

Combining Audio and Video to Predict Helpers' Focus of Attention in Multiparty Remote Collaboration on Physical Tasks

Jiazhi Ou, Yanxin Shi, Jeffrey Wong, Susan R. Fussell, Jie Yang

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

{jzou, yanxins, jeffreyw, sfussell, yang+}@cs.cmu.edu

ABSTRACT

The increasing interest in supporting multiparty remote collaboration has created both opportunities and challenges for the research community. The research reported here aims to develop tools to support multiparty remote collaborations and to study human behaviors using these tools. In this paper we first introduce an experimental multimedia (video and audio) system with which an expert can collaborate with several novices. We then use this system to study helpers' focus of attention (FOA) during a collaborative circuit assembly task. We investigate the relationship between FOA and language as well as activities using multimodal (audio and video) data, and use learning methods to predict helpers' FOA. We process different modalities separately and fusion the results to make a final decision. We employ a sliding window-based delayed labeling method to automatically predict changes in FOA in real time using only the dialogue among the helper and workers. We apply an adaptive background subtraction method and support vector machine to recognize the worker's activities from the video. To predict the helper's FOA, we make decisions using the information of joint project boundaries and workers' recent activities. The overall prediction accuracies are 79.52% using audio only and 81.79% using audio and video combined.

Categories and Subject Descriptors

H5.3. Information interfaces and presentation (e.g., HCI): Group and organizational interfaces – collaborative computing, computer-supported collaborative work

General Terms

Algorithms, Experimentation, Human Factors, Languages.

Keywords

Focus of Attention, Computer-Supported Cooperative Work, Multimodal Integration, Remote Collaborative Physical Tasks

1. INTRODUCTION

Physical collaborative tasks are tasks in which two or more people interact with 3D objects in the real world. In this research, we

focus on instructional tasks, in which *helpers* offer expertise and guidance to *workers* who perform the actual tasks. For example, in distance education, a physics teacher might use video and audio to simultaneously instruct several students located in different remote classrooms on how to do electronic experiments.

Most previous studies of remote physical collaboration focus on two-party collaboration (e.g., [8][11][13][14][15]). To extend this work to multiparty collaboration, we face many new challenges. Focus of attention (FOA) is one example. FOA is a perceptual variable that indicates the action, object, or person to which someone is attending [2]. In two-party collaborations, to identify FOA, one must figure out on which area in a visual space the helper is focusing (e.g., instruction manual, task objects, worker's face). In a multiparty collaboration, the helper also switches FOA back and forth among different workers. In this paper, we examine helpers' allocation of FOA among multiple workers.

The study of FOA in remote multi-party collaboration has both theoretical and practical impacts. By analyzing a helper's FOA, we can understand how he/she allocates time and attention among multiple collaborators. Although scene-oriented video systems have been proven very useful in remote two-party collaboration on physical tasks ([8][11][13]), problems such as bandwidth limitations and limited display size may arise when one helper assists multiple remote workers simultaneously. As a result, the helper may not be able to see all workers' workspaces at the same time. A solution is to allow the helper to switch views among workspaces as needed, such that the focal workspace is allocated maximal bandwidth and display space. If a system can predict the helper's desired FOA in real time, it can automatically switch display windows to provide the right visual information at the right time.

Much research has been directed to tracking and analyzing FOA in other domains. Although a person's visual attention is not necessarily his/her actual FOA, eye gaze or head pose has been commonly used as a good estimation of FOA ([5] [17]). However, commercial eye trackers are expensive and intrusive. Some research systems are non-intrusive but less accurate. There is some related work on predicting FOA in multi-party face-to-face meetings (e.g., [12][18]). In Stiefelhagen et al.'s work [18], the inputs are the video from a panoramic camera and/or audio, and the output is the participants' FOA in the meeting.

We are interested in predicting FOA from modalities other than gaze. In physical collaborative tasks, the hypothesis that a person's FOA is predictable from other modalities is in the spirit of research that has showed the strong correlation between the gaze of attention and task property, conversational content, and actions [15]. The work that is the most related to this paper is the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMI '06, November 2–4, 2006, Alberta, Canada.

Copyright 2006 ACM 1-59593-541-X/06/0011...\$5.00.

FOA prediction in two-party remote collaboration studied by Ou et al. [14], who designed an online puzzle task in which a helper instructed a remote worker on how to assemble puzzles. The helper’s FOA was predicted in real-time from conversational content and the worker’s actions with accuracies ranging from 69.81% to 76.62%, depending on the difficulty of the puzzle. The problem discussed in this paper differs from most previous research in that it examines one-to-many remote communication.

The remainder of the paper is organized as follows: in Section 2 we describe the research problem and related work. In Sections 3, 4, and 5, we present our algorithms for identifying addressees, recognizing workers’ activities, and performing multimodal integration. We report experimental results in Section 6 and discuss general conclusions and future work in Section 7.

