Multimed Tools Appl (2011) 55:443–481
DOI 10.1007/s11042-010-0560-9

A comprehensive study of visual event computing

WeiQi Yan · Declan F. Kieran · Setareh Rafatirad · Ramesh Jain

Published online: 6 July 2010
© Springer Science+Business Media, LLC 2010

Abstract This paper contains a survey on aspects of visual event computing. We start by presenting events and their classifications, and continue with discussing the problem of capturing events in terms of photographs, videos, etc, as well as the methodologies for event storing and retrieving. Later, we review an extensive set of papers taken from well-known conferences and journals in multiple disciplines. We analyze events, and summarize the procedure of visual event actions. We introduce each component of a visual event computing system, and its computational aspects, we discuss the progress of each component and review its overall status. Finally, we suggest future research trends in event computing and hope to introduce a comprehensive profile of visual event computing to readers.

Keywords Visual events · Search · Retrieval · Mining · Reasoning

1 Introduction

An event in the general sense is defined as something that happens at a given place and time (wordnet.princeton.edu/perl/webwn). The basic characteristics of an event to be considered in the sense of human centric computing systems are its ID, time, location and description. Events that take place at different times or positions are considered to be different events. People experience events associated with media either as a combination, e.g. video and audio, or individually. With the

This work was completed when the first author was a research scholar in UC Irvine.

W. Yan (✉) · D. F. Kieran
Institute of ECIT, Queen's University Belfast,
Belfast, UK
e-mail: w.yan@qub.ac.uk, dcsyanwq@gmail.com

S. Rafatirad · R. Jain
Department of Computer Science, University of California,
Irvine, CA 92697, USA

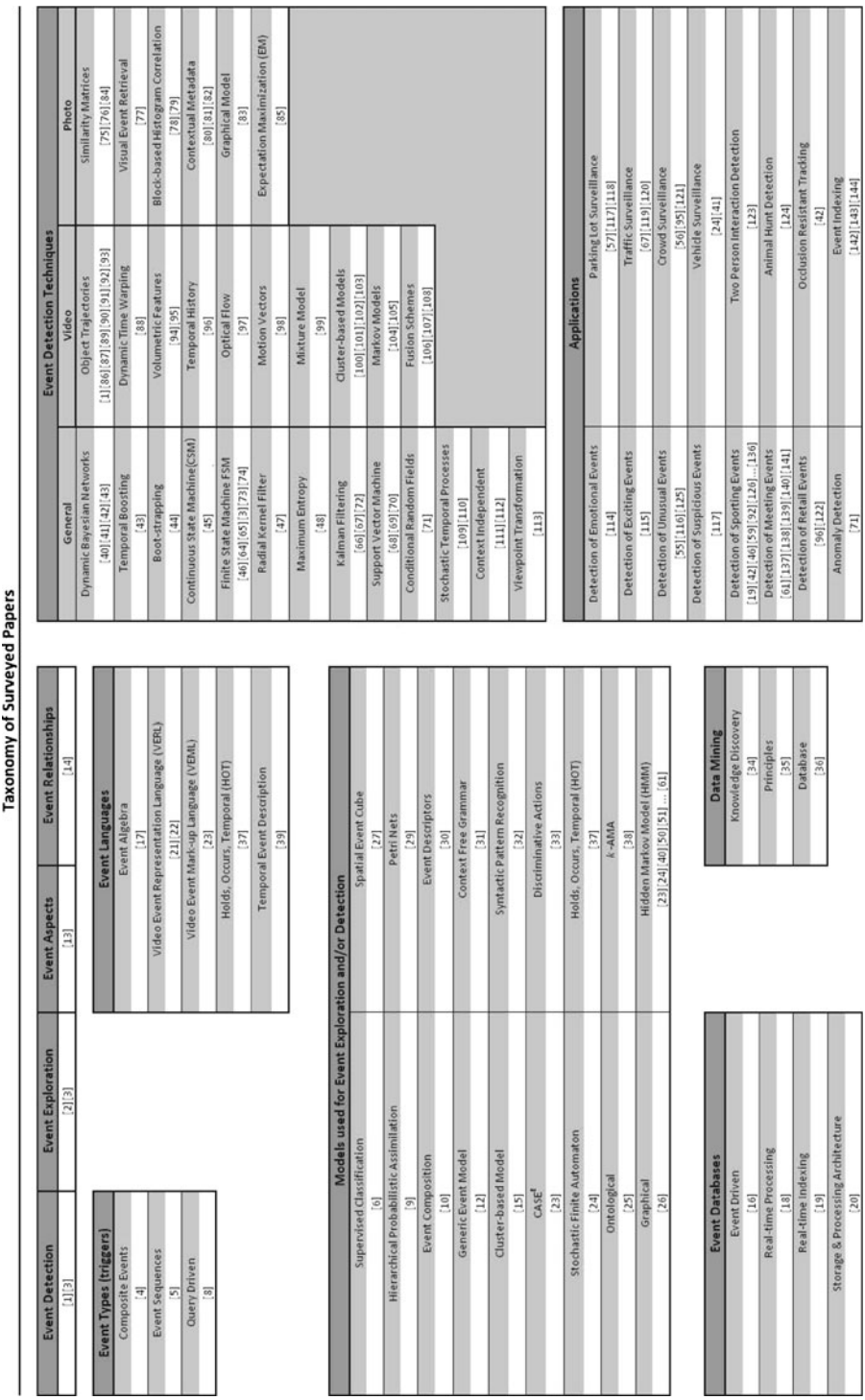


Fig. 1 Quick reference to papers surveyed

advent of new types of media, and the increasing diversity of media for capturing events, event-computing has emerged as a new research area. Previously, most work focused on event detection [44], however, recent work has started to address other functionalities such as storage, reasoning, interaction and exploration [9]. Consequently, new disciplines, such as data mining, have had to be embraced. The definition of an event in terms of visual computing is something that still has to be formally defined, throughout this paper, we aim to discuss the definition of an event in the hope of giving the reader a clear sense of what we see an event as, even though this concept is still under discussion within visual computing.

In this paper, we introduce visual event computing techniques and their computational aspects. In Section 2, the paper first describes what an event is and what characteristics are inherent in an event. This analysis describes how an event can be given a type classification. The composition of an event within a visual event computing system, including the different computational aspects and the relationships between events and their exploration. We then conclude this section by reviewing the current modeling techniques used in event detection. In Section 3, we classify events according to various data types and acquisition methods. Current applications of the event paradigm are introduced in Section 4 and we summarize our survey in Section 5. This paper is aimed at those readers who expect to have a basic knowledge of events and how events are employed in visual computing. In this paper, we limit ourselves to visual computing, however, the extension to general events is straight-forward.

An event to be captured for the purpose of further analysis includes detection, storage, reasoning, mining, exploration and actions. Therefore, event computing systems must encapsulate each step. In detection, events are detected and analyzed semantically. Event detection is a procedure to match previously well defined patterns that have a high occurrence with new incoming patterns. Event detectors sense changes in the attributes of objects which participate in an event, ultimately causing a change in event status. The detectors analyze the event, it changes in status, and thus, a semantic evaluation can be provided. Following analysis, the detected events are stored in a database with their metadata, thereby allowing users to explore and retrieve particular events. Thus mining and reasoning of events can be facilitated by a database. As soon as events are changed, or presented in a new form, they are stored in a database as new events [66].

For quick reference to the work surveyed in this paper, we provide a taxonomy of the papers that were considered in Fig. 1. While an effort to be as specific as possible in the classification of papers has been made, the reader should be aware that there is some overlap where topics may be complementary.

2 Visual event computing

2.1 Events

What is an event? Definitions of an event from various research fields are very diverse and tend to reflect the content of the assigned media. In text event detection and track, an event is something that happened somewhere at a certain time, whereas

in pattern recognition, an event is defined as a pattern that can be matched with a certain class of pattern types. Meanwhile, from a signal processing viewpoint, an event is triggered by a status change in the signal. Thus, a uniform definition is required for all media.

An event is symbolic abstraction for the semantic segmentation of happenings in a specific spatio-temporal volume of the real world. Happenings include the presence of a discernable entity or entities present throughout the temporal space of a piece of media (for the case of visual computing); however, it does not mean that all entities have to be included in a specific event. State changes in objects, or their movement in the real world, can trigger meaningful events [76, 84, 119]. To find the correct relationships between events and their related data, the relationships between sub-events should be realized. This step requires the object attributes involved with the related events and sub-events. For a better understanding, let's look back at theories about the ultimate entities which compose the real world [91]. The real world is a spatio-temporal framework that requires entities to both its dimensions: time and space. All definitions about the constituents of the world refer to their spatial, temporal or spatio-temporal aspects. From this viewpoint, events, objects, their properties, and the existing relationships between them, are the main subjects in terms of the spatio-temporal aspect of the real world.

An event is understood as a fundamental semantic concept in multimedia systems which are becoming ubiquitous. There are strong and deep conceptual, engineering, computational, and human centered design reasons to consider events as a primary structure for organizing and accessing dynamic multimedia systems. There is an intimate relationship between events and experiences in experiential computing. Events play a central role in that they are related to multiple information sources such that changes in multimedia lead to the triggering of events. Event detection requires examining objects, actions, and their inter-relationships automatically [109]. We consider events as abstract entities of the actual world's composing elements in terms of their spatial aspect. Another attribute of events is that they are the source of causal relationships; they cause other events or objects to be created, continued, degraded or terminated. They consist of time intervals, with a beginning and end point. For higher level events, there might be some blank time intervals because nothing has happened related to an event. Events should have occurrences, patterns and categories. The term *event* is very wide in its own sense; hence, it has different classifications according to different categories. In the next section, we define events as either telic or atelic, based on their temporal properties, and as either atomic or composite, based on their composition.

2.1.1 Event types

– *Telic and Atelic*
  Telic events have patterns and types which make them purposeful, while atelic events do not have such attributes. The time interval in which a telic event occurs has end points, whilst atelic events do not. To illustrate the difference between telic and atelic events, consider the following linguistic example:

  1. Tom built a house in a month (Telic);
  2. Tom built houses for a month (Atelic);

The first sentence shows that Tom has finished building a house in a month, which means that the end point for this specific event is defined. However, the second sentence is not clear about the end point, as we are not able to recognize the starting and finishing point of each house. Irrespective of whether an event is telic or atelic, it should have the attributes of a temporal sequence, and include the basic entities of an event.

– *Atomic and composite*
An event is thought as a physical reality [10]. Modeling events need knowledge about both atomic and composite events. An atomic event is an elementary one which cannot be divided into any other events and is the simplest type of event inferred from the observables in the visual data. It is the event in which exactly one object having one or more attributes is involved in exactly one activity.
To organize an event-based system, we have to consider event types, properties and the existing relationships between atomic events, and start thinking about what makes certain events related to each other so that they can be categorized based on their types and classified for easy event retrieval.
Composite events are defined by composition of two or more atomic events [76]. A composite event indicates the part-whole hierarchical relationships in events. If composite events contain simultaneously multiple single events, the events are called multithreaded. Gehani et al. [35] composed composite events from primitive or atomic ones in 1992. Atomic events are basic ones optionally qualified by a mask, which is used to hide or mask the occurrence of an event. Composite events are specified as event expressions, which are formed using event operators. The event actions in a database include "insert", "delete" and "update". An event occurrence is a 2-tuple of the form *<atomic event*, *event identifier>*, in which event identifiers are used to define a total ordering.
An event history is a finite set of event occurrences in which two event occurrences have not the same event identifier. Obviously, composite events are created by considering structural and causal combinations that are meaningful in a given context for specific people.
A multi-object event is a composite event that involves multiple objects. An atomic event is regarded as a consistent motion state of an object, and are often inferred directly from motion trajectories [46]. A multi-object event is then learned based on a coarse-to-fine strategy. An event class is detected by a bottom-up/top-down search algorithm, where some distinctive local event properties are propagated to infer a likely global event configuration. In an attempt to make the concept of an event useable in the visual computing paradigm, we must look closer at its aspects [122].

2.1.2 Event aspects

An event, defined in human centric computing systems, has six aspects; *temporal*, *causal*, *spatial*, *experiential*, *informational* and *structural*. The contents of these aspects are summarized in Table 1.

The temporal aspect contains the *Physical time*, indicating the time stamp (event-starting time and event-time-duration); *Logical time*, the temporal domain concept; and *Relative time*, which is the temporal relationship of the specific event to other events.

Table 1 The components of each entity of event aspects

Temporal	Causal	Spatial	Informational	Experiential	Structural
Physical time	Initiation	Relative location	Context	Documenting	Sub-events
Logical time	Perpetuation	Reinforcement	Tags	–	Composite events
Relative time	Facilitation	Destruction	Comments	–	Atomic events
–	Hindrance	Splitting	Logical location	–	–
–	Termination	Merger	Physical location	–	–
–	–	GPS position	–	–	–
–	–	Geographic and	–	–	–
–	–	frame region	–	–	–

The causal aspect refers to a chain of events [123] which cause other events to be produced, live longer, be degraded or be terminated. This causal relationship also applies between an event as the cause and the object as the effect.

