VEKG: Video Event Knowledge Graph to Represent Video Streams for Complex Event Pattern Matching

Piyush Yadav
Lero- Irish Software Research Centre
National University of Ireland Galway
Galway, Ireland
piyush.yadav@lero.ie

Edward Curry
Lero- Irish Software Research Centre
National University of Ireland Galway
Galway, Ireland
edward.curry@lero.ie

***Abstract*— Complex Event Processing (CEP) is a paradigm to detect event patterns over streaming data in a timely manner. Presently, CEP systems have inherent limitations to detect event patterns over video streams due to their data complexity and lack of structured data model. Modelling complex events in unstructured data like videos not only requires detecting objects but also the spatiotemporal relationships among objects. This work introduces a novel video representation technique where an input video stream is converted to a stream of graphs. We propose the Video Event Knowledge Graph (VEKG), a knowledge graph driven representation of video data. VEKG models video objects as nodes and their relationship interaction as edges over time and space. It creates a semantic knowledge representation of video data derived from the detection of high-level semantic concepts from the video using an ensemble of deep learning models. To optimize the run-time system performance, we introduce a graph aggregation method VEKG-TAG, which provides an aggregated view of VEKG for a given time length. We defined a set of operators using event rules which can be used as a query and applied over VEKG graphs to discover complex video patterns. The system achieves an F-Score accuracy ranging between 0.75 to 0.86 for different patterns when queried over VEKG. In given experiments, pattern search time over VEKG-TAG was 2.3X faster as compared to the baseline.**

Keywords—Video Representation, Pattern Matching, Complex Event Processing, Knowledge Graphs, Spatiotemporal Networks

I. Introduction

With the evolution of concept like smart cities, smart homes, and self-driving cars, there is an exponential growth in sensor devices. The world is now transitioning from Internet of Things (IoT) to Internet of Multimedia Things (IoMT) [1] where visual sensors are deployed ubiquitously and streaming massive amount of video data. Analytics is performed over these video streams to detect events of interest in applications like surveillance, traffic, agriculture and disaster monitoring for effective decisions. Middleware systems like event processing act as communication abstraction between data publishers (sensors) and subscribers (applications) and enable consistent and timely event detection from the streaming data.

The event processing paradigm is characterized by the concept of *timeliness,* which is collectively expressed with different terms like *on-the-fly*, *low-latency, high-throughput* and *real-time processing* [2]. Within event processing, Complex Event Processing (CEP) systems have been increasingly adopted in different domains like traffic monitoring, maritime surveillance and financial applications [2] [3] to detect event patterns and send notifications in real-time. CEP system detects complex events by correlating simple events based on the registered query. The CEP matching model is continuous where once the query is registered, the matching engine tries to mine patterns over incoming streams in an online setting. The system captures the recent state of the stream and applies a set of operators' rule and triggers notification as the pattern is detected. While early works have started to investigate images within event processing [4], presently CEP systems have limitations to process video streams.

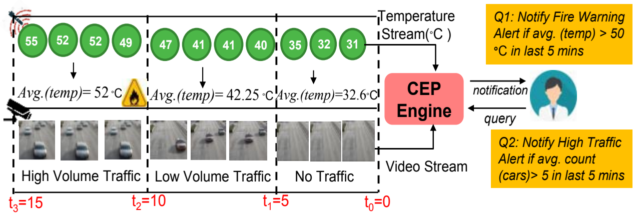


Fig. 1 Motivational scenario

A. *Motivational Scenario and Challenges*

Consider a smart city scenario where the city administrator has subscribed to CEP system for a fire warning and high volume traffic alert. As shown in Fig.1, the CEP engine is receiving two data streams, one from a building temperature sensor and another from a CCTV camera. As per the query 1 rule (Q1), if the average temperature is greater than 50°C in last five minutes then CEP system should notify a *fire warning* alert. The temperature sensor emits '*temperature event'* in every second and will be considered a simple event. The complex event *'fire warning'* is combined by averaging a simple '*temperature event*' for a given time. In Fig. 1, a CEP system will raise a *fire warning* alert at time t_2-t_3 as the average temperature of incoming streams is higher than 50°C. Similarly, for query 2 (Q2), the CEP system should notify the traffic volume, but faces multiple challenges to process video streams. Most of the existing CEP and stream processing systems work with an assumption that the incoming stream has a structured format like key-value pairs (*temperature = 52˚C* in Fig. 1) and XML [5]. However, video data are highly complex and unstructured in terms of an event model. At the machine level, contents of the video data are represented as low-level features like color, pixels, shapes and textures while humans interpret video content as a high-level semantic concept like car, chair, and person. While visualizing, human cognition can easily understand and differentiate events like *no traffic* and *high volume traffic*. It is difficult for CEP systems to reason

978-1-7281-4129-9/19/$31.00 ©2019 IEEE
DOI 10.1109/GC46384.2019.00011

over video data as 1) it has no structured representation and data model where semantic concepts boundaries are not known and organized, 2) the video event patterns spans over time and space. To process video streams queries (like Q2) in a CEP engine leads to significant challenges.

- How to *extract and represent* low-level video content and video stream into a structured data model with high-level semantic concepts?
- How to *identify relationships* between semantic concepts of video content which occurs over time and space?
- How to *match* spatiotemporal CEP query rules over the represented data model efficiently at runtime?

To overcome the above challenges, we will outline the basic requirements which are required to model the video stream.

B. *Problem Requirements*

Videos are considered a continuous sequence of image frames, which consists of *objects*. Humans perceive objects as a high-level semantic concept which occupy specific positions in an image. Technically, objects are a collection of low-level image features which have been given a high-level semantic label like *car* and *person*. Videos may have an evolving nature where different objects occur over time, generating varying nature of complex events. Modelling complex events in unstructured data like videos require detecting objects and relationships between them. We enlist four essential requirements to represent video data suitable for complex event processing.

