Event Composition with Imperfect Information for Bus Surveillance

Jianbing Ma, Weiru Liu, Paul Miller, Weiqi Yan
School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast BT7 1NN, UK
{*jma03,w.liu*}*@qub.ac.uk,*{*p.miller,w.yan*}*@ecit.qub.ac.uk*

***Abstract*—Demand for bus surveillance is growing due to the increased threats of terrorist attack, vandalism and litigation. However, CCTV systems are traditionally used in forensic mode, precluding an in-time reaction to an event. In this paper, we introduce a real-time event composition framework which can support the instant recognition of emergent events based on uncertain or imperfect information gathered from multiple sources. This framework deploys a rule-based reasoning component that can infer malicious situations (composite events) from a set of correlated atomic events. These are recognized by applying analytic algorithms to the multimedia contents of bus surveillance data. We demonstrate the significance and usefulness of our framework with a case study of an on-going bus surveillance project.**

***Keywords*-Multiple sensor, DS theory, Bus surveillance, Event composition, Event reasoning**

I. Introduction

For the past decade, the deployment of CCTV in major urban centres and cities has become well established. Recently, CCTV technology has begun to be deployed on public transport systems such as buses and trains e.g., [3], [9], etc. The deployment of this technology on moving platforms presents unique problems that are not encountered in conventional CCTV deployments, such as the uniquely harsh environments that exist on moving platforms leading to extreme temperature and humidity ranges as well as the need for mechanical robustness. Other problems are due to the sheer scale of the CCTV system which can contain 14-18 cameras on each bus, with approximately several hundred buses in a provincial UK city. The Intelligent Sensor Information System (ISIS) project is aimed at developing a state-of-the-art surveillance sensor network concept demonstrator for public transport that will overcome these problems. This will allow scalable and customizable technology for local and wide-area surveillance, and has the potential to enhance the safety of public transport systems and reduce crime.

Key to the success of CCTV technology on mobile platforms and elsewhere is the use of video analytics. Until recently, all of the research effort has gone in to the development of event detection. However, as this technology has started to migrate from the laboratory to the commercial sector, there is a growing realization of the need to manage the events generated by video analysis software. By manage we mean the representation, storage, reasoning and mining of events. A useful analogy here, is the management that is required for events generated by communication networks. The main difference here is that because events are detected in video data, their semantic content is much richer in that they are related to everyday happenings. An example of such a system is The IBM Smart Surveillance System (S3) [11], which has an open and extensible framework for event based surveillance. S3 consists of two components: the Smart Surveillance Engine, which provides the front end video analysis capabilities, and middleware for large scale surveillance, which provides data management capabilities.

However, the sheer quantity of data precludes the live streaming of such data over a wireless network to a control room. Even if this were possible, the analysis in real-time of such an amount of data is not feasible. Thus, mobile systems, more than most, need to adopt the event as the primary structure of surveillance systems, rather than data.

Therefore, one of the main tasks of event management systems is that of event composition, whereby patterns of events across a distributed network are detected. Event composition allows us to represent different events and also to instantly infer events of interest by applying rules to combine existing events. In addition, new situations can be captured by simply adding a new inference rule instead of modifying custom code, hence ensuring a flexible solution for evolving situations. For example, in a bus surveillance system, which has already modelled a passenger entering the bus doorway, a new event, such as abusive passenger behaviour towards the driver, can be inferred using a new rule that indicates if conditions related to a passenger and driver have occurred. Event composition can either be deterministic [6], [1], or probabilistic [13], [14], however, to date no one has addressed the problems of imperfect and conflicting information from different sources.

Imperfect information frequently occurs in real world applications. For example, in the case of a person entering a bus doorway, the person may be classified as male with a certainty of 85% by the classification analysis. However, the remainder does not imply that the person is female with a 15% certainty, rather, it is unknown. That is, we do not know how the remaining 15% shall be distributed on male or female. Hence by probability theory, this information can only be represented as $p(male) \geq 0.85$ and $p(female) \leq 0.15$ which is difficult to use for reasoning (e.g., Bayes Nets, etc). Imperfect information is usually caused by the unreliability

of the information sources. For example, the camera may have been tampered with, illumination could be poor, or the classifier training set may be unrepresentative. Any or all of these can result in imperfect information which cannot be represented by probability measures.

