Workshop on Activity monitoring by multi-camera surveillance systems (AMMCSS) in conjunction with 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2011

Real Time Complex Event Detection for Resource-Limited Multimedia Sensor Networks

Fadi Al Machot, Kyandoghere Kyamakya
Institute of Smart System Technologies
Klagenfurt University
{firstname.lastname}@aau.at

Bernhard Dieber, Bernhard Rinner
Institute of Networked and Embedded Systems
Klagenfurt University
{firstname.lastname}@aau.at

Abstract

This paper presents a real-time complex event detection concept for resource-limited multimedia sensor networks. A comprehensive solution based on Answer Set Programming (ASP) is developed. We show that ASP is an appropriate solution to detect a large number of simple and complex events (video-audio understanding) on platforms with limited resources e.g. power consumption, memory and processing power. We underline the major problems of the existing paradigms for complex event detection (based on e.g. logic programming and Semantic Web), with a special focus on the major challenges which reduce the performance of real-time event detection. Finally, we demonstrate the high performance of ASP compared to that of Semantic Web.

1. Introduction

Detection of different events or situations has become an important topic in audio and video surveillance systems in the last years. Especially the surveillance of public areas such as airports or train stations has been in the focus of research and development e.g. audio and visual events in sports [18] or audio events in the military [14]. Some surveillance systems have to be be installed in resource - limited areas/situations where power, memory and processing resources are limited. This limitation is a real challenge for surveillance systems researchers. It is especially the case when the system has to be robust, run on chip and detect events in real time.

The existing solutions for event detection in surveillance systems can be divided in two main groups: a) model-based detection systems (probabilistic or statistical ones), and b) rule-based detection systems (logic programming and context modeling). On the other hand some other interesting paradigms such as constraint satisfaction programming (CSP), quantified boolean formulas (QBF), or first order logic (FOL)do unfortunately not offer the expressive capability to define the knowledge necessary to model the spatial and temporal context information at stake in video understanding systems. The semantics of ASP is more expressive than those of related formalisms e.g. propositional satisfiability (SAT), constraint satisfaction problems (CSP), and integer linear programming (ILP).

In this paper we are solely dealing with the detection of complex events in short term, that is, within some seconds only (or up to maximum one minute). The events we are considering are divided in two classes, simple events and complex events:

1. Simple Event: This is the simplest form of events e.g. run, walk, shot, etc.

2. Complex Event: a complex event which is the combination of the simple events e.g. groups of persons are running, group of persons are fighting, group of persons are running in different direction, etc.

We propose a solution based on answer set programming (ASP) to realise a system able to detect complex events in real time, is robust and can easily run on chip (DSP or FPGA).

2. Related Work

Several papers on complex event detection have been published. The main approaches involves supervised learning for event detection. Many works use Bayesian Networks for event recognition such as [10], [29], [2] or [12]. Others use support vector machines [17] or hidden markov models [23] [11].

These methods show in some scenarios a high detection rate and in others very low detection rate. Still, using model based complex event detection needs many (a huge number of) trainings samples. Further, the classification involved usually does not support different types of events.

A possible way of dealing with scene understanding is an ontology based context modeling and reasoning. It is

978-1-4577-0845-9/11/$26.00

not only important to record and detect different events. An important issue is to understand the scene. Some works are based on context modeling and reasoning, see Refs [28], [26], [30], [13], [27] or [24]. The problem of these concepts remains the limitation of running on chip.

Another way to build an embedded service for complex event detection is to use logic programming whereby several approaches have been illustrated in [25] [4] [21]. Hakeem and Shah [9] have presented a hierarchical event representation for analysing videos. The temporal relations between the sub-events of an event definition are represented using the interval algebra of Allen and Ferguson [1]. In the ASP based approach for event recognition, however, the availability of the full power of logic programming is one of the main features. It further allows activity definitions to include spatial and temporal constraints. In particular, some logical programming languages do not offer arithmetic operation built-ins and numeric constraints can affect decidability.