R1- Object Detection: Objects are considered as fundamental building blocks of videos. There is a need to detect objects from low-level video content as they act as a backbone for the required data model. For example, a simple CEP query can be to *notify if any car object is present* in the video feed.

R2- Attribute Detection: An object can have specific characteristics which differentiate it from other objects. These can be termed as objects attributes. For example, in Fig.1 there are *car* objects with *color* (i.e. red and silver) and *type* (i.e. sedan and van) attributes.

R3- Spatiotemporal Relationship Identification: The objects in videos interact with each other and have a relationship across space and time. These interactions generate a spatiotemporal network giving rise to complex events. For example, in Fig.1 a *red car* is spatially located to *right* of a *silver car* (frame 4). Here *right* is a spatial relationship between two objects, i.e. two *car* with *color* attributes *red* and *silver*. Similarly, complex events such as *high traffic flow,* require the relationships across multiple objects.

Thus, there is a clear need for flexible video data model which can handle objects spatiotemporal dynamics.

C. *Contribution*

Inspired by works from computer vision, we aim to build expressive semantic representations of video data which will enable CEP engines to reason over incoming media streams. We model videos as streams of *time-evolving graphs,* where nodes and relationships change with respect to temporality and space. The main contribution of this work is as follows:

1. Video Event Knowledge Graph (VEKG), a flexible semantic representation of video streams expressed as a knowledge graph. VEKG acts as an intermediate bridge between unstructured video data and human level semantics.
2. A video event extraction method which captures detailed semantics of VEKG by modelling objects and their relationship interaction with each other.
3. VEKG-Time Aggregated Graph (VEKG-TAG), an aggregated representation for VEKG with 2.3X faster search with limited construction overhead.
4. Spatiotemporal event pattern rules to show the efficacy of video pattern detection over VEKG in CEP environment.

The rest of the paper is organized as follows. Section 2 presents the background. Section 3 discusses the proposed VEKG representation and extraction approach, while Section 4 focusses on formal concepts for relationship and aggregation. Section 5 presents operators with event rules. Section 6 shows experimental evaluation, while Section 7 discusses the related work. Section 8 concludes the paper and discusses future work.

II. PRELIMINARIES

This section conceptualises the initial background and techniques which are required for video stream representation.

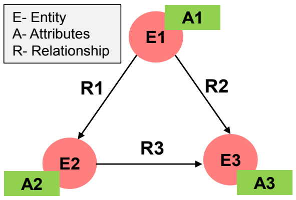


Fig. 2. Knowledge graph structure

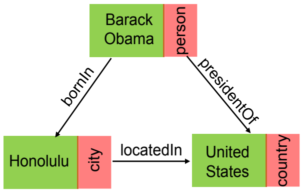


Fig. 3 Knowledge graph example

A. *Knowledge Graphs*

Knowledge Graph (KG) represents knowledge in graph form and captures entities, attributes and their relation in nodes and edges, respectively [6]. Entities relate to things which exist in real-world and have an independent existence. Attributes are the characteristics and properties of an entity, such as color and type. Within this context, Fig. 2 and 3 shows a KG structure with a simple example where person (E1) with name (A1) Barack Obama was born in (R1) city (E2) Honulu (A2) and was the president of (R2) of country (E3) United States (A3). Here the edges (R1,R2,R3) are typed relationship with high-level semantic meaning like *bornIn, presidentOf* and *locatedIn*. Some example of famous KG's are IBM Watson, Google Knowledge Vault and Facebook Graph API.

B. *Image Understanding and Object Detection*

The image understanding domain focuses on reasoning over image content to describe the image using high-level human-understandable concepts. In computer vision, the semantic representation of these high-level visual concepts are called objects (e.g. car, person). Algorithms from the vision literature can detect objects from images such as SIFT [7] (Fig. 4), and HOG [8]. Recently, Deep Neural Networks (DNN) [9] have become a state-of-the-art method to detect objects with good levels of accuracy and performance. DNN's are a supervised learning method, where a model is trained using annotated training data to detect the presence or absence of an object in the given image. DNN-based object detection models like YOLO [10] and M-RCNN [11] (Fig. 5) provides bounding boxes and segmented boundaries across the objects in the images.

Fig. 4 SIFT object detection

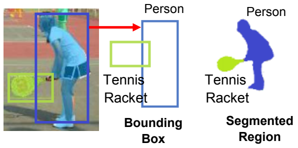


Fig. 5 YOLO and MRCNN object detection

III. Proposed Approach

Our approach to complex event pattern matching in video streams is as follows: i) First, we define the video event, ii) Second, we focus on the representation of the content of the video data streams, iii) Third, we detail on event extraction and aggregation method to represent videos, and iv) Fourth, we define event rules for video event detection.

A. *Video Event Definition*

The user perceives a video event as a high-level semantic concept observed in the change of state in video content over time [12]. Using the CEP analogy, we have defined two categories of video events:

Simple Video Event: In CEP, a simple event is the instantaneous and atomic (i.e. either exists entirely or not at all) occurrence of interest at a specific time instance [13]. We have extended this notion of the simple event for videos. Since objects are the primary visual concepts which a user can perceive from a video sequence. A Simple Video Event can be considered as an occurrence of any object which a user can identify from the video. If a user queries about the presence or absence of objects (e.g. 'car', 'person') in a video, then we consider it as a Simple Video Event.

Complex Video Event: In CEP, complex events are considered as *composed* or *derived* events which are constructed from simple events [14]. The simple events are nested with different temporal and logical operators to form a complex event. Similarly, a Complex Video Event can be built using spatial, temporal and logical operations using simple video events. For example, *high traffic flow* in a video is a complex video event which is made from simple video events such as the presence of *cars* and their count at a specific location for a given time.