Another aspect of real world applications is that events can be detected from different sources, such as video, audio, and speedometers, etc. Hence, there is a need to use an event model that can handle events from multiple sources. Lastly, in real applications, there is usually useful domain knowledge that should be taken into account by the event management system. For example, recorded crime statistics can provide a likelihood of a criminal act occurring along bus routes at different times of the day.

In this paper, we describe an approach to event composition which is able to deal with imperfect information from multiple sources and domain knowledge using the Dempster-Shafer (DS) theory of evidence [5], [10]. Such an approach also enables inferences to uncertain events. To the best of our knowledge, our approach is the first that addresses imperfect information from multiple sources for event composition. Our approach provides a sound framework for surveillance applications, such as CCTV for buses.

The rest of the paper is organized as follows. Section 2 provides the preliminaries on DS theory. In Section 3, formal definitions of our event model are given including the definitions of events, multi-source events combination, event flow and event inference. We then provide a case study in Section 4. Finally, we conclude the paper in Section 5.

II. Dempster-Shafer Theory

Here we briefly introduce the main concepts in DS theory. Let Ω be a finite set containing exclusive and exhaustive answers to a question. We call Ω the frame of discernment and we denote $\Omega = \{w_1, \ldots, w_n\}$.

Definition 1: A mass function is a mapping $m : 2^\Omega \rightarrow [0, 1]$ such that $m(\emptyset) = 0$ and $\sum_{A \subseteq \Omega} m(A) = 1$.
A is called a focal element of $m(\cdot)$ when $m(A) > 0$. Let $\mathscr{F}_m$ denote the set of focal elements of $m(\cdot)$. For instance, *the person entering a doorway is a male with certainty 85%* is represented as $m(\{male\}) = 0.85$, $m(\{male, female\}) = 0.15$ if $\Omega = \{male, female\}$. This representation can be used for subsequent handling while the probabilistic representation $p(\{male\}) \geq 0.85$, $p(\{female\}) = 1 - p(\{male\}) \leq 0.15$ cannot.

DS theory has been applied in many real-world applications. One reason is its ability to represent imperfection (like the above example $m(\Omega) = 0.15$) and another is its ability to accumulate evidence. The latter means that it has a mechanism, the *Dempster's rule of combination*, to obtain the overall effect of multiple pieces of evidence by fusing them (by means of reinforcement of consistent information implied in them and discrediting conflict information). Let $m_1(\cdot)$ and $m_2(\cdot)$ be two mass functions over Ω from two distinct sources. Combining $m_1(\cdot)$ and $m_2(\cdot)$ gives a new mass function $m(\cdot)$ as follows:

$$m(C) = (m_1 \oplus m_2)(C) = \frac{\sum_{A \cap B = C} m_1(A) m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A) m_2(B)} \quad (1)$$

In real world applications, sources may not be completely reliable, so in [8], the *Discount rate* was defined with which a mass function can be discounted in order to reflect the reliability of evidence. Let r $(0 \leq r \leq 1)$ be a discount rate and m be a mass function, then the discounted mass function m^r is defined as

$$m^r(A) = \begin{cases} (1 - r) m(A) & A \subset \Omega \\ r + (1 - r) m(\Omega) & A = \Omega \end{cases} \quad (2)$$

When $r = 0$ the source is absolutely reliable and when $r = 1$ the source is completely unreliable. The ability of reliability discounting is another advantage of DS theory compared with probability theory.