2. PROBLEM DESCRIPTION AND RELATED WORK

In this research, our goal is to develop a multimedia system that supports multiparty remote collaborations on physical tasks in which a helper (expert) assists multiple workers (novices). The system is designed as follows. The helper and workers share audio communication channels. Each worker’s workspace is captured by a video camera and the video stream is fed into the helper’s workstation. To save network bandwidth, only one workspace is displayed in full resolution at a time. In our current setup, the helper can switch manually among views of different workers’ workspaces. However, the switching process interferes with helpers’ ability to provide instructions. Thus, we aim to develop a mechanism that enables the system to automatically switch among views of different workspaces based on a helper’s FOA. Furthermore, we aim to predict the helper’s FOA from dialogue between the helper and workers and from the workers’ activities.

In this section, we first describe our prototype system that allows a helper to switch among views of different workspaces by a mouse click. We will use this system to collect the ground truth: multimodal data streams in remote multiparty collaboration that will be used to build models to predict a helper’s FOA. We then describe several issues related to predicting FOA in real time. At the end of this section, we give an overview of the algorithm we use to combine the inputs from different sources to predict FOA.

2.1 Multiparty Remote Collaboration

2.1.1 Experimental System

In order to study multiparty collaboration, we built an experimental system that provides shared audio communication and separated video communication. A helper can select a display window using a mouse click. The system shows a live video stream from the selected workspace in the main window at high resolution (640 by 480 pixels). Video streams from other workspaces are displayed in much smaller windows (48 by 36 pixels) at the bottom of the screen. The helper can get a vague sense of what is happening in the non-focal workspaces (e.g., whether there is movement present) by observing the small videos. To switch views, the helper clicks on the desired video window. A party-line audio system allows a conversation between the helper and one worker to be overheard by the other worker. In the current study, we simulate the scenario of a service center – a technical support staff member (the helper) might get calls from several customers (the workers) and help them simultaneously.

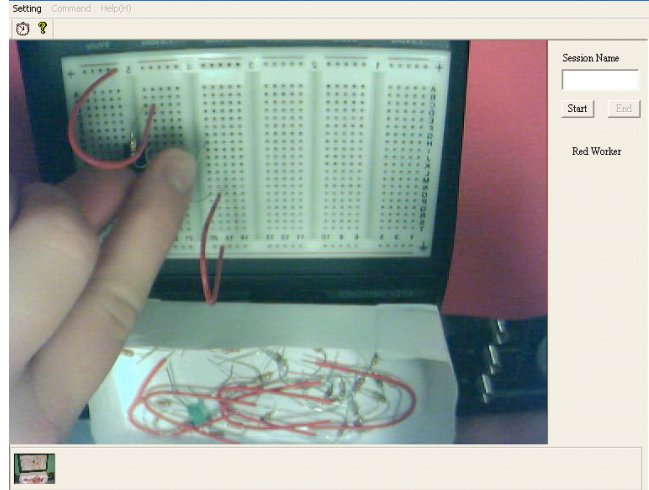


Figure 1. A video system for multiparty remote collaboration. The large window shows the video of the focal worker’s workspace. The other worker’s video is shown in the small window at the bottom left. Helpers can switch views by clicking on the small window. The workspace appears at the top of the video and the pieces bay at the bottom.

Figure 1 shows a screen shot of the interface. The task is to assemble an electronic circuit on a breadboard. The elements include IC chips, resistors, LEDs, capacitors, and wires. Similar to [11][15], the shared visual information can be divided into two areas: a *pieces-bay* that holds all the available elements and a *workspace* on which the circuit is assembled.

Although visual attention might not always match FOA (e.g., a helper might talk to one worker while monitoring the activity of another worker), for our current purposes, we define the helper’s FOA as the workspace selected in the main window.

2.1.2 Data Collection

The data used in this paper are from 12 sessions of collaborative tasks with a total number of 36 participants. In each session there were three participants: a helper with background knowledge in circuit assembly, and two workers without circuit assembly experience. Each worker in a trial was asked to assemble a different circuit, as quickly and accurately as possible. Participants were in the same room but visually separated by barriers such that they could talk freely but not see each other. The helper used our video system to instruct both workers simultaneously. Workers’ tables were colored with either red or pale green so that the helper could easily differentiate them. Task time ranged from 8 minutes to 21 minutes. For each session, we recorded the conversation, the video streams from each workspace, and, for the helper, which video stream was selected as the FOA.

2.2 Research Issues

The focus of this paper is to explore the possibility of predicting the helper’s FOA from speech and video, which is related to the following four issues.

2.2.1 Joint Projects

Clark defines a conversation between two participants as a *joint project*, which consists of an entry, a body, and an exit process ([3][7]). In multi-party collaborations, a joint project is entered

when the helper starts communicating with one worker and ends when the helper switches and starts instructing another worker. An example is shown in Figure 2. (Here, we refer to the workers by their table color: *red worker* and *green worker*.)

[A joint project is entered]
 Helper: And for the green worker, can you plug the IC chip 4049 in the middle, across column 25 and column 26 with the notch facing upward as well.
 Green Worker: Okay.
 [The joint project is exited, a new joint project is entered]
 Helper: Okay, red worker well done. Okay. Can you connect can you find a resistor that is yellow ...

Figure 2. An example of switching joint projects. A joint project between a helper and the green worker is entered when they start talking and exited when the helper addresses the red worker.

