Automated Sip Detection in Naturally-evoked Video

Rana el Kaliouby
Media Laboratory
Massachusetts Institute of Technology
20 Ames Street
Cambridge MA 02139 USA
kaliouby@media.mit.edu

Mina Mikhail
Computer Science Department
American University in Cairo
113 Kasr Al Aini Street
Cairo, Egypt
minamohebn@gmail.com

ABSTRACT

Quantifying consumer experiences is an emerging application area for event detection in video. This paper presents a hierarchical model for robust sip detection that combines bottom-up processing of face videos, namely real-time head action unit analysis and and head gesture recognition, with top-down knowledge about sip events and task semantics. Our algorithm achieves an average accuracy of 82% in videos that feature single sips, and an average accuracy of 78% and false positive rate of 0.3%, in more challenging videos that feature multiple sips and chewing actions. We discuss the generality of our methodology to detecting other events in similar contexts.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous; J.4 [**Social and Behavioral Sciences**]: Psychology

General Terms

Experimentation, Human Factors

Keywords

Event detection, Human activity recognition, Affective computing, Head gesture recognition, Spontaneous video

1. INTRODUCTION

With the increasing ubiquity of cameras and video streaming, event detection in video is being applied to a range of applications and domains including surveillance, vision-based human computer interaction, content-based retrieval and sports video analysis. This paper presents an emerging application of event detection in video: quantifying consumer experiences for marketing, product evaluation, usability, advertising and customer relationship management. We describe a natural video corpus of a sipping study where consumers, in a series of trials, are given a choice of two beverages to sip and then asked to answer some questions related to their sipping experience. One of the main events of interest is that of the sip, where we are interested in analyzing the customer's facial expression leading up to and immediately after the sip. Manually tagging the video with sip events is a time and effort-consuming task; at least two or three coders are needed to establish inter-rater reliability, requiring at least 30 minutes of coding per video per coder.

As with event detection in video in general, several challenges exist with regard to machine detection and recognition of sip events. First, a good definition of what constitutes a sip event is needed that covers the different ways with which people sip and defines the beginning and end of an event. Secondly, detecting sip events involve the detection and recognition of the person's face, their head gestures and the progression of these gestures over time. Third, events are often multi-modal, requiring fusion of vision-based analysis with semantic information from the problem domain and other available contextual cues. Finally, the sipping videos are different than those of say surveillance or sports: there are typically fewer people in the video, the amount of information available besides the video is minimal, compared to sports where there's an audio-visual track and lots of annotations. Also the events are subtler and there is typically only one camera view that is static.

In this paper, we were faced with the challenge of tagging hours of videos to quantify participants's reactions to sipping different beverages. The first step in doing so is tagging the video with sip events. Our approach combines machine perception—namely probabilistic models of facial expressions and head gestures—with top-down semantic knowledge of the events of interest. While facial expression and head gesture recognition has been around for a while, our work extends existing research in two principal ways. First, the majority of existing literature on facial expression analysis is concerned with the recognition of facial events; in this paper we show how knowledge of facial expressions and head gestures, when combined with contextual information, can be used for activity recognition and for quantifying people's experiences. Second, we test our methodology on an extensive corpus of natural videos that feature substantial head motion, occlusions, and changes in lighting (most facial analysis systems are not tested on natural videos. On such a natural, challenging corpus, we report high accuracy with low false positive rate for videos that have single as well as multiple sips. Our approach of combining machine perception with semantic knowledge about the context can

ICMI'08, October 20-22, 2008, Chania, Crete, Greece.

be generalized and applied to the detection of other events for quantifying consumer experiences.

The paper is organized as follows: section 2 surveys related work on event detection in video, especially those that address human activity recognition. Section 3 discusses the corpus of videos from the sipping study, explaining the occurrence of sips. Section 4 gives an overview of our methodology for sip detection. Sections 5 and 6 describe facial feature tracking and head gesture recognition; section 7 describes the sip detection algorithm. Experimental evaluation and results are presented in section 8. Section 9 concludes the paper and outlines future directions in the area of event detection in video for quantifying consumer experiences.

2. RELATED WORK

The problem of machine detection of sips is closest to that of human activity recognition in video, a subset of the broader area of event detection in video. Event detection may involve the detection and recognition of objects (e.g., a face or cup), actions (e.g., turning head towards the cup, picking up cup, sipping), and their relationship over time. While object detection—determining if an instance of a given class of objects is present or not in an image—is often a component of event detection, a survey of that literature is outside the scope of this paper; instead, the reader is referred to the excellent survey by Yilmaz *et al.* [12].

Event detection often fuses information from other modalities and sources, and in the majority of cases uses prior semantic knowledge that is specific to the problem domain. A few exceptions include Fleischman *et al.* [4] where unsupervised content-based indexing is used in retrieval of sports video and Mustafa and Sethi [8] where forests of hidden Markov models and random local detectors in the camera's field of view are used unsupervised event detection in videos.

