gesture detection even when the topic of the test set was significantly different than the training set.*

*Engineering Approaches to Conversational Informatics*. Edited by Toyoaki Nishida

## 1.1 Introduction

During face-to-face conversation, people use visual feedback to communicate relevant information and to synchronize rhythm between participants. A good example of nonverbal feedback is head nodding and its use for visual grounding, turn-taking and answering yes/no questions. When recognizing visual feedback, people use more than their visual perception. Knowledge about the current topic and expectations from previous utterances help guide our visual perception in recognizing nonverbal cues. Our goal is to equip an embodied conversational agent (ECA) with the ability to use contextual information for performing visual feedback recognition much in the same way people do.

In the last decade, many ECAs have been developed for face-to-face interaction. A key component of these systems is the dialogue manager, which usually provides a history of the past events, the current state, and an agenda of future actions. The dialogue manager uses these contextual information sources to decide which verbal or nonverbal action the agent should perform next. This is called context-based synthesis.

Contextual information has proven useful for aiding speech recognition. In Lemon et al. (2002), the grammar of the speech recognizer dynamically changes depending on the agent's previous action or utterance. In a similar fashion, we want to develop a context-based visual recognition module that builds upon the contextual information available in the dialogue manager to improve performance.

The use of dialogue context for visual gesture recognition has, to our knowledge, not been explored before for conversational interaction. In this chapter we present a prediction framework for incorporating dialogue context with vision-based head gesture recognition. The contextual features are derived from the utterances of the ECA, which is readily available from the dialogue manager. We highlight four types of contextual features: lexical, prosodic, timing and gesture, and select a subset for our experiment that were topic independent. We use a discriminative approach to predict head nods and head shakes from a small set of recorded interactions. We then combine the contextual predictions with a vision-based recognition algorithm based on the frequency pattern of the user's head motion. Our context-based recognition 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 Figure 1.1), or that a head nod is not likely and a head nod misperceived by the vision system can be ignored.

The following section describes related work on gestures with ECAs. Section 1.3 present a general discussion on how context can be used for different type of visual feedback. Section 1.4 describes the contextual information available in most embodied agent architectures. Section 1.5 presents our general framework for incorporating contextual information with visual observations. Section 1.6 shows how we automatically extract a subset of this context to compute lexical, prosodic, timing and gesture features. Finally, in section 1.7, we describe context-based head gesture recognition experiments, performed on 16 video recordings of human participants interacting with a robot.

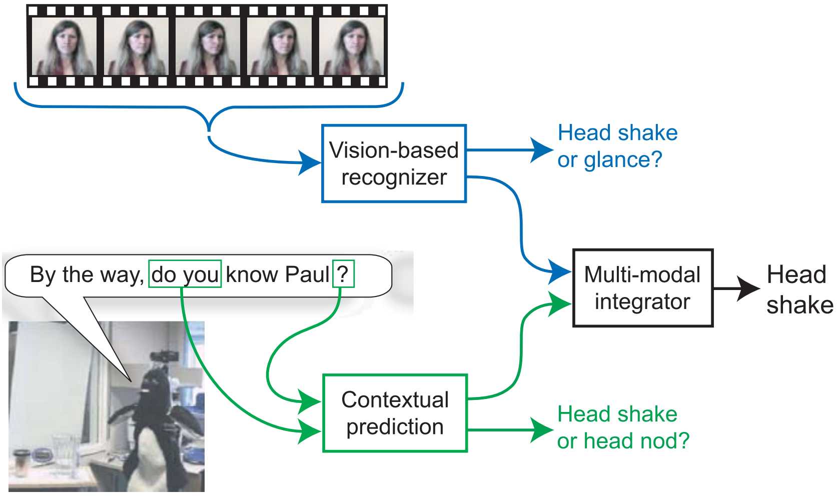

Figure 1.1 Contextual recognition of head gestures during face-to-face interaction with a conversational robot. In this scenario, contextual information from the robot's spoken utterance helps disambiguating the listener's visual gesture.

## 1.2 Background and Related Research

There has been considerable work on gestures with ECAs. Bickmore and Cassell (2004) developed an ECA that exhibited many gestural capabilities to accompany its spoken conversation and could interpret spoken utterances from human users. Sidner et al. (2005) have investigated how people interact with a humanoid robot. 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. (2003) 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. They incorporated their results in an ECA that updated its dialogue state. Numerous other ECAs (e.g. de Carolis et al. (2001); Traum and Rickel (2002)) are exploring aspects of gestural behavior in human-ECA interactions. Physically embodied ECAs—for example, ARMAR II (Dillman et al. 2004, 2002) and Leo (Breazeal et al. 2004)–have also begun to incorporate the ability to perform articulated body tracking and recognize human gestures.

