Instantiation of a textual description schema of video surveillance scenes

Wael Farid Youssef

Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :

Wael F. Youssef

le 24 Septembre 2019

Titre :

Instanciation d'un schéma de description textuel de scènes de vidéo surveillance

---

École doctorale et discipline ou spécialité :

ED MITT : Image, Information, Hypermedia

Unité de recherche :

IRIT

Directeur/trice(s) de Thèse :

Philippe Joly	Directeur	Prof., UT3 Paul Sabatier
Siba Haidar	co-directeur	Dr., Universitee Libanaise

Jury :

Sergio Velastin	Rapporteur	Prof., Université Carlos III de Madrid
Nicolas Henri	Rapporteur	Prof., University of Bordeaux
Philippe Joly	Directeur	Prof., UT3 Paul Sabatier
Siba Haidar	Co-Directeur	Dr., Université Libanaise

THESIS

For obtaining

PHD from University of Toulouse

Proposed by:

University of Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Prepared and presented by:

Wael F. Youssef

Date…

Title:

Instantiation of a textual description schema of video surveillance scenes

---

Specialty:

ED MITT: **I**mage, **I**nformation, **H**ypermedia

Unity of research:

IRIT

Thesis supervisor(s):

Philippe Joly	Supervisor	Prof., UT3 Paul Sabatier
Siba Haidar	Co-Supervisor	Dr., Lebanese University

Committee:

Sergio Velastin	*Rapporteur*	*Prof., University of Carlos III de Madrid*
Nicolas Henri	*Rapporteur*	*Prof., University of Bordeaux*
Philippe Joly	*Supervisor*	*Prof., UT3 Paul Sabatier*
Siba Haidar	*Co-Supervisor*	*Dr., Lebanese University*

Abstract

Surveillance systems are important tools for law enforcement agencies for fighting crimes. Surveillance control rooms have two main duties: live monitoring the surveillance areas, and crime solving by investigating the archives. To support these difficult tasks, several significant solutions from the research and market fields have been proposed. However, the lack of generic and precise models for video content representation make the building of fully automated intelligent video analysis and description system a challenging task. Furthermore, the application domain still shows a big gap between the research field and the real practical needs, it also shows a lack between these real needs and the on-market video analytics tools. Consequently, in conventional surveillance systems, live monitoring and investigating the archives still rely mostly on human operators.

This thesis proposes a novel approach for textual describing important contents in videos surveillance scenes, based on new generic context-free "VSSD ontology", with focus on two objects interactions. The proposed ontology presents a new generic flexible and extensible ontology dedicated for video surveillance scenes description. While analysing and understanding variety of video scenes, our approach introduces many new concepts and methods concerning mediation and action at a distant, abstraction in the description, and a new manner of categorizing the scenes. It introduces a new heuristic way to discriminate between deformable and non-deformable objects in the scenes. It also highlights and exports important features for better video objects interactions learning classifications and for better description. These features, if used as key parameters in video analytics tools, are much suitable for supporting surveillance systems operators through generating alerts, and intelligent search.

Moreover, our system outputs can support police incidents reports, according to investigators needs, with many types of automatic textual description based on new well-structured rule-based schemas or templates.

Additionally, in this thesis, many important propositions were made, driven by practical experience, to reduce the existing gaps between the surveillance systems operators' needs from one side, the research field and the commercial (industry) field from the other side. These propositions encounter the research field, and the practical one, especially at the level of future intelligence video analytics development and integration with other systems. Some of these propositions are innovative yet simple to be applied, which can bring great benefits and optimize the use for surveillance systems operators when live monitoring, investigating, and analysing the crimes.

Résumé

Les systèmes de vidéosurveillance sont des outils importants pour les agences chargées de l'application de la loi dans la lutte contre la criminalité. Les chambres de contrôle de la vidéosurveillance ont deux fonctions principales : surveiller en direct les zones de surveillance et résoudre les infractions en enquêtant les archives. Pour soutenir ces tâches difficiles, plusieurs solutions significatives issues des domaines de la recherche et du marché ont été proposées. Cependant, le manque de modèles génériques et précis pour la représentation du contenu vidéo fait de la construction d'un système intelligent et automatisé capable d'analyser et de décrire des vidéos une tâche ardue. De plus, le domaine d'application montre toujours un écart important entre le domaine de la recherche et les besoins réels, ainsi qu'un manque entre ces besoins réels et les outils d'analyse vidéo dans le marché. Par conséquence, jusqu'à présent dans les systèmes de surveillance conventionnels, la surveillance en direct et la recherche dans des archives reposent principalement sur des opérateurs humains.