Because there is no communication between the two workers, the joint project in this task determines the worker with whom the helper is interacting. Research on face-to-face communication has indicated that a person’s gaze is closely linked to the speaker or addressee. Argyle estimated that in two-person conversations, people look almost twice as much while listening (75%) compared to speaking (41%) [1]. Vertegaal et al. found that in multi-party conversations, speakers looked at the person they were talking to 77% of the time and listeners looked at the speaker 88% of the time [19]. We hypothesize that in remote collaboration, the switch of joint projects is correlated with FOA.

2.2.2 Online Segmentation and Identification

The multi-party collaboration process is a continuous stream of joint projects. Identifying joint projects will thus help us understand the communication structure. However, a joint project is a higher level unit, encompassing words and utterances, and not readily available as an input. We do not have information about the words after each time point because they have not been spoken. In this paper we address the problem of how to segment and identify joint projects in real time. This problem is related to automatic text or dialogue segmentation, which chops a stream of words (in text or conversation) into individual units, which can be topics [4], discourse [9], and dialogue-acts [10]. This is addressed in two ways. Each word is directly classified as one of these predefined units (e.g. [10]). This classification relies on topic-specific words across the body of the unit. Alternatively, each word position can be classified as either a boundary, which can be unit specific, or a non-boundary ([4][9]). This process relies on the features surrounding the boundary. Because the physical tasks of different workers are similar (but not identical), we opt for the latter method – segmenting the joint projects and then identifying them with the boundaries.

Topic segmentation has been studied over the last decade. Beeferman and colleagues proposed a feature-based exponential model for topic segmentation in text [4]. Features include the likelihood from an adaptive language model, cue words that appear in the *previous* few words or sentences, and cue words that appear in the *next* few words or sentences. A feature selection technique was applied to select the features with top most gains.

Galley et al. extended the idea of feature-based classification and added acoustic features such as silences, overlaps, and speaker change [9]. We call these strategies *offline segmentation* because given a potential boundary, features can be extracted from both *before* and *after* it.

Performing segmentation and identification online, using only information *before* the current word, is very different from offline segmentation. Take [4]’s WSJ features as an example: The top 5 features are whether a specific word (“INCORPORATED,” “SAYS,” etc.) appears *after* the current word, which means that this *future* information is most discriminative in classifying a boundary or non-boundary. If only the previous information can be used, the boundary word (with label 1) will not contain these important words and the non-boundary words afterwards will include them as features and consequentially a classifier will not make use of these words to classify a boundary. To overcome this problem, we propose a sliding window-based delayed labeling method, which will be discussed in detail in Section 3.

2.2.3 Multimodal Integration

Because our experimental system is video-based, we can use multimodal input to predict the helper’s FOA. The workers’ workspaces and activities are captured by the video. By analyzing the video, the worker’s actions can be extracted. Ou et al. showed that the worker’s actions are important clues to FOA in two-party remote collaboration [14]. In this paper we extend these findings to multi-party collaboration.

2.2.4 Problem Definition and Algorithm Overview

Let $\{(w_1, id_1), (w_2, id_2), \dots, (w_N, id_N)\}$ be a sequence of words, where w_i is the i th spoken word with the speaker identification id_i ; $\{\{frame_{11}, frame_{12}, \dots, frame_{1T}\}, \{frame_{21}, frame_{22}, \dots, frame_{2T}\}, \dots, \{frame_{D1}, frame_{D2}, \dots, frame_{DT}\}\}$ be the sequences of frames sampled from the workers’ video, where $frame_{dt}$ is the d th worker’s frame over the working area at time t ; and $\{g_1, g_2, \dots, g_N\}$ be the sequence of the helper’s FOA, which is defined as the worker whose video appearing in the main display, right after word w_i . The goal is to predict g_i from $\{(w_1, id_1), (w_2, id_2), \dots, (w_i, id_i)\}$ and $\{\{frame_{11}, frame_{12}, \dots, frame_{1t}\}, \{frame_{21}, frame_{22}, \dots, frame_{2t}\}, \dots, \{frame_{D1}, frame_{D2}, \dots, frame_{Dt}\}\}$, where t is the time before word w_i . The error rate of speech recognition varies depending on users and environment. In the current study we focus on the effect of message content on FOA and w_1, \dots, w_N are transcribed words. To align the words with the helper’s FOA and workers’ activities we ran a speech recognizer to label their exact starting and ending time.

We address this problem by breaking it into three components (Figure 3): online joint project segmentation and identification (OSI), worker activity recognition, and multimodal integration. The OSI module takes the word stream as input, determines at each word w_i whether there is a joint project boundary, and outputs the identity of the joint project jp_i as one of the workers, $jp_i \in \{worker_1, worker_2, \dots, worker_D\}$. The worker activity recognition module recognizes the recent activity for each worker. The multimodal integration module combines the outputs of the OSI and activity recognition modules and generates the predicted FOA at that time. Details of each module will be presented in the following sections.