Head pose and gesture offer several key conversational grounding cues and are used extensively in face-to-face interaction among people. Stiefelhagen (2002) developed several successful systems for tracking face pose in meeting rooms and has shown that face pose is very useful for predicting turn-taking. Takemae et al. (2004) also examined face pose in conversation and showed that if tracked accurately, face pose is useful in creating a video summary of a meeting. Siracusa et al. (2003) developed a kiosk front end that uses head pose tracking to interpret who was talking to who in conversational setting. The position and orientation of the head can be used to estimate head gaze which is a good estimate of a person's attention. 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 (Morency et al. 2002).

Kapoor and Picard (2001)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 . Fujie et al. (2004) also used HMMs to perform head nod recognition . 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.

Context has been previously used in computer vision to disambiguate recognition of individual objects given the current overall scene category (Torralba et al. 2003). While some systems (Breazeal et al. 2004; Nakano et al. 2003) have incorporated tracking of fine motion actions or visual gesture, none have included top-down dialogue context as part of the visual recognition process.

## 1.3 Context for Visual Feedback

During face-to-face interactions, people use knowledge about the current dialogue to anticipate visual feedback from their interlocutor. Following the definitions of Cassell and Thorisson (1999) for nonverbal feedback synthesis, we outline three categories for visual feedback analysis: (1) content-related feedback, (2) envelope feedback, and (3) emotional feedback. Contextual information can be used to improve recognition in each category.

**CONTENT-RELATED FEEDBACK** Content-related feedback is concerned with the content of the conversation. For example, a person uses head nods or pointing gestures to supplement or replace a spoken sentence. For this type of feedback, contextual information inferred from speech can greatly improve the performance of the visual recognition system. For instance, to know that the embodied agent just asked a yes/no question should indicate to the visual analysis module a high probability of a head nod or a head shake.

**ENVELOPE FEEDBACK** Grounding visual cues that occur during conversation fall into the category of envelope feedback. Such visual cues include eye gaze contact, head nods for visual grounding, and manual beat gestures. Envelope feedback cues accompany the dialogue of a conversation much in the same way audio cues like pitch, volume and tone envelope spoken words. Contextual information can improve the recognition of envelope visual feedback cues. For example, knowledge about when the embodied agent pauses can help to recognize visual feedback related to face-to-face grounding.

**EMOTIONAL FEEDBACK** Emotional feedback visual cues indicate the emotional state of a person. Facial expression is an emotional feedback cue used to show one of the 6 basic emotions (Ekman 1992) such as happiness or anger. For this kind of feedback, contextual information can be used to anticipate a person's facial expression. For example, a person smiles after receiving a compliment.

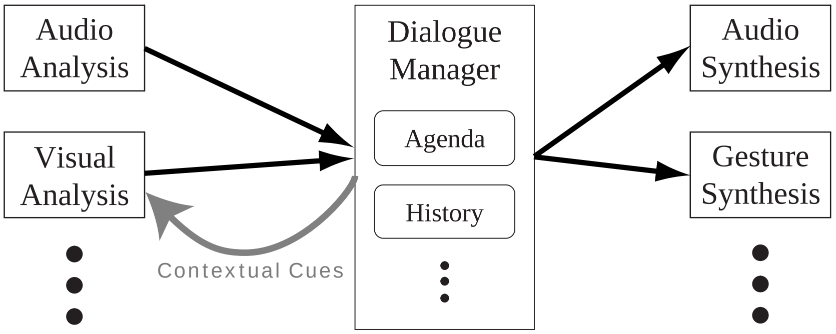

Figure 1.2 Simplified architecture for embodied conversational agent. Our method integrates contextual information from the dialogue manager inside the visual analysis module.

In general, our goal is to efficiently integrate dialogue context information from an embodied agent with a visual analysis module. We define a visual analysis module as a software component that can analyze images (or video sequences) and recognize visual feedback of a human participant during interaction with an embodied agent. The next step is to determine which information already exists in most ECA architectures.

## 1.4 Context from Dialogue Manager

Figure 1.2 is a general view of the architecture for an embodied conversational agent [1]. In this architecture, the dialogue manager contains two main subcomponents, 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 dialogue manager based on its discourse model (prior knowledge) and on the history. Some useful contextual cues can be estimated from the agenda:

- What will be the next spoken sentence of our embodied agent?
- Are we expecting some specific answers from the user?
- Is the user expected to look at some common space?

The history keeps a log of all the previous events that happened during the conversation. This information can be used to learn some interesting contextual cues:

- How did the user answer previous questions (speech or gesture)?
- Does the user seem to understand the last explanation?

[1] In our work we use the COLLAGEN conversation manager (Rich et al. 2001), but other dialogue managers provide these components as well.

