Structural and Temporal Inference Search (STIS): Pattern Identification in Multimodal Data

Chreston Miller
Center for HCI, Virginia Tech
2202 Kraft Drive,
Blacksburg,Va. 24060, USA
chmille3@vt.edu

Louis-Philippe Morency
Institute for Creative Technologies, USC
Los Angeles, CA 90094, USA
morency@ict.usc.edu

Francis Quek
Center for HCI, Virginia Tech
2202 Kraft Drive,
Blacksburg,Va. 24060, USA
quek@vt.edu

ABSTRACT

There are a multitude of annotated behavior corpora (manual and automatic annotations) available as research expands in multimodal analysis of human behavior. Despite the rich representations within these datasets, search strategies are limited with respect to the advanced representations and complex structures describing human interaction sequences. The relationships amongst human interactions are structural in nature. Hence, we present Structural and Temporal Inference Search (STIS) to support search for relevant patterns within a multimodal corpus based on the structural and temporal nature of human interactions. The user defines the structure of a behavior of interest driving a search focused on the characteristics of the structure. Occurrences of the structure are returned. We compare against two pattern mining algorithms purposed for pattern identification amongst sequences of symbolic data (e.g., sequence of events such as behavior interactions). The results are promising as STIS performs well with several datasets.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*relation systems, temporal logic*

General Terms

Algorithms

Keywords

Structural Search, Multimodal Data, Event Data, Temporal Behavior Models

1. INTRODUCTION

There is a multitude of annotated behavior corpora (manual and automatic annotations) available as research expands in multimodal analysis of human behavior. Many of these corpora and supporting visualization tools store, organize, and display multimodal data based on the structural nature of behavior. By structure we mean discrete events that hold ordered relations in time that may vary between occurrences. For example, the visualization tools MacVisSTA [30], ANVIL [11], and EXMarALDa [34] display multimodal data as interval events with support for continuous signal data. The input formats of these tools are based on discrete interval events (multivariate symbolic data). This organization strategy is also seen in domains where frequent episode mining [25, 26, 31] is applied (e.g., medical records, neural spike data,...). Frequent episode mining is normally based on identifying a sequence of atoms (e.g., symbols or descriptions) and identification of "relevant" patterns is based on frequency and/or statistical modeling. However, for analysis of multimodal data, a pattern's value to an expert may not be based on frequency or statistical significance but on subjective relevance. Hence, a search strategy designed for an expert's interest in multimodal behavior data is motivated.

We present Structural and Temporal Inference Search (STIS), a pattern search strategy for multimodal data built upon the structural nature of human behavior. A pattern defines a sequence of behaviors. Behaviors are encoded as annotated event intervals with temporal order being implicitly or explicitly defined. An example is a greeting among two individuals with the possible formulation: <A walks up to B>[within 1 second]<A shakes B's hand> and <A says "Hello">. We base STIS upon this representation using contextualized information. This is done through viewing a pattern that is of interest to an expert (i.e., a relevant pattern) as not only the focus point of analysis but also defining the search criteria. A pattern is deemed relevant by an expert dependent on the expert's interest in the behavior described by the pattern. Identification of a relevant pattern within a dataset (i.e., search) results in occurrences of the pattern.

The expert's definition of a relevant pattern incorporates his or her knowledge into the search criteria as opposed to relying on statistical modeling to bring to the surface a pattern that may or may not be of interest. Statistical models used to extract frequent and/or statistically significant patterns (episodes) [14, 26] do not address cases where a pattern may only occur a handful of times. As discussed in [21], an algorithm's results based on some automated metric (such as frequent episode mining) would require some form of explicit pattern search anyway. This motivates our interest in identifying pattern occurrences of interest to the expert.

ICMI'12, October 22–26, 2012, Santa Monica, California, USA.

The rest of our paper is organized as followed. In section §2 we present related work of multimodal corpora, analysis tools, visualizations, and temporal pattern mining approaches. The details of STIS are discussed in §3. §4 discusses our experiment methodology, implementation details, the datasets used, the baseline algorithms, the patterns tested, our results, and discussion. Conclusions and future work close our paper in §5.

2. RELATED WORK

Our data domain is multimodal. There has been a strong trend toward creation and analysis of multimodal corpora. This is no surprise as the authors of [29] argue the value and deeper understanding multi-modality adds to analysis of human behavior. Many multimodal corpora have been created in response to this observation which predominantly consist of sequences of descriptive events (behavior patterns). The VACE/AFIT [5] multimodal meeting corpus is a detailed recording of multiple sessions of Airforce officers partaking in war gaming scenarios in a meeting setting. The Semaine corpus [16] is a collection of emotionally colored conversations. The Rapport and Face-to-Face corpora [8, 24] are sets of speaker-listener interactions. One of the largest to date is the AMI corpus [3] which contains 100 hours of recorded meetings. Mörchen created a series of datasets of varying degrees of modalities [23]. These mentioned corpora and datasets are highlights of a growing community of such data.

With the increasing number of multimodal datasets, tools are needed to visualize the data for analysis. These tools have been developed to visualize multi-channel annotation information coupled with varying degrees of multi-channel support of audio and video. Well known examples of these tools are MacVisSTA [30], ANVIL [11], Theme [35], EXMARaLDA [32, 6], ELAN [38], C-BAS[1], Transformer, and VCode [9]. The AMI corpus uses a different approach through use of the Nite XML toolkit which provides extensive support for complex annotation representation and supportive interface. Nite XML toolkit visualization is centered around transcription text (e.g., dialogue) of a corpus being annotated and is linked to supportive media, e.g., audio or video.