A well-known system for activity recognition is the Chronicle Recognition System (CRS). The language includes predicates for persistence and event absence [3]. The CRS language does however not allow mathematical operators in the constraints of the temporal variables. Consequently, CRS cannot be directly used for activity recognition in video surveillance applications. Shet et al. have presented a logic programming approach for activity recognition [19]. The temporal aspects of the definitions of Shet, Davis et al. are not well represented, there are no rules for computing the intervals in which a complex event takes a place.

There are many interesting applications based on ASP [8] e.g. in planning, reasoning about action, configuration, diagnosis, space shuttle control, spatial, temporal and probabilistic reasoning, constraint programming, etc. The rest of the paper is organized as follows: Section 4 overviews a case study and a detailed description of the proposed concept based on answer set programming. Then section 5 describes the performance results of the concept developed. Finally, section 7 presents concluding remarks followed by an outlook of future works.

3. Complex Event Detection for Audiovisual Sensor Networks

Complex event detection in audio-video sensor networks is based on three main steps. The first step is the extraction of features using object recognition and object tracking algorithms. Second, scenarios and the rules to detect simple events, like walking, running or shouting must be defined. Finally, complex events are detected by combining the simple events.

Beside these main steps, we have to define the context model which describes all the information that may influence the way a scene is perceived. The state of an environment is defined as a conjunction of predicates. The environment must be modeled to retrieve the position, orientation and types of objects, as well as position, information and state of other objects from information observed in the environment.

After building the context model, we need a context interpreter which provides the context reasoning services including inferring contexts, resolving context conflicts and maintaining the consistency of context knowledge base. Different inference rules can be specified and input into the reasoning engines [28].

For the description of regions of interest in the image or to detect the coordinates of moving objects close to the important regions, geometric correction is required.

Additionally, in multi-sensor networks (e.g audio and video), the extraction of features from video-audio streams is the basis for data fusion and is needed to combine data in order to estimate or predict entity states. Data fusion techniques combine data from multiple sensors to achieve more specific inferences than what could be achieved by using a single sensor. Some proposed techniques for data fusion are presented in [6]. Most common solutions are based on the numerical properties.

If fusion of data from multiple cameras is necessary to monitor a specific area, then finding overlapping regions of multiple views adds even more complexity to the problem. Several methods for tracking in multiple views have been proposed. Most consist of two steps. The first step is performed on each single-view separately. The second step is a multi-view data fusion step. In the single-view stage, features are extracted and estimations are made. Then, data is fused between multiple views to obtain the final results. When the system predicts that the current camera no longer has a good view of the object then the system must switch to another camera [6] [16].

4. Case study: Smart Resource-Aware Multi-Sensor Network

The SRSnet project aims at constructing a smart resource-aware multi-sensor network. The goal is to deploy a sensor network consisting of both video and audio sensors that is capable of detecting complex events in an environment with limited infrastructure. This especially means that there is no access to a power grid and thus the sensor nodes must be able to operate on battery and renewable energy for as long as possible. SRSnet needs not only to record and transmit sensor information but also performs on-board data processing while running on battery and renewable energy for as long as possible. An integral part of the SRSnet project is the detection of high level events.

Low level events detected by audio and video processing are the bricks used to construct high-level complex events. Additionally, the network must react to events and to new task assignments. This requires the presence a module for dynamic network reconfiguration to reconfigure sensor parameters and nodes according to events, task assignments and resource requirements.

A resource aware multimedia sensor network like SRSnet can be deployed in environments like national parks to help protect sensitive environments. We will demonstrate our project in the National Park Hohe Tauern in Austria. To archive events and provide an interface to users, we use a multimedia data warehouse that collects detected events and multimedia artifacts. Users can then query the database for interesting events in time and space. The data warehouse is meant to be deployed outside of the sensor network itself (i.e. as a cloud service). To feed information into the data warehouse we use web services which are called from the network. This architecture enables us to save energy by only connecting to the data warehouse on demand. A persistent connection is not needed.