One approach to detecting events in video, phrases the problem as a classification problem of video segments. Features are first extracted from each frame of a video segment (which is typically a sliding window of a certain number of frames), concatenated and then fed to a classifier such as a support vector machine, Bayesian classifier or decision trees that predicts the semantic label of each segments. Hung *et al.* [7] extract scoreboard and shot transition information from baseball videos and input those into a Bayesian belief network to classify several types of baseball actions. Wang *et al.* [10] use support vector machines to detect audio events such as a whistle or audience noise which are then used to segment events in soccer videos. The drawback of treating event detection as a classification problem is that there is no explicit encoding of the temporal relationships between actions or events in a video, so it is challenging to find the boundaries—start and end time—of an event. In addition, long-term dependencies between events or actions in a video are not encoded.

An alternative approach defines the problem as a sequence learning problem where temporal information between frames and events in the video is explicitly modeled either deterministically or probabilistically. Defining an explicit relationship between events in the sequence learning approach increases the accuracy of event detection and yields more specific start and end times of an event. Hakeem and Shah [6] define a video event graph of temporally correlated sub-events to represent the frequency of occurrence of conditionally dependent sub-events. They applied the event graphs to several domains in which there are multiple agents interacting such as event detection in office meetings, railroad monitoring and surveillance. Bia *et al.* [1] predefine visual concepts in soccer games (e.g., slow motion replay, referees and score captions) that are then fused with aural concepts such as cheers or whistle and input to a finite-state machine that encodes the relationship between these semantics to detect events. Guler *et al.* [5] use background subtraction followed by foreground object segmentation with hidden to detect human motion activities such as walking or carrying actions. Finally, Wang *et al.* [11] use conditional random fields to fuse temporal multi-modal cues such as audio-visual features and keywords for event detection in soccer video.

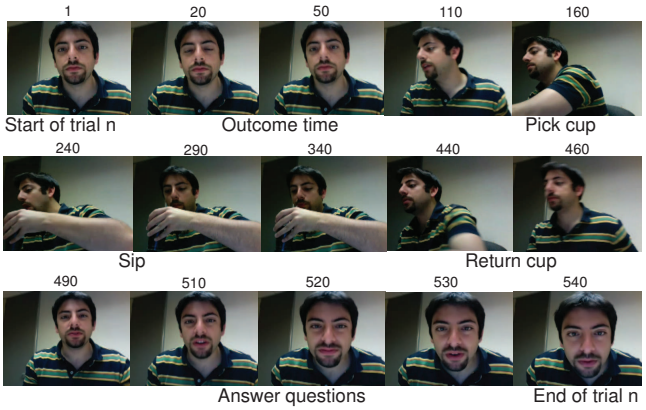

Figure 1: Semantics of a sip event. For each of the thirty trials that feature only a beverage sip, participants find out which beverage to sip, turn towards the cup, sip, return the cup and answer questions.

3. SIP STUDY

The videos used to test our algorithm are from a sip study that Affective Computing at MIT Media Laboratory conducted in collaboration with a major beverage company. Thirty-five participants were recruited (equal number of males and females), each participant is seated in front of a laptop (with a built-in webcam) and given a choice of two beverages that were located on the left and right of the laptop. Each participant is then asked to take a sip of one of the beverages and answer several online questions about their experience. This sequence of sipping then answering questions is repeated 30 times, for an average duration of 30 minutes per participant. We note that while the participants were aware of being recorded, they were not given any instructions to limit their face or body motion. As a result, there is considerable head and body motion in all the videos, especially as the participants turned to pick a beverage and there is substantial individual variances in expressiveness.

As shown in Fig. 1, each trial is defined by the `Outcome-Time`, the time when the participant is told which beverage to take a sip of, followed by the sipping event—orienting towards the cup, picking the cup, sipping and returning the cup—and then the questions time. In addition, participants also had the option to drink water or eat crisps before a sip of beverage to neutralize the taste of the previous sip: 78 out of a total of 699 featured multiple sips or chewing actions, an example of which is shown in Fig. 2.

Figure 2: An example of a multiple sip, showing the participant chewing and sipping water before sipping the beverage.

4. EVENT DETECTION METHODOLOGY

We use machine vision and machine perception of human activity combined with event information that is automatically logged by the sip application to form a semantic representation of sip events. One approach would have been to develop a cup detector (focusing on the object of interest rather than the action of sipping). Looking at the videos, we realized that in many of the sips, the cup is not visible, and different cups were presented to the consumer for the beverage and the water. Thus, we decided to focus on detecting the action of sipping using a machine perception approach, where we attempt to infer the action of sipping from the behavior of the person.

As shown in Fig. 3, our approach is hierarchical going from low-level inferences about the presence of a face in the video and the person's head gesture (e.g., persistent head turn to the left) to more abstract knowledge about the presence of a sip event in the video. Our method combines elements from both approaches to event detection—using a dynamic classification approach at the lower levels of the model to a sequence representation at the topmost level. This hierarchy of actions allows us to model the complexity inherent in the problem of sip detection, namely the multiple definitions and scenarios of a sip, as well as the uncertainty of the actions, e.g., whether the person is turning their head towards the cup or simply talking to someone else. In addition, we use semantic information from the event logs to increase the accuracy of the system.