It is common for behavior interactions to be described as a sequential sequence of observed events. Many experts in the field of behavior analysis express behaviors of interest in this fashion [3, 5, 11, 12, 16, 17, 19, 33, 36]. This is no surprise as such descriptions capture the sequence of events that define the behavior. Many large corpora have been produced to identify and understand behavior among humans interacting within a small group setting (e.g., [3, 5, 16]) One focus of the analysis of these corpora is identifying the structure of behavior patterns. However, there is limited support for searching based on the structure.

Currently, there are a few search strategies in this data domain. Some visualization tools such as Nite XML toolkit has a supportive query language for searching the annotations. Such an approach can be powerful but construction of queries can be complex and cumbersome. ANVIL supports searching amongst the text of annotated event labels. This can be useful when looking for a specific event. However, identifying a sequence of labels does not seem to be supported. Some tools, such as VCode, can export annotated events to a text file where search outside of the application can be performed. MacVisSTA has the ability to save an observation (notebook) and play it back but not find other occurrences of the observation. EXMARaLDA performs search using a tool created by the EXMARaLDA authors called EXAKT. Their search is modeled after KWIC (keyword in context) and has powerful support for regular expressions in text search (search transcription text, annotations, and descriptions). ELAN has similar search support to EXMARaLDA but has the added ability to add temporal relation constraints (Allen's constraints between two intervals [1]). Transformer is purposed for transforming data files for use in one tool to another. They do support text search in which different corpus files can be specified to search.

The search we are interested in is symbolic temporal pattern mining where the focus is discovering "interesting" patterns among *symbolic* time series data (not numerical) [13, 22]. There are a few approaches related to this aspect of STIS. The first is T-patterns developed by Magnusson [15] where a sequence of events will occur within certain time windows, e.g., $A_1 \xrightarrow{T_1} A_2 \xrightarrow{T_2} A_3$ for time intervals T_1 and T_2. T-patterns are used as the basis of pattern representation in Theme [35] where each T is set through various statistical methods. Time interval windows are used in the second related approach, Frequent Episode Mining (FEM) algorithms of [26, 31]. The FEM algorithms use one of two approaches: conditional probability or a frequency threshold, both on defined timing windows.

3. STIS METHOD

Structural and Temporal Inference Search (STIS) is founded on creating a formalism of a pattern based on structure, timing, and ordered relationships. We operate on a pattern at the semi-interval level (start or end of an interval). This representation was first introduced by Freksa in [7] and later revisited by Mörchen and Fradkin in [23]. Semi-intervals allows a flexible representation where partial or incomplete knowledge can be handled since operation is on parts of an interval and not the whole. In this section we discuss how we use semi-intervals to describe a pattern and build a structured search based on the pattern to identify occurrences within a dataset. An overview of our method can be seen in Figure 1. Given a set of event annotations (e.g., from ELAN or MacVisSTA), create a semi-interval set which is organized in a database of definitions and instances. This is done offline. Then the expert provides an event sequence that is converted into a *pattern* which contains implicit search criteria. This is given to STIS which performs structural analysis on the *pattern*, uses the results of the analysis to form search criteria, searches to identify occurrences based on the criteria and returns a set of occurrences. We will discuss the details of what occurs offline and online in turn.

Offline: Event annotations from a multimodal dataset are transformed into a set of semi-interval annotations. We define an event as:

Definition 1. An *event* is an interval $[r_i, r_j]$ with semi-intervals r_i and r_j, $i, j > 0$, representing the start and end points of the event, respectively.

Our representation of an event does not associate with a particular occurrence time of the event, i.e., r_i and r_j are not the times of the start and end points. This is necessary as many occurrences of the same event can occur.

[1]Developed at Arizona State, http://www.cmi.arizona.edu/, but the url for C-BAS is broken.

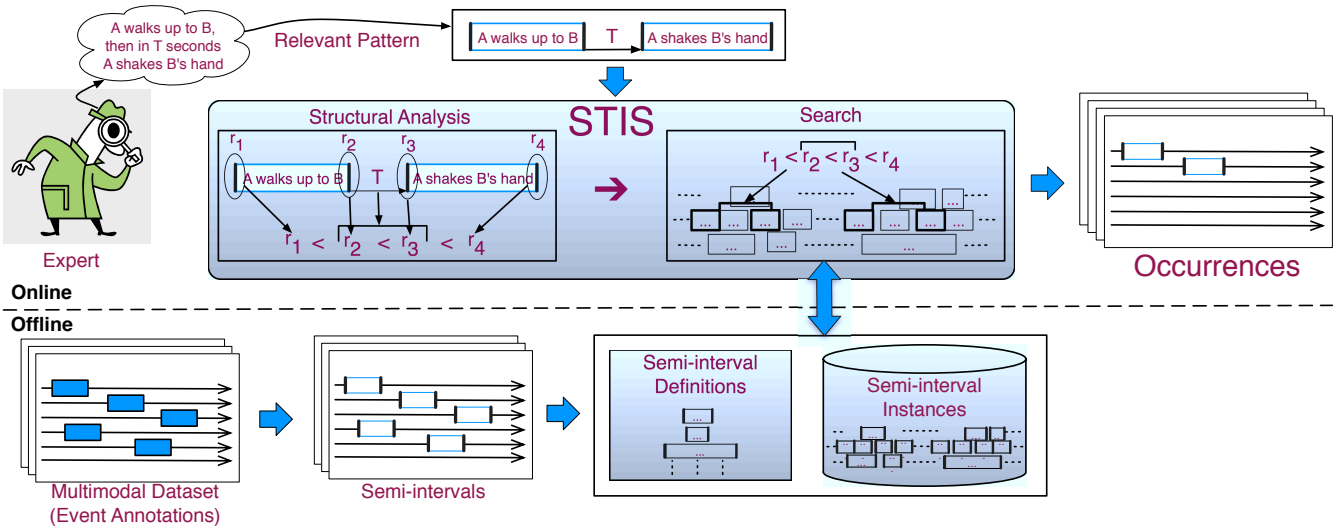

Figure 1: STIS overview: Offline, from a multimodal dataset create a semi-interval set organized as *definition* and *instance* tables. Online, an expert provides an event sequence that is converted into a *pattern* containing implicit search criteria. STIS performs structural analysis on the *pattern*, uses the results of the analysis to form search criteria, searches to identify occurrences based on the criteria and returns a set of occurrences.