B. *Video Event Representation*

Representing semantic information from video streams is a challenging task. Object detection techniques are not enough to define the complex relationships and interactions among objects and thus limits their semantic expressiveness. Videos comprise a sequence of consecutive image frames and can be considered as a data stream, where each data item represents a single image frame. These image frames have no fixed data model and need to be converted into suitable representation to be processed by the CEP engine.

We propose an object-centric representation using entity-centric Knowledge Graphs (KG). Graph-based representation for the video stream is suitable as it fits the following characteristics:

- *Scalable:* can capture multiple and diverse video objects and attributes information occurring at different time instances.
- *Complex Relationship:* can capture interaction among video objects as spatiotemporal relationships which can later be inferred as a high-level event like *high traffic volume* using event rules.
- *Maintains Hierarchy:* can handle information at different hierarchies ranging from low-level image features to their semantic mapping like object, scenes etc.
- *Semantically Queryable:* can apply event rules and define pattern-matching operations over the data.

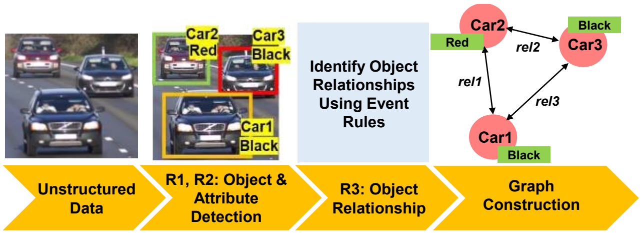


Fig. 6 VEKG extraction process

We have aligned the KG construction process with the video representation requirements (*R1, R2, R3*) listed in Section I-B. As shown in Fig. 6, the representation process is divided into two aspects- 1) Objects and Attribute Detection, and 2) Relationships among Objects.

Objects and Attribute Detection (R1 & R2): Following KG extraction, we perform object and attribute detection for video frames. A machine interprets a video frame using low-level visual features (e.g. pixels, intensity) while users perceive them as human-understandable concepts, i.e. Objects such as 'Car'. These objects can have multiple characteristics and properties which are represented as its attributes (e.g. color, type). Fig. 6 shows the extraction process for the image where *car* objects with different color attributes (red and black) are extracted.

Relationships among Objects (R3): In a video, relationships among objects can exist across time and space. They can be classified as:

- *Relationship within a frame (Intraframe):* Within an image frame, objects occupy specific positions. Thus, a *spatial relationship* can be established among the objects. Fig. 6 shows the spatial relation (*rel1*, *rel2*, *rel3*) among three *car* objects
- *Relationship across frames (Interframe):* Across frames, objects interact with each other over time. Thus, temporal

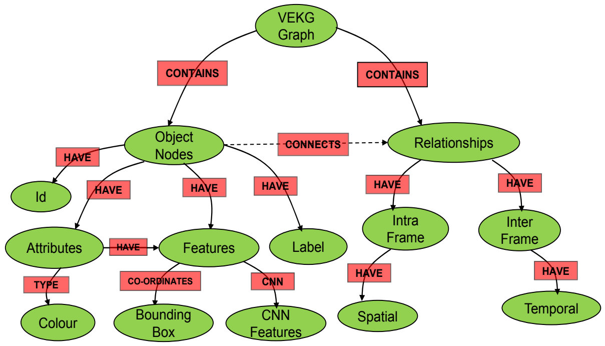


Fig. 7 Video Event Knowledge Graph (VEKG) scheme

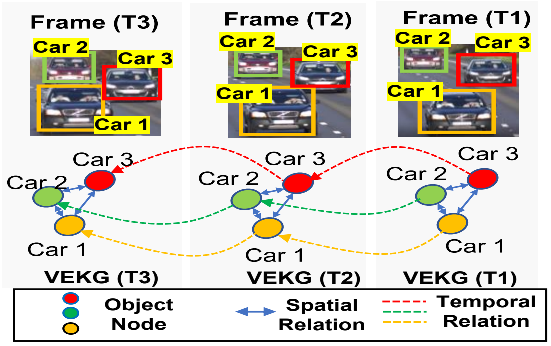


Fig. 8 VEKG graph stream for video

relationships can be established among objects across frames.

As discussed above the representation should be able to handle the object's spatiotemporal information at the frame level (intraframe) and stream level (interframe). Following this, a Video Event Knowledge Graph (VEKG) representation is proposed, where nodes correspond to objects and edges represent spatial and temporal relationships among objects (Fig. 7). A VEKG can be defined as:

Definition1 (VEKG Graph): For any image frame, the resulting Video Event Knowledge Graph is a labelled graph with six tuples represented as:

$VEKG = \{\mathbf{V}, \mathbf{E}, \mathbf{Av}, \mathbf{R_E}, \boldsymbol{\lambda_v}, \boldsymbol{\lambda_E}\}$ where

$\mathbf{V}$ = set of object nodes $\boldsymbol{O_i}$

$\mathbf{E}$ = set of edges such $\mathbf{E} \subseteq \mathbf{V}\ \boldsymbol{X}\ \mathbf{V}$

$\mathbf{Av}$= set of properties mapped to each object nodes such that $\boldsymbol{O_i}$= (id,attributes, label, confidence,features)

$\mathbf{R_E}$= set of spatiotemporal relations classes

$\boldsymbol{\lambda_v}, \boldsymbol{\lambda_E}$ are class labelling functions -$\boldsymbol{\lambda_v}: \mathbf{V} \rightarrow \boldsymbol{O}$ and $\boldsymbol{\lambda_E}: \mathbf{E} \rightarrow \mathbf{R_E}$.

Definition2 (VEKG Graph Stream): A Video Event Knowledge Graph Stream is a sequence ordered representation of VEKG such that:

$VEKG(\boldsymbol{S}) = \{(\boldsymbol{VEKG^1}, \boldsymbol{t_1}), (\boldsymbol{VEKG^2}, \boldsymbol{t_2}) \dots\dots (\boldsymbol{VEKG^n}, \boldsymbol{t_n})\}$ where $\boldsymbol{t \epsilon\ timestamp}$ such that $\boldsymbol{t_i} < \boldsymbol{t_{i+1}}$.