Based on the history, we can build a prior model about the type of visual feedback shown by the user. Based on the agenda, we can predict the type of visual feedback that will be shown by the user.

The simplified architecture depicted in Figure 1.2 highlights the fact that the dialogue manager already processes contextual information in order to produce output for the speech and gesture synthesizer. The main idea is to use this existing information to predict when visual feedback gestures from the user are likely. Since the dialogue manager is already merging information from the input devices with the history and the discourse model, the output of the dialogue manager will contain useful contextual information.

We highlight four types of contextual features easily available in the dialogue manager:

**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 certain visual feedback. For example, if the current spoken utterance started with “Do you”, the interlocutor will most likely answer using affirmation or negation. In this case, it is also likely to see visual feedback like a head nod or a head shake. On the other hand, if the current spoken utterance started with “What”, then it’s unlikely to see the listener head shake or head nod–other visual feedback gestures (e.g., pointing) are more likely in this case.

**PROSODY AND PUNCTUATION** Prosody can also be an important cue to predict gesture displays. We use punctuation features output by the dialogue 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** Gesture synthesis is a key capability of ECAs and it can also be leveraged as a context cue for gesture interpretation. As described in Cassell and Thorisson (1999), 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 gesture, the user is more likely to look at the location that the ECA is pointing to.

The following section presents our framework for integrating contextual information with the visual observations and Section 1.6 describes how we can automatically extract lexical, prosodic, timing and gesture features from the dialogue system.

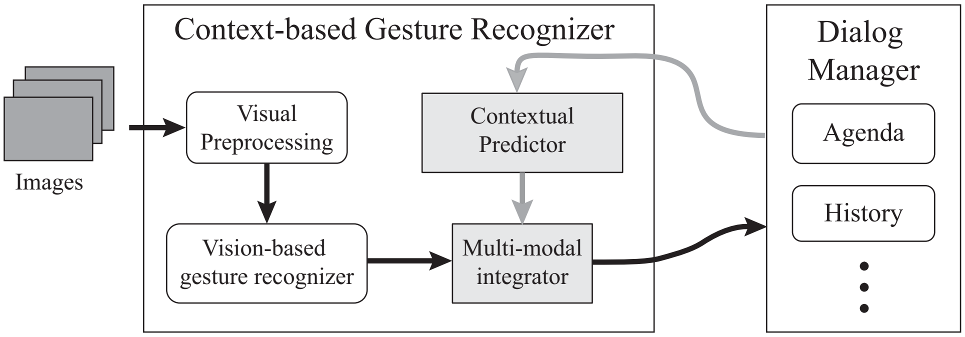

Figure 1.3 Framework for context-based gesture recognition. The contextual predictor translates contextual features into a likelihood measure, similar to the visual recognizer output. The multi-modal integrator fuses these visual and contextual likelihood measures. The system manager is a generalization of the dialogue manager (conversational interactions) and the window manager (window system interactions).

## 1.5 Framework for 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 the context predictor and vision-based recognizer independently, potentially using corpora collected at different times. Figure 1.3 depicts our complete framework.

Our context-based recognition framework has three main components: vision-based recognizer, contextual predictor and multi-modal integrator. In the vision-based gesture recognizer, we compute likelihood measurements of head gestures. In the contextual predictor, we learn a measure of the likelihood of certain visual gestures given the current contextual feature values. In the multi-modal integrator, we merge context-based predictions with observations from the vision-based recognizer to compute the final recognition estimates of the visual feedback.

The input of the contextual predictor is a feature vector $\mathbf{x}_j$ created from the concatenation of all contextual features at frame $j$. Each contextual value is a real value encoding a specific aspect of the current context. For example, one contextual feature can be a binary value (0 or 1) telling if the last spoken utterance contained a question mark. The details on how these contextual features are encoded are described in Section 1.6.

The contextual predictor should output a likelihood measurement at the same frame rate as the vision-based recognizer so the multi-modal integrator can merge both measurements. For this reason, feature vectors $\mathbf{x}_j$ should also be computed at every frame $j$ (even though the contextual features do not directly depend on the input images). One of the advantages of our late-fusion approach is that, if the contextual information and the feature vectors are temporarily unavailable, then the multi-modal integrator can recognize gestures using only measurements made by the vision-based recognizer. It is worth noting that the likelihood measurements can be a probabilities or a “confidence” measurement (as output by SVMs).

As shown in Figure 1.3, the vision-based gesture recognizer takes inputs from a visual pre-processing module. The main task of this module is to track head gaze using an adaptive view-based appearance model (see Morency et al. (2003) for details). This approach has the advantage of being 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.

The multi-modal integrator merges context-based predictions with observations from the vision-based recognizer. We adopt 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. The output from the integrator is further sent to the dialogue manager or the window manager so it can be used to decide the next action of the ECA.