Identification with a particular occurrence time is discussed later. For organizing events, two look-up tables are used. The first, a *definition table*, stores semi-interval *definitions*. A *definition* stores characteristics of the event from which it originated. These characteristics consist of a textual description, the actor involved (or source of the event), the type of event, e.g. modality, and whether it is a start or end semi-interval. These descriptive characteristics are a subset of event aspects in [37], except for start/end. Such characteristics have been used as focal aspects during analysis of event-based multimodal data [5, 17, 23]. The *definitions* are used to store descriptive information for each semi-interval without repetition (i.e., look-up table of unique *definitions*). The second look-up table, an *instance table*, stores of all semi-intervals in the dataset organized by temporal order. Each semi-interval in this table links to its *definition* in the first table. The *definitions* in the first table allows querying semi-intervals based on characteristics while the second table allows querying of events based on temporal criteria. Currently, our organization of event information is purposed to store and represent interval and semi-interval data. Point data can also be stored in which case a single semi-interval with no matching semi-interval is stored.

Online: An expert provides an event sequence to identify. The sequence is mapped to a *pattern* representation:

Definition 2. A *pattern* is a sequence S of semi-intervals r_i, $i \in \{1, ..., |S|\}$, such that for each $r_i \in S$, $\exists\ r_j$ such that r_i occurs before or is equal to $r_j\ \forall\ i \leq j$; $i, j \in \{1, ..., |S|\}$. Each r_i and r_j has an associated *temporal constraint* $\hat{t}_i$ which is a time window between r_i and r_j such that r_j occurs within $\hat{t}_i$ time of r_i where r_i's time $(t_i) \leq r_j$'s time (t_j), i.e., $t_i \leq t_j \leq t_i + \hat{t}_i$.

An example *pattern* can be seen in Figure 2A which represents one rendition of the greeting between two individuals from Section 1. The *temporal constraint* T expresses r_3 and r_4 to occur within T time units of r_2. This is useful as one may only be interested when A shakes B's hand and says "Hello" within a certain time to A approaching B. If no constraint is given, then matches that are not temporally close will be found but do not represent the desired greeting occurrence, i.e., ten minutes passes after A approaching B, then A shakes B's hand, etc., which does not represent the desired greeting structure.

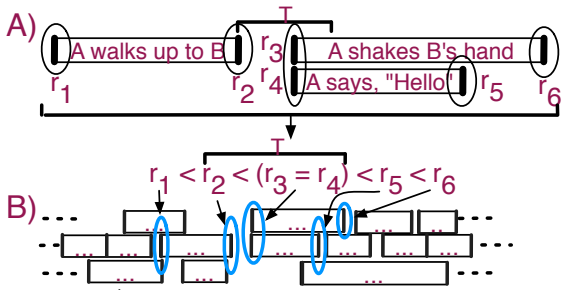

Figure 2: A) Example structure of a *pattern*. Note the temporal constraint between r_2, r_3, and r_4. B) Segmentation into *pockets* of equality.

Following from this, a *pattern* can be viewed in one of two ways: a complete *pattern* or key-parts of the *pattern*. A complete *pattern* contains complete intervals (i.e., matched semi-intervals). The key-parts represent relevant semi-intervals of the pattern that are key to identification of occurrences of the *pattern*, which could include complete intervals. For example, the *pattern* in Figure 2A is a complete *pattern* whereas r_2, r_3, and r_4 within time T represent the key-parts of the *pattern*. Note that the key-parts and the complete *pattern* could be the same and the key-parts need not be unique but their temporal constraints and relational order are relevant to identifying the *pattern*.

The expert's *pattern* is given to STIS as input. STIS then performs two steps: structural analysis and search. The structural analysis step consists of "dissecting" the *pattern* and extract ordered and temporal information. In this step, *temporal constraints* are stored and the *pattern* is segmented into *pockets* of equality. We define a *pocket* as

Definition 3. A *pocket* p of *pattern* R is a set of semi-intervals $r_i \in p$ such that $\forall i, j \in \text{indices}(p)$, $0 \leq |t_i - t_j| \leq \varepsilon$ where indices(p) is the set of semi-interval indices within p, e.g., i and j.

This use of *pockets* follows from the observation that at the semi-interval level, semi-intervals in a sequence are either equal (within a certain small time window) or not (outside the time window). Hence, semi-intervals can be grouped

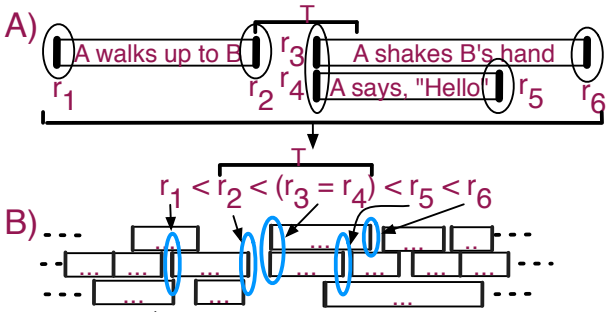

Figure 3: A) Example of search criteria and B) search within the semi-interval instances.

accordingly into *pockets* of equality. All semi-intervals that are within an ε of each other are deemed equal and grouped in a *pocket*. These groups are separated by temporal order (inequality). The segmentation into *pockets* allows a simple method with which to implicitly store ordered relational information, i.e., using the structure to provide relational information. As can be seen in Figure 2, we can see the implicit (ordered) relationships amongst the r_i's. For example, there is no need to explicitly store (remember) that r_3 and r_4 are equal or that r_2 occurs before r_6.

