Intelligently Integrating Information from Speech and Vision Processing to Perform Light-weight Meeting Understanding

Alexander I. Rudnicky, Paul E. Rybski,
Satanjeev Banerjee, and Manuela M. Veloso
Carnegie Mellon University
Pittsburgh, PA 15213
{air,prybski,satanjeev,mmv}@cs.cmu.edu

ABSTRACT

Important information is often generated at meetings but identifying, and retrieving that information after the meeting is not always simple. Automatically capturing such information and making it available for later retrieval has therefore become a topic of some interest. Most approaches to this problem have involved constructing specialized instrumented meeting rooms that allow a meeting to be captured in great detail. We propose an alternate approach that focuses on people's information retrieval needs and makes use of a *light-weight* data collection system that allows data acquisition on portable equipment, such as personal laptops. Issues that arise include the integration of information from different audio and video streams and optimum use of sparse computing resources. This paper describes our current development of a light-weight portable meeting recording infrastructure, as well as the use of streams of visual and audio information to derive structure from meetings. The goal is to make meeting contents easily accessible to people.

1. INTRODUCTION

Most organizations routinely conduct meetings which serve an important role in allowing groups of people to collaboratively develop solutions to problems and to maintain a shared understanding of goals and progress. Although everything that transpires at a meeting will not be of enduring interest, group members have occasion to seek information from previous meetings. We have found that people are interested in specific categories of information, depending on circumstances and that the information is gleaned from either artifacts (such as minutes or slides) or directly from meeting participants [2]. Often the exact nature of the information desired is not known until the time of the query, nor is the best source for this information (persons, artifacts, etc) apparent at the outset. We believe that a recording system that takes into account the potential value of the information it captures but maintains maximum flexibility in retrieval would be of great value to users.

Many approaches to meeting understanding assume that it would be desirable to capture as complete a record of a meeting as possible, so that a high-quality representation of its contents can be produced and that fairly subtle interpretations can be derived. This may not be necessary if the information that subsequently needs to be retrieved can be derived from focused extraction of relevant data from a meeting. We are particularly interested in the fact that simple data streams, taken individually and in combination appears to yield useful information about meeting structure, yet do no entail ”deep” understanding of meeting contents.

For example, we have found ([3]) that fairly simple low-level cues, such as turn-taking behavior, provide quite precise information about meeting structure and participant roles. We believe that bottom-up approaches to meeting understanding that rely on observable cues to participant behavior, and that are linked to operationally defined objectives (i.e. looking for specific types of information that are of demonstrable interest to users) will yield better systems as well as clearer understanding of meeting structure and dynamics as well as information content. Our own experience with highly instrumented environments, as well as experiments with low-level sources of information have led us to the conclusion that high-bandwidth approaches to meeting recording and understanding may not be the most effective approach to providing meeting information that is actually of interest to users.

On a more practical level, instrumented meeting rooms are costly to build and are (at least in our experience) difficult to maintain. We have also observed that people do not seem willing to use instrumented rooms on a spontaneous basis. Perhaps this is because current systems do not provide users with any clear value, certainly not enough to overcome whatever accommodation does need to be made. This limits the nature and number of meetings that can be collected and studied. Collection apparatus that is highly portable and easy to use would enable us to more easily collect data from many meetings in a variety of circumstances.

This paper describes our work on meeting understanding, based on a light-weight portable meeting recording infrastructure as well as the use of multiple streams of visual and audio information (as well as language information) to structure meetings and make their content accessible to interested parties. We describe the components of the system and discuss current research efforts to allow such a light-weight system to operate effectively.

2. CHROMOLITH: A LIGHT-WEIGHT MEETING RECORDING, UNDERSTANDING AND BROWSING SYSTEM

We have implemented the Chromolith system for the Capture and Retrieval Of Meeting knOwledge that Learns by Interacting wITh Humans. This system consists of a Meeting Recorder [1] that captures the audio, video and other streams at a meeting, the Speechalyzer that performs speech recognition and other speech processing on the captured audio, the CAMSeg that automatically provides meeting structure at different granularities, and the MockBrow meeting browsing system which can be used by an end-user to view recorded meetings and detected meeting structure.