In our experiment, we used Support Vector Machines (SVMs) to train the contextual predictor and the multi-modal integrator.

**SVM** Using Support Vector Machine (SVM), we estimate the likelihood measurement of a specific visual gesture using the margin of the feature vector $\mathbf{x}_j$. During training, the SVM finds a subset of feature vectors,called support vectors, that optimally define the boundary between labels. The margin $m(\mathbf{x}_j)$ of a feature vector $\mathbf{x}_j$ can be seen as the distance between the $\mathbf{x}_j$ and the boundary, inside a kernel space $\mathcal{K}$. The margin $m(\mathbf{x}_j)$ can easily be computed given the learned set of support vectors $\mathbf{x}_k$, the associated set of labels $y_k$ and weights $w_k$, and the bias $b$:

$$m(x) = \sum_{k=1}^{l} y_k w_k K(\mathbf{x}_k, \mathbf{x}_j) + b \tag{1.1}$$

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

$$K(\mathbf{x}_k, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_k - \mathbf{x}_j\|^2} \tag{1.2}$$

where $\gamma$ is the kernel smoothing parameter learned automatically using cross-validation on our training set.

## 1.6 Contextual Features

In this section we describe how contextual information is processed to compute feature vectors $\mathbf{x}_j$. In our framework, contextual information is inferred from the input and output events of the *dialogue manager* (see Figure 1.3 and Section 1.5).

We tested two approaches to send event information to the contextual predictor: (1) an active approach where the system manager is modified to send a copy of each relevant event to the contextual predictor, and (2) a passive approach where an external module listens at all the input and output events processed by the system manager and a copy of the relevant events is sent to the contextual predictor. In the contextual predictor, a pre-processing module receives the contextual events and outputs contextual features. Note that each event is accompanied by a timestamp and optionally a duration estimate.

In our framework, complex events are split into smaller sub-events to increase the expressiveness of our contextual features and to have a consistent event formatting. For example, the next spoken utterance event sent from the conversational manager will be split into sub-events including words, word pairs and punctuation elements. These sub-events will include the original event information (timestamp and duration) as well as the relative timing of the sub-event.

The computation of contextual features should be fast so that context-based recognition can happen online in real-time. We use two types of functions to encode contextual features from events: (1) binary functions and (2) ramp functions.

A contextual feature encoded using a binary function will return 1 when the event starts and 0 when it ends. This type of encoding supposes that we know the duration of the event or that we have a constant representing the average duration. It is well suited for contextual features that are less time sensitive. For example, the presence of the word pair “do you” in an utterance is a good indication of a yes/no question but the exact position of this word pair is not as relevant.

A ramp function is a simple way to encode the time since an event happened. We experimented with both negative slope (from 1 to 0) and positive slope (from 0 to 1) but did not see any significant difference between the two types of slopes. A ramp function is well suited for contextual features that are more time sensitive. For example, a grounding gesture such as a head nod is most likely to happen closer to the end of a sentence than the beginning.

The following sub-section gives specific examples of our general framework for contextual feature encoding applied to conversational interfaces.

### 1.6.1 Conversational Interfaces

The contextual predictor receives the avatar’s spoken utterance and automatically processes them to compute contextual features. Four types of contextual features are computed: lexical features, prosody and punctuation features, timing information, and gesture displays. In our implementation, the lexical feature relies on extracted word pair (two words that occur next to each other, and in a particular order) since they can efficiently be computed given the transcript of the utterance.

While a range of word pairs may be relevant to context-based recognition, we currently focus on the single phrase “do you”. We found this feature is an effective predictor of a yes/no question in many of our training dialogues. Other word pair features will probably be useful as well (for example, “have you, will you, did you”), and could be learned from a set of candidate word pair features using a feature selection algorithm.

We extract word pairs from the utterance and set the following binary feature:

$$f_{\text{“do you”}} = \begin{cases} 1 & \text{if word pair “do you” is present} \\ 0 & \text{if word pair “do you” is not present} \end{cases}$$