As described earlier, a sip is characterized by the person turning towards the cup, leaning forward to grab the cup and then drinking from the cup (or straw). We use face tracking and head pose estimation to identify when the person is turning, followed by a head gesture recognition system that identifies only persistent head gestures using a networks of dynamic classifiers (hidden Markov models). At the topmost level we have devised a sip detection algorithm that for each frame analyzes the current head gesture, the status of the face tracker and the event log, which in combination provide significant information about the person's sipping actions.

5. HEAD POSE ESTIMATION

For feature point tracking we use Google's `FaceTracker` [3], formerly Nevenvision's facial feature tracking SDK. `FaceTracker` uses a generic face template to bootstrap the tracking process, initially locating the position of 22 facial landmarks including the eyes, mouth, eyebrows and nose. A

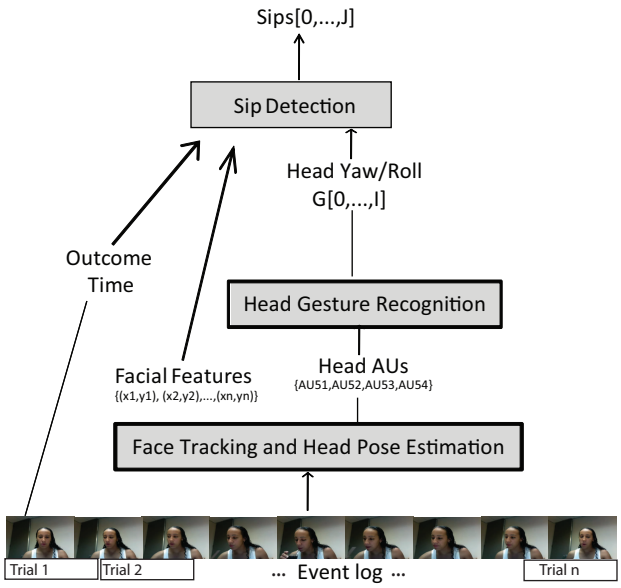

Figure 3: Hierarchical methodology for sip detection

combination of Gabor wavelet image transformations and neural networks are then used to track the position of the points over a live or recorded video stream. While tracking proceeds on 2D video input, a learned 3D model of the human face is used to correct tracking errors and cope with pose variations. `FaceTracker` deals with a wide range of face physiognomies and skin colors, and tracks users that wear glasses and/or have facial hair. The tracker also deals with non-initial neutral frames, a key feature that most other existing tracking systems do not currently support.

`FaceTracker` correctly tracks videos with head rotation speed of up to 6 degrees per frame. In the event that the confidence of the tracker falls below a certain threshold (0.6 in our case), as in a sudden large motion of the head or a head yaw exceeding 40 degrees, the tracker stops and reinitialized after 5 ms before attempting to relocate the feature points. The status of the tracker—whether it is on of or off–provides useful information regarding a person's pose especially when combined with knowledge about the person's previous position and head gestures.

Once the feature points are located for a frame t, head action units (AUs) using Ekman and Friesan's Facial Action Coding System (FACS) [2] are recognized at that frame. The following head AUs are detected by the system: the pitch actions AU53 (up) and AU54 (down), yaw actions AU51 (turn-left) and AU52 (turn-right), and head roll actions AU55 (tilt-left) and AU56 (tilt-right). The rotation along the pitch, yaw and roll, $\angle_{\text{yaw}}$, $\angle_{\text{pitch}}$ and $\angle_{\text{roll}}$ respectively, are calculated from expression invariant points. These points are the nose tip, nose root and inner and outer eye corners. For instance, $\angle_{\text{yaw}}$ is computed as the ratio of the left to right eye widths, while $\angle_{\text{roll}}$ is computed as the rotation of the line connecting the inner eye corners.

The output of this stage of analysis consists of: (1) a vector of the tracker's status `Tracker`[0,...,T], where at frame t, `Tracker`[t] is either on (a value of 1) or off (a value of 0); and (2) a vector of detected head action units that is used for head gesture recognition.

6. HEAD GESTURE RECOGNITION

Detecting single instances of head AUs is insufficient to describe a yaw (turn) or roll (tilt) event in the case of a sip. Instead, we are interested in head yaw/roll events that are *persistent* in the right or left direction to highlight that the head gesture has persisted over time. We use Hidden Markov Models (HMMs), to represent such head gestures. Each head gesture j is represented by a discrete HMM with N states, M symbols and parameters $\lambda_j = (\pi$,A,B):