III. Event Composition Framework

A. Event Model

Event definition: Definitions of an event from different research fields are very diverse and tend to reflect the content of the designated application. To make our event composition framework more general, in this paper, we define events as follows: an event is an occurrence that is *instantaneous* (event duration is 0, i.e., takes place at a specific point of time) and *atomic* (it happens or not). The atomic requirement of an event does not exclude uncertainty. For instance, when there is a person boarding a bus and this person can be a male or a female (suppose we only focus on the gender), then whether it is a male/female that boards the bus is an example of uncertainty. But *a male (resp. a female) is boarding the bus* is an atomic event which either occurs completely or does not occur at all. To represent uncertainty encountered during event detection, we distinguish an *observation* (with uncertainty) from *possible events associated with the observation* (because of the uncertainty). This can be illustrated by the above example: an observation is that *a person is boarding the bus* and the possible events are *a male is boarding the bus* and *a female is boarding the bus*. An observation says that something happened, but the *entity* being observed is not completely certain yet, so we have multiple events listing what that *entity* might be.

In the literature, there are two types of events, one type contains *external events* [1] or *explicit events* [13], [14] and the other consists of *inferred events*. External events are events directly gathered from external sources (within the application) while inferred events are the results of the inference rules of an event model.

Event Representation: Intuitively, a concrete event definition is determined by the application domain which contains all the information of interest for the application. But there are some common attributes that every event shall possess, such as

1. $EType$: describing the type of an event, such as, *Person Boarding Vehicle* abbreviated as PBV. Events of the same type have the same set of attributes.
2. $occurT$: the point in time that an event occurred.
3. sID: the ID of a source (e.g., ip addresses, here we just use numerical numbers, e.g., 0,1, to denote a source).
4. $reliab$: the degree of reliability of a source.
5. sig: the degree of significance of an event.

Formally, we define an event e as a tuple

$$e = (EType, occurT, sID, reliab, sig, v_1, \cdots, v_n)$$

where v_is are any additional attributes required to define event e based on the application. Attribute v_i can either have a single or a set of elements as its value, e.g., for attribute *gender*, its value can be *male*, or *female*, or {*male, female*} (however, it is not possible to tell the gender of a person when their face is obscured, so we introduce a value *obscured* as an unknown value for gender). Any two events with the same event type, source ID and time of occurrence are from the set of possible events related to a single observation. For example, $\mathsf{e}_1 = (\mathsf{PBV}, 20 : 05 : 31, 1, 0.8, 0.7, \mathsf{male}, \cdots)$ and $\mathsf{e}_2 = (\mathsf{PBV}, 20 : 05 : 31, 1, 0.8, 0.7, \{\mathsf{male}, \mathsf{female}\}, \cdots)$ are two events with v_1 for *gender* (we have omitted other attributes for simplicity).

Event Cluster: To represent a set of events from an observation, we introduce the concept of an *event cluster*. An *event cluster* EC is a set of events having the same event type, occurrence time and source ID, but with different $v_1, \cdots, v_n$ values. Events e_1 and e_2 above form an event cluster for the observation *someone is boarding the bus*.

For simplicity, in the following, we use $\mathscr{V}$ to denote the set of tuples $(v_1, \cdots, v_n)$ appearing in an event cluster EC and define m as the mass function over $\mathscr{V}$ representing uncertainty related to the observation.

For an event e in event cluster EC, we use $e.EType$ (resp. $e.occurT$, etc) to denote the event type (resp. time of occurrence, etc) of e, $e.v$ to denote $(v_1, \cdots, v_n)$, and $e.m$ to denote the value $m(e.v)$. By abuse of notations, we also write $EC.EType$ (resp. $EC.sID$, $EC.occurT$, $EC.reliab$) to denote the event type (resp. source ID, time of occurrence, reliability) of any event in EC since all the events in EC have the same values for these attributes.

It should be noted that the degree of significance of an event is self-evident (i.e., a function over $e.v$). For example, in bus surveillance, the event *a young man boards a bus around 10pm in an area of high crime risk* is more significant than the event *a middle-aged woman boards a bus around 6pm in an area of low crime risk*. However, due to space limitation, we will not discuss it further.