The Meeting Recorder [1] supports light-weight recording of meetings using a generalized framework that allows for recording multiple types of data streams. The framework is general enough to allow the addition of new streams of information that may become available in the future. Each sensor in the room (head-mounted microphones, table-top microphones, ceiling-mounted video cameras, etc) is considered to produce a recordable stream of discrete events (speech segments, video frames, etc). These events are time-stamped using a central time server and initially recorded onto the local disc of the computer to which the sensor is connected, for example a personal notebook computer. Recordings are asynchronously transferred to a server, subject to available processing and connectivity bandwidths.

Meeting data stored on the server is processed by the Speechalyzer to extract speech information, including end-pointing of utterances, transcription and prosodic information (duration, energy, pitch). Since speech is acquired through close-talking microphones, the system also performs unsupervised adaptation to individual speakers (currently yielding a 5% relative improvement in accuracy). As part of the larger CALO system, the Speechalizer also has access to information about participants and their environment as well as artifacts (such as agendas) that are used to improve performance.

The CAMSeg component takes the processed audio and performs three different kinds of meeting structure understanding. First it detects the states the meeting went through (discussion, briefing and presentation), and the role of each participant during each state (discussion participant, presenter, observer, etc). Next it detects the *personnel roles* each participant plays. These include identifying the participant who is running the meeting and the skills of the other participants. Finally it detects the main topics of discussion during the meeting. More details of this component are in section 3.

The MockBrow meeting browsing system allows the user to view all the recorded meeting information, as well as additional information detected/ generated by the Speechalyzer and by CAMSeg. Users can play back those streams that are of interest to them (e.g., the audio from the presenter only, the recorded slides of the presenter, etc). The topics detected by CAMSeg are available to quickly zoom in on those portions of the meeting that are of interest.

3. SPEECH BASED MEETING STRUCTURE UNDERSTANDING

Speech-based meeting structure understanding is performed in the CAMSeg component—the CALO Meeting Segmenter. This component takes the speech activity as input, and extracts three different structures from it—the state of the meeting at various times, the organizational role of each participant, and the major topics of discussion in the meeting.

3.1 Meeting State Recognition

We define three different meeting states as follows, each with different possible participant roles. The *presentation state* indicates segments during which a single participant is presenting an idea or a concept using formal mechanisms, such as presentation slides on a projector or a white-board. In this case, meeting participants are either the *presenter* or *observers*. The *discussion state* indicates segments during which discussion or brainstorming takes place. Such states are marked by quick back-and-forth between the *discussion participants*. The last state, *briefing* times when there is an information flow from a single participant to one or more other participants. This state is marked by an absence of the quick back-and-forth of discussions, and also of the formal presentation mechanisms of the presentation state. Participant roles in this state include the *information provider* and one or more *information consumers*.

CAMSeg splits the meeting into small time segments and classifies each such segment into one of the three states (and participants into applicable roles) by using decision trees learned from hand-labeled data. Decisions are based on features derived from turn-taking patterns, such as utterance durations, degree of overlap and so on. Figure 1 shows classification accuracy as a function of lagging window size. A size of about 20 sec appears to give the best accuracy. We believe that this general approach can be extended to other types of phases.

3.2 Participant Role Detection

Besides the discourse-based roles that participants play in a meeting, participants also have *institutional* roles. These include manager (project lead, budget controlling manager, human resources manager, etc) as well as organization-specific *skills based* roles (hardware acquisition expert, facilities expert, etc). The long-term goal for CAMSeg is to develop a set of common roles and eventually automatically detect roles specific to a particular organization. Determining the role(s) played by individuals in meetings can then be used to constrain interpretation of meeting content.

Currently we have a simplified version of this problem where there are three roles that participants can play: that of a manager, a hardware acquisition expert and a facilities expert. CAMSeg detects these roles by accumulating all the speech of each of the participants across multiple meetings and then classifying them by running a decision tree trained on hand labeled data. Features include the simple utterance length features used in meeting state detection, as well as unigram based features extracted from the automatically recognized spoken words (as output by the Speechalyzer). Further to quickly take advantage of the specialized vocabulary of the participants in a specific organization, these decision trees are *self-trained* by relearning on the data from some of the earlier meetings in the sequence if participants

