A Speech-Based Approach to Video Retrieval

Behrang QasemiZadeh, Jiali Shen,
Ian M. O'Neill, Philip Hanna, Darryl Stewart, Paul C. Miller, and Hongbin Wang

Paper ID ****

Abstract

This paper describes anatomy of a pilot surveillance system with a speech based interface for content based retrieval of video data. The components of this pilot system are implemented in Matlab, Java, and Prolog. The proposed system relies on an ontology based information sharing architecture and lets components of the system communicate among each other through TCP/IP communication channels. The aim of developing the pilot system was to explore dependencies between image analysis, event detection, video annotation, and speech based retrieval of the video content in the context of a broader spoken dialogue system .

1. Introduction

The ISIS project is a multidisciplinary EPSRC-funded project based at ECIT, Queen's University's institute for Electronics, Communications and Information Technology. The specific aim of the project is to create an intelligent sensor network capable of detecting threats to passenger safety on public transport. However, the individual techniques for detection, classification and information retrieval are in many cases highly generic and relevant to many application contexts. Among the initial experiments conducted for ISIS was an attempt to classify by gender individuals passing through an office doorway (the Doorway Experiment) and then to retrieve footage of a 'person', a 'male' or a 'female' passing through the doorway within a given time period. With the eventual aim of facilitating information retrieval in a busy operations centre, the retrieval enquiry took the form of a spoken natural language query.

A second scenario took as its setting a more complex scene outside an office: here hand-tagged data (based on the British Home Office's i-LIDS data set [1] and representing individuals described by clothing and gender and vehicles described by colour) were used as a basis on which to formulate enquiries concerning the presence, activity and/or location of individuals and vehicles that matched a particular description. While this scenario (the i-LIDS Experiment) proved valuable in terms of developing an underlying data model, its need for hand-tagging and a rigid set of spoken enquiries meant that the simpler Doorway Experiment was a more accurate representation of a replicable technique for automated retrieval and dynamically-phrased information retrieval.

In this paper our description of experimental activity concentrates on our Doorway Experiment, with some reference to the more advanced avenues of enquiry suggested by the forward-looking i-LIDS Experiment.

1.1. The Doorway Experiment

As presented to the members of the i-LIDS consortium in the autumn of 2008, the Doorway Experiment showed how natural language could be used to progressively refine a user enquiry. In this experiment natural language enquiries adhered to a fixed structure, corresponding to a predefined underlying syntactic grammar, with the expectation that the variables required by the system (gender, start-time, end-time and day) would occur in fixed positions in each user utterance. This template-based approach offered enough flexibility for the user to change variable values at will and so vary the nature of the enquiry over several iterations.

The underlying dataset represented six individuals (three males and three females) passing through an office doorway in the course of a few minutes.

- *Show me the keyframes of a person entering the doorway between ten thirty a.m. and eleven a.m. on October the ninth.*
- *Show me the keyframes of a person entering the doorway between ten forty-five a.m. and eleven a.m. on October the ninth.*
- *Show me the keyframes of a male entering the doorway between ten forty-five a.m. and eleven a.m. on October the ninth.*
- *Show me the keyframes of a female entering the doorway between ten forty-five a.m. and eleven a.m. on October the ninth.*

2. The Anatomy of the full system

Figure 1 shows the anatomy of the prototype system in terms of its building blocks, a set of agents run on different TCP/IP communication channels.

The Data Manager Agent gets the image analysis results from the Image Analysis Agent and it writes the time instants, and the values of annotations, in the appropriate locations in the database. The Prolog Inference Engine Agent detects events from the annotations and Time Stamps provided and it asks the Data Manager Agent to write the inferred events into the Event Annotation repository. The Natural Language Interface Agent gets spoken natural language utterances and parses the input into an intermediate representation encapsulating information about the type of query, e.g. information seeking, in addition to a first order logic representation of natural language utterances. The Knowledge Management Agent then looks at the information provided by the user's utterances and the data repositories to provide the user with a video compilation – or another mode of response, such as a spoken or written answer.