Cette thèse propose une nouvelle approche pour la description textuelle de contenus importants dans des scènes de vidéosurveillance, basée sur une nouvelle «ontologie VSSD» générique, sans contexte, centrée sur les interactions entre deux objets. L'ontologie proposée est générique, flexible et extensible, dédiée à la description de scènes de vidéosurveillance. Tout en analysant les différentes scènes vidéo, notre approche introduit de nombreux nouveaux concepts et méthodes concernant la médiation et l'action distante, la description synthétique, ainsi qu'une nouvelle façon de segmenter la vidéo et de classer les scènes. Nous introduisons une nouvelle méthode heuristique de distinction entre les objets déformables et non déformables dans les scènes. Nous proposons également des caractéristiques importantes pour une meilleure classification des interactions entre les objets vidéo, basée sur l'apprentissage, et une meilleure description. Ces caractéristiques, si elles sont utilisées comme paramètres clés dans les outils d'analyse vidéo, sont bien adaptées pour aider les opérateurs de systèmes de surveillance à travers des générations d'alertes, et une recherche intelligente.

De plus, nos sorties système peuvent prendre en charge les rapports d'incidents de police, selon les besoins des enquêteurs, avec de nombreux types de descriptions textuelles automatiques basées sur de nouveaux schémas ou modèles, bien structurés et basés sur des règles.

Enfin, dans cette thèse, de nombreuses propositions importantes ont été faites, s'appuyant sur l'expérience pratique, pour réduire les écarts existants entre les besoins des opérateurs de systèmes de surveillance d'un côté, le domaine de la recherche et le domaine commercial (de l'industrie) de l'autre côté. Ces propositions engagent le domaine de la recherche et le domaine pratique, en particulier au niveau du développement futur des produits intelligents d'analyse de vidéos et de l'intégration avec d'autres systèmes. Certaines de ces propositions sont novatrices mais simples à appliquer, ce qui peut apporter d'importants avantages et optimiser l'utilisation par les opérateurs de systèmes de surveillance lors du suivi en direct, de l'enquête et de l'analyse des crimes.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

I. Introduction

In an era of rapid technological development on many fields (social, political, economic, security, …), and the accompanying changes of the crime aspects and its methods, getting more sophisticated by the day, the law enforcement agencies duties towards their citizens are becoming more and more difficult to enforce. These duties revolve around the protection of persons, property and freedom, the maintaining of order and the strengthening of security, public safety and the application of laws and regulations. Thus, law enforcement agencies, are in continuous search for effective public safety and security strategies to help deal with criminal and terrorist acts.

The new face of dealing and fighting with crimes is through collecting data, and transforming this data into intelligence. Among the most modern public safety and law enforcement tools is the surveillance system, where the video surveillance is a big source of data and the strong point of the most investigations.

The rapid progress in technology, Multiplexing, digital technology, NVRs, storage and processing made enormous progress in surveillance systems. For that, the deployment of video surveillance systems worldwide has grown exponentially in recent years. Many of large cities have concerns about crimes, terrorist attacks, incidents and antisocial behaviour problems, such as fights, vandalism, breaking and accidents, often these cities have video cameras already installed in the streets and around the important sites. Visual surveillance is now used to monitor the security of sensitive areas, as a risk management and crime reduction tool, such as in public places, schools, banks, shopping malls, transport infrastructures (e.g. airports, underground stations), hospitals, government buildings and borders.

One of the most important duties and goals of surveillance systems is to **live monitoring** the surveillance areas, in case incident occurs, actions should be taken. Another main focus is crime solving by investigating the archives. Two tasks are difficult to achieve, due to lack of human resources for active monitoring and of accurate parameters concerning archive indexation. Most, video surveillance recordings are indexed with rough descriptors such as time, camera ID and some photometric parameters.

Surveillance systems produce large amounts of video data which are stored for immediate or future use. Years of video surveillance are recorded. A crucial need is to make sense of this massive quantity of visual data. Surveillance videos are generous in motion information which rises up as one of the most important clue to identify the dynamic content of videos. Extraction and analysis of motion information in videos are essential in content-based surveillance video analysis and understanding. Detecting and understanding an incident is a simple mission for human, but it is very complicated for machine.