A mass function m over $\mathscr{V}$ for event cluster EC should satisfy the normalization condition: $\sum_{e \in EC} e.m = 1$. That is, EC does contain an event that really occurred. For example, for the two events, e_1 and e_2, introduced above, a mass function m can be defined as $m(\mathsf{male}, \cdots) = 0.85$ and $m(\{\mathsf{male}, \mathsf{female}\}, \cdots) = 0.15$.

For atomic events that are detected from algorithms, mass values can be estimated based on the algorithms used and the accuracy of detection etc. For inferred events using rules (as shown in Section 4), mass values are mainly estimated from domain knowledge. The main focus in our surveillance application is on finding malicious events, hence there is a need to assign mass values to these events indicating to what extent an event is potentially malicious. Since whether an event is malicious is usually closely related to other factors, domain (or expert) knowledge plays a key role here. For instance, to judge whether a passenger approaching the driver's cabin from the bus saloon is malicious (with $e.v = (threatLevel, \cdots)$), domain knowledge about the safety of the bus routes is useful. Some bus routes are relatively safer than others, so for these routes we will assign a smaller $m(\mathsf{Threat}, \cdots)$ value (and a larger $m(\mathsf{noThreat}, \cdots)$ value) for the above inferred event, whilst if the bus route is through a high crime risk area, this value could be much higher.

B. Event Flow

When a set of event clusters have the same event type and time of occurrence but different source IDs, we call them *concurrent* event clusters. This means that multi-model sensors may have been used to monitor the situation. Therefore, we need to combine these event clusters since they refer to the same observed fact from different perspectives. The combined result is a new event cluster with the same event type and time of occurrence, but the source ID of the combined event will be the union of the original sources. The combination of event clusters is realized by applying Dempster's combination rule on discounted mass functions. That is, the mass function of an event cluster is discounted with the discount rate defined as the reliability of the source.

Definition 2: Let $EC_1, \cdots, EC_k$ be a set of concurrent event clusters, and $m_1^r, \cdots, m_k^r$ be the corresponding discounted mass functions over $\mathscr{V}$, m be the mass function obtained by combining $m_1^r, \cdots, m_k^r$ using the Dempster's combination rule, then we get the combined event cluster $EC = \oplus_{j=1}^{k} EC_j$ such that $\forall e \in EC$, we have $e.v \in \mathscr{F}_m$, $e.EType = EC_1.EType$, $e.occurT = EC_1.occurT$, $e.sID = \{EC_1.sID, \cdots, EC_k.sID\}$, $e.reliab = 1$, and $e.m = m(e.v)$. Conversely, for each focal element A in $\mathscr{F}_m$, there exists a unique $e \in EC$, s.t., $e.v = A$.

As stated earlier, $e.sig$ (significance) is a function over $e.v$.

Event models usually use the concept *Event History* (EH) to describe the set of all events whose occurrences fall between a certain period of time. However, in our framework, given a set of event clusters, we first carry out event combination, and then retain only the combined event clusters. So what we have is not a *history*, because of this, we call it an *event flow* and denote it as EF. We use $EF_{t_1}^{t_2}$ to represent a set of combined event clusters

whose occurrences fall between t_1 and t_2. Since an event flow contains the combined events, to some extent, we have already considered the *opinions* (of the original events) from different sources.

C. Event Inference

Inference rules: Event inferences are expressed as a set of inference rules which are used to represent the relationships between events. An inference rule R is defined as a tuple $R = (LS, EType, Condition, m_{IEC})$ where:

LS, abbreviated for *Life Span*, is used to determine the temporal aspect of the rule R [4], [1], [14]. LS is an interval determined by a starting point and an end point, or an initiator and a terminator, respectively. The starting point and end point are two points in time which can be determined by the event flow that is known at the time a rule is executed. For instance, a starting time point may refer to the occurrence time of a specific event, a prior given time, etc and an end time point can be the occurrence time of another event, a prior given time, or a time period plus the starting point, etc.