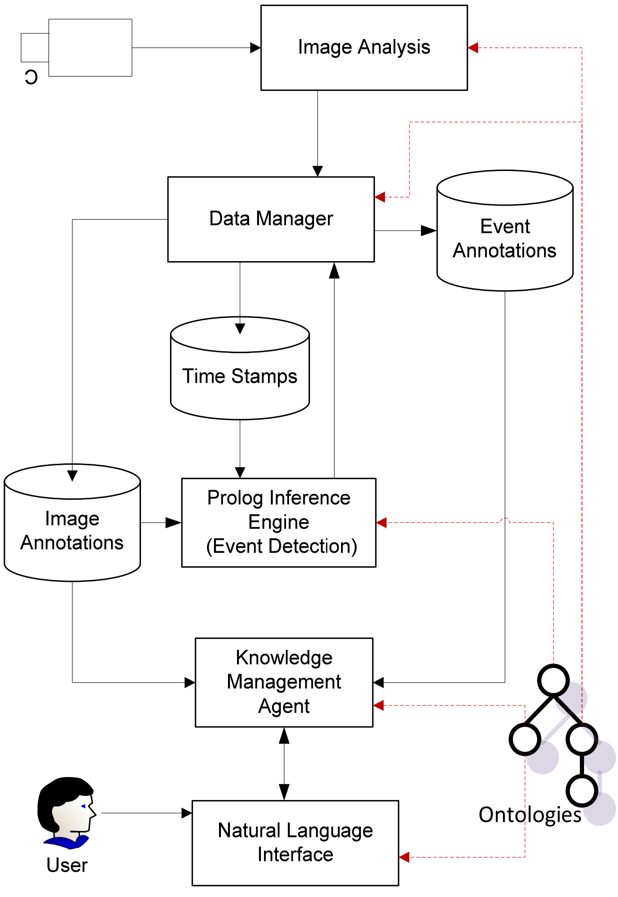


Image Analysis
Data Manager
Event Annotations
Time Stamps
Image Annotations
Prolog Inference Engine (Event Detection)
Knowledge Management Agent
Ontologies
User
Natural Language Interface

Figure 1. The Anatomy of the full system. Each block shows an agent run on a TCP/IP communication channel. The hatched lines show references, while the solid lines show the data flow among agents.

To make sure the system functions correctly in terms of semantics, a set of ontologies supports the system's agents. The ontologies ensure that the system uses the same vocabulary for the same semantic concepts in different agents. There are four classes of ontology in the system: a time ontology, properties ontologies, an object ontology, and an events ontology. The definitions in the Time Ontology are based on Allen's interval Algebra [2], and together with the time stamps, these serve the temporal needs of the system – e.g. providing temporal tags as well as an inference engine when temporal query analysis is required. A Property Ontology provides a set of vocabulary for describing an attribute of object – e.g. if colour is considered as an attribute for vehicle objects, then the system's agents will use only the vocabulary introduced by the "Colour Property" to annotate or retrieve the video. The Property Ontology provides other services, such as comparing two vocabularies or mapping them together. The Object Ontology provides a classification of objects and it may be varied from one application domain to the next. The most important role of the Object Ontology is to provide the system with vocabularies of shared concepts in a domain of interest – e.g. in the doorway scenario, the doorway, people, and stationary objects like computers might be the objects of interest. An Event Ontology like the one defined in [3] presents definitions for events, and the temporal or causal relationships between them.

In the following, we describe each of the system's agents in short.

2.1. Image Analysis Agent

The Image Analysis Agent can identify people and their properties, such as gender. At the current stage of development within ISIS gender profiling is based on the human face. The process is outlined in Fig. 2. A doorway camera is positioned to capture the faces of those who have just entered the target zone. Viola's technique [4] is used for face detection. The system is trained using AdaBoost [4] to select the most discriminative rectangle features, an approach that has been used in many applications.

Then Principal Component Analysis (PCA) is carried out on the normalized faces (24*24 pixels). In principal component analysis (PCA), eigenvalues of the matrix composed by training faces are calculated by ranking the eigenvectors with respect to their eigenvalues and selecting the first *M* principal components. Theory suggests that these components contain most of the information needed to separate the training faces. The first 5 eigenvectors are shown in Figure 3. A conversion matrix is also generated in this step, which is applied to transfer the input testing face into the new space.

In this system, the first 50 components are input to a trained Support Vector Machine (SVM) as the gender classifier. The SVM system has been trained by nearly 4000 faces with same PCA processing.