Fig. 8 shows three VEKG graphs for image frames at different time instances. The object nodes (Car1, Car2, Car3) in VEKG graphs are connected using spatial edges. VEKG is a complete directed graph, which means that each object is spatially related to another object which is present in the image frame. Thus each image frame consists of $n(n-1)$ edges where $n \epsilon\ number\ of\ objects$. The edge weights between nodes are updated as per query rule and are discussed in Section V. The temporal relation edge between object nodes is created by identifying the same object nodes in different frames using object tracking.

C. Video Event Knowledge Graph Extraction Architecture

The VEKG event extraction architecture is a computer vision pipeline that receives the video streams from different publishers and converts it to a stream of VEKG graphs. Fig. 9 shows the CEP engine architecture for VEKG extraction and matching which constitutes of the following components:

Video Frame Decoder: Receives the raw video frames and processes them to low-level feature map using video encoders.

DNN Models Cascade: A pipeline of different DNN models (object detectors, attribute classifiers) pre-trained on specific datasets. The low-level feature map from the video frame decoder is passed to the object detector for detecting objects. The Region of Interest (ROI) [10] of detected object features are then passed to attribute classifier for attribute detection. Object tracking is performed to determine if objects in different frames are the same or not.

Graph Constructor: Constructs a timestamped graph snapshot for each frame. It receives the output from the DNN models and represents them as a graph based on the VEKG schema. The VEKG graphs are then pushed to the *Pattern Matcher* as input for pattern matching.

Pattern Matcher: CEP systems work over the concept of 'state' which is the discretized snapshot of the continuous stream. In CEP, *windows* capture the *state* and apply event rules to detect patterns over that state [15]. As per eq. 1 window can be defined as:

$$\boldsymbol{TIMEWINDOW} \boxplus\ (\boldsymbol{VEKG}(\boldsymbol{S}), \boldsymbol{t}): \rightarrow \boldsymbol{S'} \qquad \textbf{(1)}$$

In equation 1, $TIMEWINDOW \boxplus$ is applied over an incoming stream $VEKG(S)$ and gives a fixed subsequence S'. In pattern matcher, windows capture the number of image frames as VEKG graph and perform spatial and temporal operations.

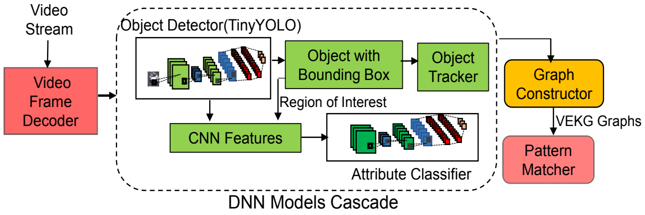


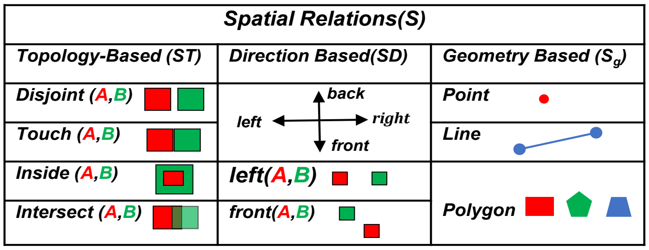
Fig. 9 VEKG extraction architecture

IV. SpatioTemporal Relation and Aggregation for Video Event Knowledge Graph

Video streams are continuous media with strict temporal and spatial relations. To define the relationship between various objects, spatial and temporal calculus is used, which are nested using logical and mathematical operators.

A. Spatial and Temporal Relations

Using spatial calculus, we have categorised spatial relations into three main classes:



Spatial Relations(S)		
Topology-Based (ST)	**Direction Based(SD)**	**Geometry Based (S_g)**
Disjoint (A,B) 🟥 🟩	back left right front	**Point** ●
Touch (A,B) 🟥🟩		**Line**
Inside (A,B)	**left(A,B)** 🟥 🟩	**Polygon** 🟥 ⬟ ⏢
Intersect (A,B)	**front(A,B)** 🟩 🟥	

Fig. 10 Spatial Relationships

- *Geometric Representation for Spatial Object (S_g):* Fig. 10 shows a spatial object can be represented using geometry-based features such as point, line and polygon. We use bounding box-based polygon to describe objects.
- *Topology-Based Spatial Relation (ST):* We use Dimensionally Extended Nine-Intersection Model (DE-9im), a 2-dimensional topological model to describe pairwise relationships between spatial geometries (S_g). The nine relationships its captures are: {*Disjoint, Touch, Contains, Intersect, Within, Covered By, Crosses, Overlap, Inside*} of which four are shown in Fig. 10.
- *Direction Based Spatial Relation (SD):* Direction captures the projection and orientation of an object in space. We use a simpler version of the Fixed Orientation Reference System (FORS) [16] which divides the space into four regions: {front, back, left, right}(Fig. 10).

For temporal modelling, we use Allen time-intervals [17]. Except for the spatial and temporal relation, we have used the logic operators {AND (∧), OR (V), NOT (¬), ANY (∃), EVERY (∀), NOR (↓) , XOR (⨁), XNOR (ʘ), Implies (⟶), Bi-Implies (⟷)}, mathematical and comparison operators {+,-,*, /, < >=} to model the relationships.