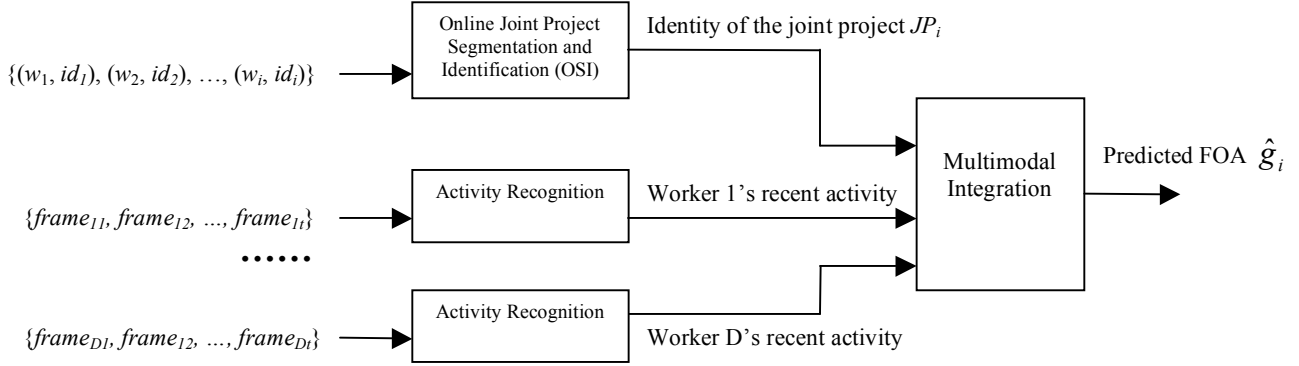


Figure 3. An Overview of the multimodal FOA prediction algorithm. w_i is the i th spoken word with its speaker identification id_i ; $\{frame_{d1}, frame_{d2}, \dots, frame_{dL}\}$ is the video sequence of the d th worker.

3. ONLINE JOINT PROJECT SEGMENTATION AND IDENTIFICATION

3.1 A Sliding Window Based Delayed Labeling Method

A number of statistical methods for text segmentation are based on a window method – for each potential boundary B_i , features F_i are extracted from a window of words surrounding it, and the label L_i is set to 1 (boundary) or 0 (non-boundary). The set $\{(F_i, L_i)\}$ is then used to train/test a classifier. This method can not be directly applied to online joint project segmentation, because (a) the words after the current word are not available yet, and (b) the characteristic words for different joint project are statistically similar so that we can only detect boundaries instead of topics, leading to a label bias problem (the number of boundaries are relatively small compared with non-boundaries). We propose a new delayed labeling method to overcome these problems.

In each word position w_i we define a window win_i , which contains the information of L words in the history (see Figure 4):

$$win_i = \{\{w_i\}, \{w_{i-1}, jp_{i-1}, id_{i-1}\}, \dots, \{w_{i-L+1}, jp_{i-L+1}, id_{i-L+1}\}\},$$

where w_i is the current word, and $\{w_{i-k}, jp_{i-k}, id_{i-k}\}$ is a triplet for word w_{i-k} where id_{i-k} is the speaker identity and jp_{i-k} is the predicted joint project.

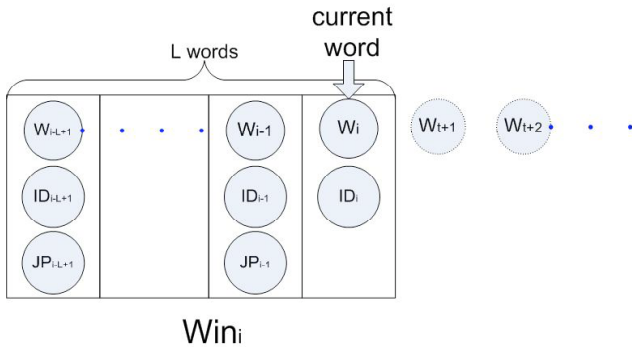


Figure 4. The window for current word.

The labeling function of win_i is defined as:

$$label(win_i) = \begin{cases} 1, & \text{if } \exists k, 0 \leq k \leq L-1, w_{i-k} \text{ is a boundary word.} \\ 0, & \text{otherwise} \end{cases}$$

In other words, we *delay* the label of a boundary word w_i to L consecutive words (which are all 1s).

We use a Support Vector Machine (SVM) as the classifier. To train the SVM, we define N features and the N -dimension attribute vector of the current window win_i is:

$$attr_vector(win_i) = (f_1(win_i), f_2(win_i), \dots, f_N(win_i)).$$

By sliding the window we get the pairs $\{attr_vector(win_1), label(win_1)\}, \dots, \{attr_vector(win_M), label(win_M)\}$, which are used to train a classifier.

The trained classifier is used to predict the current joint project online. Given a word w_i at position i , we first use the window based method to generate the attribute vector and input it to the classifier, which outputs whether a boundary has occurred within the window. We then assign a joint project ID jp_i to that word based on predicted boundary labels and the joint project ID of its previous word jp_{i-1} . After that we slide the window one word forward to predict the joint project for the next word. For the first word we can assign a random label to it. (See Figure 5 for details.)