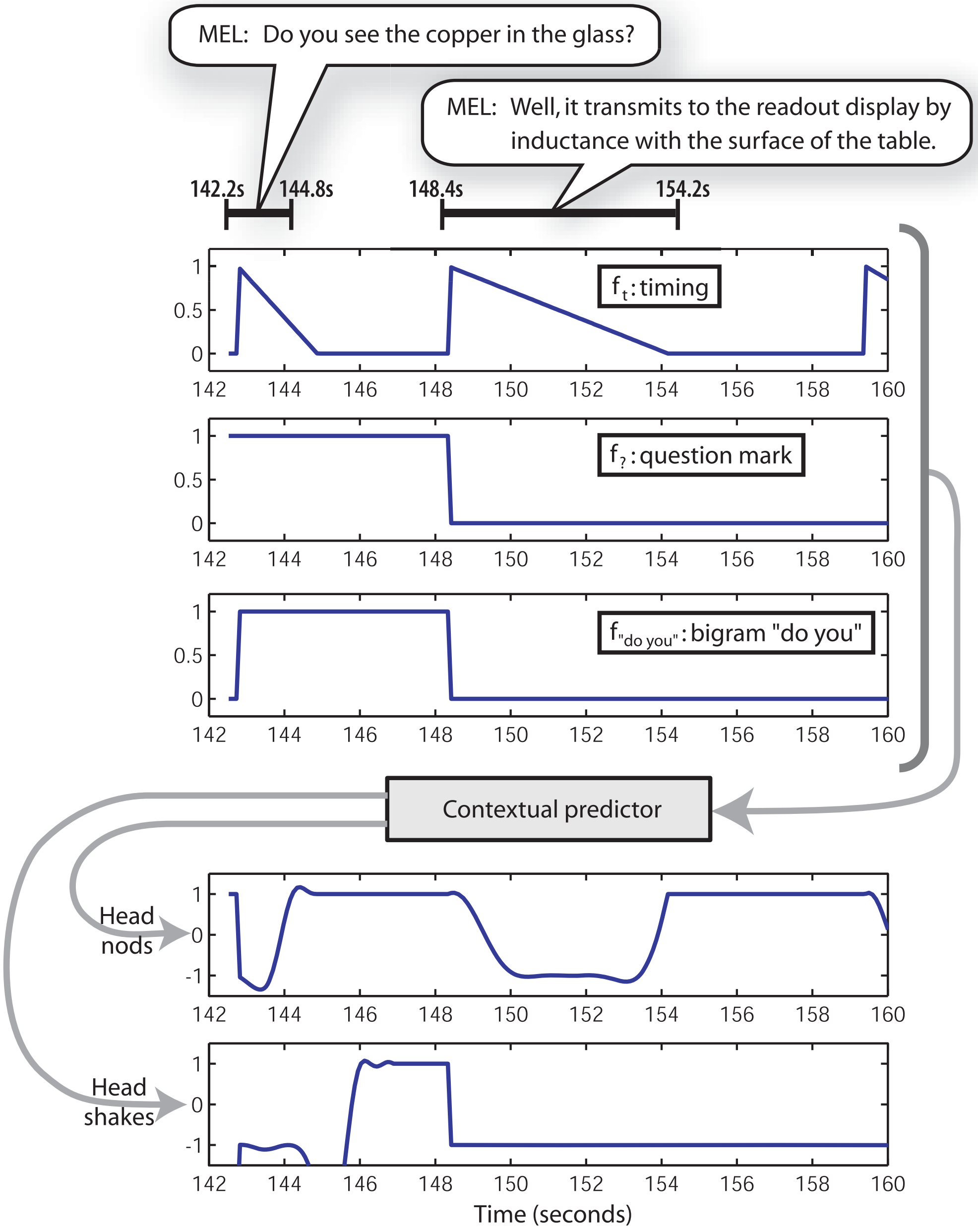

Figure 1.4 Prediction of head nods and head shakes based on 3 contextual features: (1) distance to end-of-utterance when ECA is speaking, (2) type of utterance and (3) lexical bigram feature. We can see that the contextual predictor learned that head nods should happen near or at the end of an utterance or during a pause while head shakes are most likely at the end of a question.

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 most likely occurs at pauses or just before. This feature can easily be computed given only two values: $t_0$, the utterance start-time, and $\delta_t$, the estimated duration of the utterance. Given these two values for the current utterance, we can estimate $f_t$ at time $t$ using:

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

We selected our features so that they are topic independent. This means that we should be able to learn how to predict visual gestures from a small set of interactions and then use this knowledge on a new set of interactions with a different topic discussed by the human participant and the ECA. However, different classes of dialogues 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 vision-based recognizer module. The lexical, punctuation and gesture features are evaluated based on the current spoken utterance. A specific utterance is active until the next spoken utterance starts, which means that in-between pauses are considered to be part of the previous utterance. The top three graphs of Figure 1.4 show how two sample utterances from our user study (described in Section 1.7) will be coded for the word pair “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 chapter.

Figure 1.4 also displays the output of our trained contextual predictor for anticipating head nods and head shakes during the dialogue 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 more 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).