The spatial aspect indicates the relative location, the spatial relationships to other events, the logical location, the spatial domain concept, physical location, GPS position, geographic and frame region.

The informational aspect of an event contains the information about the media content related to the captured event, which will be used in its context information. It also includes the tags and comments which users add to the related event.

The experiential aspect includes all the media files stored in a separate database with paths related to specific events. These play the role of objects occupying events.

The structural aspect refers to the structure of composite events containing sub-events. These sub-events may contain other composite events or atomic events. The structural aspect of an event can be displayed with a tree structure.

2.1.3 Event relationships

There are four kinds of relationship between events: *referential*, *structural*, *causal*, and *similar*. Hopkins et al. [48] conducted research on determining the causes of events and strove to find an effective algorithm that could determine whether one event causes another. Brute-force and intervention-proving approaches were used to prune the search space and project a casual world onto a reduced set of variables.

2.2 Content of visual event computing

In this section, we review event modelling, detection, storage, exploration, mining, reasoning and operations, etc. Event modelling is essentially a discovery problem and involves the use of pattern recognition techniques, such as cluster analysis if prior information is unknown [27].

As explained earlier, events can be related to multiple information sources with changes in those multimedia leading to the triggering of events. Using heterogeneous data types in an information system leads one to think about an event-centric model [112]. Many media-centric applications have been developed which have used events based on their specific requirements, however, no generic event model has been implemented to capture events from different applications. A generic event-model [122] would offer the opportunities for reusable components and techniques

for event visualization, exploration and event query languages. There are some new experiential applications such as chronicles, life logs, and event-centric media managers, but so far, none of these applications shares a common event model. They use specific event models based on their application requirements.

A common approach to representing events helps to reduce the different specialized event models, developed each time for a new event-based application, to one reusable model which can be used in different applications, irrespective of the media. Such a model can incorporate other event-based applications because it is generic. A common event management structure can provide reusable implementation platforms for lots of applications. It should also be extensible and adaptable in order to promote the applicability of the event model. Furthermore, it should be capable of integrating events from heterogeneous applications. Extensibility and adaptability are general concepts, so, for a better understanding, the following explanations are provided.

- *Extensibility* To perform retrieval easily, event types should be assigned to events in order to reduce the exploration time. However, there are various event types that might not be considered in the first place while designing an event-based system, or some event types may be created in future. Therefore, the best idea would be to design a system that lets users add event types, properties, and associations.
- *Adaptability* As events have different description aspects, it is desirable that the system be adaptable enough to let users choose the representation for those descriptions compatible with the applications needs.

The proposed prototype consists of three components: (1) event monitoring, (2) tagging, (3) querying/browse/search/retrieval, in order to test the fundamental event-base information storage framework and the relationship network.

2.2.1 Event operations

Event operations usually refer to two kinds of operations: unary and binary. The former includes *projection*, *selection*, and *renaming*, whilst the latter consists of *union*, *concatenation*, *conditional sequence*, *iteration* and *aggregates*. A prototype system, called Cayuga [28], has been implemented using the event operations of algebra theory. It adds built-in support for parameterization, aggregates and selection over infinite domains, and support for arbitrary streams of events and events with non-trivial duration.

2.2.2 Event storage

Databases are employed to save events. Sunrise [14] is an industrial-strength database system for real-time event processing and aggregation for telecommunication applications that has been developed in Bell Labs since 1998. It is a main-memory database platform that supports scalability and parallel processing with the service authoring environment.

The instantly indexed multimedia database system [86], as the name suggests, performs real-time indexing of real world events as they take place, called Lucentvision, it has a rich set of indices derived from disparate sources and allows domain-specific

retrieval and visualization of multimedia data. Lucentvision exemplifies an emerging paradigm of instantly indexed multimedia databases that convert's real world events in real-time into a form that enables a new multimedia experience for remote users: 1) immersion in a virtual environment where a viewer can choose to view any part of the event from any desired viewpoint and at any desired speed; 2) the ability to visualize statistics and implicit information hidden in media data; 3) the ability to search, retrieve, compare and analyze content including video sequences, virtual replays and a variety of new visualizations; 4) the ability to access this information in real-time over diverse networks.

Based on an event conceptual model, Pack et al. [82] identify a set of design requirements that guide the development of a storage and processing system architecture. The system allows for multiple methods of event detection (manual detection, web crawling, video and audio processing) that can be used to create an event summary, thereby facilitating an easy search method for heterogeneous media. The work identified the criteria of event extensibility, event persistence, search and update efficiency, and consistency. The advantages of this system include flexibility and explicitness.

Supported by a Video Event Representation Language (VERL), events can be represented by MPEG-7 semantic description schemes (DSs) as described in [115]. MPEG-7 DSs are designed primarily to describe higher-level audio-visual (AV) features such as regions, segments, objects, events and other immutable metadata related to creation and production, usage, etc. The DSs produce more complex descriptions by integrating together multiple descriptors and DSs, and by declaring relationships among the description components. In MPEG-7, the DSs are categorized as pertaining to the multimedia, audio, or visual domain. Typically, the multimedia DSs describe content consisting of a combination of audio, visual, and possibly textual data, whereas, the audio or visual DSs refer specifically to features unique to the audio or visual domain, respectively.

2.2.3 Modelling techniques to facilitate event presentation

Franois et al. [33] introduce the VERL and the Video Event Mark-up Language (VEML). VERL is used to represent video events and works with VEML as a companion annotation framework [33]. VEML [41] is a language for recording the observation of concept instances defined in VERL's video event and object ontology. VEML consists of a set of structures, compatible with the VERL definition, that allows links to physical evidence. VERL is designed to encode six items (ontology, data streams, context, objects, events and others) for events that have been automatically extracted, or interactively annotated, in a set of streaming data. The functions of VERL include: *process*, *primitive*, *single-thread*, *multiple-thread*, *subtype*, *rule*, and *sequence*. The language provides: *repeat-until*, *while-do*, *conditions*, etc. There are six possible basic relationships that can exist between two events: *before*, *meets*, *overlaps*, *begins*, *contains* and *ends* [76]. Distinguishing features of VEML are the underlying set of high-level data structure encoding and the relationships between the event ontology, scene-centric and stream-centric representations.

An ontology of events requires a means of describing the structure and function of events. The structure indicates how an event is composed of lower-level states and sub-events, the function introduces the roles an event plays in its environment

and how it in turn participates in larger-scale events. Nevatia et al. [76] represent video events using VERL to annotate instances of the events described in VERL. This paper provides a summary of VERL and VEML, as well as the considerations associated with the specific design choices. They also advocate [76] use of hierarchical decomposition and single or multiple threads to naturally represent complex spatio-temporal events, common in the physical world, by a composition of simple events. The events are abstracted into three hierarchies: *primitive events*, *single-thread composite events*, and *multiple-thread composite events*. This leads to a language, the Event Recognition Language (ERL), which allows users to conveniently define events of interest without interacting with the low-level processing in the program. The data types in this language include object, location, interval, and numerical value. Hongeng et al. [47] point out that a single-thread action is represented by a stochastic finite automation of event states, which are recognized from the characteristics of the trajectory and shape of moving blobs associated with an actor using Bayesian methods. Scenario events are modeled from shape and trajectory features using a hierarchical activity representation, where events are organized into several layers of abstraction, providing flexibility and modularity in the modeling scheme. Multi-agent events are recognized by propagating the constraints and the likelihood of event threads in the event graph. Events in the scenario library are modeled using a hierarchical event representation, in which a hierarchy of entities is defined to bridge the gap between a high-level event description and the pixel level information. Several layers of more abstract mobile object properties and scenarios are constructed explicitly by users to describe a more complex and abstract activity shown at the highest layers. The links between a mobile object property at a higher layer, to a set of properties at the lower layers, represents the relationship between them. Scenarios are defined from a set of properties or sub-scenarios, and the structure of a scenario is hierarchical. Event representation of the scenario level maps closely correspond to how humans would describe events—little expertise is expected from users.

Ontological semantics aims at building resources which would be maximally applicable for reproducing the results of human language processing ability. Malaia [67] proposed using an ontological description of the semantics of lexical entries to describe a real-world event. The typical event taxonomy deals with four types: *state*, *process*, *accomplishment* and *achievement*. Accomplishments and achievements are more complex and share several important traits: telic complete (incomplete) or complete.

In event interactions [87], highly intuitive graphical operations are used to perform event-level manipulations such as merging, altering and creating new events. Event interactions allow reasoning about the semantically hierarchical nature of events. The creation of capabilities is required for performing drill-down and roll-up operations on event hierarchies and visualizing their spatial and temporal characteristics. A collection of specialized interfaces allows users to visualize and interact with various semantically relevant event characteristics.

CASE^E [41] bridges the gap between low- and high-level events. Based on the CASE^E representation of natural language, an event is regarded as a collection of actions performed by one or more agents, and an event detection involves matching a sub-tree pattern. The detected events are represented hierarchically in terms of sub-events, case-list and temporal logic based on interval algebra.

A conceptual representation for the complex spatial arrangement of image features in large multimedia datasets is introduced in [114]. Spatial Event Cubes (SEC) are a scalable approach to mining spatial events in large image datasets based on spatial occurrence of perceptually classified image features. It not only visualizes the dominant spatial arrangements of feature classes but also discovers non-obvious configurations.

Stemming from natural language processing, a representation of activities as bags of event n-grams is introduced in [42], where the global structural information of activities using their local event statistics are analyzed. Based on these discovered sub-classes, a definition of anomalous activities is given and a way to detect the activities is provided. Making use of this representation, the work shows how activity subclasses can be discovered by exploiting the notion of maximal cliques in an edge-weighted graph. Finally, an incremental information-theoretic method of a new activity-instance detection and classification, without re-analyzing the entire activity data-set, is proposed.

Petri Nets [37] represent events by a transition, where the type of transition depends on whether the event is primitive or composite. The advantages of using Petri nets to represent events are that they can be used for both deterministic and stochastic inference of event occurrences. Petri nets have a nice graphical representation with well-defined semantics that uses only a few types of elements. The Petri can be used to represent sequentiality, concurrency and synchronization of events; they can also represent events in a top-down fashion at various levels of abstraction.

Events are represented [80] by using event descriptors, each of which has a physical quantity expression that reflects an interpretation of an event and composition rules. The event representation is a middle language between sensor reading values and natural language phrase descriptions. Observable events are represented using physical quantities of an object state such as position, velocity and temperature. This approach brings an ontological structure to a set of event descriptors.

In [120], events in video sequences are presented using reversible context-free grammars. By using the classification entropy as a heuristic cost function, the grammars are iteratively learnt using a search method. Context-free grammars, with their flexible representation, provide more expressive power with a straightforward design. A search-based iterative algorithm for learning the grammar structure and parameters for each class of motion is employed in a semi-supervised learning strategy.

In syntactic pattern recognition [50], the given data is represented by a string of discrete (terminal) symbols from an alphabet (a finite set of terminal symbols). For event recognition, the terminal symbols correspond to what we call primitive events extracted from the video. Abnormal events are detected when the input does not follow the grammar syntax, or the attributes do not satisfy the constraints in the attribute grammar to some degree.

Discriminative actions can be used to describe the fundamental units in distinguishing between events. Actions are captured first, these actions are modelled, and their usefulness in discriminating between events is estimated as a score. The score highlights the important parts (or actions) of the event from the recognition aspect [2].

2.2.4 Event mining and reasoning

Data mining [43, 56] is the principle of sorting through large amounts of data and picking out relevant information. It has been described as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" and "the science of extracting useful information from large data sets or databases" [34]. Event mining delivers a whole variety of information by searching for patterns in data.

Event mining was firstly addressed within the context of a database. Tesic et al. [114] mine spatial events to discover interesting spatial patterns in an extended image database. Their SEC data structure supports the extension of the general association rule approach to multimedia databases so as to identify frequently occurring item sets.

In [101], a propositional language, called HOT, is proposed using two sets of symbols for temporal reasoning, whose inference engine is based on qualitative and quantitative temporal constraints, and is defined over holds, occurs and temporal propositions. The language has a tractable core which allows others to make weak inferences. An alternative propositional semantics to HOT, that decouples propositional constraints from temporal constraints, is the Conditional Temporal Network. Such a network is consistent i.f.f. there exists a minimal model of the propositional constraints so that the set of temporal constraint propagation techniques is applicable to reasoning about events.

Fern et al. [5] reason about events using *k*-AMA, a sublanguage of event logic, and develop a specific-to-general algorithm for learning event definitions in *k*-AMA. The events are recognized from video input using temporal, relational and force-dynamic representations. An event-recognition component determines which events from a library of event definitions occurred in the model, and recognizes events in the video. Lower and upper bounds algorithms of the subsumption and generalization problems for two expressively powerful subsets of this logic are proposed, and a positive-examples-only specific-to-general learning method based on the resulting algorithms is used [30].

2.3 Techniques applied in event visual computing

Currently, events are detected by Dynamic Bayesian Network (DBN) [21–23, 38], TemporalBoosting [38], Bootstrapping [1], Continuous State Machine (CSM) [53] and Finite State Machines (FSM) [18], Kalman Filtering, Radial Reach Filter (RRF) [100], Maximum Entropy [107], etc. These algorithms are mostly taken from statistical pattern recognition, machine learning and artificial intelligence, etc.