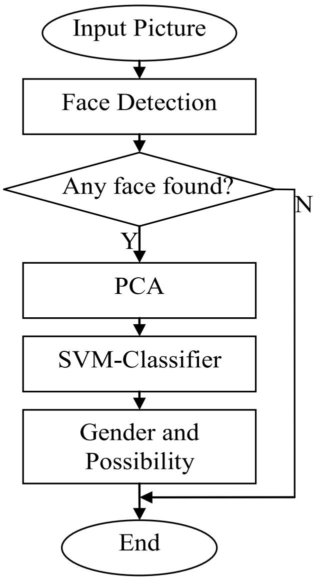


Figure 2. Human Gender Profiling

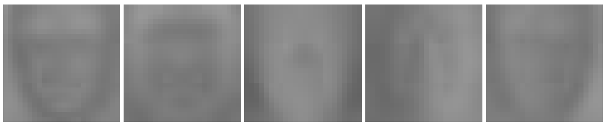


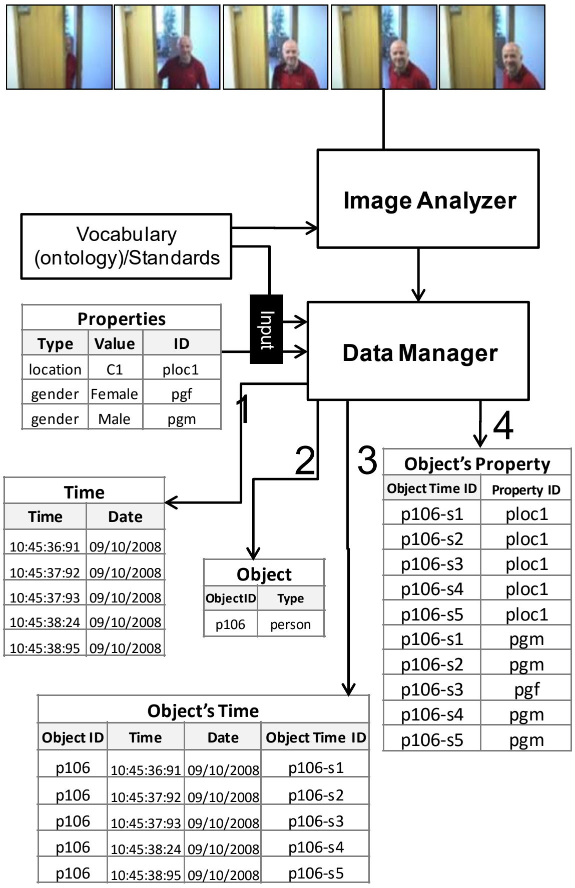
Figure 3. The first five eigenfaces.

2.2. Data Manager Agent

The Data Manager Agent is responsible for any transaction on the data repository. This includes usual data administrative tasks in addition to checking data consistency, and compatibility against the system's ontologies. For example, the Data Manager checks whether annotations provided by the Image Analyzer Agent are valid according to the property and object ontologies. In addition, the Data Manger Agent records all the transaction times from the Image Analyzer Agent and keeps track of temporal order of annotations. The following shows an example of a annotation provided by Data Manger Agent.

```
<object type="vehicle" id="v1"time="8">
    <properties>
      <colour>
        blue
      </colour>
        <location>
          Gp12528
        </location>
      </properties>
</object>
```

Figure 4 shows the steps the Data Manager Agent takes when it gets a new input from the Image Analyzer Agent.



Properties

Type	Value	ID
location	C1	ploc1
gender	Female	pgf
gender	Male	pgm

Time

Time	Date
10:45:36:91	09/10/2008
10:45:37:92	09/10/2008
10:45:37:93	09/10/2008
10:45:38:24	09/10/2008
10:45:38:95	09/10/2008

Object

ObjectID	Type
p106	person

Object's Time

Object ID	Time	Date	Object Time ID
p106	10:45:36:91	09/10/2008	p106-s1
p106	10:45:37:92	09/10/2008	p106-s2
p106	10:45:37:93	09/10/2008	p106-s3
p106	10:45:38:24	09/10/2008	p106-s4
p106	10:45:38:95	09/10/2008	p106-s5

Object's Property

Object Time ID	Property ID
p106-s1	ploc1
p106-s2	ploc1
p106-s3	ploc1
p106-s4	ploc1
p106-s5	ploc1
p106-s1	pgm
p106-s2	pgm
p106-s3	pgf
p106-s4	pgm
p106-s5	pgm

Figure 4. From Images to the high level Facts. Figure shows the steps the Data Manger Agent takes when it receives new annotations from Image Analyzer.