After structural analysis, STIS creates a set of search criteria applied to the *instance table* in which instances are identified. The search criteria contain ordered relationships among the semi-intervals and any defined temporal constraints. An example of search criteria can be seen in Figure 3A, where the greeting *pattern* from Section 1 is revisited. Here, the ordered relational information and a temporal constraint are extracted from the *pattern* and a set of search criteria are created. Then in Figure 3B, these criteria are used to identify occurrences within the semi-interval instances (*instance table*). As can be seen, the search criteria collected from the *pattern* reflect the *pattern's* structure. STIS uses this criteria to scan all semi-interval instances finding occurrence matches and returning a set of matches. The situated context of each occurrence is preserved as is important for multimodal analysis. This is accomplished through storing the occurrences' times and the semi-intervals within a certain defined time window for each occurrence. This process is based on the work of [20].

4. EXPERIMENTS

For multimodal data organized as multi-channel temporal events, we pose the following question: Can search based on a defined *pattern* structure identify occurrences of the *pattern* with greater accuracy than search based on conditional probability thresholds? We first outline our methodology for experimentation followed by a description of the implementations of the search strategies used. After which we describe the datasets used and the behavior categories our experimentation focuses on. The baselines are then discussed. Then we present our results and provide discussion.

4.1 Methodology

Since behavior analysis has many variables to consider, testing our search strategy must be done in a controlled environment. To accomplish this, we introduce occurrences of *patterns* with variation into several datasets at known locations, apply the search techniques, then see if the *patterns* can be identified. This is also necessary as exact known ground truth instances for the datasets used is limited. The techniques used are STIS, FEM frequency, and FEM conditional probability. We chose 5 categories of *patterns* in a meeting room setting deemed important by experts, i.e., relevant *patterns*. These categories come from analysis reported in [5, 17]. We then apply the same search techniques to unaltered real datasets with known ground truth.

We experiment with three datasets from the domain of behavior analysis in a meeting room setting. The first is a generated (synthetic) dataset that is created based on the parameters of real datasets similar to bootstrap aggregating (bagging) [2]. The other two datasets are real datasets consisting of two sessions within a corpora (details in §4.3).

We introduce into each dataset occurrences of relevant *patterns* with variation based on the 5 behavior categories. Each *pattern* is based on relational structures observed by experts. For each *pattern*, we introduce 10 instances into its own copy of each dataset, i.e., there is no interference between the *patterns* of different behavior categories. We also ensure that none of the 10 overlap. Then we search each dataset copy for its respectively inserted *patterns*. The results are compared to the known inserted locations for accuracy. Power/penalty analysis is used as a metric (described in §4.6). We then take the two real datasets unaltered and search for occurrences of known ground truth. We conduct these searches using two versions of each *pattern*: the complete *pattern* and key-parts. This allows a comparison between using complete knowledge of a *pattern* and the relevant pieces according to the expert (sometimes complete knowledge is not needed or unattainable).

Since one of our datasets is generated, there is some concern that the *pattern* instances introduced already exist due to random generation. However, the probability that the generated dataset has many relevant *pattern* instances present is very low. This probability was explored in [21].

4.2 Implementation

STIS is implemented in C++ using Qt 4.7 [27] for the user interface and a SQLite database for the datasets. The current interface of our system is not shown as it is not the focus of this paper. The FEM frequency algorithm (*FEM1*) is implemented in C++ and the FEM conditional probability algorithm (*FEM2*) is implemented in Java. Both FEM algorithms are part of TDMiner (http://people.cs.vt.edu/patnaik/software). In deciding the appropriate *temporal constraints*, the choice depends on the events involved, what events mean to an expert, and the kind of data. Ultimately, it is up to the one performing the search. For our experiments we chose to use a global 3 second window as a *temporal constraint* between each consecutive semi-interval being matched. The timeframe of behavior patterns is normally temporally tight (on the order of milliseconds up to seconds). Using a 3 second window allows the identification of instances that are temporally tight and those a little longer without a flood of results with many potentially being irrelevant. However, this window can be user set.

4.3 Datasets

Here we describe the datasets used for our experiments. Our experiments were conducted using the VACE/AFIT multimodal meeting room corpus [5, 17].

VACE/AFIT: This dataset consists of several meeting sessions of Airforce Officers from the Airforce Insitute of Technology (AFIT) partaking in war-gaming scenarios. We

Table 1: Datasets' Contents.

Data-set	Length (min)	# Semi-interval	# Unique Semi-intervals	Speech Length (secs)			Gaze Length (secs)			Gesture Length (secs)			# Gaze	# Speech	# Gesture
				Ave	Min	Max	Ave	Min	Max	Ave	Min	Max			
Generated	~45	25590	240	1.3	0.06	5	1.27	0.1	5	1.29	0.1	5	10626	7438	7526
										Nodding, Phrase					Nodding, Phrase
AFIT 1	~45	7802	342	1.59	0.1	64.46	2.16	0.1	158.86	0.99, 0.84	0.23, 0.3	10.71, 11.68	4704	1414	1018, 666
AFIT 2	~42	13362	226	2.28	0.03	124.62	1.09	0.07	58.22	0.89, 1.0	0.27, 0.27	13.78, 9.21	11456	1126	610, 170