– *Event detection using Dynamic Bayesian Network*
  A Bayesian network is a graphical model for representing conditional independencies between a set of random variables [36]. A Bayesian approach starts with a priori knowledge about the model structure and model parameters. The initial knowledge is updated using the data to obtain a posterior probability distribution over both models and parameters that usually peaks around the likelihood maxima. The expectation-maximization (EM) algorithm is used to estimate the likelihood maxima [105] with hidden variables. A DBN is used to

represent sequences of variables. These are often a time-series or a sequence of symbols. The hidden Markov model (HMM) can be considered as the simplest type of DBN. DBNs generalize two well-known signal modelling tools: Kalman filters for continuous state linear dynamic systems, and HMMs for classification of discrete state sequences [105].

HMMs [92] have been widely used in visual event detection [3, 8, 13, 19, 21, 41, 47, 74, 96, 128, 141]. The main reason for this is that events can be regarded as continuous and having temporal coherence, which can be well modelled by HMMs.

A HMM model is usually formulated as a triple $< A, B, \pi >$, where $A = (a_{ij})_{n\times n}$ is the state transition matrix, $a_{ij} = \sum \mathbf{1}_{(i\to j)}/F$ is the probability of the j-th state given the i-th state, $\mathbf{1}_{(i\to j)}(\cdot) = 1$ solely inference i from j. $B = (b_i)_{n\times 1}$ is the matrix of overall observation symbol probability, $b_i = \sum \delta_{si}/F$ is distribution of the i-th state, $\delta_{si}(\cdot)$ is the Kronecker function. π is the primal state sequence, it is generated from local states in the detecting procedure.

A HMM is a statistical model in which the system being modelled is assumed to be a Markov process with unknown parameters, the challenge being to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis. In a HMM, the state is not directly visible, but variables are influenced by the state. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by an HMM gives information about the sequence of states. In event detection, a HMM is regarded as a generative model within the maximum-likelihood framework. Each event class is described by several state models. These states are used to represent different sub-events of different event types. The HMM parameters are trained by the Baum-Welch algorithm, whereas the Viterbi algorithm is used for classification [105].

A semi-supervised approach [137] for event recognition is provided in situations where there is not enough labelled training data, and the high dimensionality of the observation space requires a large amount of labelled data to capture the event characteristics. The framework is general and can be easily applied to many cases in which collecting labelled data is difficult, but collecting a large amount of unlabeled data is easy. The corresponding sequence of events is obtained by applying the Viterbi decoding algorithm.

However, whilst HMMs provide a good method of modelling temporal sequences, they suffer from overfitting when faced with a large number of parameters, long and complex temporal sequences, and relatively small amounts of training data [3]. HMMs also have difficulty modelling long term temporal relationships in data. This is due to the state transition distribution which obeys the Markov assumption where the current state only depends on the previous state. To overcome this drawback, layered HMMs were proposed in [13, 97]. In [13], the first layer, namely the feature HMM, is used to produce a posterior probability for each of the midlevel clusters at each time t in the sequence. This layer is built by using unsupervised clustering and segmentation of the training data. The second layer is trained using the output of the first layer. This is supervised training using higher level events. So the higher level event HMM produces a probability of a higher level event at time t. In [97], individual action features are extracted as vectors, which are subsequently fed forward

to a continuous Gaussian mixture HMM that provides a segmentation and recognition of events via the Viterbi algorithm.

A Bayesian network is one of the main tools in video event detection. Hongeng et al. [47] recognize events by computing the probabilities of simple events at each frame based on Bayesian analysis. Simple events are represented by a Bayesian network and are inferred from the properties of moving objects. Complex events are defined as a temporal sequence of sub-events, and are represented by a finite-state automation. The recognition process consists of decoding the pattern of sub-events and segmenting them into their corresponding state. Multiple-thread events are recognized by combining the probabilities of complex event threads whose temporal segmentations satisfy the logical and time constraints.

Multiple agents have been applied to detect and represent events embedded in a long video. Hakeem et al. [41] first classify the multiple agent based event detection algorithms into three categories: 1) manual encoding or constraints (grammar, rules); 2) modelling single person activities, or requiring prior knowledge and data variation (HMM, Bayesian network, etc); 3) graph partitioning of the weight matrix. Most recently, detection of unusual and surprising events has become a promising research direction. Hereafter, unusual and/or surprising events refer to events that seldom occur. In [40], events are detected in an event graph. For training events, they adopt a directed acyclic graph approach for representing temporal relationships, and use an event correlation graph to represent temporal conditional dependencies between sub-events. Based on a video event graph, an event correlation graph, and the weight matrix, events are detected using normalized cuts, which is an unbiased method of partitioning a graph into two segments. Using the learned event models, event detection in video is preceded by estimating the weight matrix for each event model. Normalized cuts are then applied to obtain event clusters. These event representations capture temporal variations of only those sub-events, via applying normal cuts to the training video and sub-event alignment of the segmented events.

In [21], events are observed as a sequence of binarized distance relationships amongst objects. The goal of event detection is to find the occurrence of interesting events in a large corpus of video footages [74]. Events are modelled using semantic spatial primitives that enable generalization well beyond the training data. Semantic primitives are defined as Bayesian networks learned from the training data. A HMM is used to model spatial-temporal relationships of objects participating in an event of interest, with observables consisting of a sequence of semantic primitives derived from the binarized distance relationships. Semantic observables outperform direct continuous observables in terms of generalizing to unseen data with little training data. This enables the detection of rare events in video footage [21].

Another approach to event detection is to exploit different kinds of information by segmenting videos into two groups and to progressively select interesting video segments using a maximum likelihood criteria. Piriou et al. [88] use probabilistic image motion (camera and image motion) models to detect events in tennis videos. The motion models associated to pre-identified classes of meaningful events are learned from a training set of video samples. Three kinds of events are taken into consideration: *rally*, *serve*, and *change of side*.

For events embedded in news videos, Boykin et al. [19] compare an FSM segmentation system with an automatically induced one using HMMs. The Viterbi algorithm is employed to predict the segmentation of a video shot from assigned states such as *start*, *stop*, and *advertisement*.

– *Event detection using Finite / Continuous State Machine*
A FSM [57] contains a finite number of states, and produces outputs on state transitions after receiving inputs. There are two types of FSM: Mealy machines and Moore machines. The machines have the attributes of equivalence, isomorphism and minimization. A FSM is represented by a state transition diagram, a directed graph whose vertices correspond to the states of the machine and whose edges correspond to the state transitions. Each edge is labelled with the input and output associated with the transition. Events are regarded as a state change. The FSM approach has been proven to be robust in modelling temporal transition patterns and has the advantage of not requiring a training process [16]. FSMs [66] can easily represent temporal or logical relationships between simple events and have the power to store extra information about each state, such as how long an object has been in that particular state. Another useful property is the ease with which state transitions can be based on multiple events.
A CSM [53] has a state transition rule in a continuous state space and classifies time-varying patterns from different types of single sources. CSMs support dynamic time warping and robustness against noise. They integrate sequential optimization, such as a Kalman filter, with class discrimination methods based on recurrent neural networks. The state transition rule is derived as the minimization of a time-varying energy function that integrates external inputs and predicted states. Time-varying patterns are embedded in a state space as their corresponding trajectories in the learning phase. Since a basic HMM represents just one class, each class is modeled as an independent state space and a state transition rule. On the other hand, CSMs embed the classes in a single state space as trajectory attractors. When a time-varying pattern is observed, the CSM changes its state along one of the learned trajectories. As a result, the time-varying pattern is classified by the trajectory to which the state is attracted. Kawashima et al. [53] detect and recognize events from multiple sources using CSMs that act as simplified Kalman filters. The interaction enables the system to dynamically focus over the multiple sources, and improves reliability and accuracy of event classification in dynamically changing situations. To design a system that recognizes events robustly in an unconstrained environment, the system integrates information from distributed multiple sensors. The dynamics of individual CSMs and the interaction among them are controlled by a certainty distribution during the recognition phase.

– *Event detection using Filtering Algorithms*
Kalman filtering [121] has been regarded as the optimal solution too many tracking and data prediction tasks. The filter is constructed as a mean squared error minimizer, and related to maximum likelihood statistics. It is optimal in the mean-squared error sense, but it is limited from a practical viewpoint by the quality and accuracy of the embedded model. Kalman filtering [51] uses a moving object's center position and size as tokens $t(k)$at time k. In order to estimate the

token change $\triangle t(k)$, Kalman filtering is applied to the system state $x(k)$, which is defined as a 4-D vector of the positional change per unit time interval of the target object and its change in size. Once the system and measurement models are defined, a recursive Kalman filtering operation can be applied to obtain optimal linear minimum variance estimates of the motion parameters.
The RRF [100] has been used to determine, on a pixel-by-pixel basis, similar and dissimilar areas between a background image and a current image of a scene. It evaluates the local texture at pixel-level resolution while reducing the effects of lighting variations. Satoh et al. [100] use this approach to detect events from real world image sequences using grayscale information, which is much more stably acquired compared with color information. The RRF is firstly calculated with reached points, these are then used to calculate evaluation points, and finally the similarity of the RRF value is obtained.

– *Event detection using Support Vector Machine*
A Support Vector Machine (SVM) [20, 90] defines basic functions that are composed of a subset of the training data, which is selected during training. The advantage of an SVM is that although the training involves nonlinear optimization, the objective function is convex and the solution of the optimization problem is relatively straightforward. SVM training always finds a global minimum, and their simple geometric interpretation provides fertile ground for further investigation. An SVM is characterized by the choice of its kernel. In [99], a generic framework for event detection in field-sports broadcast video using an SVM is outlined. Given a shot, the corresponding feature data is aggregated into a Shot Feature Vector. An event model is inferred from evidence derived in turn from feature detectors, which are chosen such that they are recyclable across multiple sports within the field-sport domain. The five feature detectors are *crowd image detection*, *speech-band audio activity*,*on-screen graphics tracking*, *motion activity measure* and *field line orientation*.
– *Event detection using Conditional Random Fields*
The Sequential Monte Carlo (SMC) method provides a finite dimensional approximation to a posterior probability given past observations [26]. Conditional Random Fields (CRFs) are undirected probabilistic models designed for segmenting and labelling sequence data. Compared with the traditional approaches of SVM and HMMs, CRF based event detection offers several unique advantages. To detect an event, a three-level framework, based on multimodality fusion and mid-level keywords, is adopted. The first level extracts audiovisual features, the mid-level detects semantic keywords, and the high-level infers semantic events from multiple keyword sequences.

3 Nature of visual data

Currently, visual events can be considered from within the context of photographs, video streams, and archived video. Photo events are well studied in recent years since photographs are easily acquired and are ubiquitous. Furthermore, detected events help us in photograph organization, search and retrieval. In live streams, surveillance and broadcasting videos are the subject of much research. In particular, research into event detection in sports video constitutes a large portion of the research effort.

3.1 Live data: video event capturing

A video event can be looked as a kind of semantic unit for expressing a story. The ultimate purpose of most tracking systems is the extraction of symbolic descriptions of scene activity. In [85], each event consists of a light-weight data structure that contains the identifier and parameters of the objects involved, as well as further information such as frame number and time stamp. Object trackers are combined into a distributed infrastructure for visual surveillance applications. The tracker generates application independent events on the basis of generic incidents and target interactions detected in the video stream. These events can then be received and interpreted by application specific clients.

An approach for event detection in live sports, based on the analysis and alignment of web-casting text and broadcast sports video, is presented in [129]. The contributions are: (1) detecting live events using only partial content captured from the web and TV; (2) extracting detailed event semantics and detecting exact event boundaries; (3) creating a personalized summary related to a certain event, player, or team, according to a user's preference. Three modules are provided: live text/video capturing, live text/video analysis, and live text/video alignment.

A patent, filed by Engle and Odutola, concerned a method and system for identifying commercial segments of a video signal [29]. This method mainly involved monitoring a digital bit stream comprising the video signal, detecting a change in a control field of the digital bit stream, and then selectively generating a commercial event notification in response to the detection step. In addition, the method may also involve detecting a change in an informational parameter of the video signal, exclusive of the audio-visual content.

3.2 Archived data

3.2.1 Photo event detection

Photographs associated with an event often exhibit little coherence in terms of both low-level image features and visual similarity [68]. The purpose of photo event detection is to integrate metadata and content-based information for automatic photograph organization. The basic framework is to quantitatively assess structure in the collection at multiple scales and feed this data into several different classifiers. Three event detection algorithms are provided: scale-space analysis of the raw timestamp data, time-based similarity analysis, and time and content-based analysis. For event clustering, confidence scores, combined from the average similarity between photographs within a cluster (average intra-cluster similarity) and the similarity between adjacent clusters (average inter-cluster similarity), are calculated.

Digital photograph collections are automatically organized into event-based clusters in [25]. In this work, an automatic unsupervised algorithm for partitioning a collection of digital photographs based on either temporal or content-based similarity is proposed. A learning vector quantization codebook is employed to discriminate between "event boundary" and "event interior" classes. The codebook vectors for each class are used for nearest-neighbor classification of the novelty features for each photograph in the test set. Dynamic programming and the Bayesian information criterion are introduced to select clustering boundaries.