$EType$ is the event type of the inferred event cluster. For example, SAD (abbr. for Shout At Driver) is an inferred event type.

$Condition$ is a conjunction of a set of conditions used to select appropriate events from the event flow to infer other events. The conditions in $Condition$ can be any type of assertions w.r.t the attributes of events. For example, let e_1 and e_2 both denote a male loitering event and e_3 denote a person shouting event, then

"$e_1.pID = e_2.pID \wedge e_1.gender = male$
$\wedge e_1.location = e_2.location = DriverCabin$
$\wedge e_2.occurT - e_1.occurT \geq 10s \wedge e_1.occurT \leq e_3.occurT \leq e_2.occurT \wedge e_3.volume = shouting$"

is a valid $Condition$. Note that for each inference rule, we only select events in the event flow within the lifespan LS (denoted by $LS(EF^t_{t'})$). We denote the set of events referred to by a ***Condition*** as $Evn(Condition)$.

m_{IEC} is the mass function for the inferred event cluster and it is in the form of $(< v^1_1, \cdots, v^1_n, mv_1 >, < v^2_1, \cdots, v^2_n, mv_2 >, \cdots, < v^k_1, \cdots, v^k_n, mv_k >)$ where each mv_i is a mass value and $\sum_{i=1}^{k} mv_i = 1$. We will explain this in detail when discussing rule semantics next.

To differentiate inferred events from other events, we use -1 to denote the source ID of an inferred event cluster and the occurrence time is set as the point in time an inference rule is executed. Moreover, the reliability is set to 1 as we assume that the inference rules are correct.

Semantics Intuitively, the semantics of using an inference rule R is defined as follows. Given an event flow $EF^t_{t'}$, if $Condition$ of any rule R is true at some time point $t^* > t'$, then an event cluster is inferred from rule R with mass function m_{IEC}. Otherwise, no events are inferred.

Formally, for any vector $< v^i_1, \cdots, v^i_n, mv_i >$, if $Condition(LS(EF^t_{t'})) = true$ (meaning that $Condition$ can be satisfied by events selected from $LS(EF^t_{t'})$), then a corresponding event e_i is generated whose event type is $EType$, source ID is -1, occurrence time is the time of rule execution, reliability is 1, $e_i.v = (v^i_1, \cdots, v^i_n)$ (and $e.sig$ is a function over $e_i.v$), and $m_{IEC}(e_i.v) = mv_i, 1 \leq i \leq k$.

Since events in $Evn(Condition)$ are associated with uncertainty, to get the joint degree of certainty of these events, we need to consider their corresponding mass values. This is done by calculating the *likelihood* of a set of events for which the details are omitted due to space limitation.

IV. A Case Study on Bus Surveillance

We use a simplified real world example on bus surveillance as a case study to illustrate how our event reasoning framework can be applied. First, we describe some scenarios for atomic events that can be obtained from various sources (cameras via gender classification algorithms, microphones via shouting detection). Some exemplar scenarios are:

Passenger boarding a bus from front/back door
Passenger gender: Male, Female
Passenger exiting a bus
Location of the door: Front door, Back door
Passenger ascending/descending stairway on a bus
when Bus is stationary, moving
Passenger loitering on a bus
location: Stairway, Driver's Cabin
Passenger sitting down in saloon area
Passenger standing in saloon area
Passenger shouting on a bus

Atomic events from these scenarios are actual events that happened at certain time with sufficient details (attributes). Below are some atomic events of these scenarios.

For **Passenger boarding a bus from front/back door** we have $e_1 = (\mathsf{PBV}, 21:05:31, 1, 0.9, 0.7, \mathsf{male}, 3283, \mathsf{fDoor}, \mathsf{double\ decker\ bus}, \mathsf{Bus1248}, 45, \mathsf{GPS_p}, \mathsf{v})$ where PBV is for event type *Person Boarding Vehicle*, male is the value of attribute *gender*, 3283 is the *person ID*, fDoor means that the person boards the bus from the *Front door*, double decker bus, Bus1248, 45 stand for the *Bus type, Number*, and *Route*, respectively. GPS_p records the position of the bus by GPS, and v indicates the velocity by the speedometer.