- **N**, the number of states in the model $S = \{S_1, \ldots, S_N\}$; each state maps to a temporal segment of the head gesture or facial expressions. The state at time t is denoted as q_t.
- **M**, the number of different observation symbols $V = \{v_1, \ldots, v_M\}$; each symbol maps to a head or facial action that constitutes that gesture or facial expressions. For example, the feature space of the head nod HMM consists of three symbols: head-up, head-down and null.
- **A**= $\{a_{ij}\}$, an N x N matrix that specifies the probability that the model's state will change from state i to state j, where $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$, $1 \leq i, j \leq N$
- **B**= $\{b_i(k)\}$, an N x M matrix, the observation symbol probability matrix depicts the output observation given that the HMM is in a particular state i, where $b_i(k) = P(v_k | q_t = S_i)$, $1 \leq i \leq N$ and $1 \leq j \leq M$
- π= $\{\pi_i\}$ is an N-element vector that indicates the probability of initially being in state i, where $pi_i = P(q_0 = S_i)$ $1 \leq i \leq N$

We use left-right, also referred to as Bakis models [9] to implement the head gesture recognition. In left-right HMMs, the state sequence begins from the left at state 1 and ends on the right at the final state N. As time increases, the observable symbols in each sequence either stay at the same state or increase in a progressive manner. The output of this stage is a vector I of detected head yaws/rolls `Gestures`[0,...,I], where each `Gestures`[i] represents a single instance of a persistent head gesture, with a specified start and end time.

7. SIP DETECTION ALGORITHM

Semantically, a sip event consists of orienting towards the cup, picking the cup, taking a sip and returning the cup before turning back towards the laptop to answer some questions. The input to the topmost level of our sip detection methodology consists of the following:

- `Gestures`[0,...,I], the vector of I persistent head turns and tilts;
- `Tracker`[0,...,T], describes the status of the tracker (on or off) at each frame of the video $0 < t < T$, which is needed because the face tracker stops when the head yaw or roll exceeds 30 degrees, which typically happens in sip events;
- `EstStartofSip`, which denotes the time within each trial when the participant is told which beverage to take a sip of (note that this is logged by the application and not manually coded)—this time is offset by a few seconds `WaitTime` to allow the participant to read the outcome and begin the sipping action;
- `TurnDuration` is the minimum duration of a persistent head gesture that indicates a sip;
- `EstQuestionDuration` is the average time it takes to answer the questions following a sip event.

The face is tracked and the vector of I persistent head gestures are detected as described in Sections **??** and 6. The rest of the sip detection algorithm is described in Algorithm 1. As described in the algorithm, there are three cases of sip detection:

Algorithm 1 Sip detection algorithm.

Input: `Tracker`[0,...,T], head yaw/roll gestures `Gestures`[0,...,I], `EstStartofSip`, `TurnDuration`, `EstQuestionDuration`
Output: `Sips`[0,..,J]

```
SipFound ← FALSE
for all Gestures[i] from 0 to I do
  if (Gestures[i].start <= EstStartofSip <= Gestures[i].end) then
    Sips[j].start ← Gestures[i].start
    Sips[j].end ← Gestures[i].end
    SipFound ← TRUE
  end if
end for
if ¬SipFound then
  for all Gestures[i] from 0 to I do
    if (Gestures[i].end <= EstStartofSip) and (Gestures[i].duration > TurnDuration) and (Tracker[t]=0) then
      Sips[j].start ← Gestures[i].start
      Sips[j].end ← Gestures[i].end
      SipFound ← TRUE
    end if
  end for
end if
if ¬SipFound then
  G ← GetLongest(Gestures[0,...,I])
  Sips[j].start ← G.start
  Sips[j].start ← G.end
end if
```

- In the first case—shown in Fig. 4—`Gestures` is parsed for a tilt or a turn event such that `EstStartofSip` elapses between the start and end frames of the gesture. In this case, the start and end frames of the sip correspond to that of the gesture;
- In the second case (Fig. 5), if a head gesture `Gestures`[i] that persists for `TurnDuration` ends before `EstStartofSip` is found, the status of the face tracker is checked. A sip is detected if the tracker was off for at least M frames following the end of `Gestures`[i]. The parameter M ensures that any case where the tracker is off for a short period of time is ignored;
- If the first two cases do not return a head gesture before or around `EstStartofSip`, the rest of the trial is

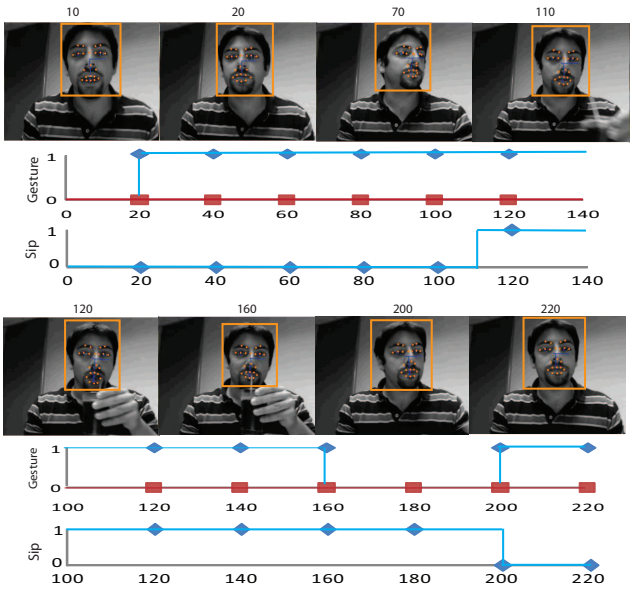

Figure 4: Case 1: an example of sip detected using a combination of event log heuristics as well as observed head yaw/roll gestures. At each frame, if the tracker is on, the facial feature points and rectangle around the face are shown. For each row of frames, the recognized head yaws and rolls are shown in the top chart, while the output of the sip detection algorithm is shown in the bottom chart.

searched for head turns and tilts. The tilt or turn with the longest duration is considered to be the sip (Fig 6).