From a user study and survey, Lim et al. [62] divide events, based on the visual content of photographs, into gatherings, family activities and a place visit. The author's model visual events in order to automatically extract relevant semantic tokens using visual event graphs and a visual vocabulary. The event models are generated from labelled photographs, and semantics extracted from photographs annotated with events.

Loui et al. [63] provide event segmentation based on date/time metadata information, as well as the color content of pictures. Two photographic events are found to be popular: chronicle and subject order. An event clustering algorithm organizes pictures into events and sub-events based on date and time of picture capture, and content similarity between pictures. Loui et al. [64] refined previous work and provided event clustering and screening of low-quality images to deal with problematic photos.

O'Hare et al. organize personal digital photograph collections based on date/time and GPS location [79]. This mediAssist project demonstrates a tool for users to manage a personal archive of digital photos in both a web based and a mobile interface. They utilize the automatically generated contextual metadata for organizing and searching personal photograph collections.

An event can be thought as a series of consecutive photographs that were taken in the same context. Naaman et al. [73] leverage time and location context to resolve identity in their PhotoCompas system. As many users annotate their identities, patterns and events, the system uses these patterns to generate label suggestions for identities that were not annotated. Naaman et al. [72] describe the contextual metadata automatically assembled for a photograph, as well as a browser interface that utilizes the metadata. PhotoCompas adopts time and location information to automatically group photos into hierarchies of location and time-based events. The categories subsume *outdoor/indoor*, *who*, *day/night*, *camera*, *mood*, *captions*, *camera settings*.

An event is considered to be a semantically meaningful human activity, taking place within a selected environment, and containing a number of necessary objects [58]. Events are classified into static images by integrating scene and object categorizations. This is achieved by integrative and holistic recognition through a generative graphical model. Similar to object and scene recognition, event classification is both an intriguing scientific question as well as a highly useful engineering application. Event classification is part of the ongoing effort of providing effective tools to retrieve and search semantically meaningful visual data. Event classification is also particularly useful for the automatic annotation of images.

A hierarchical clustering of photographs, based on a similarity matrix of color histograms, and summarization of photographs, based on a novel contrast context histogram technique, is employed in [59]. In [90], Principal Component Analysis (PCA) is employed to reduce the dimensionality of a histogram based on color descriptors. An SVM is then used to classify the images into various high level categories corresponding to histogram subspaces.

Photographic events are characterized by the coherence of multimodality including time, content and camera settings [69]. An event is taken as a latent semantic concept, and discovered by fitting a generative model using Expectation-Maximization (EM). This approach is general and unsupervised, without any training procedure or predefined threshold. The multimodal metadata used in [69] includes contextual

information about the time and camera parameters, and perceptual image content such as color, texture and face number. The parameters for photo clustering are iteratively estimated using an EM algorithm.

3.2.2 Video event detection

Video encapsulates spatio-temporal features, as well as content and contextual information. Object trajectories and motion are the salient features for video event detection.

– *Event detection using trajectory and action cylinder*
A trajectory is a set of time-indexed locations combined into a single hypothesized entity. It is at a higher level of abstraction than time-ordered locations used by other correspondence algorithms. The use of trajectories for event detection has a relatively long history [44, 45, 49]. In the latter, a statistical model of object trajectories is learned from image sequences. Both simple and complex events are recognized by attaching meaning to prototypes representing instantaneous movements and complete trajectories. Trajectory prediction can be achieved in a similar way by labeling nodes whose prototypes represent complete trajectories—with information acquired automatically in a further learning phase. Partial trajectories can then activate the node representing the most similar complete trajectory.
More recently [94], dynamic time warping was employed to match trajectories using a view invariant similarity measure. It works in an unsupervised manner for motion capturing, action representation and learning. In earlier work [93], the spatio-temporal curvature of a trajectory was represented by a sequence of dynamic instants and intervals. Video is automatically segmented into individual actions, and a view invariant representation for each action is calculated. The proposed model also learns parameters from different actions, and discovers different instances of the same actions performed by different people.
Syeda-Mahmood et al. [110] recognize action events based on a trajectory cylinder obtained from multiple views. The shape formed from successive perspective projections of an object is visualized as a generalized cylinder, called action cylinder. This is a spatio-temporal solid formed by combining successive cross-sections obtained by the intersection of the 3D body with a plane. The action cylinder can be viewed as the perspective projection of the object as it undergoes motion.
In [102], automatic event detection is performed using the trajectories of individual objects. For each bacteria cell, the initial position, area and orientation in space are calculated, and the movements of cells tracked. For swimming bacteria, events such as forward swimming, tumbling, and stopped are detected.
An effective representation of sports tactics, called aggregate trajectory, is constructed of multiple trajectories obtained using a novel analysis of the spatio-temporal interaction among players and the ball in [144]. The interactive relationship between the playing region information and hypothesis testing for the spatio-temporal distribution of trajectories is exploited to analyze tactical patterns in a hierarchical coarse-to-fine framework.
A trajectory that records an object's position from entering to exiting a scene is one of the most useful information types for embedding a moving object's

behaviour [138]. A generic rule induction framework, based on trajectory series analysis, is proposed to learn event rules. The trajectories acquired by a tracking system are mapped into a set of primitive events that represent basic motion patterns of moving objects. A grammar induction algorithm, based on the minimum description length (MDL) principle, is adopted to infer meaningful rules from the primitive event series. Grammar induction from artificial intelligence and natural language processing aims at identifying a set of grammar rules from a set of training sentences. PCA and Euclidean distance are adopted to compute the similarity between two trajectory segments. A spectral clustering algorithm is then used to partition the segments into several motion patterns. This allows trajectory classes, corresponding to different driving lanes, to be separated correctly. A HMM that takes into account the uncertainty of the low level processing is trained on each cluster. These HMMs are used as the detectors of primitive events. For a given trajectory segment, the HMM yields the maximum likelihood.

A standard approach to detect events is to break the model into parts, allowing the parts to move independently, and to measure the joint appearance and geometric matching score of the parts. Allowing parts to move makes the template more robust to the spatial and temporal variability inherent in actions. Examples of this approach are given in [54, 55]. Specifically, integral video and box features of a sequence are extracted. A detected volume over all locations in space and time is then scanned at different spatial and temporal scales and windows. The detector is trained and tested on real videos with the actions *sit-down*, *stand-up*, *close-laptop* and *grab-cup* actions having different camera view, scale variations, and changing speeds at which actions are performed. The efficient matching of event models is via over-segmented spatio-temporal volumes. The models are derived from a single example and are manually constructed. Automatic generation of event models from weakly-labeled observations is a related interesting problem [55]. The model is derived from a single exemplar of the event, however, it can detect events in crowded videos. The key point is in the use of shape matching with over-segmented regions.

– *Event detection using temporal and spatial features*

A visual event is frequently related to a moving object with constraints on its size, colour or shape. A sequence of events is represented by the tracked trajectory of the object of interest. These trajectories are further clustered to form typical trajectory templates. Therefore, the entire modelling process relies critically on the accuracy and consistency of segmentation and tracking, which are often ill-conditioned due to the presence of multiple objects, occlusion and non-linearity of the trajectories. Visual event detection and classification may be performed without explicit object-centred segmentation and tracking [124]. Events are represented and detected first at the pixel level and then at a blob level (grouped pixels) autonomously. A pixel change history is proposed to characterise pixel-wise temporal visual information in order to detect pixel-level events, and is based on the temporal history of each pixel intensity. Crucially, it is combined with an adaptive mixture background model to form a new representation for detecting and classifying pixel-level events. It also provides an important cue for characterising blob-level events which are defined on the basis of grouped pixel-level events. Blob-level events are computed by unsupervised clustering

with automatic model order selection. The EM algorithm is employed to cluster events with MDL used for automatic model order selection. Although no explicit object-centred segmentation and tracking were performed, meaningful event clusters can be consistently formed. The detected blob-level events can be classified into meaningful classes without object-centred tracking.

Recognizing a motion event requires choosing the most likely spatio-temporal model. Black [17] employs optical flow with parameterized spatio-temporal models for representing motion events. Within a Bayesian framework, the phase, rate, spatial position, and scale are taken into account to deal with image variations. The computational mechanism, based on the condensation algorithm, incrementally estimates a distribution over model parameters. The approach automatically detects and recognizes motion events based on image derivatives.

Xu et al. [128] detect video events by representing video motion as the responses of frames to a set of motion filters. Motions between two video frames are represented by an energy redistribution function. The redistribution function is filtered by a set of motion filters, each of which is designed to be most responsive to a type of dominant motion. Such a filter process converts a video into a temporal sequence of filter responses in which distinct temporal patterns, corresponding to high level concepts, are presented. The motion content of a video is used to extract meaningful dynamic events by using probabilistic models. Only low-level motion features are exploited to maximise generality and increase efficiency. The motion activity models, namely the residual motion with a causal Gibbs or Gaussian mixture model, are determined by analysing the distribution of local motion-related measurements. These are derived from a weighted mean of normal flow magnitude.

In [135], events are detected from moving blobs of MPEG video in the compressed domain. Feature vectors from a video clip form a high dimensional curve, simplification of which allows one to browse the video clip at places where events have occurred. The camera motion is computed from motion vectors, and the residual vectors are regarded as moving blobs. Interesting events, such as a new object appearing, objects interacting, or an object changing shape, are detected from these moving blobs. The event detection module builds feature vectors from 2D histograms of stepwise motion vectors and finds discontinuities in the trajectories of the feature vectors. Therefore, dynamic event detection requires four main techniques: identification of camera motion, segmentation of moving blobs, tracking of moving blobs, and analysis of their respective motions for classification of their interaction. The limitations using motion vectors to detect events are: 1) the object speed cannot be fast; 2) a motion vector cannot represent the motion properly when the blobs are too small. For event identification, a vocabulary of dynamic events, based on relative motions and the size changes of tracked moving blobs, is required.

Dynamic events can be considered as being comprised of spatio-temporal atomic units, called actions. Events can be represented as a mixture of actions and the transitions among these actions. In [4], a mixture model learns an optimal combination of various components representing actions. This approach can also be interpreted as a unifying framework for combining appearance and temporal features in events. The composition of the feature content is controlled by the number of mixtures in the model.

Leonard is the first system that goes all the way from video to event classification using recovered force dynamics [104]. Event classification is efficiently performed on this preferred subset of models using prioritized cardinality and temporal circumscription. In earlier work, the maximum-likelihood approach for visual event classification was used [105]. Siskind et al. proposed a technique to classify events by recovering changing support, contact, and attachment relationships between participant objects, using a kinematics simulator driven from the output of 3D tracking. Kinematics describes object motion without considering the masses and forces that bring about the motion. To represent the idea, several movies of simple spatial-motion events were taken, including: *picking-up*, *putting-down*, *pushing* and *pulling boxes*, and *dropping erasers*. An edge detector and line finder were then applied to each of the movie images and an animated output obtained. The event recognition task is partitioned into two independent sub-tasks: a lower-level task which detects object orientation, shape, and size, and an upper-level task, which uses the 2D pose stream produced by the former, to classify an instance of a given event type.

In [65], a double threshold multidimensional segmentation algorithm is proposed to automatically decompose a complex human motion into a sequence of simple linear dynamic models, without prior knowledge of the number of dynamic models. Event classification was performed using cluster analysis with the model parameters as input. The dynamic model parameters form a compact representation of the motion data, which is amenable to cluster analysis for event classification.

In [107], a time interval multimedia event (TIME) framework is presented as a robust approach for semantic event classification in multimodal video documents. The presentation used in TIME extends the Allen temporal interval relationships. For automatic classification of semantic events, three different machine learning techniques are employed: the C4.5 decision tree, maximum entropy and SVM. The framework explicitly handles context and synchronization and yields a robust approach for multimodal integration. Events are presented in patterns. To model such a framework, the relationships between any two time intervals are considered. There are thirteen relationships: *precedes*, *meets*, *overlaps*, *starts*, *duration*, *furnishes*, *equals and inverse*, etc. The events in soccer videos are *goal*, *yellow card*, *red card*, *substitution*.

Another approach to event detection in video, is the use of an event-inference module. Using this approach, Haering et al. [39] propose a three-level video-event detection methodology and apply it to animal-hunt detection in wildlife documentaries. The first level involves color, texture, motion features, shot boundaries and moving objects. The second level uses a neural network to determine the object class of the moving blobs. The third level detects video segments that match user-defined event models. The work aims at detecting animal hunt events in wildlife documentaries, such as a moving animal or stopping animal, and detecting the rapid chase of a fleeing or running animal. Osadchy et al. [81] use an anti-face method to detect events in both the gray scale and feature domain. The algorithm was applied to detect *activity curves* corresponding to sketched symbols in two and three dimensions. Using two basis views, it was possible to successfully detect sketches in views that substantially differ from the training set. Three advantageous features of the technique include: 1) the method

is robust to rotation, scale, and speed of the event; 2) the proposed method is capable of discriminating a given word from very similar words; 3) the method is used for motion feature recognition.