focus on two sessions (*AFIT 1* and *AFIT 2*). Each session is a scenario in which five officers (C, D, E, F, and G) are given a problem to discuss and resolve. The room where the sessions took place was instrumented for multi-channel audio and video along with motion capture of the officers (details of instrumentation in [5]). The officers in *AFIT 1* are discussing potential candidates for a scholarship. The scenario is that C, D, F, and G are department heads meeting with the institute commandant E to select three scholarship award recipients. The officers in *AFIT 2* are discussing options for exploiting an enemy missile that has been discovered. Each session is approximately 45 minutes with manual and automated annotations for speech, gaze fixations, F-formations, and several gestural forms (including gesture phrases) for each officer. F-Formations, or focus formations, were first identified by Adam Kendon to be units of topical cohesion marked by gaze cohesion of participants to common objects or spaces [10]. Gesture phrases are atomic gestural motions marking a single motion trajectory that typically coincide with atomic speech phrases [18]. These annotations are events that were extracted from the audio, video, and motion capture data and describe the officers' interactions. The sum of the annotations is a dataset consisting of multiple channels (21 for *AFIT 1*, 19 for *AFIT 2*) of overlapping event data extracted from various synched media streams. The sequences of behavior described by the annotations are rich and descriptive. Each dataset is summarized in Table 1. For our experiments, *A1* and *A2* are altered versions (i.e., *patterns* introduced) of *AFIT 1* and *AFIT 2*, respectively, and *A3* and *A4* are unaltered versions (i.e., original) of *AFIT 1* and *AFIT 2*, respectively.

Generated: We generated a dataset based on the parameters of the VACE data. For five fictitious people, we randomly generate 45 minutes of annotations for parallel channels of speech, gaze fixations, and gesture phrases. For each person and each channel, we generate a timeline of events with varying lengths and gaps between them totaling 15 parallel channels. This is fewer channels but much denser as can be seen in Table 1 with significantly more semi-intervals. We label this dataset as *G*.

4.4 Relevant Patterns

Here we describe the general *patterns* that were deemed interesting by experts and introduced into *G*, *A1*, and *A2*, plus the ground truth *patterns*. The *pattern* structures used can be seen in Figure 4. The outlined semi-intervals represent the key-parts of the *pattern*. The actors used for the *patterns* introduced into *A1* and *A2* were chosen so that they did not match the original actors reported by experts in [4, 17]. For example, if a *pattern* we want to introduce was reported involving C gazing at F, then we did not use C and F but instead G and D. This was done to prevent interference from the actual *patterns* observed by the experts.

Mutual Gaze (MG): In the AFIT sessions, different participants controlled the floor at different times (i.e., leading the discussion for the moment). When the control passed from one participant to the next, there was a mutual gaze exchange between the current holder of the floor to the next.

Gaze Coalition (GC): It was discovered in *AFIT 1* that the social interaction amongst the participants had as much to do with the outcome of the meeting as the specific merits of the scholarship candidates being discussed. The participants dynamically formed coalitions to support each-other's candidates through a process of mutual gaze fixations and back-channel expressions of support during discussions [19].

A coalition to support a proposed idea is initialized when the proposer seeks to make eye-contact with other participants while he is presenting the idea. Participants who supported the idea would return the eye-contact, while those who disagreed would avert gaze. When a return gaze was attained, the presenter's gaze moved to another member.

Floor Control (FC): During a session, a participant would gain floor control through a hand movement (gesture) and start speaking. This was deemed 'floor capturing' in [4].

Turn Taking (TT): As detailed above, each session had a meeting manager who normally was the dominant participant and facilitated the meeting. When a participant sought to take a turn speaking, the participant might look at the meeting manager while the current floor controller spoke. Once the current floor controller finished speaking, the participant seeking a turn would then begin speaking.

F-formation (Ff): F-formations were observed throughout the sessions. The defining behavior for F-formations observed were concurrent focus on the same person (or object).

Ground Truth: In *A3* a GC model was known to exist from unpublished analysis. In *A4*, a TT and Ff model were reported in [17]. These represent the only occurrences of a known ground truth *pattern* in which the exact timing within the datasets can be verified (3 in total). The ground truth *patterns* consisted entirely of semi-interval key-parts.

4.5 Baselines

We compare STIS against two Frequent Episode Mining (FEM) algorithms [26, 31]. The motivation behind the particular kind of FEM algorithms (*FEM1* and *FEM2*) we use is the discovery of pattern sequences within temporal event data. The authors of *FEM1* and *FEM2* applied their algorithms to neural spike data (i.e., firing patterns of neurons in the brain). The patterns represented by this data have many similarities to our data domain: a sequence of firing times of neurons in sequence, i.e., a sequence of discrete events governed by temporal constraints.

Since FEM algorithms are meant for mining and not searching, we compromise by tuning them similarly to STIS's parameters, and then search their results for relevant *patterns*. This is necessary as there are no other approach like STIS for comparison. The closest is *FEM1* (discussed shortly). In a FEM algorithm, there are several methods for specifying whether a pattern occurrence is deemed important. The two approaches most pertinent to our problem is frequency (*FEM1*) and statistical significance (*FEM2*). *FEM1* sets a frequency threshold reporting a *pattern* if seen at least as many times as the threshold. *FEM2* is based on the conditional probability of one event given another within a time window. In other words, if interested in A following B within 2 seconds, we would look for $Pr(B|A) \geq \alpha$ where B is within 2 seconds of A and α is a significance (connection strength) threshold. For more details see [31].