There is a fundamental need to extract automatically meaningful content and produce high-level scheme or descriptions of the activities. Such a system can help effectively to generate alerts to assist the live monitoring, and can help intelligently to index, organize, and retrieve valuable information from surveillance video databases, and finally to automatically generate useful reports.

Several significant solutions from the research and market fields have been proposed. However, the lack of generic and precise models for video content representation make building of fully automated intelligent video analysis and description a challenging task. This lack is due to the high complexity of video scenes, and its diversity from the

context to the objects and actions types. Furthermore, for these reasons and many others, the application domain still shows a big gap between the research field and the real needs, also show a lack between these real needs and the on-market video analytics tools. Unfortunately, till now, both tasks of crime solving and live monitoring in conventional visual surveillance systems rely mostly on human operators; either to dig hard threw hundreds of hours in the archives, or to monitor actively hundreds of cameras.

This thesis proposes a novel approach for textual description of the video scenes. We claim it to be a new approach for a general knowledge-based context-independent applicable in real-world surveillance video. Our approach is based on a proposed ontology, which combines objects features and derive high-level information; our ontology is methodologic and easily expendable.

From the perspective of an experienced Major, head of CCTV control room, the main two concerns of the approach were, to: first automatically extract useful information for investigating the archives and setting up alerts in real-cases, and second, to present it in an understandable way for the system operators by proposing new sentence representations. For that, well-structured schemas can be applied to generate incident scene descriptions similar to ones used in police reports. These schemas are also called templates.

"On Friday 17/05/2019 at 15:06:39: A person "2" moves, in intersection spot "Hamra-Rome" (33.895245, 35.487536), on the left of "Hamra" street, heading immediately north, toward the person "1", occurring respectively irregularity in its shape, and big changes occurring respectively on its surface having now smaller one, and having respectively considerable decreasing of its Speed.

"The two objects are approaching; a distant aggressive interaction occurs between them."

Example of scene description and of object interaction description.

A description as the one in the above example can be very helpful for the surveillance system operators. It is based on many useful parameters such as objects characteristics, and inter-objects parameters. Those characteristics can be set up to generate alerts. They also can be queried for, in the archives.

This thesis also highlights the existing gap between the research field and the surveillance system operators' needs from one side and the gap between the latter one with the commercial (industry) field. Many propositions were made driven by my personal experience as a researcher and a CCTV Control Room manager.

The research works presented in this thesis have been conducted under high constraints of applicability in a professional context. They took place in the framework of cooperation between IRIT Laboratory in the University of Paul Sabatier – Toulouse, France, and the Lebanese university, Beirut, Lebanon, and Beirut CCTV control room, Lebanon.

I.1. State of the art

Understanding and textually describing a video scene is an easy task for most humans, but is still a complex task to the computer. Automatic video scene description includes understanding and differentiating between the multiplicity of the backgrounds, the objects, the interactions, the scenes types, and the temporal order of incidents and events. Moreover, it requires a translation of the information into a comprehensible textual description or what is known as natural language.

The textual description, in general, can be used to improve wide range of applications like human-robot interaction, scene descriptor for blind, summary of (web-) videos, medical diagnosis, surveillance systems, robotics, military systems and others. In specific, for surveillance systems and traffic surveillance in cities, generating alerts for the observers and intelligently investigating the archives for the investigators.

In the last decades, researchers have studied multiple strategies and ontologies to bridge the gap existing between visual content and textual description. For that, Computer vision and Natural language Processing (NLP) fields addresses such a problem, separately and also, some workshops have been held on both areas (Andrei et al., 2018).

Being stretched, over more than two decades, and having so many applications have made this area of research very wide. Therefore, it's quite challenging to recapitulate and categorize all the works done in this area, especially as each contribution might differ according to the needs, outputs, methodologies, automation degree, used methods, and even sometimes trends.

In addition, researches in the related fields to video description like connecting words to pictures, image captioning, video to text, narrating images in natural language sentences, video captioning, video summarization, behaviour descriptions, natural-language video descriptions, and visual recognition and description, may share common methods or follow similar methodology.