– *Event detection using AV features*
For detecting events in a meeting, the usual approach is to extract a set of standard audio-visual features from three cameras. In a meeting event scenario [137], visual features consist of head vertical centroid position and eccentricity, hand horizontal centroid position, eccentricity, and angle. The motion magnitude for head and hand blobs were also extracted. The average intensity of different images computed by background subtraction are extracted. For audio features from a microphone array, a speech activity measure was computed. Three acoustic features, namely energy, pitch and speaking rate, were then estimated on speech segments.
Detecting semantic events from audio-visual data with spatio-temporal support is a challenging multimedia understanding problem. A Duration Dependent Input and Output Markov Model to detect events based on multiple modalities was proposed by Naphade et al. [75]. It provides a hierarchical mechanism to map media features to output decision sequences through intermediate state sequences. It also supports discrete non-exponential duration models for events. Combining these two features, the Viterbi algorithm is used to infer events. Kristjansson et al. [116] present an extension of the forward-backward algorithm that can be used for inference and learning in event-coupled HMMs. They present results on a simplified multimedia indexing task, where the objective is to detect an event whose onset is loosely coupled in audio and video.
Specific types of events occurring in a classroom or lecture environment, can be detected using a query-driven approach which combines visual and audio cues derived from an image and the textual content of presentation slides [109]. Visual events are detected using the displayed slide, or are captured from a video stream by region hashing. A region of a video frame can be recognized as containing a specific slide if the affine intervals of corresponding region pairs are identical. To do this, affine coordinates of features in a region are computed first. The range in which these coordinates lie is noted in the corresponding affine interval, and the affine interval information consolidated and represented in an index structure called the interval hash tree. Audio events are detected by the slides on the audio track using audio word script (IBM Via Voice). A topical audio event is a set of contiguous time points in an audio track, where there is spoken evidence for the maximal number of textual phrases listed on a slide.
Both internal AV features [125–127], and various types of external information sources can be utilized for event detection in team sports videos. In the case of a soccer video, the event types are *goal*, *save*, *shot-off target*, *penalty-goal*, *corner-kick*, and *free-kick*. Three fusion schemes are proposed: rule-based scheme, aggregation, and Bayesian inferences. The use of multiple sources of information based on intrinsic AV features and external knowledge helps to detect events in the soccer video. Comparisons show that Bayesian inference has the best capabilities to tackle asynchronism among the three schemes.

– *Event detection using stochastic processes*
In [139, 140], events are regarded as stochastic temporal processes, with two events being considered as similar if they could have been generated by the

same stochastic process. A simple statistical distance measure between video sequences captures the similarities in their behavioral content. This measure is nonparametric and can thus handle a wide range of complex dynamic actions. A behavior-based distance measure between sequences can be used for a variety of tasks, including: video indexing, temporal segmentation, and action based video clustering. By presenting events in a nonparametric way, periodic and nonperiodic activities, isolated occurrences, and multiple repetitions can be recognized utilizing a single framework for both structured video and dynamic textures.

– *Event detection using other features*
Amera et al. recognise context-independent events using key-image extraction [6]. Context-independent events refer to events having a fixed meaning. Amera et al. rigorously define a set of context independent events including *enter*, *appear*, *exit*, *disappear*, *move*, *stop*, *occlude*, *remove*, *depositor*, etc [7]. These are automatically detected using feature extraction following segmentation, motion estimation and object tracking. Events are automatically detected by combining trajectory information and spatial features, such as size and location. When specific conditions are met, events related to these conditions are detected.
MediaTE (Media to Everyone) is able to create videos of higher narrative or aesthetic quality with a complete mobile lifecycle [1]. It proposes an at-capture bootstrapping of event information from which all system guidance flows. Bootstrapping focuses on extracting entities from the user input. The event can have global attributes, both physical and discourse related, as well as similar attributes inherited implicitly from actors and objects that it contains. The goal of characterizing and populating an event is to enable the creation of shot suggestions specifically moulded to a user's context, and to obtain a sufficient amount of information about an event from the user at capture time in a natural manner. The bootstrap consists of three aspects: a setting attribute, relevant human actors, and relevant objects.
In [108], robust event recognition is achieved by recovering the viewpoint transformation and time correspondence between a query action and a given action segment in the video. This can be used to efficiently deal with viewpoint changes, execution style changes and occlusions.
Fern et al. recognize events from video input using temporal, relational and force-dynamic representations [5]. The raw input is the video-frame sequence, segmentation and tracking components then transform this input into polygon movies in which objects are marked with polygons. A model-reconstruction component then transforms the polygon movies to a force dynamical model. Finally, an event-recognition module determines which events, from a library of event definitions, occurred in the model. The detected events are: *pick-up*, *put-down*, *stack*, *unstack*, *move*, *assemble* and *disassemble*.

4 Applications

An event is a fundamental semantic concept in multimedia systems, which are fast becoming ubiquitous. There are strong and deep conceptual, engineering,

computational and human-centred design reasons to consider events as a primary structure for organizing and accessing dynamic multimedia systems. Consequently, event-based applications are under development in different fields. In this section, we give an overview of these applications, in particular, we consider event detection in surveillance and sports videos. Surveillance videos mainly contain events that have happened in scenes such as car parks, airports, lobbies, traffic, checkpoints etc. Several special events, mostly related to surveillance, have also been widely studied, such as emotion events [52], exciting events [132], unusual events [141, 143], and suspicious events [31]. Event detection in live broadcast videos from sports, such as baseball, soccer, football etc., has been widely studied in recent years due to the tremendous commercial potential.

4.1 Event detection from surveillance video

Event detection from surveillance and monitoring videos plays a practical role in personal security. Typical vehicle-related events in unmanned airborne vehicle surveillance [47] include: *approach checkpoint*, *stop short before arriving*, *car goes through checkpoint*, *car avoids checkpoint*, *move inside*, and *leaving*. Another example is theft at a phone-booth. The events include: *bringing to object*, *attacking a person*, *using phone*, *taking away the object*, *passing by*, etc. Temporal interval logical relationships are used to compute multiple agents, and multiple threaded events.

Anomalies in individual and interactive event sequences are an important issue in surveillance. In [26], an SMC method is employed to track an event sequence in discrete state space for anomaly detection. A Markov Random Field (MRF) is used to extend SMC for both individual and interactive events. An adaptive temporal differencing method is used to describe pixel changes, and an effective and efficient event representation approach, employing SMC and subspace methods, are combined to implement event tracking in probabilistic manifolds.

Chan et al. [22] studied event recognition in a busy scene consisting of a refuelling airplane being serviced. Events recognised included: *close-to*, *contained-in*, *appear-near*, *disappear-near* and *moving*. For this application, object tracks are often fragmented, therefore, the level of track fragmentation best for event recognition was investigated. The approach was to use DBNs to model events, with observed nodes corresponding to the spatio-temporal semantic relationships between event actors and elements. Interpolation over track gaps, in both space and time, was then performed. The model represents complex events defined by interactions between multiple object tracks. The main contribution in [23] is the combination of track linking with event recognition in a joint formulation that optimizes both simultaneously. The advantage is that events can be recognized despite highly fragmented tracking due to long occlusions, in scenes with many non-involved movers, under different scene viewpoints and/or configurations.

Parking lot events are monitored in [32]. The event recognition module receives input information, such as location, tracking and classification of moving objects, and classifies an event as standard or dangerous on the basis of pre-defined object motion models. The work consisted of three parts: object classification using an adaptive high order neural tree, object tracking based by the mean shift algorithm, and recognition of normal, suspicious and dangerous events. The functionalities of

a car park surveillance system usually include the online classification and detection of abandoned objects [31], and the automatic detection and indexing of video event shots showing the cause of an alarm [96]. The method for video-event shot detection and indexing is obtained by integrating the metadata of the three subsystems, as well as addressing video-object layers represented by blobs.

Events are also detected from traffic surveillance videos [134]. The detected events have three levels: low, medium, and high (traffic jam-low, lane change-medium, and traffic rule violation-high, respectively). The low-level module detects moving objects from captured images; the middle-level module analyzes the relationships between the input image and the road surface in the real world; the high-level module calculates parameters for each passing vehicle. The main feature is that no prior information of the capture conditions is required. Nishida et al. [78] develop a tracking algorithm, based on a spatio-temporal MRF, in order to acquire and visualize events from traffic images with occlusion and clutter problems. The detected events are vehicle counts of traffic direction, velocities, frequent paths and so on. Jung et al. [51] track moving objects using Kalman filtering and occlusion reasoning. The trajectory of the moving object is approximated by polynomial functions and is described by motion trajectory descriptors.

Events can also be detected from crowd scenes [8, 55, 98]. Crowd events are usually difficult situations containing highly-cluttered dynamic backgrounds. Khan et al. [98] present a planar homography constraint to resolve occlusions and to robustly determine locations on the ground plane corresponding to people's feet. The algorithm is able to accurately track people in all views maintaining correct correspondences across views. The algorithm is ideally suited to situations when occlusions between people would seriously hamper tracking, or if there are simply not enough features to distinguish between different people. The major contribution is the detection of ground plane locations of people and the resolution of occlusion using a planar homography constraint. Combining foreground likelihoods from all views into a reference view and using the homography constraint ensures that the blobs representing feet are segmented out. The feet are tracked by clustering them over time into spatially coherent worms. In [8], crowd behaviour is characterized by observing the crowd flow, with unsupervised feature extraction to encode normal crowd behaviour. The unsupervised feature extraction applies spectral clustering to find the optimal number of models required to represent normal motion patterns. Using projections of the eigenvectors in the sub-space spanned by normal crowd scenes, the proposed technique applies spectral clustering to automatically identify the number of distinct motion segments in the sequence. The features in the clustered motion segments are used to train different Multiple Observation HMMs for normal sequences, which compose a bank of models for the simulated training video.

Another application is the detection of events in commercial spaces such as retail stores [71]. The framework is evaluated in a retail environment for detecting trollies entering or leaving the back door of a store and the opening or closing of a cashier's cash drawer. Five different event classes were automatically learned, in terms of their location and temporal order, through unsupervised clustering, with the following events manually labelled: *can taken*, *entering* and *leaving*, *shop keeper*, *browsing* and *paying*. In other work, learned mixture models were utilized to recognize detected blob-level events online [124].

Events involving two-person interactions have also been the subject of research [83]. Two-person interactions are a combination of single-person actions, which are themselves composed of a human body-part gesture. Each gesture is an elementary motion event and is composed of a sequence of instantaneous poses at each frame. The method is based on a hierarchy of action concepts: static pose, dynamic gesture, single-person action, and person to person interactions. Human actions are represented by multiple triplets aligned according to spatial-temporal constraints between actions.

In [89], a three-level video event detection algorithm is applied to animal hunt detection in wildlife documentaries. Low-features include Gabor filters, co-occurrence matrix measures, fractal dimension measures, and color features. A multi-layer perception neural network is trained using the back-propagation algorithm with a total of 9 animals and 5 non-animal labels. A simple color histogram technique decomposes video sequences into shots, which are summarized in terms of object, spatial and temporal descriptors. Hunt events are detected by an inference module that utilizes domain-specific knowledge and operates on the generated shot summaries. The event inference module looks for a prescribed number of shots.

Zhong et al. [142] detect unusual activities by dividing the video into equal length segments and classifying the extracted features into prototypes, from which a prototype segment co-occurrence matrix is computed. A correspondence relationship between prototype and video segments, which satisfies the transitive closure constraints, is sought. The main feature of this algorithm is that it utilizes extremely simple features that are automatically selected from the signal. Zhou et al. [143] detect unusual events via multiple camera mining. The unusual event detection uses two-stage training to bootstrap a probabilistic model for common events. An event not classified as common is considered unusual. Zhang et al. [141] proposed a semi-supervised adaptive HMM framework, in which common event models are initially learned from a large data set, whilst unusual event models are learned by Bayesian adaptation.

The IBM Smart Surveillance System (S3) has an *open and extensible architecture* for video analysis and data-management. Its role in video analysis is to encode the camera streams and send them to the video or streaming database, and also to analyze the camera streams for events and send the resulting metadata in XML format to the metadata database. Its role in data management is in providing a human-interface layer for queries, alerts, events and real-time event statistics. The system consists of middleware for use in surveillance systems, and provides video-based behavioral analysis capabilities. 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.

4.2 Event detection from sports video

– *Event detection from football video*
  Babaguchi et al. [12, 77] propose a combination of methods for event detection in sports video. Firstly, multimodal information is processed by tracking the dependency between media streams based on the concurrency of their related events. This process, called inter-modal collaboration, establishes links between

visual and linguistic streams. Secondly, domain knowledge is exploited to extract specific visual objects. An event is detected by object occurrence analysis in the visual streams. Typical events include: *touch down*, *field goal*, *point after touch down*, *safety*, etc. In [77], four extra events are detected: *players*, *motion*, *referees gesture*, *change of score*, and *keywords from auditory*.