In SRSnet we want to detect complex event that are interesting in a National Park environment like shooting, persons in forbidden areas (e.g. areas with sensitive flora), or wildlife. The events of multiple sensors (audio and video) will be combined by the complex event detection system in order to gain a global view on the complex events.

4.1. Complex Event Detection based on Answer Set Programming

A logic program in the language of AnsProlog (also known as A-Prolog) is a set of rules of the form:

$$a_0 \leftarrow a_1, ..., a_m, nota_{m+1}, ..., nota_n \qquad (1)$$

where $0 \leqslant m \leqslant n$,each a_i is an atom of a_i propositional language and not represents $negation - as - failure$. A negationas- failure literal (or naf-literal) has the form $nota$, where a is an atom. Given a rule of this form, the left and right hand sides are called the $head$ and $body$, respectively. A rule may have either an empty head or an empty body, but not both. Rules with an empty head are called constraints, while those with an empty body are known as $facts$. A definite rule is a rule which does not contain naf-literals, and a definite program is composed solely of definite rules [20].

Let X be a set of ground atoms. The body of a rule of the form (1) is satisfied by X if $\{a_{m+1}, ..., a_n\} \bigcap X = \phi$ and $\{a_1, ..., a_m\} \subseteq X$. A rule with a non-empty head is satisfied by X if either its body is not satisfied by X, or$a_0 \in X$. A constraint is satisfied by X if its body is not satisfied by X. Given an arbitrary program, Π and a set of ground atoms, X, the reduct of Π w.r.t. X, Π^X, is the definite program obtained from the set of all ground instances of Π by:

1. deleting all the rules that have a naf-literal not a in the body where $a \in X$, and
2. removing all naf-literals in the bodies of the remaining rules.

A set of ground atoms X is an answer set of a program Π if it satisfies the following conditions:

1. If Π is a definite program, then X is a minimal set of atoms that satisfies all the rules in Π.
2. If Π is not a definite program, then X is the answer set of Π^X. (Recall that Π^X is a definite program, and its answer set is defined in the first item [20].

Logic programming can be extended to allow us to represent new options for problems in the head of the rules. ASP gives us this ability with ordered disjunctions. Using ASP under specific conditions, reasoning from most preferred answer sets gives optimal problem solutions.

Through Logic Programs with Ordered Disjunction (LOPDs) such as normal logic programs we are able to express incomplete knowledge through the use of default negation. This allows us to represent performances among intended properties of problem solutions which depend on the current context [5]. Also, expressing properties in NP (i.e. properties whose verification can be done in polynomial time), where answer sets of normal logic programs can be generated through solutions and polynomial time proofs for such properties. The solution of such problems can be carried out in two steps [7]:

1. Generate a candidate solution through a logic program
2. Check the solution by another logic program

ASP provides the combination between spatial and temporal relationships among sensor nodes, where this combination helps to detect different scenarios in a logic sequence of events.

Complex event detection (generally) in audio video sensor networks is based on three main steps: a) the first step is the extraction of features using object recognition and object tracking algorithms; b) then the definition of scenarios and of the rules to detect simple events, like walking, running, shouting, etc.; c) finally, is the detection of complex events by combining the simple events together to detect a complex scenario. In the development of video understanding systems (outdoors systems) the geometric correction is needed for the description of region of interests in the image or to detect the coordination of moving objects nearby important regions. However, in multi media sensors network (

i.e., audio and video) the extraction of features from video-audio streams is the basic processing. Then data fusion is needed to combine data or in formation to estimate or predict entity states. Data fusion techniques combine data from multiple sensors to achieve more specific inferences than could be achieved by using a single, some proposed techniques are in [6]. Most common solutions are based on numerical properties. Still, fusion of multiple cameras views is also needed to monitor a specific area; hereby the problem of overlapped individual views is a complex problem. For building an event detection system based on Answer Set Programming (ASP). We have to design a knowledge base (KB) and to define the ASP rules to detect the desired events.