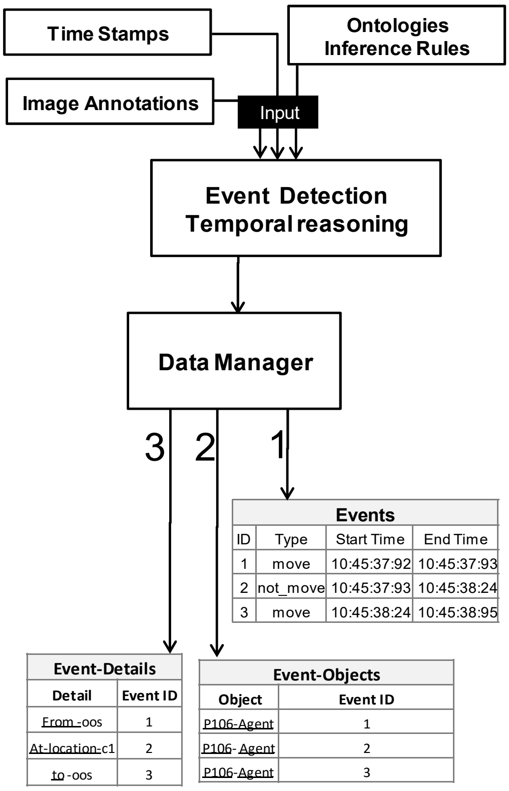
2.3. Prolog Inference Engine

The Prolog Inference Engine, or Event Detection Agent, detects events by monitoring annotations of objects in time instants. The Event Detection Agent recognizes events in a similar way to [5]. Then it asks the Data Manager Agent to assert inferred facts into the preserved data repository. An example of annotation for an event can be as follows:

```
<event type="move" id="jzglkncoos">
  <event_objects>
    <subject>
```

```
      v1
    </subject>
  </event_objects>
  <event_details>
    <from_location
inherited_from="location">
      oos
    </from_location>
  </event_details>
  <time stime="0" etime="2"/>
</event>
```

Figure 5 shows the steps the Data Manager takes when asserting inferred events by Event Detection agent.


Time Stamps
Ontologies Inference Rules
Image Annotations
Input
Event Detection Temporal reasoning
Data Manager
3
2
1

Events			
ID	Type	Start Time	End Time
1	move	10:45:37:92	10:45:37:93
2	not_move	10:45:37:93	10:45:38:24
3	move	10:45:38:24	10:45:38:95

Event-Details	
Detail	Event ID
From -oos	1
At-location-c1	2
to -oos	3

Event-Objects	
Object	Event ID
P106-Agent	1
P106- Agent	2
P106-Agent	3

Figure 5: decomposing and asserting an inferred event into data repository

2.4. Natural Language Interface

The main components of Natural Language Interface are a speech recognizer and a semantic parser. The speech recognizer used in this case was Sphinx [6]. The semantic parser is formed around ontology-related vocabularies. When the Semantic Parser gets an input, it looks for a proper mapping from the natural language utterances to the semantics that are defined in the ontologies of the system. This is done by a set of Definite Clause Grammar rules implemented in Prolog. The following is an example parse of a natural language phrase:

```
<Event prolog_rule_id="eventlexical1"
event_type="get_into" id="49702660200">
    <lexme pos="verb" class="occurance"
tense="none" aspect="progressive">
      entering
    </lexme>
</Event>
```

In the above example, the semantic parser detected the word "entering" as an event of type "get into" and it provides additional linguistic information

2.5. Knowledge Manager

The Knowledge Manager agent gets a semantics parse tree as an input; and it maps the parse information on to the facts previously asserted by the Data Manager Agent. The Knowledge Manager may use additional inference rules – e.g. to detect higher level types of events or to map linguistic variables on to their canonical forms. The process is as follows: 1. Find pattern of input parse tree, 2. Extract facts from Parse tree and assert them into the memory, 3. Translate facts into their canonical forms, 4. Locate video annotations with the criteria specified by extracted facts, and 5. Video Compilation of located video annotations. At the third step, translation of facts into their canonical forms, knowledge manager may use ontologies at different level of granularity to perform its task. For example, it may use definitions for higher level types of events, or it may use definitions at a finer level of granularity to translate facts and locate video annotations.

Looking ahead

Future enhancements to the experimental system, for which technical solutions have already been identified, include enabling the system speak a commentary on the information retrieved – the commentary will correspond to the selection criteria uttered by the user – as well as displaying the user's selection criteria as confirmatory on-screen text.

Our longer-term goal is to develop the experimental natural language interface into a comprehensive interactive natural language dialogue system. A dialogue-based information retrieval system will:

- prompt operators for important search criteria that they have omitted;
- suggest more discriminating constraints that the operator may wish to use for narrower searches;
- make and confirm inferences concerning vaguely formulated operator enquiries;
- and confirm turn-by-turn modifications to search constraints that the operator may introduce in the course of an evolving dialogue.

3. References

[1] Home Office Scientific Development Branch Imagery library for intelligent detection systems (i-LIDS), http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/video-based-detection-systems/i-lids/

[2] James F. Allen, 1983. "Maintaining Knowledge About Temporal Intervals," Communications Of ACM 26, vol.11.

[3] Francois, A. R., Nevatia, R., Hobbs, J., and Bolles, R. C. 2005. VERL: An Ontology Framework for Representing and Annotating Video Events. IEEE MultiMedia 12.

[4] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", in *Proc. of IEEE CVPR 2001*, Vol. 1 2001, pages 1-511 – 1-518

[5] Shet, V. D., Harwood, D., and Davis, L.S. 2005. VidMAP: Video Monitoring of Activity with Prolog, AVSS.

[6] Sphinx-4, A speech recognizer written entirely in the Java, http://cmusphinx.sourceforge.net/sphinx4/

[7] B Qasemizadeh, I O'Neill, P Hanna, and D. Stewart. A Data Model for Content Modelling of Temporal Media. FMN, 2009.