In further work on inter-modal collaboration [11], the temporal correspondence between the visual and closed caption (CC) streams is exploited to improve the reliability and efficiency of video content analysis. The proposed method attempts to seek time spans in which events are likely to take place, through keyword extraction from the CC stream. These are then used to index shots in the visual stream. Detected events include: *touch down* and *field goal*. Miyauchi et al. [70] also adopt inter-modal collaboration to detect semantic events in three stages: closed caption analysis, auditory analysis, and visual analysis. Key words are related to events from the CC stream and feature parameters characterising cheering and shouting from the auditory stream. Multimodal streams consist of visual, auditory and textual information.

In [13], three level events are detected from rugby video using layered HMMs. The first of these is structural events of a shot, e.g., medium shot, medium shot low, angle close up, person in a close up, long shot, miscellaneous. The second are play events, e.g., play, non-play and replay. Lastly, are the action events, e.g., running and passing, maul, line-out, kick, penalty, scrum and try.

– *Event detection from soccer video*

In [117], soccer video events are detected using a three-layer event detection scheme. A probabilistic framework, based on Bayesian inference, is used to reason whether interesting events are presented. A short video segment composed of consecutive frames that contain a special cue comprises an intermediate-level semantic descriptor which is at a semantic level above low-level features and shots. Six semantic units are considered: SMR, close-up, audience, caption, goalmouth and close-up & audience/caption unit. When evidences are observed, they are inserted into the network and the posterior probabilities of events are calculated using model parameters, priors and conditional probabilities. *Shooting* and *red/yellow card* events in soccer are detected based on a Bayesian network. Furthermore, Tang et al. [111] presents a content-adaptive transmission system for streaming reconstructed soccer goal events over networks. The reconstructed event consists of one panoramic image or a sequence of panoramic images. The system constructs a field model by detecting landmarks. Each transmission scheme defines the video content to be transmitted, how many images are to be reconstructed, and where the goal reconstruction event will take place. For each frame of the goal event sequence, the positions of the ball and players are detected, and segmentation is performed on the rectangular region around them. The positions of the extracted segments are localized on the field model and the segments are pasted accordingly.

Tactic patterns can be discovered from goal events in broadcast soccer videos based on the tactic clues extracted from players and ball trajectories. In [144], a multiobject detection and tracking algorithm is employed to obtain player and ball trajectories during a goal event. Goal events are extracted with far-view shots based on the analysis and alignment of web-casting text and broadcast video. An aggregate trajectory is constructed, based on multiple trajectories,

using an analysis of the spatio-temporal interaction between players and ball. The interactive relationship, information from the playing region, and hypothesis testing for trajectory spatio-temporal distribution are exploited to analyze the tactic patterns in a hierarchical coarse-to-fine framework.

A semantic video indexing algorithm based on a FSM and low-level motion indices, extracted from the MPEG-2 compressed bit-stream, is presented in [18]. The proposed algorithm can detect sports events, such as scoring of a goal in a soccer game and other relevant events using fast pan and fast zoom-in.

A wide range of player actions and game events that are derived from a hierarchical entity-relationship model representing the prior knowledge of soccer events, is presented [118]. In this approach, information on the players and ball from multiple monitoring points is combined to derive their positions via triangulation. A list of observed events, interpreted events, and soccer actions are provided. Knowledge using in detecting the various events includes the laws of the game, understanding of how a soccer match progresses, and all the possible events which may happen during the course of a game. In [23], complex events are detected based on the detection of basic actions. Once an event is detected and marked as valid, the algorithm will invoke another set of heuristic rules to determine which specific actions are involved in the event.

One approach to sports video annotation is to integrate text and image streaming [77]. From the text and image data, actors, actions and events of each scene are extracted using linguistic cues and domain knowledge. The linguistics are segmented and used to extract parts where an event has a high probability to have taken place. This is done by utilizing key phrases to get the elements of each story. An action is independent of the type of sport involved and is of a general nature. An event is a result of those actions and is specific to the particular type of sport being played.

– *Event detection from baseball video*

A feature of baseball sports videos is that they usually have a well-defined structure that contains segments of pitching and batting [60]. A baseball event can be defined as the portion of a video clip between two pitches: a play is a concatenation of many events, and a baseball video is composed of a series of plays. The recognized caption is inferred to find possible semantic categories of play in the first step; visual features of the video are utilized to find out the type of play, from one of *non-hitting*, *infield*, and *outfield*, in the second step; and the resulting information is combined to find the exact semantic meaning of the play. It is then semantically classified using an algorithm that integrates caption rule-inference and visual feature analysis.

LucentVision is a multimedia system for live and real world event detection [86]. The system can integrate between 2–8 cameras, incorporates enhanced analysis and visualization, and includes object tracking and virtual replays. LucentVision sends out broadcast grade graphics over the air, generates Virtual Reality Modelling Language environment models, and detects changes in those models throughout an event. The queries in the database include those based on scores, statistics, space, and of a historical nature. LucentVision provides live web updates to the ATP tour official website (http://atptour.com). The system periodically updates the site and offers a selection of LucentVision visualization options, including a map, statistics and a virtual replay. Multiple types of baseball

video event detection, using superimposed caption text detection and recognition, are proposed in [136]. These include *out*, *run*, *walk* and *score*, and also event boundary detection. Events, and the associated game state information, are extracted using a videotext detection and recognition module. The event boundary detection is based on video view recognition. An event occurrence is detected by the caption changes. The pitching view and non-active views surrounding the event are detected to determine the beginning and end points of an event.

In [61], *non-hitting*, *in-field*, and *out-field* events from a baseball game are detected using video motion vectors. These are used as features for a three-layer feed-forward neural network, which is shown to be adept at correct classification. The neural network is trained using the back-propagation algorithm. In another approach to event detection in baseball videos [24], a rule-based decision module infers what happened by checking the information changes in the caption. With the help of official baseball rules, the rule-based decision module detects events occurring between two consecutive pitch shots. In addition, a model-based decision module further classifies events that could not be explicitly determined by checking the caption information. The four shot context features from the test sequence are classified as events by the *k*-nearest neighbour algorithm. The following events are detected: *hit*, *double*, *triple*, *home run*, *stolen base*, *caught steal*, *fly out*, *strikeout*, *base on balls*, *sacrifice bunt*, *sacrifice fly*, *double play*, and *triple play*.

4.3 Meeting event detection from indoors scenes

Meetings are social events where people exchange ideas. An information system for indoor-group oriented activities is provided in [103]. This involves the storage and indexing of multimedia data consisting of video, audio, PowerPoint files and other media-based information. Two steps are used for the meeting system: aspects of the event model are specified based on user requirements and domain semantics; the second involves implementation of the model. A natural approach would be to model both the static and dynamic aspects of the information simultaneously with one model.

Meeting events are detected from projected documents based on document image analysis for integrating non-temporal documents into multimedia meeting archives [15]. The approach takes advantage of the observable events related to documents that are visible during meetings. Slide changes are detected using the Synchronized Multimedia Integrating Language.

In [95, 97], Bayesian Network, Gaussian Mixture Model, Maximum Likelihood Pixel, Radial Basis Network, and SVM classifiers are evaluated for detecting meeting events such as discussion, monologue, note-taking, white board activities and presentations. Segmentation and classification of meeting events are implemented using multiple classifier fusion and dynamic programming. Within the DBN, a multistream HMM is coupled with a linear dynamical system to compensate for disturbances in the data. Three audio and visual modalities are fused in the multi-stream HMM. The DBN shows a significantly higher recognition performance compared to a single-stream HMM. Combining artificial neural networks and a HMM result in a highly discriminative system which outperformed conventional models [97].

Meeting event labelling is both laborious and time-consuming [137], since meetings are often lengthy and events are jointly defined by audio-visual patterns. A meeting is modelled as a sequence of exclusive events taken from the following set: *discussion*, *monologue*, *note-taking*, *presentation* and *white-boarding*. Given a sequence of audiovisual features extracted from a meeting, the Viterbi algorithm produces the sequence of states, in other words events, most likely to have generated the features.

Events in an office environment are detected in [106]. These events include *talking on a phone*, *checking voicemail*, *bringing a cup to a face*, *scratching/rubbing face*, *yawning and hand at mouth*, *putting on glasses/earphones*, and *rubbing eyes*. Temporal boosting was used to improve weak classifiers by allowing them to use the previous classifiers response in evaluating the current frame, and making use of the temporal continuity of video at the classifier and detector level. In addition, the framework is able to combine information from multiple cameras to increase overall system performance.

In [133], two kinds of lecture events are detected: *a speech period* and *a chalkboard writing period*. A speech period is detected by voice activity detection with LPC cespectrum, and classified into speech or non-speech using the Mahalanobis distance. The chalkboard writing periods are detected by using a graph-cut technique to segment a precise region of interest, such as an instructor, in order to detect a change of characters on the chalkboard.

4.4 Event for video indexing and adaption

Teraguchi et al. [113] propose a construction method for personalized video digests based on generated indexes. The semi-manual content-based indexes are automatically generated by manually flagging predefined and easily recognizable events while watching the video. Hence, the time required for flagging is reduced and rapid video indexing is achieved. Event-based indexes are generated by using either a single- and/or multiple-trigger index.

An event scene in either stored or live feed video can be described by using the semantic DSs of MPEG-7. Both theme and content can be described using one or more of the following object DSs: AgentObject DS, Event DS, Concept DS, SemanticPlace DS, Semantic Time DS and SemanticState DS. Thawani et al. [115] detect an ad using event driven semantics. Both ads and events are represented using MPEG-7 semantics DSs and the most relevant ads are selected based on program events. The selection and presentation of customized targeted advertisements, based on program events, are detected and derived from the semantic analysis of the scene and objects in the scene. The Event Recognition System specifically performs two kinds of analysis, prediction and recognition. Depending on the number of clusters formed, an appropriate number of ads are selected and cached. Caching an Ad selection, based on an event class description, results in a higher chance of utilization.

An event on demand video adaptation system, which follows the MPEG-21 digital item adaptation framework, is reported in [130, 131]. This system provides a generic adaptation solution to take account of user's preferences, semantic aspects and network environments etc. Events are detected by audio/video analysis and annotated by the MPEG-7 DSs. The system has two primary processes: event

identification & annotation, and MPEG-21 Digital Item Adaptation. The system mainly consists of an adaptation decision engine and an adaptation operation engine. The former decides the source parameters by considering AQoS, usage environment description and constraints. The latter changes the generic Bitstream Syntax Description (gBSD), based on the adaptation decision, and adapts the video resource according to the newly adapted gBSD. The event information is parsed from an MPEG-21 annotated XML file together with its bitstream to generate a gBSD. The user's preference, network characteristics and Adaptation QoS (AQoS) influence the adaptation decision. Event selection and frame dropping are effective and efficient ways to meet a user's preference, and to adapt to variation in the network condition. The MPEG-21 digital item adaptation helps to reduce computational complexity through XML manipulations. Moreover, XML based adaptation provides a generic solution for all video formats. Compared to other adaptation methods, XML based adaptation provides a quick, affordable and convenient solution. Events for video adaption are also modelled as FSMs in [16], where combinations of feature values determine a person's transition from one emotional state to another.

5 Summary and conclusion

In this paper, we have summarized recent work in visual event computing. We began by defining an event and specifying different types of events. We discussed properties of events and the different types of relationships between them. We then reviewed the use of prominent techniques from statistical pattern recognition, machine learning and artificial intelligence for event detection. The use of different media, such as archived photographs, and live/archived video, for event detection was reviewed. In particular, the use of different features, such as trajectories, spatial, temporal, audio-visual and others, for event detection and recognition, mining and reasoning were reviewed. We also reviewed the use of events for video indexing and adaptation.

Up to now, most of the work on event computing has been primarily concerned with detection from archived video, and on capturing events from live visual streams within a multimodal multimedia environment. Future work will pay more attention on other event operations such as classification, exploration, reasoning, mining and prediction. When events are captured and detected, they will be stored in a database specifically related to events. In addition, other media associated with the event will be stored in another database. To access events with their related data, the information which binds them must be stored too. Another database will be required to keep track of common usage aspects among clients who manipulate events. Search and retrieval using events is another key area of future research. Events can be used in a search as a semantic unit, as useful semantic clusters can be derived from event entities. The retrieval system can then utilize these semantic clusters to refine the re-ranking process in the presence of pseudo relevance feedback. Filed events are also very helpful for learning, training, analysis, influencing further actions and prediction. Events can also be used for video summarization and abstraction. In order to register the temporal event of an object, the original video clips and a set of their descriptors are abstracted to be stored in the database for search and retrieval. When

a query is invoked through the interface, it is interpreted in a predefined descriptor format and compared with all the stored descriptors. After retrieving the interested video clip, the system returns the best matching events to the user. Lastly, research will also be needed for an event browser and in the presentation of events.

Acknowledgements We appreciate for the great help from the colleagues of Queen's University Belfast(QUB): Prof. Danny Crookes, Dr. Weiru Liu, Dr. Paul Miller, and Dr. Xiwu Gu etc. This work was partially supported by QUB research project: Unusual event detection in audio-visual surveillance for public transport (NO.D6223EEC).