4.2. The Structure of the Knowledge Base

The structure of our knowledge data base consists of different entities entities, the most important two parts are: the object entity and the sound entity.
Object entity: object types are observed from sensors e.g. person, dog, car, etc.

Object entity properties	*Sound entity properties*
objectId	hasSoundType
hasObjectType	hasSoundArrayID
hasSpeed	hasSoundTime
hasDate	hasSoundDate
hasTime	hasSoundCorX
hasDirection	hasSoundCorY
hasCameraId	hasUncertaintySoundcCor
hasFrameId	hasUncertaintySoundType
hasX	
hasY	
hasUncertaintyType	
hasUncertaintyCorType	

Table 1. Object properties and the additional properties of sound entities

Sound entity: sound types are observed from sensors e.g. shot, scream and howl, etc. The properties of object and sound entities are shown in Table 1.

The knowledge base is used as input for the solver to generate the answer sets, which present the detected simple and complex events.

4.3. The defined rules based on Answer set Programing

The scene context plays a major role during the detection of an event. The objects have two different types of features, sound features and video features. These features are extracted from audio/video subsystem. The rules consist of:

1. The specification of the direction (the directions of the objects are divided in 8 different directions, south, southEast, southEastEast,..etc).

2. The specification of the zones (the project area is divided in different zones like children zone, forbidden zone,...etc).

3. The specification of the sound entity and the object entity.

Uncertainty can not be avoided in practical visual surveillance applications. We consider now one class of uncertainty, the one called detection uncertainty. Detection uncertainty is a part of our knowledge base. We consider two types, the first one is the uncertainty of the localization and the second one is the uncertainty of the object type classification. We are getting these uncertainty values from the low level feature extraction. In the actual phase of our project we do not have to consider the logic uncertainty since our rules in the KB are always true. We use a real-value weight to represent the confidence of each rule in the knowledge base.

As an example, to detect a complex event such as a running group of persons, we need to identify at least two persons. If the distance between these two persons is less than 3 meters and both are running, then the whole group is running. The condition on the distance is specified in the predicate $near(X_1, X_2)$, where X_1 and X_2 present the observed persons:

```
near(X1,X2):-
X1!=X2,
dist(X1,X2,D),
D<3,
hasObjectType(X1,OT1),
hasObjectType(X2,OT2),
OT1=person,
OT2=person,
hasTime(X1,T1),
hasTime(X2,T2),
T1=T2.
```

As test scenario, for illustration, the detection of the complex event that (a group of people is running in different directions) is happening when at least there are three persons nearby each other and they are moving in three different directions , i.e. $northWest$, $northEast$, $southEast$. and at the same time. The last three conditions make sure that the detected persons are not the same.

```
diffDirections9(X1,X2,X3):-
northWestWest(X1),
northEast(X2),
southEast(X3),
```

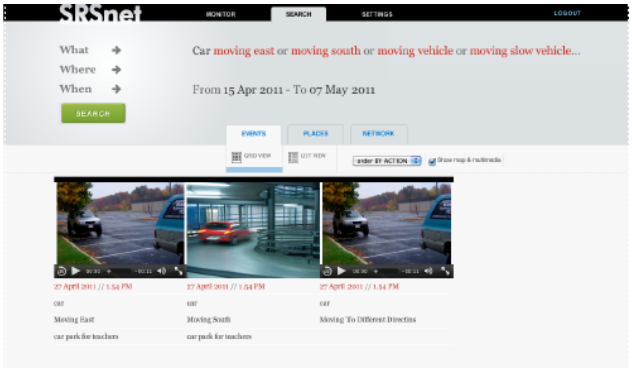

Figure 1. A screen shot of the proposed system

```
near(X1,X2),
near(X2,X3),
hasTime(X1,T1),
hasTime(X2,T2),
hasTime(X3,T3),
T1=T2,
T2=T3,
T3=T1.
X1!=X2,
X2!=X3.
X1!=X3.
```

5. Results

The evaluation of our system is done for different scenes (see Figure 2). The test environment is a park place, equipped with several cameras and audio arrays. The events we defined are divided in two groups: simple and complex events. After conducting experiments on a benchmark dataset, we realized that, whenever the accuracy of the detection is high, then our detection ratio is over 94% for all complex events, see Table 1.