Interestingly enough, *FEM1* supports search by *pattern* definition where occurrences of a specified *pattern* are counted

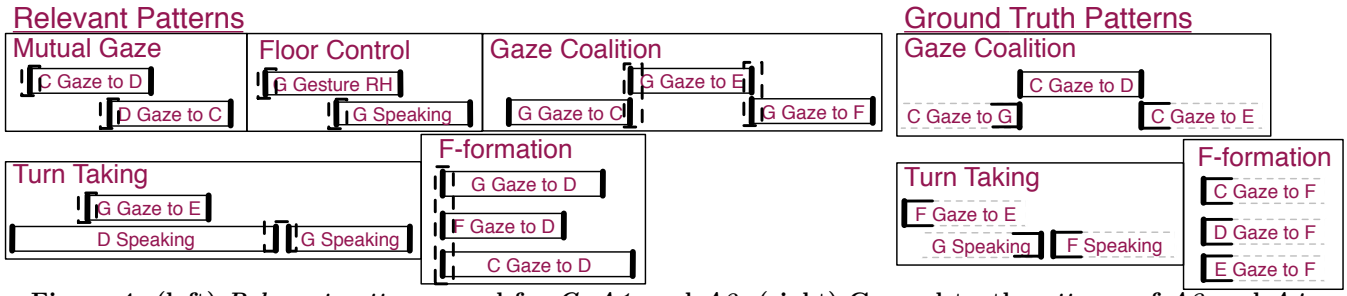

Figure 4: (left) *Relevant patterns* used for *G*, *A1*, and *A2*. (right) Ground truth *patterns* of *A3* and *A4*.

Table 2: Overall Power/Penalty Analysis

	Complete	Key-Part	Ground Truth
Misses			
STIS	1	0	0
FEM1	9	17	2
FEM2	22	2*	3
Mean Power			
STIS	99.33	100	100
FEM1	94	88.67	33.33
FEM2	85.33	76*	0
Mean Penalty			
STIS	18.67	36.3	75.56
FEM1	19.68	39.87	91.67
FEM2	40.22	23*	100

without using a frequency threshold. This is analogous to an expert defining a *pattern* to search. The results of this kind of search in *FEM1* would be the same if using *FEM1* to search by *pattern* frequency. The main difference is that using frequency results in a long list of *patterns* to sift through for an exact *pattern* (plus choosing an appropriate threshold) whereas defining a *pattern* to count is focused on the exact *pattern* of interest. Definition of a *pattern* in *FEM1* for counting is closely related to how STIS operates, hence, this approach of *FEM1* is used for comparison.

For *FEM2* , the conditional probability threshold ranged from 0.03 to 0.1 depending on the size of the dataset and the *pattern*. We noticed for smaller datasets, in general, a larger threshold could be used. We use the same 3 second window between semi-intervals for *FEM1* and *FEM2*. *FEM1* functioned on all the datasets in semi-interval form. However, *FEM2* had some limitations requiring the use of interval datasets in most cases. With an interval dataset, *FEM2* performs operations with respect to an interval's start.

4.6 Results

The performance of STIS is tested through *power/penatly* analysis of [28]. This is done for datasets *G*, *A1*, and *A2*. We then look at the results for the three ground truth *patterns* also with power/penalty analysis. In total, STIS was run on 33 *patterns*, *FEM1* on 33, and *FEM2* on 30; in total 96 *pattern* searches were performed. For simplicity, we use the naming scheme *X_Y* to reference each *pattern* where *X* is the dataset and *Y* is the *pattern* abbreviation. For example, *A1_MG* is mutual gaze *pattern* from *A1*.

For describing the results, we use power/penalty analysis which reports a power and penalty percentage. The idea behind power/penalty analysis is that if there are x known instances of a phenomenon in a dataset, y instances identified by a method or algorithm, and z number of instances common amongst the known and identified, $z \leq x$, then the power percentage is $z/x*100$. For example, if $x = 10$, $y = 18$ and $z = 7$, then the power is $7/10 * 100 = 70\%$, i.e., the method's power is 70% in identifying the relevant instances. The other 11 identified instances are part of the penalty. These are extra instances the expert must go through and, in turn, are an extra cost. The penalty $= (y - z)/y * 100$, in our example, penalty $= (18 - 7)/18 * 100 = 61.11\%$.

A precision/recall approach is not applied as such approach provides how accurate your model is in identifying an instance. However, in our case, we not only want to accurately identify an instance, but whether that instance represents a specific behavior of interest. For example, one can create a detection system for hand waving. However, an expert may not only be interested in hand waving, but when A waves at B. What is detected will either be related to the behavior of interest (power) or not (penalty).

Table 2 presents the overall power/penalty analysis results. STIS was able to identify nearly all the occurrences (99.33% - only 1 missed). FEM1 and FEM2 missed a number more (94% and 85.33%, respectively). STIS had a higher mean power and a lower mean penalty for the complete *pattern* case. STIS also performed better than *FEM1* for the key-part case. The '*' for *FEM2*'s key-part results signify that these are only partial results. We were only able to run *FEM2* on key-part *patterns* for *A1* due to some limitations of *FEM2* (discussed later). Hence, the results presented in Table 2 are for this case. The corresponding sub-set of results for STIS and *FEM1* are 0, 100, 26.46 and 5, 90, 27.61 for misses, mean power, and mean penalty, respectively. For this case, *FEM2* had a lower mean penalty.

For the ground truth, STIS was the only approach that was able to identify all 3 known ground truth occurrences. We would like to emphasize for the ground truth identification STIS's ability to search for a *pattern* and one of the results be a ground truth occurrence. The high penalty is due to verifying the identification of only one occurrence for each *pattern*. STIS returned at max 5 occurrences for the ground truth *patterns* whereas *FEM1* and *FEM2* returned up to 22 occurrences and for some, did not return any.

In Figure 5 we can see the details of the penalty for the complete *pattern*, key-part *pattern*, and ground truth *pattern* cases. As can be seen, STIS and *FEM1* have competing results. The limitations of *FEM2* caused it to struggle with the ground truth case. Not surprisingly, all approaches had their worst performance for the generated data. There is a noticeable difference between the complete and key-part *pattern* cases. The penalty increased for key-part. The reason for this is most likely because the key-part *patterns* contain mostly semi-intervals (not intervals) leading to a greater chance of having more matching occurrences. This is one of the characteristics of the semi-interval representation as a pattern defined using semi-intervals can match a greater number of *patterns* than an interval representation [23].