Fig. 7 shows the distribution of cases of our algorithm for each participant in our corpus—case 1 accounts for 45% of the detected sips; case 2 accounts for 25%, while case 3 accounts for the remaining 30% of sips.

The algorithm above only deals with a single sip per trial. However, as described earlier, the participants often chewed or drank water before taking a sip of the beverage. Thus, any number of sips could occur within `EstStartofSip` right upto `EstQuestionDuration` before the start of the next trial, which is the time it takes the participant to answer questions related to their sipping experience. To handle multiple sips within a trial, persistent head gestures that: (1) occur after `EstStartofSip`; (2) start within `EstQuestionDuration` before the start of the next trial and (3) last for at least `TurnDuration` are all returned as possible sips.

8. EXPERIMENTAL EVALUATION

We test our methodology with 25 out of the 35 participants with each video lasting 30 minutes for a total of 12 plus hours of video. Ten videos were discarded because in seven of the participants only the first few minutes of the video was available and in the remaining three there was no accompanying event log available. All the videos have a resolution of 320x240 and were recorded at 25fps. Each video contains 29 sip trials each, except for one participant, which only had 11 valid sip trials (in the remaining 18 trials the participant ignored the sipping instructions, thereby invalidating these trials).

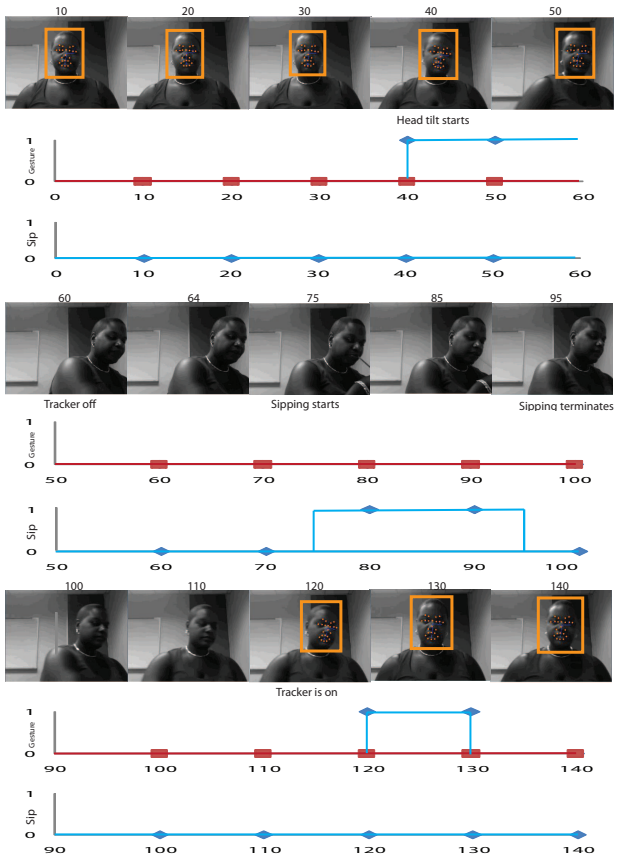

Figure 5: Case 2: an example of a sip detected by a temporal sequence of detecting a head yaw/roll gesture followed by the tracker turning off.

The videos featured 777 sips, which are part of single sip or multiple sip trials. In the latter, the user may perform more than one sipping and/or chewing actions. A sipping event lasts around 10-30 seconds and starts when the participant picks up the cup containing the beverage or water and terminates when the participant returns the cup back to its place. In this study, the beverage was provided with a cup and straw, while the water was provided in a cup only. Two manual coders coded the sipping times. While there was 80-90% inter-rater agreement about the occurrence of a sip event, it was harder for the raters to agree on the exact start and end times of a sip. The only additional information available besides the raw video was the `OutcomeTime`, the time when the user finds out which beverage to sip.

8.1 Face Tracker results

The video corpus presents a very challenging test for any facial analysis systems as there was substantial head motion and rotation, recurring face occlusions due to sipping as well as hand-over-face gestures, inconsistent changes in lighting across video. In addition, some of the participants wore accessories such as sunglasses which affected the face tracking, while others wore a head cap, which often occluded the face (as with the participant in Fig. 6).