References

1. Adams B, Venkatesh S (2005) Situated event bootstrapping and capture guidance for automated home movie authoring. In: Proc. of ACM Multimedia'05, Singapore, pp 754–763
2. Alahari K, Jawahar C (2006) Discriminative actions for recognising events. In: Proc. of ICVGIP'06. LNCS, vol 4338, India, pp 552–1563
3. Al-Hames M, Rigoll G (2005) A multi-modal mixed-state dynamic bayesian network for robust meeting event recognition from disturbed data. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 45–48
4. Alahari K, Jawahar C (2006) Dynamic events as mixtures of spatial and temporal features. In: Proc. of ICVGIP'06. LNCS vol 4338, India, pp 540–551
5. Alan Fern RG, Siskind JM (2002) Learning temporal, relational, force-dynamic event definitions from video. In: Proc. of AAAI'02, Palo Alto, California, pp 159–166
6. Amer A, Dubois E, Mitiche A (2002) Context-independent real-time event recognition: application to key-image extraction. In: Proc. of IEEE ICPR'02, Quebec, Canada, pp 945–948
7. Amera A, Duboisb E, Mitichec A (2005) Rule-based real-time detection of context-independent events in video shots. Real-Time Imaging 11(3):244–256
8. Andrade EL, Blunsden S, Fisher RB (2006) Modeling crowd scenes for event detection. In: Proc. of ICPR'06, Hong Kong, China, pp 175–178
9. Appan P, Sundaram H (2004) Networked multimedia event exploration. In: Proc. of ACM multimedia. New York City, USA, pp 40–47
10. Atrey PK, Kankanhalli MS, Jain R (2006) Information assimilation framework for event detection in multimedia surveillance systems. Springer/ACM Multimedia Syst J 12(3):239–253
11. Babaguchi N, Kawai Y, Kitahashi T (2002) Event based indexing of broadcasted sports video by intermodal collaboration. IEEE Trans Multimedia 12(3)68–75
12. Babaguchi N, Sasamori S, Kitahashi T, Jain R (1999) Detecting events from continuous media by intermodal collaboration and knowledge use. In: Proc. of IEEE ICMCS'99, Firenze, Italy, pp 782–786
13. Barnard M, Odobez J-M (2005) Sports event recognition using layered hmms. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 1150–1153
14. Baulier J, Blott S, Korth HF, Silberschatz A (1998) A database system for real-time event aggregation in telecommunication. In: Proc. of VLDB'98, pp 680–684, New York, USA
15. Behera A, Lalanne D, Ingold R (2004) Looking at projected documents: event detection & document identification. In: Proc. of IEEE ICME'04, Taipei, pp 2127–2130
16. Bertini M, Bimbo AD, Cucchiara R, Prati A (2004) Object-based and event-based semantic video adaptation. In: Proc. of IEEE ICPR'04, Cambridge, UK, pp 987–990
17. Black MJ (1999) Explaining optical flow events with parameterized spatio-temporal models. In: Proc. of IEEE CVPR'99, Ft Collins, USA, pp 326–332
18. Bonzanini A, Leonardi R, Migliorati P (2001) Event recognition in sport programs using low-level motion indices. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 2127–2130
19. Boykin S, Merlino A (2000) Machine learning of event segmentation for news on demand. Commun ACM 43(2):35–41
20. Burges CJ (1998) A tutorial on Support Vector Machines for pattern recognition. Data Min Knowl Disc 2:121–167

21. Chan MT, Hoogs A, Schmiederer J, Petersen M (2004) Detecting rare events in video using semantic primitives with HMM. In: Proc. of IEEE ICPR'04, Cambridge, UK, pp 150–154
22. Chan MT, Hoogs A, Sun Z, Schmiederer J, Bhotika R, Doretto G (2006) Event recognition with fragmented object tracks. In: Proc. of IEEE ICPR'06, HongKong, China, pp 412–416
23. Chan MT, Hoogs A, Bhotika R, Perera AGA, Schmiederer J, Doretto G (2006) Joint recognition of complex events and track matching. In: Proc. of IEEE CVPR'06, New York, USA, pp 1615–1622
24. Chu W-T, Wu J-L (2005) Integration of rule-based and model-based decision methods for baseball event detection. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 137–140
25. Cooper M, Foote J, Girgensohn A, Wilcox L (2005) Temporal event clustering for digital photo collections. ACM Trans on TOMCCAP 1(3):269–288
26. Cui P, Sun L, Liu Z-Q, Yang S (2007) A sequential monte carlo approach to anomaly detection in tracking visual events. In: Proc of IEEE CVPR'07, Minnesota, USA
27. Dai S, Dhawan AP (2007) Adaptive learning for event modeling and characterization. Pattern Recogn 40(5):1544–1555
28. Demers A, Gehrke J, Hong M, Riedewald M, White W (2005) A general algebra and implementation for monitoring event streams. Cornell University, Tech Rep TR2005-1997
29. Engle JC, Odutola A (2006) Control field event detection in a digital video recorder. US Patent 5699124
30. Fern A, Givan R, Siskind JM (2002) Specific-to-general learning for temporal events. In: Proc. of AAAI'02, Palo Alto, USA, pp 152–158
31. Foresti GL, Marcenaro L, Regazzoni CS (2002) Automatic detection and indexing of video event shots for surveillance applications. IEEE Trans Multimedia 4(4):459–471
32. Foresti GL, Micheloni C, Snidaro L (2004) Event classification for automatic visual-based surveillance of parking lots. In: Proc. of IEEE ICPR'04, Cambridge, UK, pp 314–317
33. Franois ARJ, Nevatia R, Hobbs JR, Bolles RC (2003) VERL: an ontology framework for representing and annotating video events. IEEE Multimed 76:269–288
34. Frawley GP-S W, Matheus C (1992) Knowledge discovery in databases: an overview. AI Mag 13(3):213–228
35. Gehani NH, Jagadish HV, Shmueli O (1992) Composite event specification in active databases: model & implementation. In: Proc. of VLDB'92, Vancouver, Canada, pp 327–338
36. Ghahramani Z (1998) Adaptive processing of sequences and data structures, lecture notes in artificial intelligence. ch. Learning Dynamic Bayesian Networks. Springer-Verlag, Berlin, pp 168–197
37. Ghanem N, DeMenthon D, david Doermann, Davis L (2004) Representation and recognition of events in surveillance video using Petri nets. In: Proc. of workshop on event mining, Madison, USA, vol 7, no 7, p 112
38. Gu H, Ji Q (2004) Facial event classification with task oriented dynamic Bayesian network. In: Proc. of IEEE CVPR'04, Reno, USA, pp 870–875
39. Haering NC, Qian RJ, Sezan MI (2000) A semantic event-detection approach and its application to detecting hunts in wildlife video. IEEE Trans Circuits Syst Video Technol 6(10): 857–868
40. Hakeem A, Shah M (2005) Multiple agent event detection and representation in videos. In: Proc. of AAAI'05, Pittsburgh, USA, pp 89–94
41. Hakeem A, Sheikh Y, Shah M (2004) Casee: a hierarchical event representation for the analysis of videos. In: Proc. of AAAI'04. San Jose, USA, pp 263–268
42. Hamid R, Johnson AY, Batta S, Bobick AF, Isbell CL, Coleman G (2005) Detection and explanation of anomalous activities: representing activities as bags of event n-grams. In: Proc. of IEEE CVPR'05. San Diego, USA, pp 1031–1038
43. Hand HMD, Smyth P (2001) Principles of data mining. MIT Press, Cambridge, USA
44. Haynes S, Jain R (1984) Low level motion events, trajectory discontinuities. In: Proc. of the first conference on artificial intelligence applications. San Diego, USA, pp 251–256
45. Haynes S, Jain R (1984) Event detection and correspondence. In: Proc. of Optical engineering, San Diego, USA, pp 251–256
46. Hongeng S (2004) Unsupervised learning of multi-object event classes. In: Proc. of the 15th British machine vision conference (BMVC'04). London, UK
47. Hongeng S, Nevatia R (2003) Large-scale event detection using Semi-Hidden Markov Models. In: Proc. of IEEE ICCV'03. Nice, France, pp 1455–1462

48. Hopkins M (2002) Strategies for determining causes of events. In: Proc. of AAAI'02. Palo Alto, California, pp 546–552
49. Johnson N, Hogg DC (1995) Learning the distribution of object trajectories for event recognition. In: Proc. of the 6th British conference on machine vision, Surrey, UK, pp 583–592
50. Joo S-W, Chellappa R (2006) Attribute grammar-based event recognition and anomaly detection. In: Proc. of CVPRW'06, New York, USA, pp 107–115
51. Jung Y-K, Lee K-W, Ho Y-S (2001) Content-based event retrieval using semantic scene interpretation for automated traffic surveillance. IEEE Trans Intell Transp Syst 2(3):151–163
52. Kang H-B (2002) Analysis of scene context related with emotional events. In: Proc. of ACM Multimedia'02, Juan Les Pins, France, pp 311–314
53. Kawashima H, Matsuyama T (2002) Integrated event recognition from multiple sources. In: Proc. of IEEE ICPR'02, Quebec, Canada, pp 785–789
54. Ke Y (2005) Efficient visual event detection using volumetric features. In: Proc. of IEEE ICCV'05, Beijing, China, pp 166–173
55. Ke Y, Sukthankar R, Hebert M (2007) Event detection in crowded videos. In: Proc of IEEE ICCV'07, Rio de Janeiro, Brazi
56. Krzysztof W, Cios P, Swiniarski R (1998) Data mining methods for knowledge discovery. Kluwer, Norwell, MA
57. Lee D, Yannakakis M (1996) Principles and methods of testing finite state machines—a survey. Proc IEEE 84(8):1090–1122
58. Li L-J, Li F-F (2007) What, where and who? classifying events by scene and object recognition. In: Proc of IEEE ICCV'07, Rio de Janeiro, Brazi
59. Li C-H, Chiu C-Y, Huang C-R, Chen C-S, Chien L-F (2006) Image content clustering and summarization for photo collection. In: Proc. of IEEE ICME'06, Canada
60. Lie W-N, Shia S-H (2005) Combining caption and visual features for semantic event classification of baseball video. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 1254–1257
61. Lie W-N, Lin T-C, Hsia S-H (2004) Motion-based event detection and semantic classification for baseball sport videos. In: Proc. of IEEE ICME'04, Taipei, Taiwan, pp 1567–1570
62. Lim J-H, Tian Q, Mulhem P (2003) Home photo content modeling for personalized event-based retrieval. IEEE Multimed 10(4):28–37
63. Loui AC, Savakis AE (2001) Automatic image event segmentation and quality screening for albuming applications. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 1125–1128
64. Loui AC, Savakis AE (2003) Automated event clustering and quality screening of consumer pictures for digital albuming. IEEE Trans Multimedia 10(4):390–402
65. Lu C, Ferrier NJ (2004) Repetitive motion analysis: segmentation and event classification. IEEE Trans PAMI 26(2):258–263
66. Ma Y, Bazakos M, Miller B, Buddharaju P (2006) Activity awareness: from predefined events to new pattern discovery. In: Proc. of ICVS'06, p 11
67. Malaia E (2006) Event structure representation in ontological semantics. In: Proc. of MLMTA (international conference on machine learning models, technologies & applications), Las Vegas, USA, pp 36–42
68. Matthew AG, Cooper D, Foote J, Wilcox L (2003) Temporal event clustering for digital photo collections. In: Proc. of ACM multimedia'03, Berkely, USA, pp 364–373
69. Mei T, Wang B, Hua X-S, Zhou H-Q, Li S (2006) Probabilistic multimodality fusion for event based home photo clustering. In: Proc. of IEEE ICME'06, Canada, pp 1757–1760
70. Miyauchi S, Hirano A, Babaguchi N, Kitahashi T (2002) Collaborative multimedia analysis for detecting semantical events from broadcasted sports video. In: Proc. of ICPR'02, Tokyo, Japan, pp 1009–1012
71. Mustafa A, Sethi I (2005) Detecting retail events using moving edges. In: Proc. of AVSS 2005, pp 626–631
72. Naaman M, Harada S, Wang Q (2004) Context data in geo-referenced digital photo collections. In: Proc. of ACM multimedia, New York, NY, USA, pp 196–203
73. Naaman M, Yeh RB, Garcia-Molina H, Paepcke A (2005) Leveraging context to resolve identity in photo albums. In: Proc. of the 5th ACM/IEEE-CS joint conference on digital libraries, Denver, CO, USA, pp 178–187
74. Naphade M, Huang T (2002) Discovering recurrent events in video using unsupervised methods. In: Proc. of IEEE ICIP'02
75. Naphade MR, Garg A, Huang TS (1997) Duration dependent input output markov models for audio-visual event detection. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 369–372