Comparing the penalty trends across datasets and *pattern* types, we see that STIS has a similar penalty trend between complete and key-part cases for each dataset. STIS seems to be least affected between the datasets but suffers from the same errors between them also. STIS and *FEM1* display

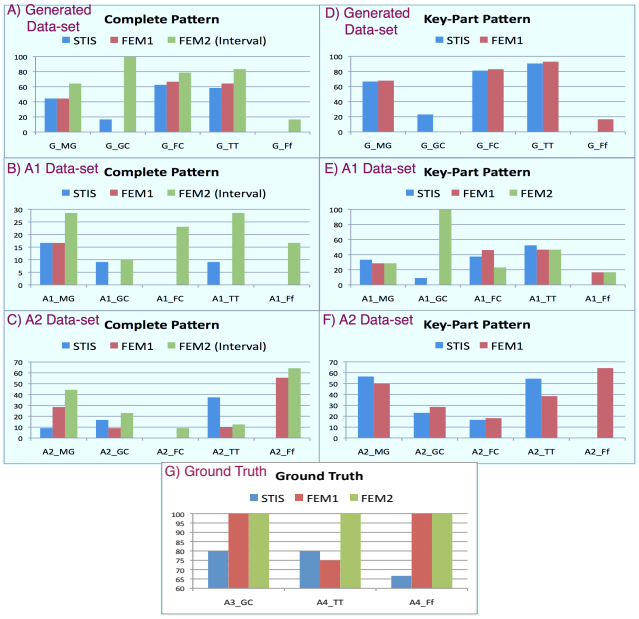

Figure 5: Penalty for *G*, *A1*, *A2*, and ground truth.

similar trends for *G*, *A1* key-part, and *A2* key-part. This suggests that they may have similar strengths and weaknesses. *FEM1* and *FEM2* had strongly correlated trends for *G* and *A2* complete but not for the other cases. Overall, the penalty trends suggest that there are commonalities in the algorithms that aid in identification but also shared pitfalls that hinder. Potential pitfall causes are the necessity to tune the *temporal constraints* and *pocket* size based on characteristics of the dataset and/or the kind of *pattern*, and making sure semi-intervals that are matched to an occurrence actually make sense, e.g., a start and end are matched to the same event occurrence and not two different occurrences of the same kind of event. Current measures in STIS to minimize this is to verify through the *instance table* that the semi-interval occurrences are matched appropriately. Further work in this area is needed to provide more robust matching. For ground truth, there was no trend seen other than STIS had the lowest penalty overall.

4.7 Discussion

In answer to the question posed previously, the results of STIS and *FEM1* confirmed that search based on event structure can identify *patterns* with high-accuracy, and search for *patterns* in multimodal data organized as multi-channel temporal events can benefit from expert input and specification as opposed to a conditional probability method (*FEM2*).

During our experimentation, we observed some limitations in the different algorithms. For *FEM1*, the ground truth *patterns GC* and *FF* could not be found as some semi-intervals of the *patterns* occur at the exact same time. *FEM1* does not handle this case, which was also observed in [21]. For *FEM2*, significant effort was required to obtains results as we had to continually try different conditional probability thresholds (some as low as 3%). This challenge came from the frequency of the *patterns* being searched for. Compared to the size of the data set, 10 occurrences (or 1 for each ground truth *pattern*) is very small. Hence, why search based on event structure with expert input and specification performed well. The *FEM2* implementation had an inherent limit in the number of reported *patterns* that could be outputted for verification (∼50K). When this limit was exceeded, verification could not be performed as results could not be outputted. The larger the dataset, the greater the number of reported *patterns*, hence, the use of the interval versions of the datasets as they were half the size of their semi-interval counterparts. However, *FEM2* using intervals suffered since the algorithm would only match according to start times and not (seemingly) use the end times. This left *FEM2* operating as if only start semi-intervals were specified leaving a greater possibility for more matches.

For STIS, we encountered an initial identification error with ground truth *Ff*. Our default size of a *pocket* was temporally very tight as the original analysis of the behavior within *A3* and *A4* was focused at a small time scale (milliseconds). One of the gaze events of the *Ff pattern* was outside of our *pocket* size. Hence, we had to slightly change the structure of the *pattern* used to search in order to identify the ground truth *pattern*. This highlights the necessity for greater flexibility when searching using a structural approach, which is an observation we were aware of and this situation confirms such. Another observation is that *FEM1* had less identified patterns than STIS but still high power. We believe this is because *FEM1* returns non-overlapping *patterns*, i.e., *patterns* that do not overlap each other. STIS does not filter for non-overlapping *patterns* as such *patterns* may contain variations of potential interest to the expert.

FEM2's search strategy is based on identifying *patterns* using defined parameters. We are interested in identifying *patterns* that match parameters *and* also match specific content. By content we mean what the events involved in the *pattern* mean. For example, the structure of the ground truth models (Figure 4) can match any number of *patterns* in the data. It is the provided content along with the structure that allows an expert to pinpoint occurrences of interest. This kind of search is supported by STIS and *FEM1* and despite the limitations observed, they performed well. Overall, STIS outperformed *FEM1* and *FEM2* and poses to be a beneficial search approach in multimodal analysis tools.