Not restricted to video surveillance, two main approaches can be noticed:

1- **Behaviour understanding and sentences generation** (Barbu et al., 2012) (Thomason, Venugopalan, Guadarrama, Saenko, & Mooney, 2014) (Guadarrama et al., 2013), (A. Rohrbach et al., 2014), (R. Xu, Xiong, Chen, & Corso, 2015)**:** this approach's name was chosen because most of the researches following this approach are mainly focusing on two stages:

   a. Behaviour understanding and content identification: also known as Knowledge-based or deterministic approaches. Extracting all needed features for identifying the semantic content and understanding the behaviour and the scene. A variety of approaches, techniques and methods have been proposed, we mention, object detection and segmentation, object classification, object recognition, multi-object tracking, trajectory analysis, action analysis, activity classification and recognition, and others. Typically, this part may involve training individual classifiers to identify background, objects, and actions in the scenes. As our main approach will focus on extracting many of the features, we will furthermore discuss the state of art of each of the corresponding methods in chapter IV.

   b. Sentence generation: generating a sentence, normally, based on a template with syntactical structure (like Subject-Verb-Object SVO tuples, place, and scene). It may

uses also a probabilistic model to map the most important visual content results from the video for each of the template categories to generate a sentence.

Some approaches from the literature are explained in the next section:

- In (S. Park & Aggarwal, 2004), the authors present a method to describe two-person interactions in a semantic NL description. For that they detect body posture after integrating individual body parts (head, torso, arms and legs) recognized using Bayesian networks (BNs). To recognize specific interactions, they used decision tree with rule-based spatial and temporal constraints. Then they map it into verb phrases using sequential and simultaneous recognitions of the predefined interactions. Human interactions are then represented as cause–effect semantics between syntactical <agent-motion-target> triplets.
- In (Farhadi et al., 2010), the authors proposed a system based on the detection of one object per image, then map it to the corresponding textual descriptions using a predefined language templates (triplet of S-V-O).
- In (Barbu et al., 2012), the authors use a dynamic programming approach combined with Hidden Markov Models to obtain verb labels for short video clips, for producing sentential descriptions.
- In (Guadarrama et al., 2013), the authors used semantic hierarchies to indicate the appropriate level of the accuracy and specificity of sentence fragments.
- In (M. Rohrbach et al., 2013), the authors incorporated semantic unaries and hand-centric features and utilized a CRF-based approach to generate video descriptions. Their method is composed mainly of two steps; first to generate semantic representation models, they feed a Conditional random field CRF using dense trajectories and SIFT features and temporal context reasoning. Second they translate it to natural language using Statistical Machine Translation (SMT).
- In (R. Xu et al., 2015), for video sentence generation the authors designed a deep joint video-language embedding model.
- In (Hanckmann, Schutte, & Burghouts, 2012), the authors proposed a hybrid method to generate textual descriptions of video actions. Their system has mainly two parts, an action classifier and a description generator. They detect and classify 48 actions in a video using the Bag-of features. The description generator, a rule-based method, finds the actors (persons or objects) in the video and connects these to the appropriate verbs.
- Krishnamoorthy et al. (Krishnamoorthy, Malkarnenkar, Mooney, Saenko, & Guadarrama, 2013) they were the first to introduce early works of describing open domain short videos data (YouTube videos). They used knowledge mined from webscale text copora to determine the best likelihood of various combinations of subject-verb-object triplets. They use a template-based approach to present the textual description, as: “Determiner (A, The) - Subject - Verb (Present, Present Continuous) - Preposition (optional) - Determiner (A, The) - Object.” They evaluate the system automatically and by human evaluation.

Another interesting works were presented by (Guadarrama et al., 2013), (Das, Xu, Doell, & Corso, 2013).

2- **Sequence learning approach** (A. Rohrbach et al., 2017), (Jeff Donahue et al., 2017), (J. Xu, Mei, Yao, & Rui, 2016), (Venugopalan et al., 2014): Known also as deep learning, probabilistic or data-driven approaches. This approach directly learns to map between

video content and textual sentence. This approach can be mainly divided into two stages:

a. Video encoding stage: also known as Visual recognition where the visual features are directly extracted, more accurately, learnt, using different types of deep neural network algorithm, like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM). The produced result composes a fixed or dynamic real-valued vector.
b. Video decoding stage: also known as the sequence generation or text generation, where the vector result of the first stage is fed for text generation, as single or multiple sentences. For decoding, first RNN were used, RNN is a neural network adding extra feedback connections to feed-forward networks, enabling it to work with sequences of inputs. Then, the network is updated grounded on every input item and the preceding hidden state. They are networks with loops that allow persevering information. These networks, mainly, have been used in many fields such as speech recognition, language modelling, image captioning, translation and more. Different types of deep neural network are now in use, most commonly, deep RNN, Bi-directional RNN, Long Short-Term Memory LSTM, Gated Recurrent Units (GRU) or others.