B. VEKG Aggregation

In videos, objects may exist for some time across multiple frames. Since objects are modelled as VEKG nodes, this leads to a high increase in the number of duplicate nodes which increase the VEKG construction and search time. To reduce this overhead, we propose the Time Aggregated Graph (TAG) [18] method over the VEKG stream. VEKG-TAG models time-series relationship across the edges of single aggregated graph to accommodate the time-varying object interactions. VEKG-TAG gives an aggregated view of a video state for a given time interval that preserves all required relationships. VEKG-TAG can be defined as:

Definition3 (VEKG-TAG): For a given time T, having n video frames represented as VEKG graph, the VEKG-Time Aggregated Graph is a labelled complete directed graph with 7 tuples such that VEKG-TAG $= \{\mathbf{V}, \mathbf{E}, \mathbf{Av}, \mathbf{R_E}, \mathbf{T}, \boldsymbol{\lambda}_\mathbf{V}, \boldsymbol{\lambda}_\mathbf{E}\}$. VEKG-TAG is similar to VEKG with an additional temporal dimension (T) adding to its edges in a single aggregated view. It requires $O(n^2T)$ memory to represent the VEKG stream of time T.

Fig. 11 shows a VEKG stream (left-side) and a VEKG-TAG (right-side) for time T1, T2 and T3 with a distance relationship

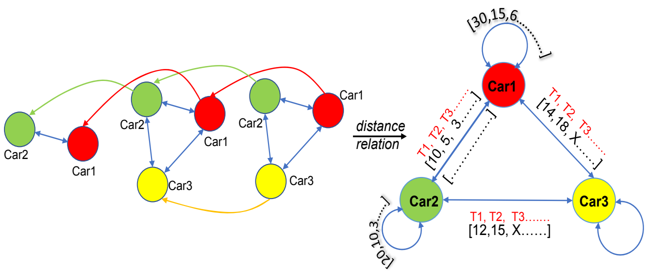


Fig. 11 VEKG stream and its aggregated form VEKG-TAG

for three car objects. VEKG-TAG shows unique object nodes (*car1, car2* and *car3*) and the distance among them over time T1, T2 and T3. The distance between *car1* and *car2* decreases over time, which means *car1* is approaching *car2*. The distance between *car2* and *car3* increases at T1 and T2, but since there is no car3 at time T3 it is represented by a don't care (X) condition. Each object node in VEKG-TAG has a *self-loop* which stores its initial position with respect to the image frame. This helps in capturing object dynamics such as an object is stationary or moving over time. Thus, VEKG-TAG consists of total of $[\,n(n-1)+n(self-loops)]$ edges which is equivalent to total n^2 edges. In next section, we introduce example event rules to reason over VEKG graphs.

V. Example Event Rules for Video Patterns

As discussed in the motivational scenario, some query operator rules have been proposed which act as a query for video event detection.

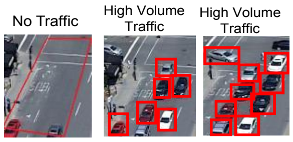


Fig. 12 High volume traffic

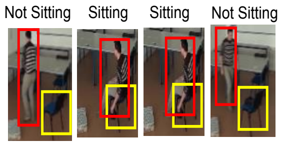


Fig. 13 Person sitting on chair

A. High Volume Traffic

The '*High Volume Traffic'* query rule is defined as: '*the average count of objects at a given space is greater than a certain threshold for a specific time range*'. For example, if the average number of cars is greater than 5 in every frame at a specific location of the road for more than 5 minutes, then we have a *high traffic volume* situation for that location. It can be defined as:

$$\exists\eta \in G \text{ and } \forall ti \in T\ if$$

$$\left(\boldsymbol{\mathcal{M}}(\mathbf{O})_{\boldsymbol{\eta}}^{\boxplus[\boldsymbol{t_1},\boldsymbol{t_2}]}\right) = \begin{cases} >\boldsymbol{r}\ \ \boldsymbol{traffic} \\ <\boldsymbol{r}\ \ \boldsymbol{not\,traffic} \end{cases} \tag{2}$$

$$where\ G\ is\ a\ space\ and\ T\ is\ time\ such\ that\ \mathcal{M} = Avg.COUNT, O = car\ \ and\ r \in \mathbb{Z}$$

In eq. 2, a spatial function $\mathcal{M}$ is applied which counts the average number of objects in every frame for a time window of ⊞ $[t_1, t_2]$ for a specific location (η).

B. Person Sitting on Chair

We define a *sitting* as 'if the overlap of a person and a chair is greater than some threshold for a given time (e.g. 10 sec) then we can say that the person is sitting on the chair. The sitting

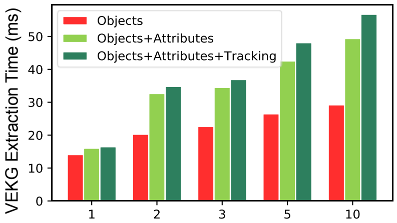


Fig. 14 VEKG extraction time from frames with different number of objects

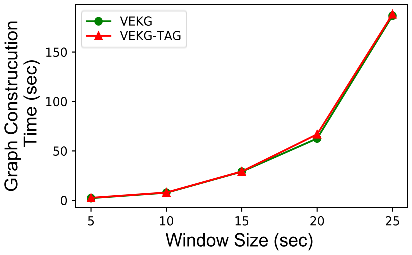


Fig. 15 Graph construction time with the change in window size

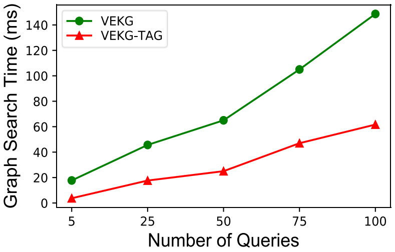


Fig. 16 Graph search time over multiple queries

rule can be written as:

$$[\boldsymbol{Overlap}\ (\boldsymbol{o_1}, \boldsymbol{o_2})]^{\boxplus[t_m, t_n]} > \boldsymbol{\alpha}\ \boldsymbol{where}\ \boldsymbol{\alpha} = \boldsymbol{overlap\ threshold}\ , \boldsymbol{o_1} = \boldsymbol{person\ and}\ \boldsymbol{o_2} = \boldsymbol{chair} \tag{3}$$

As per eq. 3 for time interval $[t1, t2]$ if object *chair* and *person* overlap value higher than $\boldsymbol{\alpha}$ then we can say that the *chair* is been occupied by the *person*.