For **Passenger exiting a bus**
$e_2 = (\mathsf{PEV}, 21:07:12, 1, 0.9, 0.3, 3283, \mathsf{bDoor}, \mathsf{double\ decker\ bus}, \mathsf{Bus1248}, 45, \mathsf{GPS_p}, \mathsf{v})$ where bDoor indicates the person exits from the *Back door*.

For **Passenger ascending/descending stairway**
$e_3 = (\mathsf{PADS}, 21:07:12, 1, 0.9, 0.3, 3283, \mathsf{double\ decker\ bus}, \mathsf{Bus1248}, 45, \mathsf{GPS_p}, \mathsf{v}, \mathsf{Crd})$ where Crd denotes the coordinates of the person on the bus.

For **Passenger loitering**
$e_4 = (\mathsf{PL}, 21:07:12, 1, 0.9, 0.3, 3283, \mathsf{DriverCabin}, \mathsf{double}$

decker bus, Bus1248, 45, $\mathsf{GPS_p}$, v, Crd) where DriverCabin stands for the person is within the *Driver's Cabin*.

For **Passenger sitting down/standing in saloon area** $e_5 = (\mathsf{PSS}, 21:07:12, 1, 0.9, 0.3, 3283, \mathsf{Stand}, \text{double decker bus}, \mathsf{Bus1248}, 45, \mathsf{GPS_p}, \mathsf{v}, \mathsf{Crd})$ where Stand stands for the person is standing.

For **Passenger shouting** $e_6 = (\mathsf{PS}, 21:07:12, 1, 0.9, 0.3, \mathsf{DriverCabin}, \mathsf{volume}, \text{double decker bus}, \mathsf{Bus1248}, 45, \mathsf{GPS_p}, \mathsf{v})$ where volume indicates the volume of the shouting. Note that shouting is recorded by a microphone which cannot indicate the specific person who is shouting.

Suppose analysis of crime data on buses reveals a high crime risk on bus route 45, then the surveillance system needs to focus on some potentially malicious events. For simplicity, in the following, we omit all the bus details in these events description for this specified bus. Now we use a set of rules to illustrate how these atomic events can be correlated to determine composed events (i.e., inferred events) which are more meaningful and significant. For example, a person boarding a bus with their face obscured may imply that a passenger assault or vandalism is about to occur. Hence the event of someone boarding a bus with their face obscured can be combined with other subsequent events to infer a composite event of high significance.

Rule 1: This rule describes that a person, either a male or with their face obscured, who has moved from saloon to drivers cabin and is now loitering, could be exhibiting abusive behaviour towards the driver. The details of the rule are: $R_1 = (LS_1, EType_1, Condition_1, m_1)$ such that $LS_1 = (0, TPL)$ where TPL stands for the occurrence of a PL event, $EType_1 = OPD$ abbreviated for Obscured Person-Driver, $Condition_1$ is defined as $e_i.EType = PBV \wedge e_j.EType = PSS \wedge e_l.EType = PL \wedge e_i.gender \in \{male, obscured\} \wedge e_i.pID = e_j.pID = e_l.pID \wedge e_l.location = DriverCabin$ and m_1 could be $< 3283, Crd, hasThreat, 0.3 >$,
$< 3283, Crd, \{noThreat, hasThreat\}, 0.7 >$.

Rule 2: This rule describes that a person, who is male or with their face obscured, is loitering at the drivers cabin and there is shouting, could be exhibiting abusive behaviour towards the driver. This rule is defined as $R_2 = (LS_2, EType_2, Condition_2, m_2)$ such that $LS_2 = (TPL, TPL{+}120)$ where TPL stands for the occurrence of an PL event, $EType_2 = SAD$ abbreviated for Shouting AT Driver, $Condition_2$ is $e_i.EType = PBV \wedge e_j.EType = PL \wedge e_l.EType = PS \wedge e_i.gender \in \{male, obscured\} \wedge e_l.volume = shouting \wedge e_i.pID = e_j.pID \wedge e_j.location = e_l.location = DriverCabin$,
and m_2 could be $< 3283, 0.3 >, < \{3281, 3282\}, 0.7 >$.