In general, a sequence learning approach eludes the two steps of content identification and sentence generation by learning to match directly videos frames to human sentences. Different combination of encoding-decoding algorithm may be used, to mention CNN–RNN, RNN–RNN, and deep reinforcement networks. An example of a common architecture for video captioning using sequence learning approach is given in Figure I-1, where 2D or 3D CNNs are exploited, on a video sequence, to extract features on optical flow images, video frames, or others… The video-level representations are then produced by mean pooling or soft attention over these visual features. Then, on the level of representations, an LSTM is trained for generating a sentence.

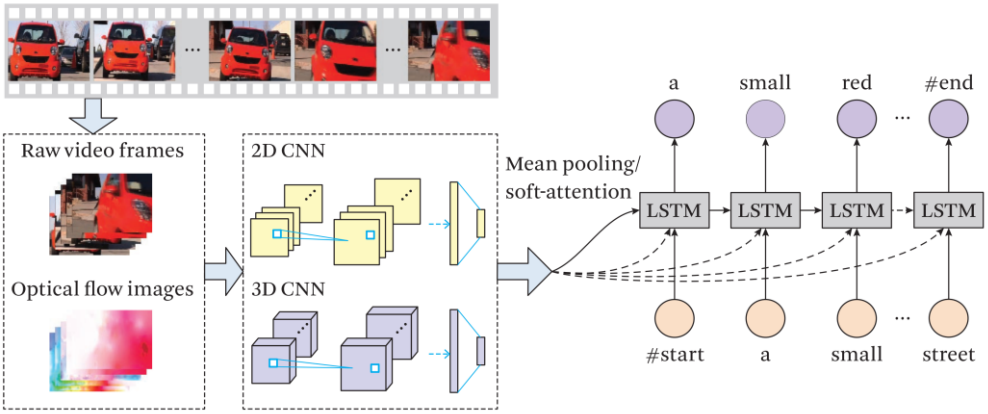


Figure I-1: An example of video captioning architecture using sequence learning approach, taken from (Z. Wu, Yao, Fu, & Jiang, 2017)

Examples of some important works on sequence learning approach:

- Some of the researches on template-based video representation used statistical machine translations (Jeffrey Donahue et al., 2015), (Barbu et al., 2012), (Atsuhiro Kojima, Tamura, & Fukunaga, 2002), (M. Rohrbach et al., 2013). These approaches map semantic sentence representation (e.g. key objects, locations, and scenes), with

a Conditional Random Field (CRF) model, to high-level concepts such as the actors, actions and objects, for generating sentences. In (Jeff Donahue et al., 2017), the authors then improved their system by learning the output sequence representations into an LSTM model to translate it to a natural sentence.

- In (Venugopalan et al., 2014), the authors proposed an end-to-end neural network to generate video descriptions. By mean pooling, the features over all the frames are represented by one vector, to be used as an input of an LSTM model to generate sentences. For better modelling results, not only video contents and their spatio-temporal relationships were used, but also the syntactical structure. (Venugopalan et al., 2015) they extended their work by adding to the input frames and optical flow images to feed an encoder-decoder framework based on two LSTM modules. The encoding converts video into a compact representation, followed by the decoding to converts the output into a caption.
- In (Yao et al., 2015), the authors proposed to utilize a temporal attention mechanism and a spatio-temporal convolutional neural network to obtain action features. The resulting video representations were used as input into the text-generating RNN.
- In (Pan, Mei, Yao, Li, & Rui, 2016), the authors proposed to model video content and textual semantics as a regularizer in Long Short-Term Memory architecture. (Pan, Yao, Li, & Mei, 2017) presented LSTM with transferred semantic attributes (LSTM–TSA) architecture where the semantic features were extracted from both images and videos using the CNN plus RNN framework for enhancing video sentence generation.
- In (Yu, Wang, Huang, Yang, & Xu, 2016), the authors used a hierarchical RNN (hRNN) to describe long video containing more than one event. The notion of hierarchical framework is to make use of the temporal dependency and semantic context between the sentences in a section. Mainly, they used two generators; a single sentence generator produced by a Gated Recurrent Unit (GRU) layer, using spatial and temporal information present in a precise time interval of a video, and a section generator models dependency between the sentences. As output, they generate a mundane description using multiple sentences in a section.
- In (Long, Gan, & de Melo, 2018), the authors proposed an LSTM with two multi-faceted attention layers which export temporal, motion and semantic properties, using nearest neighbor (NN) search, Support Vector Machine (SVM) and hierarchical recurrent neural encoders (HRNE) for a subject and verb prediction based on the temporal features.
- In (Das et al., 2013), the authors proposed to generate dense captions using sparse object stitching; their work for the description is not data-driven, however it is based on top-down ontology.