Domain experts can develop intuitive event rules to define complex events facilitating video pattern detection. The above rules are converted to a query graph which is used to perform graph-based matching over VEKG graphs.

VI. Experimental Setup and Results

Table 1 Dataset and query specification

Video	Dataset	FPS	Query
P1	Pexels[1]	30.8	Q1: {Car}
P2	Pexels[1]	30.2	Q2: {Car ∧ color: black}
P3	YouTube	31	Q3: {High Traffic Volume (Car)}
P4	Le2i[2]	30	Q4: Sitting (Person ∧ Chair)

A. Implementation and Datasets

The prototype system is implemented in Java 8 and experiments were performed on a 16-core Linux machine running on 3.1 GHz processor, Nvidia Titan Xp GPU with 12 GB of RAM. For initial video preprocessing, we have used the Java OpenCV [19] library, and for video content retrieval Deeplearning4j [20] was used. For object detection, we have used DNN based YOLO [10] model. For attribute extraction, the features based on bounding box coordinates were fetched from YOLO model layer and passed to the attribute classifier, which is a simple color filter. JGraphT [21], a Java library for graphs was used for VEKG graph construction.

Table 1 shows a list of videos collected from different datasets. The videos were selected by visually analysing that a given pattern (e.g. high traffic flow) is present or not and at what instances. We crawled different videos related to different query operators and created the ground truth dataset manually. Table 1 also shows a list of event query patterns. The query pattern is listed as per their increasing complexity. In Q1 the subscriber is only interested in an object *car* while in Q2 the subscriber is interested in both object and its attribute (color: black). In Q3 and Q4 the subscriber has queried for a complex spatiotemporal pattern which the system will detect for the defined pattern rules. We performed different experiments on these queries using different publishers (video streams) to understand the efficiency of the proposed model.

B. VEKG Extraction Time

It is an initial pre-processing time to extract objects and attributes from the video frames. Eq. 4 shows the VEKG extraction time which is the time taken by video frame decoder ($t_{frame-decode}$) and DNN model cascade time ($t_{DNN-model}$) to extract objects and attributes. As the frame decoding time was very low (0.5-1 milliseconds (ms)), we have focussed only on time required by DNN models cascade.

$$t_{VEKG-extraction} = t_{frame-decode} + t_{DNN-model} \tag{4}$$

Fig. 14 shows the VEKG extraction time for video frames with different numbers of objects. We have focused on three key stages of the DNN model cascades, i.e. 1) Object detection time, 2) Object and attribute detection time, and 3) Object, attribute detection, and object tracking time. These three characteristics were compared with video frames having the number of objects ranging between 1 (F1) to 10 (F2). Fig. 9 shows the average object detection time lies between 14.1ms to 29.2 ms for F1 and F2. The difference between object detection time is very low because of the shared computation principle over which object detectors work [10]. The object and attribute average detection time for F1 is 16.03 ms, which increases to 49.3 ms for F2. This is because of the extra overhead where each object needs to be passed to the attribute classifier and so with an increase in object number the attribute classification time increases. Tracking is cheap process, thus including tracking time results in an overall detection time of 16.4 ms, and 56.7 ms for F1 and F2, respectively. The VEKG extraction time is one of the main bottlenecks in system performance due to the computationally intensive DNN models. This is an area of future work.

C. VEKG Graph Construction and Search Time

Graph construction is the time to create VEKG graphs for a given time window. This includes the time for creating nodes and edges relations as per the query rules. The construction process will be repeated *n* times for *n* number of registered queries. The graph search time is the time to search the event pattern as per the query rule. Fig. 15 shows the graph construction time for VEKG and VEKG-TAG for different time window sizes. The construction process for both graphs is nearly the same with a subsecond increase in VEKG-TAG. This is due to the reason that VEKG-TAG is aggregated over VEKG and the extra time it requires is for node and edge initialization only, as all others relationships are already calculated during the VEKG construction process. In Fig. 15 the graph construction time increases with the increase in window size as

[1] https://www.pexels.com/

[2] http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr

there will be more objects creating more nodes. For VEKG and VEKG-TAG, the construction time for a 5 second (sec) window was 2.2 and 2.57 sec, which increases to 186.7 and 188.2 sec respectively for 25 sec time window. Fig. 16 shows the search time for both methods for different numbers of queries. VEKG-TAG performs better in search as it is the summarized version of VEKG with non-redundant nodes and edges. For five queries the search time of VEKG and VEKG-TAG is 17.7 and 3.7 ms respectively. For 100 queries VEKG-TAG search requires only 61.7 ms as compared to VEKG which have search time of 148.6 ms respectively. Thus, VEKG-TAG search is nearly 2.3X faster for 100 queries and this performance will increase with the increase in number of queries. The performance shown here is under a worst-case scenario where all the nodes and edges were traversed for both graph methods. The search time, i.e. latency for each specific query is discussed in next section.