The complex event	Specificity	Sensitivity
A group of persons are running	100%	98.4%
A group of persons are fighting	89%	94.4%
A group of persons are running in different directions	92.1%	96.2%

Table 2. The Performance of the reasoning system

To measure the runtime behavior of the answer set programming approach, we performed several tests on an embedded platform that will also be used in the SRSnet project. We use Atom-based embedded boards as example platforms. We tested all algorithms on pITX-SP 1.6 plus board manufactured by Kontron[1]. It is equipped with a 1,6 GHz Atom Z530 and 2GB RAM.

The following results have been obtained: the average execution time to detect all simple and complex events is 0.40 seconds, the minimum execution time is 0.39 seconds and the maximum execution time is 0.46 seconds. Sixteen simple and eight complex ASP rules were used. In average, 980 features were in the knowledge base. The results show that a complex event detection can be executed once or twice a second; this enables the audio/video subsystem to collect sufficient data for detecting the next complex events. For the evaluation we use $iClingo$[2] as a solver for ASP [8]. It is written in C and can run under Windows and Linux. The reason of the high performance of ASP on chip that the representation of the knowledge base and the solver size are not expensive. The Solver is 47 Kilo byte and is written in C, where most of the existed hardware platforms are able to execute it.

6. Acknowledgments

This work has been developed in the frame of Smart Resource-aware Multi-sensor Network project (SRSnet). It is funded by the European Regional Development Fund, Interreg IV Italia-Austria Program.

7. Conclusion

The detection of different events or situations has become an important topic in audio and video surveillance systems in the recent years. In this work we have demonstrated the advantages and disadvantages of the most important technologies. We have also shown, that the use of Answer Set Programming can significantly reduce the effort needed to detect complex events while obtaining the same level of quality in the detected events. ASP is expressive, convenient, and supports formal declarative semantics. We showed that ASP can be used to detect a large number of simple and complex events within a reasonable time frame that allows for real-time operation. We proved that ASP is an appropriate solution for complex event detection systems in multi sensors networks with limited resources.

In our future work, we will use rule decisions systems, which generate decision rules based on decision tables. By using Rough-set theory and genetic algorithms, we integrate the generated rules in ASP for detecting events where it is not possible to describe the related behavior.

References

[1] J. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531, 1994. 2

[2] D. Arsic, F. Wallhoff, B. Schuller, and G. Rigoll. Video based online behavior detection using probabilistic multi stream fusion. *IEEE International Conference on Image Processing*, pages 606–609, 2005. 1

[1] http://www.kontron.com

[2] http://potassco.sourceforge.net

[3] A. Artikis and G. Paliouras. Behaviour recognition using the event calculus. *Artificial Intelligence Applications and Innovations III*, pages 469–478, 2009. 2

[4] A. Artikis, M. Sergot, and G. Paliouras. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, pages 3–8. ACM, 2010. 2

[5] G. Brewka. Logic programming with ordered Function. In *Proceedings of the National Conference on Artificial Intelligence*, pages 100–105. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002. 3

[6] J. Crowley and Y. Demazeau. Principles and techniques for sensor data fusion. *Signal processing*, 32(1-2):5–27, 1993. 2, 4

[7] T. Eiter and A. Polleres. Towards automated integration of guess and check programs in answer set programming: a meta-interpreter and applications. *Theory and Practice of Logic Programming*, 6(1-2):23–60, 2006. 3

[8] M. Gebser, R. Kaminiski, B. Kaufmann, M. Ostrowsky, T. Schaub, and S. Thiele. Using gringo, clingo and iclingo. September 2008. 2, 5

[9] A. Hakeem and M. Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, 171(8-9):586–605, 2007. 2

[10] R. Howarth. Interpreting a dynamic and uncertain world: task-based control. *Artificial Intelligence*, pages 5–85, 1998. 1