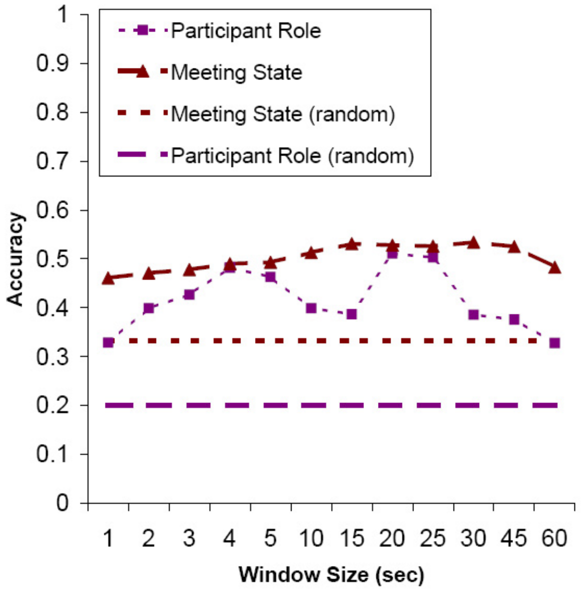

Figure 1: Classification accuracy for meeting phase, as a function of window size used for feature extraction.

attend multiple meetings. In an evaluation of this algorithm, we found that a trained system gives 67% initial accuracy on a test set, rising to 83% after two meetings of unsupervised learning.

3.3 Topic Detection

One of the most important ultimate goals of meeting recording is to provide a mechanism by which users can quickly find information in a meeting record. Specifically, given a user query, we wish to select those segments of the meeting that are relevant to the query. Currently CAMSeg has the capability of taking the name of a topic (as represented by a short string of words) and finding the segment(s) in the given meeting that corresponds to a discussion on that topic. For example, the user may be interested in viewing the portion of the meeting that contained discussions of "printers". Topic detection is performed using a variation on the text tiling algorithm [5]: a cosine similarity is computed between feature vector derived from adjacent pairs of meeting segments. Features include, in addition to words, utterance lengths and discourse markers associated with topic boundaries. Given the number of topics in a meeting (say derived from an agenda) the system places boundaries, on average, within 13 sec of a human-annotated boundary. Topics segments containing words in the query string are returned as being the meeting segments most relevant to the user query.

4. VISION BASED MEETING STRUCTURE UNDERSTANDING

Inferring the state of activities from visual information takes place at two levels. The first level is the state classification of the individual people attending the meeting. The second level is the classification of the global state of the meeting, which is done after the individual states of the people are determined. Instead of attempting to solve the image understanding problem purely from data, we construct a set of Dynamic Bayesian Network classifiers from *a priori* knowledge about meetings and about how people interact in those meetings.

4.1 Human Activity Recognition

We assume that the objects we want to recognize are used by humans and therefore, we want to detect those human activities. We detect people by searching a monocular panoramic video stream for faces [8]. The relative displacements of the face positions are calculated (difference from one frame to the next) and given as input to the activity recognition algorithm. We have defined a finite state diagram that encapsulates a coarse level of possible activities that can be detected in the data stream of visually tracked faces. Figure 2 illustrates the state diagram of the activities we recognize.

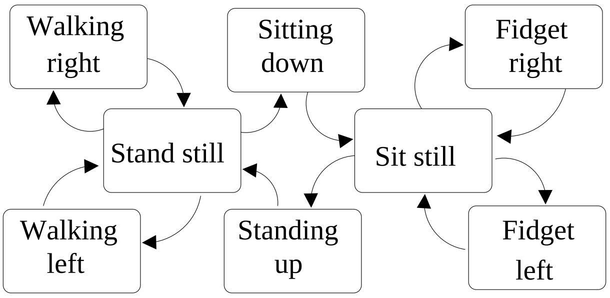

Figure 2: Example finite state diagram for a single person in a meeting.

The "Sit still" state represents instances when a person is sitting in their chair and is stationary. The "Fidget left" and "Fidget right" states represent motion of those people in their chair, such as if they look around or lean to one side to talk to someone. The "Standing up" and "Sitting down" states represent the transitions from a "Sit still" and "Stand still" state. These are the actual activities involved with getting up from one's chair and taking one's seat, respectively. Finally, once a person is in the "Stand still" state, they can "Walk left" and "Walk right".