Other rules can be defined similarly to infer events of interests. Depending on the significance and likelihood (calculated from these events' mass values) of the composite event, live video from the event source shall be put at the front of the queue for streaming back to the surveillance operations centre for human analysis.

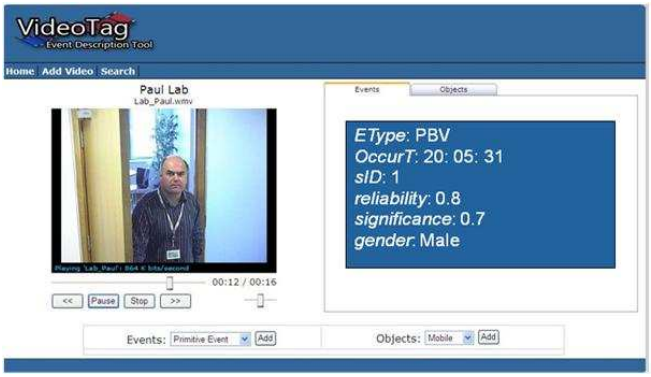

Figure 1. Main Page of the VideoTag Tool

A typical scenario of using rules can be as follows. Assume that a person (denoted as P) boards the bus with their face obscured and sits down in the saloon area. After a period P stands up and moves to the driver's cabin. After a short period, shouting is recorded. With the bus surveillance system, atomic events generated by P's activities and behaviours are detected and analyzed. Conditions of rules in the system are constantly checked with the current collections of events in the event flow. When the Condition of a rule is met, this rule is triggered. For this example, rules R1, R2 are triggered and a new event is generated describing that a face-obscured person is being abusive to the driver and the video is streamed live to the control centre. As a consequence, the analyst in the control room can verify whether or not an event is occurring and if the former, engage with the offender over a two way audio-visual link and warn them to desist from their continuing abuse.

In addition to deploy rules for reasoning, atomic event descriptions must be effectively stored for efficient retrieval and use. To facilitate this, we have implemented an event description system called *VideoTag* which is a web based tool implemented using C#. It allows users to describe atomic events in XML format. Figure 1 shows the main screen of the tool.

Example 1: Atomic event e_1 defined in Section 3.1 (with $m(e_1.v) = 0.7$) can be represented as

```
⟨event type = "PBV"
      id = "c6caa61347bf43db921454bc2a485c3e"⟩
  ⟨occurT⟩21 : 05 : 31 pm 18/02/2009⟨/occurT⟩
  ⟨source⟩1⟨/source⟩
  ⟨reliability⟩0.8⟨/reliability⟩
  ⟨significance⟩0.7⟨/significance⟩
  ⟨gender⟩male⟨/male⟩
  ⟨personID⟩3283⟨/personID⟩
  ⟨door⟩front⟨/door⟩
  ...
```

⟨massValue⟩0.7⟨/massValue⟩
⟨/events⟩
Here the id is automatically generated as an universal event ID by the program.

V. Conclusion

In this paper, we introduced an event composition framework which is able to represent and reason with events with uncertainty and imperfection from multiple sources. A rule based inference system is used to derive composed events of significant interests from atomic events that are directly detected. We demonstrated the main functions of the framework using a case study from a real world example in a bus surveillance application. Fully implemented, this framework should be a great assistance for detecting potentially dangerous behaviors and actions on public transport.