[11] Y.-P. Huang, C.-L. Chiou, and F. Sandnes. An intelligent strategy for the automatic detection of highlights in tennis video recordings. *Expert Systems with Applications*, 36:9907–9918, 2009. 1

[12] X. Jiang, D. Neill, and G. Cooper. A bayesian network model for spatial event surveillance. *International Journal of Approximate Reasoning*, 2009. 1

[13] K.-E. Ko and K.-B. Sim. Development of context aware system based on bayesian network driven context reasoning method and ontology context modeling. *International Conference on Control, Automation and Systems*, pages 2309–2313, October 2008. 2

[14] M. Maroti, G. Simon, A. Ledecz, and J. Sztipanovits. Shooter Localization in Urbain Terrain. *Computer*, 37(8):60–61, August 2004. 1

[15] C. Matheus, K. Baclawski, M. Kokar, and J. Letkowski. Using swrl and owl to capture domain knowledge for a situation awareness application applied to a supply logistics scenario. In A. Adi, S. Stoutenburg, and S. Tabet, editors, *Rules and Rule Markup Languages for the Semantic Web*, volume 3791 of *Lecture Notes in Computer Science*, pages 130–144. Springer Berlin / Heidelberg, 2005.

[16] N. Pham, W. Huang, and S. Ong. Probability hypothesis density approach for multi-camera multi-object tracking. In *Proceedings of the 8th Asian conference on Computer vision-Volume Part I*, pages 875–884. Springer-Verlag, 2007. 2

[17] C. Piciarelli, C. Micheloni, and G. Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11), November 2008. 1

[18] D. Sadlier and N. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transaction on Circuits and Systems for Video Technology*, 15(10):1225–1233, October 2005. 1

[19] V. Shet, D. Harwood, and L. Davis. Vidmap: video monitoring of activity with prolog. In *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, pages 224–229. IEEE, 2005. 2

[20] Baral, C. and Gelfond, G. and Son, T.C. and Pontelli, E., "Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge," *Proceedings of the 18th international joint conference on Artificial intelligence, Toronto, Canada, pp. 259–266, 2010* 3

[21] V. Shet, D. Harwood, and L. Davis. Multivalued default logic for identity maintenance in visual surveillance. *Computer Vision–ECCV 2006*, pages 119–132, 2006. 2

[22] L. Snidaro, M. Belluz, and G. Foresti. Representing and recognizing complex events in surveillance applications. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 493 –498, 2007.

[23] J. Snoek, J. Hoey, L. Stewart, R. Zemel, and A. Mihailidis. Automated detection of unusual events on stairs. *Image and Vision Computing*, 27(1-2):153–166, 2009. 1

[24] T. Strang. A context modeling survey. *In Workshop on Advanced Context Modelling, Reasoning and Management associated with the Sixth International Conference on Ubiquitous Computing*, 2004. 2

[25] S. Tran and L. Davis. Event modeling and recognition using markov logic networks. *Computer Vision–ECCV 2008*, pages 610–623, 2008. 2

[26] B. Truong, Y.-K. Lee, and S.-Y. Lee. Modeling and reasoning about uncertainty in context-aware systems. *Proceedings of the 2005 IEEE International Conference on e-Business Engineering*, 2005. 2

[27] G. Wang, J. Jiang, and M. Shi. A context model for collaborative environment. *Proceedings of the 10th International Converence on Computer Supported Cooperative Work in Design*, 2006. 2

[28] X. Wang, D. Zhang, T. Gu, and H. Pung. Ontology based context modeling and reasoning using owl. *In Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications*, pages 18–22, March 2004. 2

[29] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. *Complex Event Processing Over Uncertain Data*, pages 253–264, 2008. 1

[30] X. Ying and X. Fu-yuan. Research on context modeling based on ontology. *International Conference on Computational Intelligence for Modelling Control and Automation,and International Conference on Intelligent Agents,Web Technologies and Internet Commerce*, 2006. 2