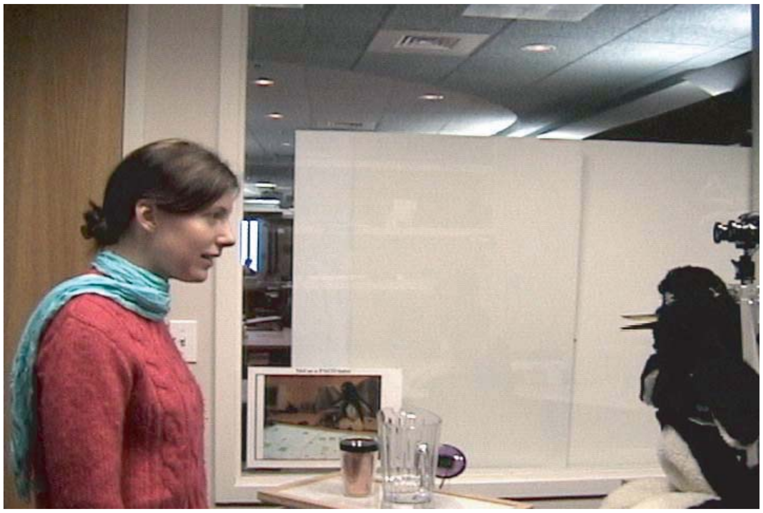

Figure 1.5 Mel, the interactive robot, can present the iGlassware demo (table and copper cup on its right) or talk about its own dialog and sensorimotor abilities.

## 1.7 Context-based Head Gesture Recognition

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

For this experiment, a first data set was used to train the contextual predictor and the multi-modal integrator (the same data set as described in Section 1.5), while a second data set with a different topic was used to evaluate the head gesture recognition performance. In the training data set, the robot interacted with the 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 the iGlassware invention ($\sim$340 utterances).

For both data sets, human participants were video recorded while interacting with the robot (see Figure 1.5). The vision-based head tracking and head gesture recognition was run online ($\sim$18Hz). The robot's conversational model, based on COLLAGEN (Rich et al. 2001), determines the next activity on the agenda using a predefined set of engagement rules, originally based on human–human interaction (Sidner et al. 2005). Each interaction lasted between 2 and 5 minutes.

During each interaction, we also recorded the results of the vision-based head gesture recognizer as well as the contextual cues (spoken utterances with start time and duration) from the dialogue manager. These contextual cues were later automatically processed to create the contextual features (see Section 1.6.1) necessary for the contextual predictor (see Section 1.5).

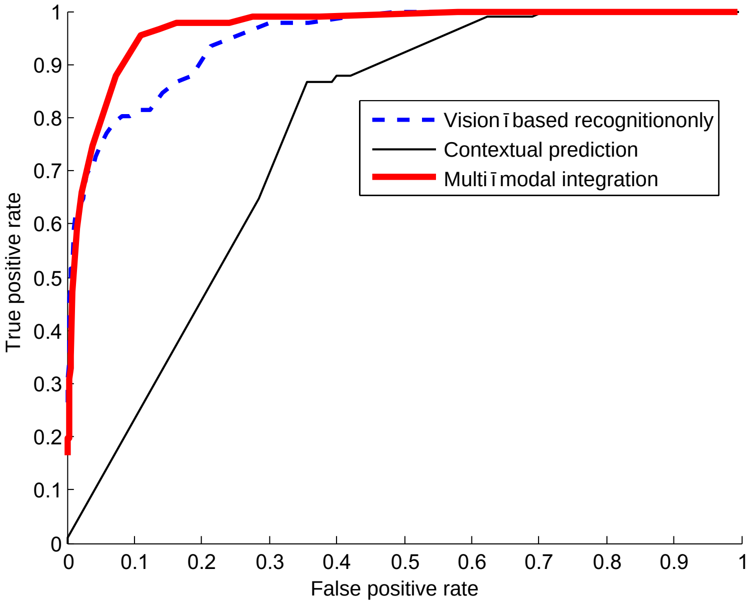

Figure 1.6 Head nod recognition curves when varying the detection threshold.

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.

### 1.7.1 Results

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).

Figure 1.6 shows head nod detection results for all 9 subjects used during testing. The ROC curves present the detection performance each recognition algorithm when varying the detection threshold. The area under the curve for each techniques are 0.9482 for the vision only, 0.7691 for the predictor and 0.9678 for the integrator.

Figure 1.7 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.