A comparative review of existing sequence learning approach in video description methods can be found in: (J. Xu et al., 2016), (Ryoo, Chen, Aggarwal, & Roy-Chowdhury, 2010), (Awad et al., 2018), and (Graham, Awad, & Smeaton, 2018). An example of video description (dense captioning) is shown in Figure I-2.

Ground-truth
Event 0: A man is writing something on a clipboard.
Event 1: A man holds a ball behind his head and spins around several times and throws the ball.
Event 2: People use measuring tape to measure the distance.

Masked Trans. (ours)
Event 0: a man is seen standing in a large circle and leads into a man holding a ball and
Event 1: the man spins the ball around and throws the ball
Event 2: the man throws the ball and his throw the distance

Figure I-2: An example of dense video captioning taken from (Zhou, Zhou, Corso, Socher, & Xiong, 2018). The colour bars represent different events. Coloured texts highlight relevant content to the event.

A comprehensive and interesting literature review on video description can be found in (Z. Wu et al., 2017), and (Aafaq, Mian, Liu, Gilani, & Shah, 2018).

Nevertheless, both approaches; Behaviour understanding and sentences generation and Sentences learning, have some major flows:

1- The **Behaviour understanding and sentences generation approach:** according to (Venugopalan et al., 2015) this approach is insufficient to model the richness of language used in human descriptions – e.g., which attributes to use and how to chain them effectively to generate a good description. Also, according to (Z. Wu et al., 2017), the missing, erroneous and misidentified information extracted from the video frames leads to disjointed descriptions. In plus, the handcrafted templates risk being non-generic for the variety of scene types (J. Xu et al., 2016).

2- The **Sentences learning approach:** according to (Aafaq et al., 2018) "The majority of current literature on video description focuses on **domain** specific **short** video clips with **limited vocabularies** of **objects** and **activities**". And so, current state of-the-art methods may not be suitable for long video sequences because they mainly focus on short topic-coherent ones. For that, the description of longer videos and scenes having variety of types remains a challenge, due to the need of large vocabularies and training data. In this domain, there is a lack of rich models that can learn the sentences to the appropriate features in the frames sequence.

Despite the tremendous work done in the field of video description in general, the existing state of art on video surveillance scene description as has it is particularity, still, however not deeply prospected. For example, video description for movies, broadcast news, or sports, can unveil practical drawbacks for video surveillance (Jiangung Lou, Qifeng Liu, Tieniu Tan, & Weiming Hu, 2002), (C. Fernández, Baiget, Roca, & Gonzàlez, 2011).
But even though that most of the researches focus mainly on short non-surveillance videos, some of the advancements made in the approaches can be used in the field of video surveillance.

Next we mention some of the most influencing works on video surveillance understanding and description:

- In Remagnino et al. (Remagnino, Tan, & Baker, 1998a), (Remagnino, Tan, & Baker, 1998b), the authors mainly focus on traffic scene to represent the behaviour of pedestrians and vehicles, where their system is based on Bayesian network to give annotations for some events in natural language. They handle also some cases of

interaction between two objects when the distance between the two is below a threshold.

- In (Aishy, 2001), the authors proposed a system for object and event extraction for video processing and representation. They mainly targets videos having realistic environment (with objects occlusions, artefacts). They proposed three processing levels: video enhancement, video analysis and meaningful content extraction (spatio-temporal features), and video events interpretation. They tested the system on real-time videos. Nerveless they did not work on textual description, but they highlighted in their approach, many interesting aspects in this field, especially concerning the real-case features extracted, and the logical relations.