In our algorithm, the joint project prediction error for one word will not be propagated, since our online algorithm includes the predicted joint project for the previous word as one *feature* in the attribute vector for SVM transition prediction for the current word. In this way, for example, if w_i is a word in worker 1's joint project, and predicted as in worker 2's joint project, then at the first true transition after w_i , our algorithm will not predict this transition, since its input attribute vector "tells" it that current joint project is "joint project of worker 1". After this false prediction of a transition, the joint projects for the following words will be corrected. The random assignment for the first word is an example beneficiary of this *stable model*. We can randomly assign the joint project to the first word. If this assignment is wrong, it will be rectified when the algorithm sees the feature words of the correct joint project.

3.2 Feature Selection

There are two types of features we used to segment and identify joint projects: lexical features and non-lexical features. Bi-grams

are used as lexical features:

$$f_{\langle w_a, w_b \rangle}(win_i) = \begin{cases} 1, & \text{if bigram } \langle w_a, w_b \rangle \text{ exists in } win_i \\ 0, & \text{otherwise} \end{cases}$$

To avoid over-fitting, we collected all possible bi-grams from the training data and sorted them by information gain:

$$IG(f) = H(Label) - H(Label | f),$$

where $H(Label)$ is the entropy of the label and $H(Label | f)$ is the conditional entropy. Bi-grams having the top N information gains were chosen as lexical features.

OSI Algorithm

Training:
 Feature selection based on information gain;
 Generate attribute vectors $attr_vector(win_i)$ and $label(win_i)$ sliding window based delayed labeling method;
 Use $\{attr_vector(win_i), label(win_i)\}$ pairs to train an SVM
 Output: Learned model, selected features

Prediction:
 Input: Learned model, selected features functions, $\{(w_1, id_1), \dots, (w_i, id_N)\}$
 Read w_i .
 Assign a random joint project, jp_i to w_i :
 $i := 2$
 Do until all words have been processed:
 Read w_i ;
 Generate attribute vector, $attr_vector(win_i)$, by window method;
 Predict by SVM, $Label_i=1$ if the window contains the transition boundary, $Label_i=0$, otherwise;
 Assign JP_i to w_i by:
 If $JP_{i-1} == worker\ 1$
 If $Label_{i-1}=0$ and $Label_i=1$
 $JP_i == worker\ 2$
 else
 $JP_i == worker\ 1$
 end if
 else
 If $Label_{i-1}=0$ and $Label_i=1$
 $JP_i == worker\ 1$
 else
 $JP_i == worker\ 2$
 end if
 end if
 $i := i + 1$
 Output : $\{JP_1, \dots, JP_N\}$

Figure 5. The online joint project segmentation and identification (OSI) algorithm.

There are two non-lexical features, f_{inter} and f_{lastJP} :

$$f_{inter}(win_i) = \begin{cases} 1, & \text{if } \exists k \text{ in } win_i, id_k \neq jp_k \\ 0, & \text{otherwise} \end{cases}$$

$$f_{lastJP}(win_i) = jp_{i-1}$$

Examples of selected features are presented in Section 6.1 The OSI algorithm for the one-helper-two-worker condition is summarized in Figure 5.

4. WORKER ACTIVITY RECOGNITION

Workers' activities in the circuit assembly task can be classified into one of three categories—idle, searching/picking up a part, and assembling a part on the breadboard—based on hand position in the video: none, pieces-bay, and workspace. (Figure 6)

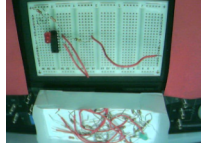
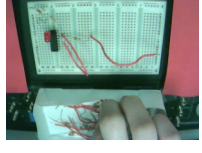
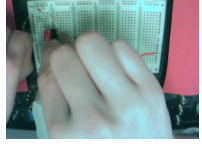
Sample Frame	Worker's Activity	Hand Position
	Idle	None
	Searching/picking up a part	Pieces-bay
	Assembling a part	Workspace

Figure 6. Three categories of worker activity with corresponding hand positions.

We applied background subtraction and modeled the background as a dynamic average of the previous frames:

$$B_t = \alpha * F_t + (1 - \alpha) * B_{t-1},$$

where B_t is the updated background after the t th frame F_t is processed. α is the learning rate ($\alpha = 0.1$ in our experiments).

Given a new frame F_{t+1} , we first compute its foreground image FG_{t+1} by subtracting the background B_t :

$$FG_{t+1}(i, j) = \begin{cases} 1, & \text{if } |F_{t+1}(i, j) - B_t(i, j)| > \text{thred} \\ 0, & \text{otherwise} \end{cases}$$

Because an activity category depends on hand position (upper/lower/none) on the screen, we sum over the row elements of FG_{t+1} and use the resulting vector as the feature vector of the input frame:

$$col_vector(F) = (\sum_j FG(1, j), \dots, \sum_j FG(h, j)),$$

where h is the image height.

We then train a classifier with the feature vectors and corresponding labels (*Idle/Searching/Assembling*). The trained model is used to recognize the worker's activity. Again we use SVM as the classifier.

5. MULTIMODAL INTEGRATION

The OSI identifies joint projects immediately after a word is spoken and provides information about the worker with whom the helper is interacting. A naïve way to predict FOA is to directly map the output of OSI to FOA, i.e., the prediction of FOA is changed whenever there is a change of joint projects. However, as noted earlier, a helper might visually attend to one worker while giving instructions to the other. This strategy might make collaboration more efficient when a helper assists multiple workers at the same time. In three out of the first five sessions, the number of FOA segments is only half the number of joint projects (see Table 1). In this section we present how we combined identified joint projects and workers' activities to predict FOA.