Table 1.1 summarizes the results from Figures 1.6 and 1.7 by computing the true positive rates for the fixed negative rate of 0.1. Using a standard analysis of variance (ANOVA) on all the subjects, results on the head nod detection task showed a significant difference among

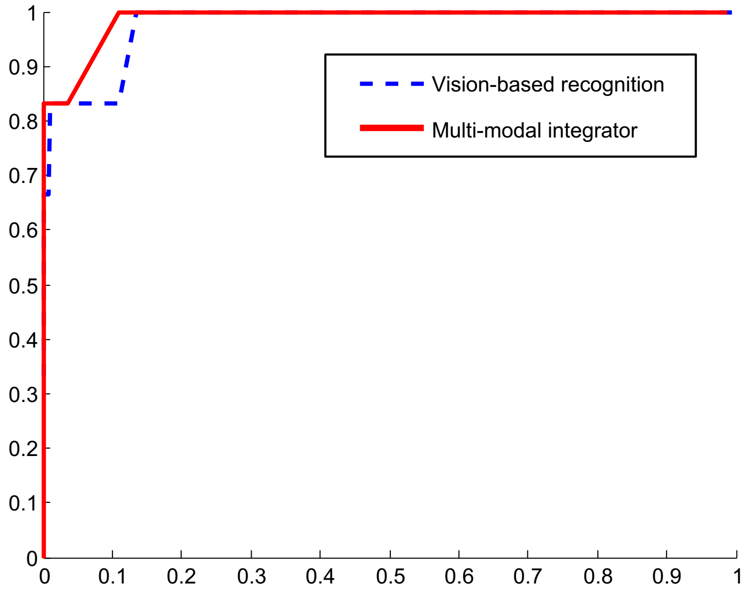

Figure 1.7 Head shake recognition curves when varying the detection threshold.

Table 1.1 True detection rates for a fix false positive rate of 0.1.

| | Vision | Predictor | Integrator |
|---|---|---|---|
| Head nods | 81% | 23% | **93%** |
| Head shakes | 83% | 10% | **98%** |

the means of the 3 methods of detection: $F(1,8) = 62.40$, $p < 0.001$, $d = 0.97$. Pairwise comparisons show a significant difference between all pairs, with $p < 0.001$, $p = 0.0015$, and $p < 0.001$ for vision-predictor, vision-integrator, and predictor-integrator respectively. A larger number of samples would be necessary to see the same significance in head shakes.

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). For the iGlass dataset, the total numbers of ground truth gestures were 91 head nods and 6 head shakes.