Time synchronization is important as we need to merge data from multiple sensors. In the bus surveillance scenario, it is reasonable to assume that all sensors/cameras on a single bus are well synchronized, hence in this paper, we will not consider this issue further. In addition, we will use data mining tools to find association rules in a realistic bus surveillance database. For parameter estimation, we find there are four kinds of parameters used in our paper: rule parameters, event parameters, reliabilities and attribute values. Rule parameters (e.g., mv_i, LS, etc) can be calculated (or learned) by association softwares; event parameters (e.g., a person is classified as male with 85% certainty) can be set by algorithms; reliabilities can be obtained by historical data and attribute values (e.g., bus speed, time point) can be obtained by relative instruments. In summary, different parameters can be estimated in different ways.

Historically, computer vision domain has focused on the learning and recognition of events (e.g., [2], [7], etc). Recently, a few papers, e.g., [15], [12], address the issue of composite events. [15] simply uses hand-crafted rules to define a composite event rule using IF/AND/OR/THEN while [12] uses empirical background knowledge as rules. No attempt is made to deal with uncertainty in the atomic events and rules in both papers.

For future work, we want to (A) systematically define the whole range of different types of events and rules within the bus surveillance application area; (B) use SIMWALK to simulate pedestrian flows on bus to trigger these defined events and rules and tune the parameters used in the description of events and rules, e.g., rule life span, reliability, mass values, all the possible values of an attribute, etc; (C) elicit domain knowledge from either domain experts or learned from history data, such as crime statistics; (D) consider biased reliability values when we cannot tell what the reliability of a source is at a certain time, e.g., when the background noise is high, the audio classifier will be biased very strongly towards male; and (E) extend our framework to handle uncertainty in the atomic event timestamp.

Acknowledgement

We would like to thank Stephen Russell for VideoTag and Mr Andrew Stevens, Andrew Newton for valuable discussions on bus surveillance. This research work is partially sponsored by the EPSRC projects EP/D070864/1 and EP/E028640/1 (the ISIS project).

References

[1] A. Adi and O. Etzion. Amit - the situation manager. *VLDB J.*, 13(2):177–203, 2004.

[2] F. Bremond, M. Thonnat, and M. Zuniga. Video understanding framework for automatic behavior recognition.

[3] Bsia. Florida school bus surveillance. In *http://www.bsia.co.uk/LY8VIM18989_action;displays tudy_sectorid;LYCQYL79312_caseid;NFLEN064798*.

[4] S. Chakravarthy-S and D. Mishra. Snoop: an expressive event specification language for active databases. *Data and Knowledge Engineering*, 14(1):1–26, 1994.

[5] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Ann. Stat.*, 28:325–339, 1967.

[6] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Compostite event specification in active databases: Model & implementation. In *Proc. of VLDB*, pages 327–338, 1992.

[7] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129162, 2004.

[8] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential reasoning systems. In *Proc. of 5th AAAI*, pages 896–903, 1986.

[9] Gardiner Security. Glasgow transforms bus security with ip video surveillance. In *http://www.ipusergroup.com/doc-upload/Gardiner-Glasgowbuses.pdf*.

[10] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[11] C. F. Shu, A. Hampapur, M. Lu, L. Brown, J. Connell, A. Senior, and Y. Tian. Ibm smart surveillance system (s3): a open and extensible framework for event based surveillance. In *Proc. of IEEE Conference on AVSS*, pages 318–323, 2005.

[12] L. Snidaro, M. Belluz, and G. L. Foresti. Domain knowledge for surveillance applications. In *Proc. of 10th Intern. Conf. on Information Fusion*, 2007.

[13] S. Wasserkrug, A. Gal, and O. Etzion. A model for reasoning with uncertain rules in event composition. In *Proc. of UAI*, pages 599–608, 2005.

[14] S. Wasserkrug, A. Gal, and O. Etzion. Inference of security hazards from event composition based on incomplete or uncertain information. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1111–1114, 2008.

[15] N. Zouba, F. Bremond, M. Thonnat, and V. T. Vu. Multi-sensors analysis for everyday activity monitoring. In *Proc. of SETIT*, pages 25–29, 2007.