- (Atsuhiro Kojima et al., 2002), (A. Kojima, Izumi, Tamura, & Fukunaga, 2000) are two of the early works that proposed, on human activities, generating a hierarchical concept of actions for natural language description appearing in real image sequences. The authors primarily describe videos of a one person performing a single action. They detect humans head and hands by a probabilistic approach, and then they use their positions and the head direction to estimate the human posture. Meanwhile, the most appropriate verbs and many syntactic elements are selected. As last step, they used machine translation method to generate natural language text.

- In (Jiangung Lou et al., 2002), the authors propose an approach for semantic interpretation of pedestrian and vehicle's behaviours for visual traffic surveillance. The trajectories recorded are then analysed using dynamic clustering, they introduce, based on HMM, a trajectory segment analysis method to every trajectory class. Then, in each segment, they assign the action of the tracked target to four basic types: Move Forward, Turn Right, Turn Left and Stop. Then they perform classification to feed the natural language semantic interpretation. For that, they use a simple grammar rule template: (The Obj) (Action) in (The place name) [at (high/low/middle) speed]. The system output module is only activated when:
  1. A new action is occurring (Move Forward, Turn Right, Turn Left and Stop).
  2. The object is entering a new region
  3. An abnormal event is occurring

  Their system is restricted to one scene type, and one object type, and did not take interactions into consideration.

- In (C. Fernández et al., 2011), (Carles Fernández, Baiget, Roca, & Gonzàlez, 2008), the authors present a supervised ontology-based methodology. Their ontology shown in (Carles Fernández et al., 2008) present interesting ideas and intersect with our ontology in many concepts. They first perform image segmentation for agent trajectories detection, body postures, and facial expressions, and targets identification. Then these information passes by a user for data filtering. They made this data for each detected agent available within a ground-plane representation of the controlled scenario. The uses XML for data exchange among the modules. Their approach considers, for video surveillance, different scene type, indoor and outdoor scenarios.
  Their proposed taxonomical events include basic actions and events (e.g. walk, run, turn) and some scenario-specific interpretation of behaviours (e.g. meeting, giving way, chasing). Their textual output is presented like: turn (Agent 20, left, crosswalk).

- In (Z. Xu, Zhi, Liang, Lin, & Luo, 2014), (C. Hu, Xu, Liu, & Mei, 2015), (Z. Xu, Hu, & Mei, 2016) the authors proposed approaches were called as Video structural description VSD. VSD targets at describing video content in text sentences. Firstly, they extract the semantic content from the video relying on spatiotemporal segmentation, feature selection, object recognition. Secondly, VSD aims at organizing resources in the video according to their semantic relations. The proposed method is based on ontology; which, between barracks, is highly recommended for a video structured description, it defines a number of concepts including vehicle, people, and traffic sigh, and their spatial-temporal relations, which allow users to annotate traffic events. In their approaches, they did not consider objects interactions description in the scene.

- A very recent and interesting work was presented in (Ahmed, Dogra, Kar, & Roy, 2019), where the authors present template-based technique that generates natural language descriptions of surveillance events. First, they track moving objects, and then they perform classification using CNN on the output into four classes: pedestrian, car, Bike, and Cycle. Finally, their system generates natural language description based on template: “A {color} {size} {type} in {speed}, coming from {entry zone} toward {exit zone}”, as “A white medium vehicle in normal speed, coming from Main Building toward Residential Zone”.
  Two important points were noticed, concerning the authors and their “experts” insist on the importance of structured templates, and human experts’ assessment. Their system assumes a surveillance scene with some prior region information, and they did not consider interaction.

Other interesting research is presented in (Tu, Meng, Lee, Choe, & Zhu, 2014), (L. Xu & Song, 2016), (W. Hu, Xie, Fu, Zeng, & Maybank, 2007), (Gerber, Nagel, & Schreiber, 2002).

I.2. Goals and Challenges

Our primary goal, in this thesis, is to describe textually video surveillance scenes, in a comprehensive way to support police incidents reports. We focus on scenes containing exactly two objects. The secondary goal will be extracting valuable features useful for generating alerts and investigating intelligently the archives by surveillance system operators.

Automatic systems that can assist police and law enforcement agencies still need improvements in order to cope the existing needs when working with video surveillance. While the video analytics companies focus on big in appearance deliveries dissipating small basic issues that are the real police needs, the research field suffers from the miss-integration, discontinuity of researches and missing the accuracy needed from the field (real cases); and the managers of such systems struggle from lack of knowledge in the “how”, “how much” and “what” they really need to assist their systems in an efficient way.