The pattern structures investigated in this paper are the beginning of our research into creation of a set of temporal relationship principles for describing interaction patterns in multimodal corpora. A subset was used in this paper but expansion is underway into representing more complex patterns. This expansion includes negation, pre- and post-conditions, interrupts, and many renditions of repetition of specific events. However, the creation of more complex *patterns* may result in very few matches, which may or may not aid the current analysis (flexible vs. rigid *pattern* definition). Another potential venue of pursuit is using STIS to search partially annotated corpora. Some events are easier to annotate than others (e.g., when someone is speaking, or a person's position in the scene), hence, searching partial annotations can provide likely probable occurrences of events not yet annotated. This would identify focal areas where efforts can be applied for more detailed annotation creation.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a search strategy for multimodal data based on the structural and temporal aspects of human behavior. We were able to show that a search strategy based on these principles performs well. STIS demonstrated the ability to accurately identify occurrences of *patterns* with an expert defined structure with some (or all) the

occurrences identified being ones sought after. *FEM1* was a tough competitor which motivates future investigations of potential incorporation of *FEM1* aspects into STIS. An example being support for non-overlapping events if desired by the expert. Another focus of future work is supporting flexible timing windows (for *pockets* and temporal constraints). Support for such is merely a question of implementation.

6. ACKNOWLEDGMENTS

We would like to thank Debprakash Patnaik for his help in using FEM and Christa Miller for invaluable input as an outside observer and editor. This research was partially funded by FODAVA grant CCF-0937133, NSF IIS-1053039, and NSF IIS-1118018.

7. REFERENCES

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[3] J. Carletta et al. The ami meeting corpus: A pre-announcement. In *MLMI*, volume 3869 of *LNCS*, pages 28–39. Springer Berlin / Heidelberg, 2006.

[4] L. Chen et al. A multimodal analysis of floor control in meetings. In *MLMI'06*, pages 36–49.

[5] L. Chen et al. Vace multimodal meeting corpus. *MLMI '06*, pages 40–51.

[6] EXMARaLDA. *http://www.exmaralda.org*, Last Checked: May, 2012.

[7] C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1-2):199 – 227, 1992.

[8] J. Gratch et al. Creating rapport with virtual agents. In *Intelligent Virtual Agents*, volume 4722 of *LNCS*, pages 125–138. Springer Berlin / Heidelberg, 2007.

[9] J. Hagedorn, J. Hailpern, and K. G. Karahalios. Vcode and vdata: illustrating a new framework for supporting the video annotation workflow. In *AVI '08*, pages 317–321. ACM.

[10] A. Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Cambridge: Cambridge University Press, 1990.

[11] M. Kipp. Anvil - a generic annotation tool for multimodal dialogue. In *Eurospeech*, 2001.

[12] M. Kipp. Spatiotemporal coding in anvil. In *LREC*, 2008.

[13] S. Laxman and P. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 04 2006.

[14] S. Laxman, P. Sastry, and K. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: a formal connection. *TKDE*, 17(11):1505 – 1517, Nov. 2005.

[15] M. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods*, 32:93–110, 2000.

[16] G. McKeown et al. The semaine corpus of emotionally coloured character interactions. In *ICME '10*, pages 1079 –1084.

[17] D. McNeill. Gesture, gaze, and ground. In *MLMI'06*, volume 3869 of *LNCS*, pages 1–14.

[18] D. McNeill. *Hand and Mind: What gestures reveal about thought*. Chicago: U. of Chicago Press, 1992.

[19] D. McNeill et al. Mind-merging. In *Expressing oneself / expressing one's self: Communication, language, cognition, and identity*, 2007.

[20] C. Miller and F. Quek. Toward multimodal situated analysis. In *ICMI '11*.

[21] C. Miller and F. Quek. Interactive data-driven discovery of temporal behavior models from events in media streams. In *ACM MM*, 2012.

[22] F. Mörchen. Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explor. Newsl.*, 9(1):41–55, 2007.

[23] F. Mörchen and D. Fradkin. Robust mining of time intervals with semi-interval partial order patterns. In *SIAM Conference on Data Mining (SDM)*, 2010.

[24] L.-P. Morency, I. de Kok, and J. Gratch. Context-based recognition during human interactions: automatic feature selection and encoding dictionary. In *ICMI '08*, pages 181–188. ACM.

[25] D. Patnaik et al. Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges. In *KDD '11*, pages 360–368. ACM.

[26] D. Patnaik, P. S. Sastry, and K. P. Unnikrishnan. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Scientific Programming*, 16(1):49–77, January 2007.

[27] Qt. *http://qt.nokia.com/*, Last Checked: Mar., 2012.

[28] F. Quek. The catchment feature model: a device for multimodal fusion and a bridge between signal and sense. *EURASIP J. Appl. Signal Process.*, 2004:1619–1636, Jan. 2004.

[29] F. Quek, T. Rose, and D. McNeill. Multimodal meeting analysis. In *IA*, 2005.

[30] R. T. Rose, F. Quek, and Y. Shi. Macvissta: a system for multimodal analysis. In *ICMI '04*, pages 259–264.

[31] P. S. Sastry and K. P. Unnikrishnan. Conditional probability based significance tests for sequential patterns in multi-neuronal spike trains. 2008.

[32] T. Schmidt. The transcription system exmaralda: An application of the annotation graph formalism as the basis of a database of multilingual spoken discourse. In *Proceedings of the IRCS Workshop On Linguistic Databases*, pages 219–227, 2001.

[33] T. Schmidt et al. An exchange format for multimodal annotations. In *Multimodal corpora*, pages 207–221. Springer-Verlag, Berlin, Heidelberg, 2009.

[34] T. Schmidt and K. Wörner. Exmaralda – creating, analysing and sharing spoken language corpora for pragmatic research. *Pragmatics*, 19, 2009.

[35] Theme. http://www.noldus.com/human-behavior-research/products/theme.

[36] M. Voit and R. Stiefelhagen. 3d user-perspective, voxel-based estimation of visual focus of attention in dynamic meeting scenarios. In *ICMI-MLMI '10*, pages 51:1–51:8. ACM, 2010.

[37] U. Westermann and R. Jain. Toward a common event model for multimedia applications. *IEEE MultiMedia*, 14:19–29, January 2007.

[38] P. Wittenburg et al. Elan: a professional framework for multimodality research. In *LREC*, 2006.