76. Nevatia R, Hobbs J, Bolls B (2004) An ontology for video event representation. In: Proc. of CVPRW'04, Washington, USA, vol 9, no 27, p 119
77. Nitta N, Babaguchi N, Kitahashi T (2000) Extracting actors, actions and events from sports video—a fundamental approach to story tracking. In: Proc of IEEE ICPR'00, Barcelona, Spain, pp 4718–4721
78. Nishida T, Kamijo S, Ikeuchi K (2001) Automated system of acquiring and visualizing track event statistics from track images. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 169–172
79. O'Hare N, Gurrin C, Lee H, Murphy N, Smeaton AF, Jones GJ (2005) Digital photos: where and when? In: Proc. of ACM multimedia'05, Singapore
80. Okadome T (2006) Event representation for sensor data grounding. International Journal of Computer Science and Network Security 6(10):129–162
81. Osadchy M, Keren D (2004) A rejection-based method for event detection in video. IEEE Trans Circuits Syst Video Technol 4(14):534–541
82. Pack D, Singh R, Brennan S, Jain R (2004) An event model and its implementation for multimedia information representation and retrieval. In: Proc. of IEEE ICME'04, Taipei, Taiwan, pp 1611–1614
83. Park S, Aggarwal JK (2004) Event semantics in two-person interactions. In: Proc. of IEEE ICPR'04, Taipei, Taiwan, pp 227–230
84. Peyrard N, Bouthemy P (2003) Detection of meaningful events in videos based on a supervised classification approach. In: Proc. of IEEE ICIP'03, pp 621–625
85. Piater JH, Richetto S, Crowley JL (2002) Event-based activity analysis in live video using a generic object tracker. In: Proc. of third IEEE international workshop on performance evaluation of tracking and surveillance, Copenhagen, pp 1–8
86. Pingali GS, Jean Y, Opalach A, Carlbom I (2001) Lucentvision: converting real world events into multimedia experiences. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 1433–1436
87. Pinzon J, Singh R, Taube W, Galan J (2006) Designing interactions in event-based unified management of personal multimedia information. In: Proc. of IEEE ICME'06, Canada, pp 337–340
88. Piriou G, Bouthemy P, Yao J-F (2004) Learned probabilistic image motion models for event detection in videos. In: Proc. of IEEE ICPR'04, Tokyo, Japan, pp 207–210
89. Qian RJ, Haering NC, Sezan MI (1999) A computational approach to semantic event detection. In: Proc. of IEEE CVPR'99, Ft Collins, USA, pp 200–206
90. Qiu G, Feng X, Fang J (2004) Compressing histogram representations for automatic color photo categorization. Pattern Recogn 37:2177–2193
91. Quinton A (1979) Objects and events. Mind 88(350):197–214
92. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
93. Rao C, Shah M (2001) View-invariant representation and learning of human action. In: Proc. of IEEE workshop on detection and recognition of events in video, Vancouver, Canada, pp 55–63
94. Rao C, Shah M, Syeda-Mahmmod T (2003) Invariance in motion analysis of videos. In: Proc. of ACM multimedia'03, Bekerley, USA, pp 518–527
95. Reiter S, Rigoll G (2004) Segmentation and classification of meeting events using multiple classifier fusion and dynamic programming. In: Proc. of IEEE ICPR'04, Cambridge, UK, pp 434–437
96. Remagnino P, Jones G (2001) Classifying surveillance events from attributes and behaviour. In: Proc. of British machine vision conf, Manchester, UK, pp 685–694
97. Reiter S, Schuller B, Rigoll G (2006) Segmentation and recognition of meeting events using a two-layered hmm and a combined mlp-hmm approach. In: Proc. of IEEE ICME'06, Canada, pp 953–956
98. Saad MS, Khan M (2006) A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: Proc. of ECCV'06, Graz, Austria, pp 133–146
99. Sadlier D, O'Connor NE (2005) Event detection based on generic characteristics of field-sports. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 759–762
100. Satoh Y, Tanahashi H, Wang C, Kaneko S, Niwa Y, Yamamoto K (2002) Robust event detection by radial reach filter (RRF). In: Proc. of IEEE ICPR'02, Quebec, Canada, pp 623–626
101. Schwalb E, Kask K, Dechter R (1994) Temporal reasoning with constraints on fluents and events. In: Proc. of AAAI'94, Seattle, USA, pp 1067–1072

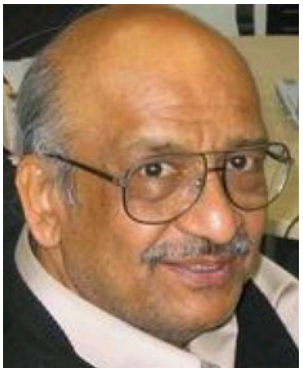
102. Shotton DM, Rodríguez A, Guil N, Trelles O (2000) Object tracking and event recognition in biological microscopy videos. In: Proc. of IEEE ICPR'00, Seattle, USA, pp 4226–4229
103. Sinha SN, Pollefeys M (2005) Synchronization and calibration of a camera network for 3D event reconstruction from live video. In: Proc. of IEEE CVPR'05, San Diego, USA, p 1196
104. Siskind JM (2002) Visual event classification via force dynamics. In: Proc of AAAI'02, San Diego, USA, pp 149–155
105. Siskind JM, Morris Q (1996) A maximum-likelihood approach to visual event classification. In: Proc. of ECCV'96. LNCS, vol 1065, London, UK, pp 347–360
106. Smith PN, da Vitoria Lobo, Shah M (2002) Temporalboost for event recognition. In: Proc. of IEEE ICCV'05, San Diego, CA, USA, pp 733–740
107. Snoek C, Worring M (2006) Multimedia event-based video indexing using time intervals. Trans Multimedia 10(4):638–647
108. Syeda-Mahmood T (2002) Retrieving actions embedded in video. In: Proc. of ACM Multimedia'02, Juan Lins Pins, France, pp 513–522
109. Syeda-Mahmood T, Srinivasan S (2000) Detecting topical events in digital video. In: Proc. of ACM multimedia'00. Marina del Rey, Los Angeles, USA, pp 85–94
110. Syeda-Mahmood T, Vasilescu A (2001) Recognizing action events from multiple view points. In: Proc. of IEEE workshop on detection and recognition of events in video 2001, Las Palmas, USA, pp 64–72
111. Tang Q, Koprinska I, Jin JS (2005) Content-adaptive transmission of reconstructed soccer goal events over low bandwidth networks. In: Proc. of ACM Multimedia'05, Singapore, pp 271–274
112. Teisseire M, Poncelet P, Cicchetti R (1994) Towards event-driven modelling for database design. In: Proc. of VLDB'94. Santiago de Chile, Chile, pp 285–296
113. Teraguchi M, Masumitsu K, Echigo T, Sekiguchi S, Etoh M (2002) Rapid generation of event-based indexes for personalized video digests. In: Proc of IEEE ICPR'02, Quebec, Canada, pp 1041–1044
114. Tesic J, Newsam S, Manjunath B (2002) Scalable spatial event representation. In: Proc. of IEEE ICME'02. Lausanne, Switzerland, pp 229–232
115. Thawani A, Gopalan S, Sridhar V (2004) Event driven semantics based ad selection. In: Proc. of IEEE ICME'04. Taipei, Taiwan, pp 1875–1878
116. Trausti TSH, Kristjansson T, Brendan Frey J (2001) Event-coupled hidden Markov models. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 385–388
117. Tong X-F, Lu H-Q, Liu Q-S (2004) A three-layer event detection framework and its application in soccer video. In: Proc. of IEEE ICME'04, Taipei, Taiwan, pp 1551–1554
118. Tovinkere V, Qian RJ (2001) Detecting semantic events in soccer games: towards a complete solution. In: Proc. of IEEE ICME'01, Tokyo, Japan, pp 1551–1554
119. Vassiliou A, Salway A, Pitt D (2004) Formalizing stories sequences of events and state changes. In: Proc. of IEEE ICME'04. Taipei, Taiwan, pp 587–590
120. Veeraraghavan H, Papanikolopoulos N, Schrater P (2007) Learning dynamic event descriptions in image sequences. In: Proc. of IEEE CVPR'07, Minnesota, USA, pp 1–6
121. Welch G, Bishop G (2001) An introduction to the Kalman filter. In: Proc. of ACM SIGGRPH'01, Los Angeles, USA
122. Westermann U, Jain R (2006) Toward a common event model for multimedia applications. International Journal on Semantic Web & Information Systems 14(1):19–29
123. Worboys MF, Hornsby K (2004) From objects to events: gem, the geospatial event model. In: Proc. of GIScience'04, Adelphi, USA
124. Xiang T, Gong S, Parkinson D (2002) Autonomous visual events detection and classification without explicit object-centred segmentation and tracking. In: Proc. of British machine vision conference, Cardiff, UK, pp 685–694
125. Xu H, Chua T-S (2004) The fusion of audio-visual features and external knowledge for event detection in team sports video. In: Proc. of ACM SIGMM international workshop on multimedia information retrieval, New York, USA
126. Xu H, Chua T-S (2006) Fusion of AV features and external information sources for event detection in team sports video. ACM TOMCCAP 2(1):44–67
127. Xu H, Fong T-H, Chua T-S (2005) Fusion of multiple asynchronous information sources for event detection in soccer video. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 1242–1245

128. Xu G, Ma Y-F, Zhang H, Yang S (2002) Motion based event recognition using HMM. In: Proc. of IEEE ICPR'02, Quebec, Canada, pp 831–834
129. Xu C, Wang J, Li Y, Wan K, Duan L-Y (2006) Live sports event detection based on broadcast video and web-casting text. In: Proc. of ACM multimedia'06, Santa Barbara, CA, USA, pp 221–230
130. Xu M, Li J, Hu Y, Chia L-T, Lee B-S, Rajan D, Cai J (2006) An event-driven sports video adaptation for the MPEG-21 DIA framework. In: Proc of IEEE ICME'06, Canada, pp 1245–1248
131. Xu M, Li J, Chia L-T, Hu Y, Lee B-S, Rajan D, Jin JS (2006) Event on demand with MPEG-21 video adaptation system. In: Proc. of ACM multimedia'06, Santa Barbara, USA, pp 921–930
132. Ye Q, Huang Q, Gao W, Jiang S (2005) Exciting event detection in broadcast soccer video with mid-level description and incremental learning. In: Proc. of ACM Multimedia'05, Singapore, pp 455–458
133. Yokoi T, Fujiyoshi H (2006) Generating a time shrunk lecture video by event detection. In: Proc. of IEEE ICME'06, Canada, pp 641–644
134. Yoneyama A, Yeh CH, Kuo CCJ (2004) Robust traffic event extraction via content understanding for highway surveillance system. In: Proc. of IEEE ICME'04, Taipei, Taiwan, pp 1679–1682
135. Yoon K, DeMenthon D, Doermann DS (2000) Event detection from MPEG video in the compressed domain. In: Proc. of IEEE ICPR'00, Singapore, pp 1819–1822
136. Zhang D, Chang S-F (2002) Event detection in baseball video using superimposed caption recognition. In: Proc. of ACM multimedia'02, Juan Les Pins, France, pp 315–318
137. Zhang D, Gatica-Perez D, Bengio S (2005) Semi-supervised meeting event recognition with adapted HMMs. In: Proc. of IEEE ICME'05, Amsterdam, The Netherlands, pp 1102–1105
138. Zhang Z, Huang K, Tan T, Wang L (2007) Trajectory series analysis based event rule induction for visual surveillance. In: Proc. of IEEE CVPR'07, Minnesota, USA
139. Zelnik-Manor L, Irani M (2001) Event-based analysis of video. In: Proc. of IEEE CVPR'01, Hawaii, USA, pp 123–130
140. Zelnik-Manor L, Irani M (2006) Statistical analysis of dynamic actions. IEEE Trans Pattern Anal Mach Intell 28(9):1530–1535
141. Zhang D, Gatica-Perez D, Bengio S, McCowan I (2005) Semi-supervised adapted HMMs for unusual event detection. In: Proc. of IEEE CVPR'05, San Diego, USA, pp 611–618
142. Zhong H, Shi J, Visontai M (2004) Detecting unusual activity in video. In: Proc of IEEE CVPR'04, Washington, DC, USA, pp 819–826
143. Zhou H, Kimber D (2004) Unusual event detection via multi-camera video mining. In: Proc. of IEEE ICVR'04, Cambridge, UK, pp 1161–1166
144. Zhu G, Huang Q, Xu C, Rui Y, Jiang S, Gao W, Yao H (2007) Trajectory based event tactics analysis in broadcast sports video. In: Proc. of ACM Multimedia'07, Augsburg, Germany, pp 58–67

WeiQi Yan received his Ph.D. degree from Academia Sinica, China in 2001. His research interests include Multimedia Systems and Multimedia Security. He has published nearly 80 papers. Dr. Yan is serving as an associate editor of Journal of Multimedia, associate editor of International Journal of Digital Crime Forensics, a guest editor of the Springer Trans. on data hiding and multimedia security (DHMMS).

Declan F. Kieran received his MEng degree from Queen's University Belfast in 2009. He is a Ph.D. student with Queen's University Belfast. His research interests are biomedical image processing and visual event computing for surveillance.

Setareh Rafatirad received her master degree from UC Irvine in 2009, she is a PhD student at School of Information and Computer Science, University of California Irvine, USA. Her research focuses on event-based data modeling and query language.



Ramesh Jain is the first Bren Professor in Bren School of Information and Computer Sciences at University California Irvine, USA. Prof. Jain's interests combine multimedia information systems, visual computing, and intelligent systems, and include multimedia search and experiential computing for live as well as archived data. He is the former president of ACM SIGMM, an ACM Fellow, IEEE Fellow, AAAI Fellow, IAPR Fellow and SPIE Fellow.