To get a sense of how much of these natural videos were being tracked (and in turn being analyzed for head gestures),

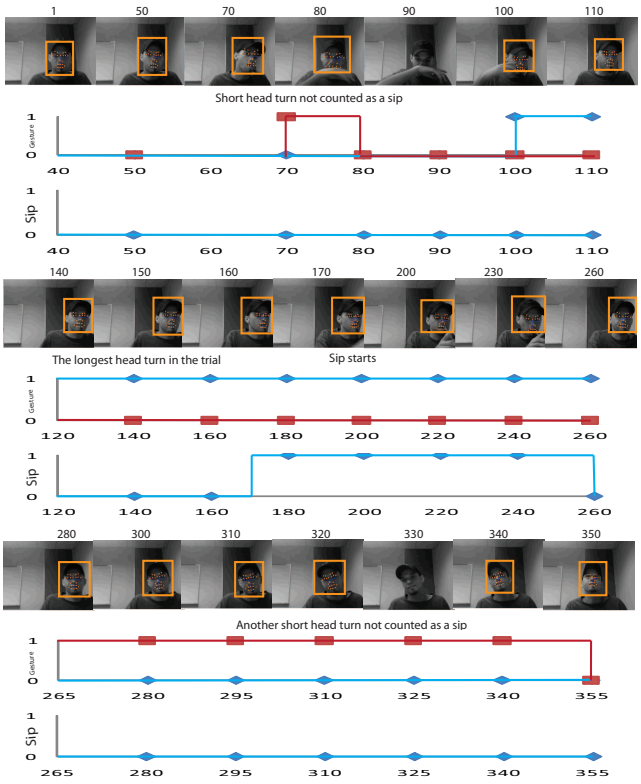

Figure 6: Case 3: an example of detecting a sip by finding the longest head yaw/roll gesture within a specified time frame.

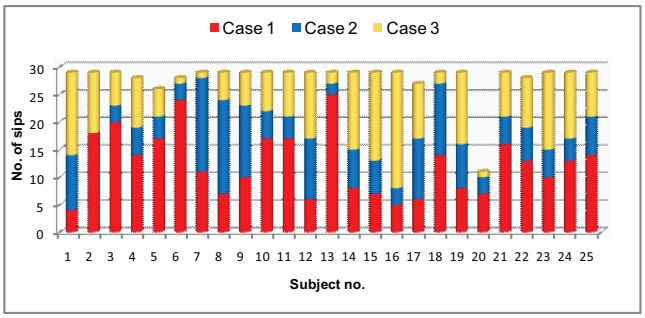

Figure 7: Breakdown of our sip detection algorithm for each participant. Case 1 looks for head yaws and rolls around `EstStartofSip` and account for 45% of our sip detection; Case 2 looks for a head yaw or roll followed by the tracker turning off, accounting for 25% of our sips; Case 3 looks for the longest duration of a sip and accounts for 30% of our sips.

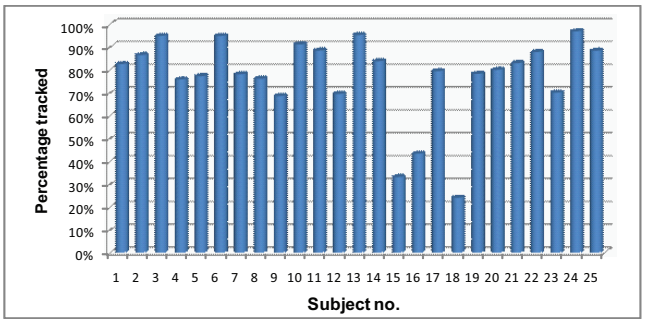

Figure 8: Percentage of each video that is successfully tracked. On average, the videos were successfully tracker 77% of the time.

we ran the 25 videos through the system and logged the frames when the tracker was on. We note that the tracker was forced to switch off when the tracking confidence fell below 0.6 (range is from 0 to 1)—thereby erring on the conservative side of tracking and ensuring that there is almost no false positive results (where a non-face object would be mistakenly picked as a face) in the tracking.

Fig. 8 shows the percentage of the video tracked for each of the 25 videos. On average, the videos were successfully tracked 77% of the time; 20% of the videos were tracked for more than 90% of the duration—in these videos, lighting was stable, the head and body motion of the participants was limited and they stayed at a relatively constant distance from the camera. Approximately half the videos were tracked 80% of the time. The videos of participants 15, 16 and 18 were tracked the least—less than 50% of the time—because either the participants wore sunglasses or frequently moved out of the camera view.

8.2 Single sips

We recorded the actual number of sips in each video and the number of sips that are detected by the algorithm. The accuracy of the algorithm for each participant is computed by dividing the number of detected sips by the number of sips. As shown in Table 1, the algorithm yielded an average recognition accuracy of 82%. Note that in the case of single sip detection, there are no false positives because the sip is either detected correctly or not.

We compare our algorithm to a heuristics-only algorithm

Table 1: Accuracy of sip detection algorithm.