We use the Viterbi algorithm [7] to infer the person's state given the sequence of data from the detected face positions. Viterbi is a dynamic programming algorithm which takes into account the state transition probabilities to generate the most likely state sequence that could have generated the observation sequence. This is more formally defined as:

$$q_{1:t} = arg \max_{q_{1:t}} P(q_{1:x}|y_{1:t}) \tag{1}$$

where q_t is the most likely state at time t. Thus, for each timestep, Viterbi computes a term $\delta_t(j)$, defined as:

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i) P(X_t = j|X_{t-1} = i)\right] P(y_t|X_t = j) \tag{2}$$

which is initialized as:

$$\delta_0(i) = P(X_0 = i)P(y_0|X_0 = i) \tag{3}$$

Additionally, the index of the most likely state at time $t-1$ to transition into each state at time t is computed and stored in a table $\psi_t(j)$ as follows:

$$\psi_t(j) = arg \max_i \left[\delta_{t-1}(i) P(X_t = j | X_{t-1} = i)\right] \qquad (4)$$

Finally, after the entire observation sequence has been analyzed in this fashion, the state sequence is obtained by backtracking over the table of ψ values ranging from $t = 0$ to $t = T$:

$$\begin{aligned} q_T &= arg \max_i \left[\delta_T(i)\right] \\ q_t &= \psi_{t+1}(q_{t+1}) \end{aligned} \qquad (5)$$

After the most likely activity state sequence has been computed for each tracked face, the location of the face is returned by the action recognition algorithm along with the specific activity name.

4.2 Meeting State Recognition

The global state of the meeting is determined by examining all of the states of the individual meeting participants. Allowable state transitions are defined as a first-order (fully-observable) Markov model which takes into account a minimum duration for a state transition. Let such a meeting model be defined as $M = \{S, T, D\}$, where S is the vector of allowable states, T is the transition matrix between states, and D is a minimum duration for being in that state. The state duration is useful to avoid noise in the model caused by the occasional misclassification of individual person states.

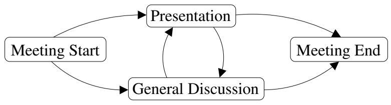

Figure 3: Example finite state machine showing the transitions for the global meeting state.

Figure 3 illustrates an example of a simple finite-state machine representation of the meeting Markov model. Each of these states is inferred from the actions of the people in the meeting. If at any point, an unknown state is is encountered, the classification system will report that this finite state machine is not valid. In this fashion, several different meeting finite state machines can be compared against the data to see which one best matches the meeting.

In order to evaluate the meeting-state classifier in a controlled fashion, we made use of our meeting simulator to generate extended test sequences. Data-driven simulation as a method for testing multi-agent learning and multi-agent state inference has been used very successfully in the domain of robot soccer [4] where the groups of small-size robots are simulated with very high-fidelity. The use of a simulator allows for careful control of the data so that different instances of events can be produced in any sequence.

The simulator was used to generate data from a meeting that consisted of three people, where one of the people gave a presentation. The presentation was preceded and followed by general discussion. The example Markov model in Figure 3 was used to classify the states of the meeting. Figure 4 shows the results of the meeting state classification. For this simple meeting model, all states were classified correctly.

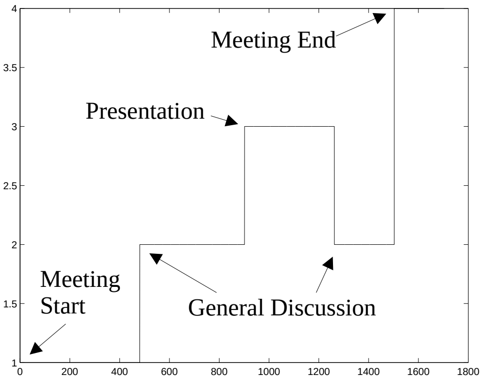

Figure 4: Classified states from a simulated meeting. State Meeting Start=1, General Discussion=2, Presentation=3, and Meeting End=4. The horizontal axis is time in seconds.

5. LIMITATIONS OF SPEECH- & VISION-ONLY SYSTEMS

5.1 Limitations of Meeting Structure Understanding using Only Speech

So far CAMSeg has focused on deriving meeting structure using information only in the audio channel. While most of the useful information is indeed in the speech channel—that is, in what is being *spoken*—there are some pieces of information that cannot be gleaned from this channel alone. In a face-to-face meeting, humans make use of non-verbal cues and gestures to communicate with each other. For example, a participant can address his utterance to another participant by merely looking at them. Purely speech based automatic methods to detect the addressee of an utterance would not be able to detect such an event (particularly if the addressee is silent). Similarly, people might agree or disagree with the speaker by merely nodding or shaking their head—again in this case the speech channel does not contain appropriate information.