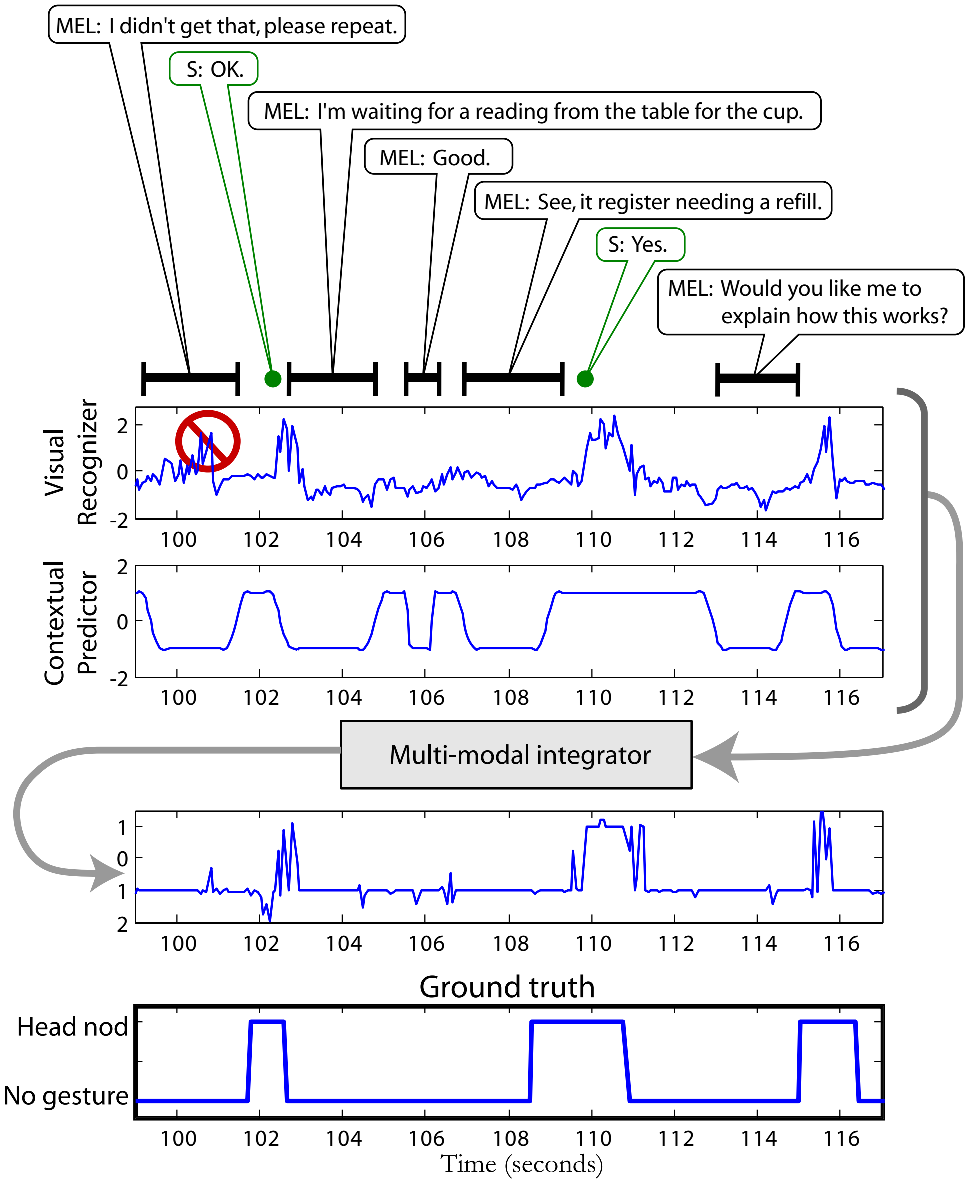

Figure 1.8 Head nod recognition results for a sample dialogue. The last graph displays the ground truth. We can observe at around 101 seconds (circled and crossed in the top graph) that the contextual information attenuates the effect of the false positive detection from the visual recognizer.

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. For the iGlass dataset, the total number of non-gestures frames was 18246 frames and the total number of frames for all 9 interactions was 20672 frames.

Figure 1.8 shows the head nod recognition results for a sample dialogue. When only vision is used for recognition, the algorithm makes a mistake at around 101 seconds by false detecting a head nod. Visual grounding is less likely during the middle of an utterance. By incorporating the contextual information, our context-based gesture recognition algorithm is able to reduce the number of false positives. In Figure 1.8 the likelihood of a false head nod happening is reduced.

## 1.8 Conclusion and Future Work

Our results show that contextual information can improve user gesture recognition for interactions with embodied conversational agents. We presented a prediction framework that extracts knowledge from the spoken dialogue of an embodied agent to predict which head gesture is most likely. By using simple lexical, prosodic, timing and gesture context features, we were able to improve the recognition rate of the vision-only head gesture recognizer from 81% to 93% for head nods and from 83% to 98% for head shakes. As future work, we plan to experiment with a richer set of contextual cues including those based on gesture display, 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.

## Bibliography

Bickmore T and Cassell J 2004 *J. van Kuppevelt, L. Dybkjaer, and N. Bernsen (eds.), Natural, Intelligent and Effective Interaction with Multimodal Dialogue Systems* Kluwer Academic chapter Social Dialogue with Embodied Conversational Agents.

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

Cassell J and Thorisson KR 1999 The poser of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*.

de Carolis B, Pelachaud C, Poggi I and de Rosis F 2001 Behavior planning for a reflexive agent *Proceedings of IJCAI*, Seattle.

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

Dillman R, Ehrenmann M, Steinhaus P, Rogalla O and Zoellner R 2002 Human friendly programming of humanoid robots–the German Collaborative Research Center *The Third IARP Intenational Workshop on Humanoid and Human-Friendly Robotics*, Tsukuba Research Centre, Japan.

Ekman P 1992 An argument for basic emotions. *Cognition and Emotion* **6**(3–4), 169–200.

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

Kapoor A and Picard R 2001 A real-time head nod and shake detector *Proceedings from the Workshop on Perspective User Interfaces*.

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

Morency LP, Rahimi A and Darrell T 2003 Adaptive view-based appearance model *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 803–810.

Morency LP, Rahimi A, Checka N and Darrell T 2002 Fast stereo-based head tracking for interactive environment *Proceedings of the Int. Conference on Automatic Face and Gesture Recognition*, pp. 375–380.

Nakano Y, Reinstein G, Stocky T and Cassell J 2003 Towards a model of face-to-face grounding *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

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

Sidner C, Lee C, Kidd CD, Lesh N and Rich C 2005 Explorations in engagement for humans and robots. *Artificial Intelligence* **166**(1–2), 140–164.

Siracusa M, Morency LP, Wilson K, Fisher J and Darrell T 2003 Haptics and biometrics: A multimodal approach for determining speaker location and focus *Proceedings of the 5th International Conference on Multimodal Interfaces*.

Stiefelhagen R 2002 Tracking focus of attention in meetings *Proceedings of International Conference on Multimodal Interfaces*.

Takemae Y, Otsuka K and Mukaua N 2004 Impact of video editing based on participants' gaze in multiparty conversation *Extended Abstract of CHI'04*.

Torralba A, Murphy KP, Freeman WT and Rubin MA 2003 Context-based vision system for place and object recognition *IEEE Intl. Conference on Computer Vision (ICCV)*, Nice, France.

Traum D and Rickel J 2002 Embodied agents for multi-party dialogue in immersive virtual world *Proceedings of the International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, pp. 766–773.