#	# Single Sips	Accuracy	# Multiple Sips	Accuracy
1	29	0.86	32	0.88
2	29	0.83	29	0.83
3	29	0.86	37	0.86
4	28	0.82	36	0.81
5	26	0.88	34	0.82
6	28	0.93	29	0.90
7	29	0.93	30	0.93
8	29	0.90	30	0.87
9	29	0.83	33	0.73
10	29	0.79	30	0.77
11	29	0.66	30	0.67
12	29	0.66	35	0.66
13	29	0.79	29	0.93
14	29	0.93	31	0.65
15	29	0.69	33	0.70
16	29	0.66	31	0.62
17	27	0.81	28	0.79
18	29	0.97	40	0.80
19	29	0.90	35	0.80
20	11	0.73	12	0.75
21	29	0.83	34	0.79
22	28	0.75	29	0.73
23	29	0.69	30	0.67
24	29	0.79	30	0.77
25	29	0.93	30	0.90
Total	699	0.82	777	0.78

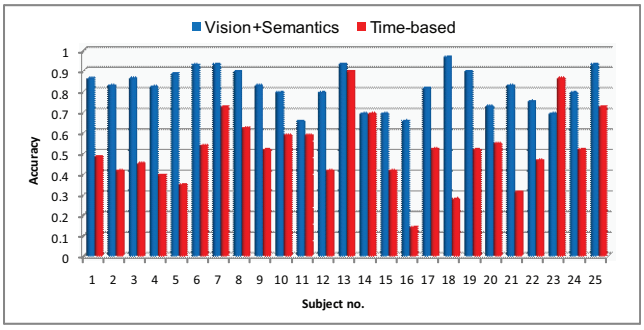

Figure 9: Comparison between our algorithm, which uses a combination of vision-based processing and semantic knowledge of our problem, to a heuristic, time-based estimate of a sip event.

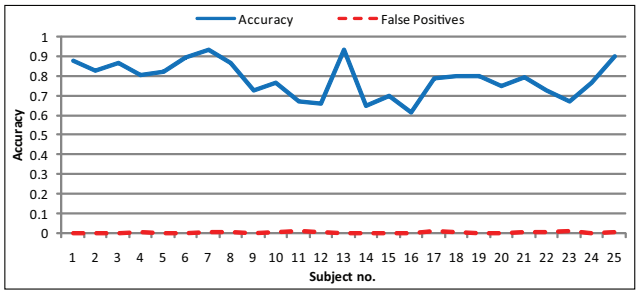

Figure 10: Accuracy and false positive rate in the case of multiple sips per trial. The average detection rate is 78% for a false positive rate of 0.3%.

that uses `EstStartofSip`—an estimation of the predicted time where the sip is likely to take place. This parameter was determined empirically by observing several videos. The comparison between our methodology and a heuristics only approach is shown in Fig. 9, demonstrating that the accuracy of our algorithm is substantially better than the heuristic based one mostly because the heuristic-based approach does not consider or accommodate the multiple paths to a sip event (e.g., that the participant may drink water before taking a sip of the beverage).

8.3 Multiple sips

The accuracy of our multiple sip detection algorithm is computed as the total number of detected sips that are true sips divided it by the actual number of multiple sips. As shown in Table 1, our algorithm yielded an accuracy of 78% for multiple sip detection. The false positive is computed as the number of falsely detected sips as a ratio of the total number of detected gestures, since that is the pool from which candidate sips are chosen. Our algorithm achieves a negligible false positive rate of 0.3%. Fig. 10 shows both the accuracy and false positive rate for multiple sip detection.

8.4 Discussion

We have shown that our methodology successfully detects single and multiple sips in over 700 examples of sip events. Our methodology fails to detect sips when the person remains in a frontal position while picking the cup and sipping, or when the head yaw or roll are undetected because

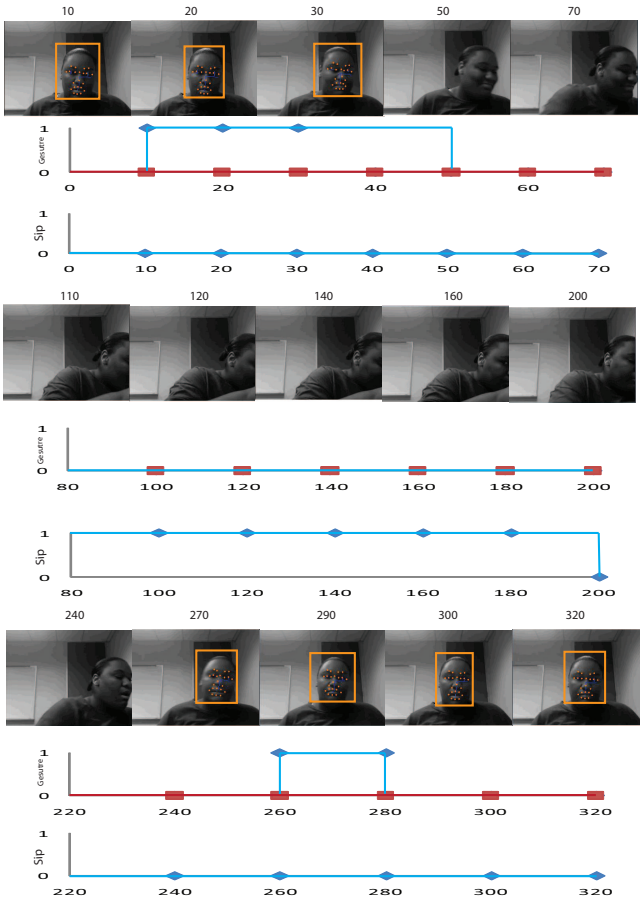

Figure 11: An example where the participant moves outside of the camera view throughout the sip event. Our approach is successful in detecting the sip event, where an alternative approach of detecting sips by finding cup objects in the video would have failed to identify this as a sip event.