Table 1. Number of joint projects and FOA segments for the first five sessions in the study (with different participants).

Session ID	# of Joint Projects	# of FOA Segments
S01	24	12
S02	14	6
S03	14	20
S04	25	10
S05	30	39

Because we did not have sufficient data to train sophisticated models such as hidden Markov models, we proposed a simple *Winner-Takes-All* strategy: when a switch of joint project is detected, we take the workers' activities as inputs, predict the majority of the FOA in the started joint project, and label the FOA in the rest of the joint project as this majority. To train/test a classifier, we use features that summarize the recent activities of all workers at word w_i .

$$\text{act_vector}(w_i) = (P_{1_Idle}, P_{1_Searching}, P_{1_Assembling}, \dots, P_{D_Idle}, P_{D_Searching}, P_{D_Assembling}),$$

where $P_{d_Activity}$ is the percentage of time the d th worker is in Activity (*Idle*, *Searching*, or *Assembling*) over the last 10 seconds.

We use a KNN classifier (K=3 in the experiments), which is memory based and effective when the size of training data is small. The distance metric is CHI-Square:

$$\text{CHI-Square} = \sum_r \frac{(a_i - b_i)^2}{|a_i + b_i|}.$$

This algorithm is summarized in Figure 7.

6. EXPERIMENTAL RESULTS

We use the data collected from five sessions of the experiments to evaluate two individual modules (OSI, activity recognition) and the final FOA prediction performance. There are a helper and two workers in each session ($D = 2$).

6.1 Joint Project Segmentation/Identification

Experiments were conducted with a 12-fold cross validation: Each time, four sessions were used to train the model and the last session was used as the testing data. The SVM classifier was implemented by Chang and Lin [6]. The true joint project label for each word was coded by hand.

The multimodal integration algorithm

Training:

Do until all joint projects have been processed:

 Generate $\text{act_vector}(w_i)$, where w_i is the ending word of the joint project;

 Compute the majority of the FOA in the next joint project;

 Save the pair ($\text{act_vector}(w_i)$, majority) into the memory

Prediction:

$i := 1$

Do until all words have been processed:

 Read w_i ;

 Call OSI and get the predicted joint project jp_i ;

 if $jp_i \neq jp_{i-1}$

 Generate $\text{act_vector}(w_i)$;

 Find the majority of FOA for the next joint-project use KNN

 Label the FOA $\hat{g}_i = \text{majority}$

 else

$\hat{g}_i = \hat{g}_{i-1}$

 end if

Output: $\{ \hat{g}_1, \dots, \hat{g}_N \}$

Figure 7. The multi-model integration algorithm.

One way to evaluate text segmentation is using the *WindowDiff* method [16]. However, because our focus is on how accurately the algorithm can identify joint projects, we use the percentage of time that the algorithm correctly identifies whether the current joint project involves worker 1 or worker 2 as our evaluation metric. For our baseline, we chose an algorithm that always assigns the joint project to the worker with whom the helper interacted most frequently. As shown in Figure 8, our algorithm achieved a high accuracy in segmenting and identifying joint projects online, and was significantly better than the baseline ($t[11] = 18.49, p < 0.001$).

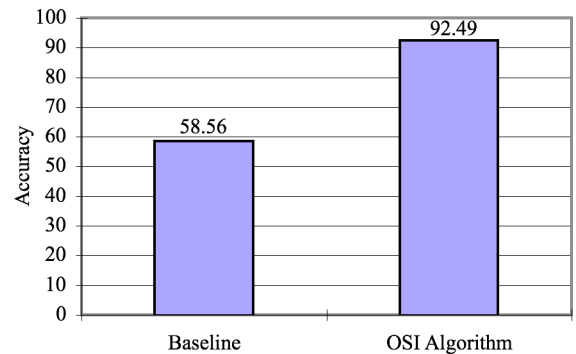


Figure 8. Accuracies of the OSI algorithm and the baseline.

To understand the algorithm's performance, we examined the lexical features automatically selected by OSI. Among 227 candidates (bi-grams that appear more than three times in the training corpus) we chose the 25 features with the highest information gain. The number of features selected was determined by cross validation. Table 2 shows the top 10 features and

indicates which ones are boundary words (indicating a boundary should exist L words before them) and which ones are not.

Table 2. Top 10 selected features.

Rank	Bi-gram	Type
1	“ <i>red worker</i> ”	Boundary
2	“ <i>for red</i> ”	Boundary
3	“ <i>for green</i> ”	Boundary
4	“ <i>and for</i> ”	Boundary
5	“ <i>green worker</i> ”	Boundary
6	“ <i>find the</i> ”	Boundary
7	“< <i>red/green</i> > <i>find</i> ”	Boundary
8	“ <i>and put</i> ”	Non-Boundary
9	“ <i>one of</i> ”	Non-Boundary
10	“ <i>step <digit></i> ”	Boundary

Helpers often used the color of the workers’ tables to address them. (e.g., “And now, the *red worker* ...”, “Step 2 for *green* ...”) When helpers wanted to switch joint projects, they usually started by calling for the attention of the worker to whom they wanted to speak by using “*red worker*”, “*green worker*”, “*for red*”, and “*for green*”. The bi-gram “*and for*” is the first two words in “and for red/green (worker)” and is a good marker of the entry into a new joint project. The physical task can be decomposed into several steps, each of which can be further divided into two sub-steps: searching for a part, and putting it in the right place. Therefore, “*step <digit>*”, “*find the*”, “<*red/green*> *find*” always appear in the beginning of a joint project (boundary word) whereas “*and put*” appears in the middle (non-boundary word). Figure 9 shows an example of how some of these bi-grams were used.