D. Event Query Accuracy and Latency

The event query accuracy examines how many relevant event patterns were detected for each query as compared to the ground truth. Query accuracy is evaluated using F-score (eq. 5), which is a harmonic mean of precision and recall. The precision is the ratio of $relevant\ events\ matched$ and $matched\ events$ while recall is the ratio of $relevant\ events\ matched$ and $relevant\ events$.

$$F - score = \frac{2 * Precison * Recall}{Precision + Recall} \qquad \textbf{(5)}$$

Table 2 shows the mean F-score for different queries which are averaged across the state time window of 5 sec. Object detection query (Q1) is run on two publishers, P1 and P2. The F-score of Q1_P1 is 0.80 is less as compare to Q1_P2 (0.89). This is because the number of objects in video P1 is high as compare to P2 leading to more occlusion thus reducing overall accuracy. The accuracy of the object and attribute detection query (Q2_P2) is 0.79 because of low accuracy of attribute classifier. The F-score of the *high traffic volume* query (Q3_P3) is 0.86 and high because it is just the overall count of object nodes and mainly depends on accuracy of state-of-the-art YOLO object detector. The sitting query (Q4_P4) has the least F-score of 0.75 because of more false positives. There were instances where a person was standing near a chair, and the bounding box overlap rule was determining it as a sitting event. To reduce such false positives there is a need for more complex event rules to deal with such use cases.

Fig. 17 shows the average processing time of each state for different query pattern. This is the time when the CEP engine receives the state and performs graph matching. For consistency, all the query patterns were run on the same video and the object nodes labels were replaced with the dummy values. The Q1 latency is higher as compared to Q2 and ranges between 0.5 ms to 1.4 ms respectively. This small rise in Q2 latency is because we are accessing object nodes and then its attribute values. Similar is the case of Q3 latency (0.9-2.7 ms) where a count function overhead is added after accessing the nodes. Since Q4 tries to extract the edges as sitting is a relation between two object nodes thus its latency is highest as compared to other queries and ranges from approximately 1.2 to 3.1 ms. Initial spike at the start is due to the extra time added in the state formation as the DNN models load into memory.

Table 2 Query accuracy

Query	Precision	Recall	F-Score
Q1_P1	0.90	0.72	0.80
Q1_P2	0.92	0.87	0.89
Q2_P2	0.86	0.73	0.79
Q3_P3	0.91	0.81	0.86
Q4_P4	0.80	0.71	0.75

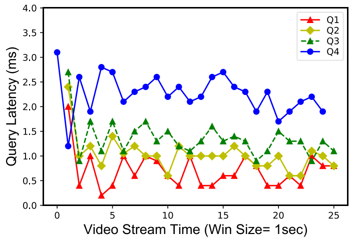


Fig. 17 Event latency for different queries

E. Limitations

Our work has some limitations which are as follows: 1) DNN models are basic building blocks for our CEP system, and any prediction failure in them will decrease VEKG representation quality. 2) We can only detect the event pattern of objects on which the DNN object detector is trained. 3)The spatial calculation was performed in the 2-dimensional plane while in the real-world, the relations are complex and spread in 3-dimensions, leading to many patterns misses.

VII. Related Work

A. Multimedia Event Representation

Initially, Westermann et al. [22] proposed an E event model and discussed high-level characteristics which a multimedia applications should possess. IMGpedia [23] added low-level features of the image to create a linked dataset of images, but it does not capture semantic relationships among them as shown in VEKG. In OVIS [24], the authors have developed a video surveillance ontology for large volumes of the video in databases while VEKG representation can be deployed both in database and streaming scenario. Xu et al. [25] present a Video Structural Description (VSD) technology for discovering semantic concepts in the video with no CEP focus. MSSN-Onto [26] focuses on event schema for multimedia sensor networks with visual descriptors, motion descriptor, spatial and temporal(camera duration) aspects instead of high-level semantic concepts and relationships in videos. SPARQL-MM [27] defines events in terms of spatial(point, line, shape) and temporal, thing(instant, interval). In [28] representation of videos was limited to discussion of physical objects like id, 2d positions, minimum bounding box, but there was no discussion on relationships of objects. Yadav et al. [29] focused on pattern detection like ‘wildfire’ from images in CEP using crowd knowledge without focusing on schema representation. Jain et al. [30] focused on complex event detection in a multimedia communication system, but they assumed event as a high-level entity without any multimedia content extraction. Zaarour et al. [31] focused on filtering visual content such as images in distributed publish-subscribe systems with no focus on complex pattern matching.

B. Visual Relation Detection

Visual relation detection techniques like Scenegraph [32] work on static data such as images where relationships are annotated among objects manually [33], and then the model is trained to detect pattern relationship among objects in the images. VEKG, on the other hand, detects relation among objects over space and time using event rules. Lee et al. proposed Region Adjacency Graphs (RAG) [34] for videos where the same segmented regions within the image frames are connected using common boundaries. Instead of focusing on low-level features like in RAG, VEKG is built over high-level semantic labels (objects) extracted from DNN models capturing spatiotemporal relation among them. In [35], the authors use neural network and capture the relationship among objects in a video, where relationships were encoded in the training data manually and later trained to predict relation. Recently, Herzig et al. [36] proposed a Spatio-Temporal Action Graph (STAG) to identify collision events using an end-to-end deep learning model. VEKG is a more generalized version of STAG and can be used both in DNN and rule-based models and can handle multiple relationships within a single representation.

C. Graph Aggregation

George et al. proposed TAG [19], an aggregated data model for spatiotemporal networks. VEKG-TAG is an addition to the above work where we used it as an aggregation method over VEKG streams for a given time instance for detecting video event patterns. Kwon et al. [37] detect rare events in videos using a graph editing framework. They decompose video into a graph where a node represents a spatiotemporal event and have connected edges to its neighbors. In contrast, VEKG captures more detailed video information where each frame is initially a graph of objects with spatial information, which is then aggregated to VEKG-TAG over temporal dimension for different queries. Adhikari et al. proposed NETCONDENSE [38], which merges adjacent node-pair and time-pair for time-varying graph. The time-pair merge loses initial edge information which is preserved in VEKG-TAG summary.