5.2 Limitations of Meeting Structure Understanding using Only Vision

Meetings are primarily about the exchange of ideas between attending parties. Much of this information is exchanged verbally and as such a vision-only recognition system will miss this information completely. In our approach, vision is intended as a compliment to speech so that the the verbally-communicated information can be augmented with non-verbal cues such as an individual's body position, a coarse notion of physical activity, and relative proximity of each meeting attendee to another. If the individual components of a meeting can be characterized by human motion and changes in position, then visually recognizing these motions can assist in meeting state understanding. However, if the meeting states consist of topics of discussion where the individual participants do not visibly change positions or activities in any detectable fashion, then visual cues will not be of much assistance.

Visual methods tend to be somewhat brittle in how well they adapt from environment to environment. For instance, distance from the camera, lighting differences, and background patterns can all negatively affect the performance of pattern matching and recognition algorithms. Algorithms that adapt to their environment robustly will require additional computational resources to operate properly.

Another challenge comes from the kind of visual information that must be extracted about people in the scene. Extracting only faces from the scene is much less computationally expensive than attempting to extract complete body pose including information about the limbs, even though the latter information may be much more informative. Additionally, as the complexity of the information increases, the possibility for error also increases dramatically. As an example, if gesture information is to be extracted, but a pointing gesture returns incorrect data, the error may corrupt the inferred meaning and confound the final result dramatically.

5.3 Combating These Limitations

Combining audio and visual cues can help clarify meeting state where information from a single modality would be ambiguous. Audio cues can be used to identify specific meeting states [3] based on the speech activity. In contrast, visual cues can identify whether the meeting has actually started, or whether the participants are still entering the room and finding their chairs. They can also be used to differentiate between an information presentation and a formal presentation by determining whether the speaker is sitting at a table with the other attendees or whether they are standing at the podium [10]. Similarly the addressees of specific utterances can be discovered by following the gaze of the speaker from the video information, and by tracking his head movements and body gestures, more of their non-verbal communicative acts can be tracked.

6. USING INTEGRATION TO COMBAT LIMITED BANDWIDTH ISSUES

In order for an intelligent meeting system (such as a smart room) to passively record all aspects of speech and video in a meeting, many sensors are needed. In addition to microphones for each of the individual participants, in order to obtain good visual records of all of the participants, cameras must be placed in several locations to cover all activities. Each sensor needs a dedicated data cable and power run to it. The placement of sensors can be expensive and time-consuming, but an often overlooked aspect of computer vision and speech processing is the problem of computational resources necessary to process the resulting information. For real-time analysis, a single computer needs to be dedicated to each sensor (microphone, camera, etc.) Thus, as the number of sensors increases, the amount of computational hardware also increases. This greatly limits the utility of such an installation as it is permanently affixed to that room and can be very difficult to manage and upgrade. Additionally, such omnipresent sensors are not feasible or practical for "spontaneous" meetings which occur outside of the specially-equipped meeting room. Systems that integrate information across sensors and modes can potentially produce information equivalent in quality to systems that attempt the same within single modalities.

Our current research efforts are focused on the challenges involved with understanding how a light-weight sensor suite can capture enough useful information from a meeting. Because this involves a much smaller number of sensors, the system must be very careful in how it decides what aspects of the meeting to pay attention to. The following subsections describe several approaches by which a light-weight system can make the most efficient use of its limited resources. These include intelligently deciding how to maximize CPU usage from ad-hoc computational resources, actively focusing the attention of the available sensors, and using information from the meeting to learn how best to follow its dynamics.

6.1 Ad-Hoc Computational Resources

The idea of integration can be extended to the computing devices that are brought together for a meeting, to ensure that the overall system is still capable of recording the important details aspects of the meeting. We assume that each person has their own laptop computer or PDA that is capable of recording speech from a head-mounted microphone. Additionally, we assume that there exists at least one wide-angle camera system such as a CAMEO [8] panoramic imaging system. The CAMEO can be run from another laptop computer and can be treated like a speaker phone in how it is set up for use in a meeting scenario.

Instead of relying on a fixed set of computers to do all of the information processing, a lightweight meeting analysis system should be able to use the available CPUs as efficiently as possible. This includes dynamically identifying available CPUs and recruiting them to be part of a collective pool. Thus, the number of networked computers that are brought to the meeting can be used to create an ad-hoc distributed compute server where spare CPU cycles from idle machines are used to process important data as needed. This technique has been successfully demonstrated in [6], a CORBA-based distributed software system that dynamically balanced its processing load by migrating processes among available CPUs.