[The conversation with the last worker ends]
Helper: Now <i>Step 2 for red find the</i> brown, black, orange resistor.
[More details of the resistor]
Helper: <i>And put</i> one end on number 2 of the IC ...

Figure 9. A short excerpt of conversation. Italicized words are bi-grams selected as features. “step 2”, “for red”, “find the” are markers of the beginning a joint project and “and put” is a marker of being within a joint project.

6.2 Workers’ Activity Recognition

We collected 61572 images from 24 workers in 12 sessions. To test activity recognition performance, we applied 10-fold cross validation. Overall accuracy in predicting the three classes was 86.13%. The confusion matrix is shown in Table 3. The algorithm performs best in predicting the *Assembling* category, where the worker’s hand is positioned on the breadboard.

Table 3. The confusion matrix for activity recognition.

Predict \ True	Idle	Searching	Assembling
Idle	87.87%	4.01%	8.12%
Searching	16.40%	67.92%	15.68%
Assembling	5.56%	0.98%	93.45%

6.3 Focus of Attention Prediction

In this section we present the results of our FOA prediction algorithm. The inputs of the algorithm, as discussed in Section 2.2.5, are conversations between the helper and two workers and the workers’ video streams. We first used the OSI algorithm and activity recognition to extract higher level semantics from audio and video. We then combined the outputs, joint project ID and workers’ activity categories, to predict the helper’s FOA (either worker 1 or worker 2) online. The predicted FOA was then compared with the true worker ID, as indicated by which worker was in focus in the main window. As shown in Table 1, we found substantial variation among helpers—some switch FOA much more frequently than others. Thus, we built user dependent models: for each session we trained and tested the FOA prediction with 2-fold cross validation. Accuracy was measured by the percentage of time the predicted FOA matched the true FOA. We compared the performances of three models:

- The baseline, which always predicts the output as the more frequent worker
- Uni-modal method, which directly maps the predicted joint project to FOA
- The proposed multimodal algorithm

Overall accuracies are shown in Figure 10. Both the uni-modal ($t[11] = 2.20$, $p=0.05$) and multi-modal ($t[11] = 4.28$, $p=0.001$) methods were better than the baseline. The multi-modal method outperformed the uni-modal method ($t[11] = 1.98$, $p=0.07$).

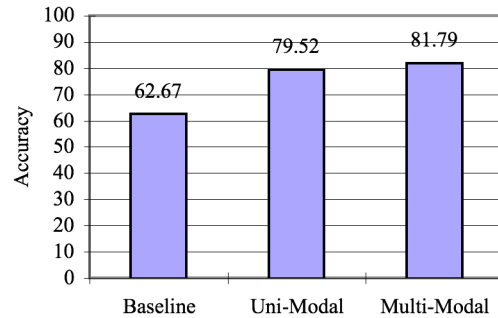


Figure 10. Accuracies of the baseline, uni-modal method, and multi-modal method.

Combining audio and video leads to significant improvement when the joint project and FOA do not match well. The proposed algorithm decides whether there is a change in FOA when a new joint project is entered by using video information. For example, a helper might not need to focus on a worker whose hands are idle, or might want to continue watching a worker to ensure that a step is completed correctly after a joint project is exited. When joint project and FOA do match, our algorithm predicts changes in FOA at every joint project transition. Since these helpers did in fact change FOA at every joint project transition, both the proposed and baseline algorithms work well.

We define an activity category as *dominant* if it comprised more than 66.6% of the sampled time period (the last 10 seconds). Because the number of FOA segments is smaller than the number of joint projects for S01, S02, and S04 (Table 1), when a new joint project was entered, the helper chose between switching FOA to the other worker or remaining with the current worker. We

calculated switches vs. holds of FOA for times in which the *Assembling* activity was dominant at a joint project transition for S01, S02, and S04. When helpers held FOA, it was much more likely that the worker's hand position was in the workspace than when they switched FOA to a new worker (Figure 11). This suggests that when a worker is busy assembling the circuit, the helper will continue gazing at that worker to ensure that the task is performed correctly, despite having switched to a new joint project with another worker. The classifier enhances the accuracy of FOA prediction by taking advantage of this phenomenon.