VIII. Conclusion and Future Work

We present VEKG a knowledge graph driven representation of video streams. VEKG enhances CEP systems capability to detect patterns from video streams, enabling visual semantic queries in CEP. The paper details the design, extraction and online construction approach for VEKG. We detail VEKG-TAG an aggregated method for VEKG streams which perform 2.3X faster query execution within the construction bounds. The paper sheds light on different spatial and temporal constructs to detect video events from the video streams using different event query rules with good F-scores (0.75-0.89) and sub-second matching latency (0.5-3.1 ms). Future extension of this work will focus on different optimization techniques to improve VEKG extraction process. Next, we will focus on event enrichment techniques by leveraging the VEKG structure to improve event detection capability.

Acknowledgement

The work is supported, in part, by Science Foundation Ireland grant 13/RC/2094.

References

[1] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, “Internet of multimedia things: Vision and challenges,” Ad Hoc Networks, 2015.

[2] S. Hasan and E. Curry, “Tackling Variety in Event-based Systems,” 9th ACM Int. Conf. Distrib. Event-Based Syst. (DEBS 2015), 2015.

[3] A. Demers, J. Gehrke, M. Hong, M. Riedewald, and W. White, “Towards Expressive Publish/Subscribe Systems,” in EDBT, 2006.

[4] A. Aslam and E. Curry, “Towards a Generalized Approach for Deep Neural Network Based Event Processing for the Internet of Multimedia Things,” IEEE Access, vol. 6, pp. 25573–25587, 2018.

[5] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” ACM Comput. Surv., 2003.

[6] F. Van Harmelen, V. Lifschitz, and B. Porter, Handbook of knowledge representation. Elsevier, 2008.

[7] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” in ICCV, 1999.

[8] C. Review, C. Do, and N. O. T. Distribute, “Fast human detection using a cascade of Histograms of Oriented Gradients,” in CVPR, 2006.

[9] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature. 2015.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in IEEE CVPR, 2016.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in ICCV, 2017.

[12] J. R. Smith, R. J. Alexandre, J. Hobbs, R. C. Bolles, and J. R. Smith, “Standards VERL : An Ontology Framework for Representing and Annotating Video Events,” IEEE Multimed., 2005.

[13] E. Wu, Y. Diao, and S. Rizvi, “High-performance complex event processing over streams,” in ACM SIGMOD, 2006.

[14] G. Cugola and A. Margara, “Processing flows of information,” ACM Comput. Surv., vol. 44, no. 3, pp. 1–62, 2012.

[15] A. Arasu and S. Babu, “The CQL continuous query language: semantic foundations and query execution,” VLDB J., 2006.

[16] D. Hernández, “Relative Representation of Spatial Knowledge: The 2-D Case,” in Cognitive and Linguistic Aspects of Geographic Space, 1991.

[17] J. F. Allen, “An Interval-Based Representation of Temporal Knowledge,” in IJCAI, 1981.

[18] B. George and S. Shekhar, “Time-Aggregated Graphs for Modeling Spatio-temporal Networks,” J. Data Semant. XI, 2008.

[19] “OpenCV Java.” [Online]. Available: https://bit.ly/2Un7GgI.

[20] “Deeplearning4j.” [Online]. Available: https://deeplearning4j.org/.

[21] “JGraphT.” [Online]. Available: https://jgrapht.org/.

[22] U. Westermann and R. Jain, “Toward a common event model for multimedia applications,” in IEEE Multimedia, 2007.

[23] S. Ferrada, B. Bustos, and A. Hogan, “IMGpedia: A linked dataset with content-based analysis of wikimedia images,” in ISWC, 2017.

[24] M. Y. Kazi Tani, A. Ghomari, A. Lablack, and I. M. Bilasco, “OVIS: ontology video surveillance indexing and retrieval system,” Int. J. Multimed. Inf. Retr., vol. 6, no. 4, 2017.

[25] C. Hu, Z. Xu, Y. Liu, and L. Mei, “Video structural description technology for new generation video surveillance systems,” Front. Comput. Sci, 2015.

[26] C. Angsuchotmetee, R. Chbeir, and Y. Cardinale, “MSSN-Onto: An ontology-based approach for flexible event processing in Multimedia Sensor Networks,” Futur. Gener. Comput. Syst., 2018.

[27] T. Kurz, K. Schlegel, and H. Kosch, “Enabling Access to Linked Media with SPARQL-MM,” in WWW, 2015.

[28] T. Lan Le, M. Thonnat, A. Boucher, F. Bremond, T.-L. Le, and F. Brémond, “A Query Language Combining Object Features and Semantic Events for Surveillance Video Retrieval,” in MMM, 2008.

[29] P. Yadav, U. U. Hassan, S. Hasan, and E. Curry, “The Event Crowd: A novel approach for crowd-enabled event processing,” in DEBS, 2017.

[30] M. Gao, X. Yang, R. Jain, and B. C. Ooi, “Spatio-temporal event stream processing in multimedia communication systems,” in SSDM, 2010.

[31] T. Zaarour and E. Curry, “Adaptive Filtering of Visual Content in Distributed Publish/Subscribe Systems,” in NCA, 2019.

[32] J. Johnson et al., “Image Retrieval using Scene Graphs,” in CVPR, 2015.

[33] R. Krishna et al., “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations,” in IJCV, 2017.

[34] J. Lee, J. Oh, and S. Hwang, “STRG-Index: Spatio-Temporal Region Graph Indexing for Large Video Databases,” in ACM SIGMOD, 2005.

[35] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua, “Video Visual Relation Detection,” in ACM Multimedia - MM ’17, 2017.

[36] R. Herzig, E. Levi, H. Xu, E. Brosh, A. Globerson, and T. Darrell, “Classifying Collisions with Spatio-Temporal Action Graph Networks,” in arXiv preprint,arXiv:1812.01233, 2018.

[37] J. Kwon and K. M. Lee, “A unified framework for event summarization and rare event detection,” CVPR, 2012.

[38] B. Adhikari, Y. Zhang, A. Bharadwaj, and B. A. Prakash, “Condensing Temporal Networks using Propagation,” in SIAM, 2017.