6.2 Active Focus of Attention

The high bandwidth needed to process the activities of each person in a panoramic video stream makes it desirable to have a method that identifies where important activity is occurring; this information can be used to focus processing on those components of the video stream that are of greatest interest, reducing processing load. An "active" approach to meeting physical awareness that combines both bottom-up and top-down information processing can focus the attention of the sensors to where they are needed most. The bottom-up information processing consists of low CPU-usage algorithms as a mechanism to decide where "interesting activity" is occurring so that the computationally more expensive algorithms can be brought to bear on those data. As an example, simple morphological operations such as the motion of objects in the images (easily computed through frame differencing) can be used to identify where physical activity is taking place. Additionally, low CPU usage speech processing can be used to detect speech activity or to monitor for a small vocabulary of keywords that are relevant to the meeting agenda (provided *a priori*). The top-down information processing uses the information obtained from high-CPU usage algorithms to decide whether the specific conversation being observed is still relevant. More details on these two components are included in the discussion below.

6.3 Adapting to Meeting Dynamics

The limited focus of attention that existing sensors can bring to bear on the meeting invariably means that some information will be missed. Our goal is to identify techniques that will minimize the amount of information that is missed. When the system is uncertain as to where to focus its attention, it can "sample" from the existing conversations to decide which ones to pay the most attention to. A plausible solution is to set up a scheduling system by which the attention of the system rotates among several different candidates. This approach has been successfully demonstrated on a distributed robotic surveillance system which had to contend with limited bandwidth communication resources [9]. As interesting conversations occur and top-down information processing helps to identify the speaker and conversation roles, the scheduler can prioritize (or bias) certain conversations over others. This will cause those to be revisited more often. If a highly weighted conversation suddenly loses relevance, that conversation's weight can drop accordingly.

7. CONCLUSIONS

We have described our research efforts to integrate information obtained from speech and vision to perform meeting understanding. This includes light-weight algorithms to identify speaker roles from conversational cues, as well as algorithms to identify physical activities from single people as well as groups. Finally, we have described some of the real-world integration problems that must be addressed when putting together such a system and have discussed some possible solutions that we are currently exploring.

8. ACKNOWLEDGMENT

This research was supported by the National Business Center (NBC) of the Department of the Interior (DOI) under a subcontract from SRI International. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, by the NBC, DOI, SRI, or the US Government.

9. REFERENCES

[1] S. Banerjee, J. Cohen, T. Quisel, A. Chan, Y. Patodia, Z. Al-Bawab, R. Zhang, P. Rybski, M. Veloso, A. Black, R. Stern, R. Rosenfeld, and A. I. Rudnicky. Creating multi-modal, user-centric records of meetings with the Carnegie Mellon meeting recorder architecture. In *Proceedings of the ICASSP Meeting Recognition Workshop*, Montreal, Canada, 2004.

[2] S. Banerjee, C. Rose, and A. I. Rudnicky. The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the Tenth International Conference on Human-Computer Interaction*, Rome, Italy, September 2005.

[3] S. Banerjee and A. I. Rudnicky. Using simple speech-based features to detect the state of a meeting and the roles of the meeting participants. In *Proceedings of the 8th International Conference on Spoken Language Processing (Interspeech 2004—ICSLP)*, Jeju Island, Korea, 2004.

[4] B. Browning and E. Tryzelaar. ÜberSim: A multi-robot simulator for robot soccer. In *Proceedings of the Second International Joint Conference on Auton omous Agents and Multiagent Systems (AAMAS)*, pages 948–949, Melbourne, Australia, July 2003.

[5] M. Hearst. TextTiling: Segmenting text into multi–paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[6] C. P. McMillen, K. Stubbs, P. E. Rybski, S. A. Stoeter, M. Gini, and N. Papanikolopoulos. Resource scheduling and load balancing in distributed robotic control systems. *Robotics and Autonomous Systems*, 44:251–259, 2003.

[7] L. R. Rabiner. A tutorial on Hidden Markov Models and selected application s in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[8] P. E. Rybski, F. de la Torre, R. Patil, C. Vallespi, M. M. Veloso, and B. Browning. Cameo: The camera assisted meeting event observer. In *International Conference on Robotics and Automation*, New Orleans, April 2004.

[9] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos. Performance of a distributed robotic system using shared communications channels. *IEEE Transactions on Robotics and Automation, special issue on Multi-Robot systems*, 18(5):713–727, October 2002.

[10] P. E. Rybski and M. M. Veloso. Using sparse visual data to model human activities in meetings. In *Workshop on Modeling Other Agents from Observations, International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.