the tracker is off. An example of an undetected sip is shown in Fig. 12. We note that one advantage of our method over an alternative approach such as using a cup detector, is that often the cup is outside of the camera view or may be occluded by the participant's hand. For instance, our approach still detects a sip even when the cup is not visible (Fig. 11).

While we apply our methodology for sip detection in video, the methodology can be easily modified and applied to other event detection in consumer studies by changing the parameter set to the problem. For instance, instead of tagging head yaw and roll events in video for sip detection, we could detect head nod or head shake events for agreement/disagreement in like/dislike studies or detect smiles for quantifying customer satisfaction. The application's event log could also be modified to suit the problem at hand, for instance logging events in a banking interaction.

9. CONCLUSION

This paper describes a methodology for detecting events of interest in studies of consumer preferences and product evaluation. In this particular study, one of the main events

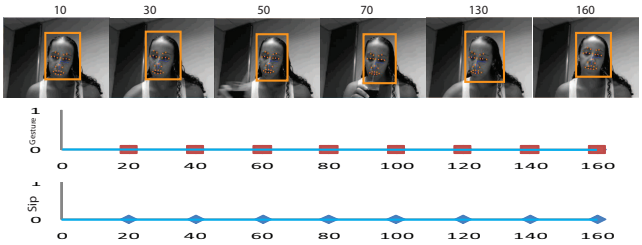

Figure 12: An example of an undetected sip event using our methodology.

of interest is that of a sip, since we are interested in analyzing the participant's facial expressions leading up to and immediately after the sip. Using a hierarchical model that combines bottom-up processing of the face videos, namely real-time face tracking and head gesture recognition, with top-down knowledge about sip events and task semantics, we present a robust algorithm for sip detection. Our algorithm achieves 82% accuracy in naturally-evoked videos that feature single sips, and 78% accuracy (false positive rate of 0.3%) in more challenging (also natural) videos that have multiple sip and chewing actions.

There are several future directions of this work. First, we would like to formulate this problem using dynamic Bayesian networks, where the presence of a sip event is probabilistic. We also plan to use the sip-time markers to automatically process facial expression before and after the sip event and correlate the results with liking and disliking data from the self-report measures. We are also developing an interface that makes the process of detecting events of interest for quantifying consumer experiences more turnkey. Finally, we would like to apply this methodology to other event detection problems in the domain of consumer preference studies.

10. ACKNOWLEDGMENTS

The authors would like to thank Hyungil Ahn and Rosalind W. Picard who in collaboration with a major beverage company designed and conducted the sip study and made the corpus available for this work. The authors would also like to thank Abdel Rahman Mahmoud and Youssef Kashef for their help developing the facial analysis API, and Google for making the face tracker available to our research. This research is supported by MIT's Things that Think Consortium.

11. REFERENCES

[1] L. Bai, S. Lao, W. Zhang, G. Jones, and A. Smeaton. A Semantic Event Detection Approach for Soccer Video based on Perception Concepts and Finite State Machines. *Image Analysis for Multimedia Interactive Services. Eighth International Workshop on*, pages 30–30, 2007.

[2] P. Ekman and W. V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists, 1978.

[3] FaceTracker. *Facial Feature Tracking SDK*. Google (formerly Nevenvision), 2006.

[4] M. Fleischman. Unsupervised content-based indexing of sports video. *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 87–94, 2007.

[5] S. Guler, W. Liang, and I. Pushee. A Video Event Detection and Mining Framework. *Computer Vision and Pattern Recognition Workshop*, 4, 2003.

[6] A. Hakeem and M. Shah. Multiple agent event detection and representation in videos. *The 20th National Conference on Artificial Intelligence*, 26, 2005.

[7] M. Hung, C. Hsieh, and C. Kuo. Rule-based Event Detection of Broadcast Baseball Videos Using Mid-level Cues. *Innovative Computing, Information and Control, 2007. Second International Conference on*, pages 240–240, 2007.

[8] A. Mustafa and I. Sethi. Unsupervised Event Detection in Videos. *Tools with Artificial Intelligence, 2007. 19th IEEE International Conference on*, 2:179–182.

[9] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[10] J. Wang, C. Xu, E. Chng, K. Wah, and Q. Tian. Automatic replay generation for soccer video broadcasting. *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 32–39, 2004.

[11] T. Wang, J. Li, Q. Diao, W. Hu, Y. Zhang, and C. Dulong. Semantic Event Detection using Conditional Random Fields. *Computer Vision and Pattern Recognition Workshop*, 2:109–109.

[12] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.