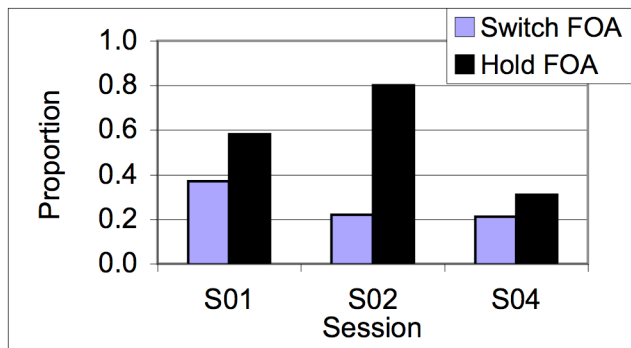


Figure 11. Proportion of switches and holds of FOA at joint project transitions in which *Assembling* activity is dominant.

7. CONCLUSIONS

In this paper we analyzed and predicted helpers' FOA in a multiparty remote physical collaborative task. We implemented a video based system to support one-to-many remote collaboration and collected multimodal data that we used to build FOA models. Dialogue and video were inputs to these models; and the helper's FOA was the output. We presented three modules: a sliding window based delayed labeling method to segment and identify joint projects online; an activity recognizer based on an adaptive background model and an SVM; and a memory-based multimodal integration algorithm. Experimental results showed that our joint project identification (with an overall accuracy of 92.45%) and worker's activity recognition algorithms (with an overall accuracy of 86.13%) were reliable. By combining their outputs, the multimodal integration algorithm achieved an accuracy of 81.79% in predicting the helper's FOA online.

There are several limitations to our study, which we plan to address in future work. First, we used party-line audio rather than a private line system. How the results will differ in a private-line system has to be explored. Second, because patterns of FOA vary among users, our FOA prediction model parameters are user dependant. When a new user comes, user enrollment and adaptation can be applied in practice. Finally, we make decisions only about joint project boundaries. With more data we can try to build finer-grained models to predict FOA.

8. ACKNOWLEDGEMENTS

This research was funded by National Science Foundation grants #0208903 and #0329077. We thank Joseph Illoreta, Shannon Pereira, and Victoria Yew for help running experiments.

9. REFERENCES

- [1] Argyle, M. & Cook, M. (1976). *Gaze and Mutual Gaze*. Cambridge University Press.

- [2] Augmented Multi-party Interaction, Recognition of Attentional Cues in Meetings, *State-of-the-art overview, Annual Reports*.
- [3] Bangerter, A., Clark, H. H., & Katz, A. R. Navigating joint projects in telephone conversations. *Discourse Processes*, 37, 1-23.
- [4] Beeferman, D., Berger, A., & Lafferty, J., Statistical models for text segmentation. *Machine Learning*, 34, 177-210.
- [5] Campbell, C. S. & Maglio, P. P. (2001). A robust algorithm for reading detection. In *Proceedings of PUI '01*.
- [6] Chang, C. & Lin, C., (2001). LIBSVM : a library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [7] Clark, H. H. (1996). *Using language*. Cambridge, UK: Cambridge University Press.
- [8] Fussell, S. R., Kraut, R. E., & Siegel, J. (2000). Coordination of communication: Effects of shared visual context on collaborative work. *Proceedings of CSCW 2000* (pp. 21-30). NY: ACM Press.
- [9] Galley, M., McKeown, K., Fosler-Lussier, E., Jing, H. (2003). Discourse segmentation of multi-party conversation. *Proceedings of ACL-03*.
- [10] Ivanovic., E., (2005). Automatic utterance segmentation in Instant Messaging dialogue. *Proceedings of the Australasian Language Technology Workshop* (pp. 241-249).
- [11] Kraut, R. E., Gergle, D., & Fussell, S. R. (2002). The use of visual information in shared visual spaces: Informing the development of virtual co-presence. *Proceedings of CSCW 2002* (pp. 31-40). NY: ACM Press.
- [12] Otsuka, K., Takemae, Y., Yamato, J., & Murase, H. (2005). A probabilistic inference of multiparty-conversation structure based on Markov-switching models of gaze patterns, head directions, and utterances. In *Proceedings of ICMI 2005*.
- [13] Ou, J., Fussell, S. R., Chen, X., Setlock, L. D., & Yang, J. (2003). Gestural communication over video stream: Supporting multimodal interaction for remote collaborative physical tasks. In *Proceedings of ICMI 2003*.
- [14] Ou, J., Oh, L. M., Fussell, S. R., Blum, T., & Yang, J. (2005). Analyzing and predicting focus of attention in remote collaborative tasks. *Proceedings of ICMI '05*.
- [15] Ou, J., Oh, L. M., Yang, J., & Fussell, S. R., (2005). Effects of task properties, partner actions, and message content on eye gaze patterns in a collaborative task. *Proceedings of CHI-2005*.
- [16] Pevzner, L. & Hearst, M. A., A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19-36.
- [17] Salvucci D. (1999). Inferring intent in eye-based interfaces: tracing eye movements with process models. *Proceedings of CHI 1999*.
- [18] Stiefelhagen, R., Yang, J., & Waibel, A. (2002). Modeling focus of attention for meeting indexing based on multiple cues. *IEEE Transactions on Neural Networks*, 13, 928-938.
- [19] Vertegaal, R., Slagter, R., van der Veer, G., & Nijholt, A. (2001). Eye gaze patterns in conversations: There is more to conversational agents than meets the eyes. *Proceedings of CHI 2001* (pp. 301-308). NY: ACM Press.