The path leading to achieve these goals is vast, and contains many details. These details are with significant challenges and involve questions that need to be answered. Therefore, to direct this research, we could sum up the goals by asking the following:

The first question is about the best approach for a good video surveillance semantic representation.

Which one of the two approaches is best suited for video surveillance? Behaviour understanding and sentences generation approach or sentences learning approach? It depends on several points.

- The sentences learning approach is still a developing field. It works well with images, but has limited vocabularies of objects and activities in videos, and is still not completely suitable for dealing with the variety of scene types and the long video sequences (Aafaq et al., 2018), (B.-C. Chen, Chen, & Chen, 2017). In plus, in our research, there is an indispensable need for structural description and behaviour understanding features, like speed, trajectory, direction, shape and others; and working with learning approaches, till now, did not encounter all these aspects together. However, we believe that this evolving field will meet finally all the needs.

- For the Behaviour understanding and sentences generation approaches, as mentioned before, there is three main difficulties:
  1) The handcrafted templates risk being non-generic for the variety of scene types. However, on video surveillance researches, one big advantage is that we knows exactly the needed output templates for the system; as, it is similar to the real case reports that already take into consideration the scenes varieties.
  2) It is insufficient to model the richness of language used in human descriptions. Again, in the surveillance field, it is sufficient for the video surveillance reports to have a simple structured sentence as output.
  3) The missing, erroneous and misidentified information extracted from the video frames leads to disjointed descriptions, which means the semantic content identification approaches for extracting all the needed features still not up to norm, and need a lot of improvements. Dealing with that, many enhancements appeared recently on many levels in computer vision field, especially after benefiting from the rapid and increasing machine learning field. And so, focusing on improving the semantic content extraction with machine learning, and then combining their advantages with the advantages of handcrafted features (X. Wu, Li, Cao, Ji, & Lin, 2018), (Cilla, Patricio, Berlanga, & Molina, 2014) it can improve the resulting content extraction.

Dealing with video surveillance system, from our perspective, requires many improvements to be made on many levels. But, in no case, we should lose the content understanding outputs because it is the main core of video surveillance analysis.

Therefore, a description system suitable for video surveillance can be built using the behaviour understanding and sentences generation approach. However, building a good system cannot be without making some improvement, on different levels, by taking advantages of the emerging machine learning field.

Many other questions are important and essential for this research. However, solving and answering them, is significantly challenging.

a) How should the video description system be built in order to be more efficient and useful for different surveillance systems?

b) How to decide what visual information to extract from video?

c) What should a system describe? How to decide when generating a textual description along the time dimension? How much the description is practical and responds to the user needs?

d) How to design a powerful sentence generation model? What an adequate textual/sentence representation contains? What is the best combination of different components of a representative sentence?

The following chapters will be answering these questions and a summary of these answers is presented in the chapter VI section VI.2.

I.3. Thesis outline

This section provides an outline of the entire thesis, which mainly consists of the next five chapters in order to achieve our goals.

In chapter II, trying to enable more integration between the high level semantics into the low level features automatically extracted, we present a new generic knowledge-based ontology, the "Video-Surveillance-Description Ontology", for describing video surveillance scenes.

As one of the most important features that can rule the way that an object can do the action, interaction or reaction, is its deformability, we present a method to classify, in chapter III, the deformable/non-deformable nature of a video object, using heuristic approach.

In chapter IV, we present our approach for textual description of surveillance scenes containing mainly two objects with main focus on the interaction occurring between the two objects. For this, we present how we produce activity matrixes of useful characteristics which can be used for generating alerts and querying the scenes, and how to generate textual descriptions of these matrixes.

In Chapter V, we highlight, based on our research and practical experience, the existing gaps between the surveillance systems operators' needs from one side, and the research field and the commercial (industry) field from the other side. Consequently, we present many propositions about how to address these drawbacks.

Finally, in Chapter VI, a general conclusion and future works of this thesis are presented.

II. Generic video surveillance description Ontology

II.1. Introduction

Multimedia content, particularly videos, are big data source. While current video browsing methodologies are mainly time-based, there is a crucial need to develop intelligent methods for effective storing, indexing, organizing, mining and retrieval from surveillance video databases. However, until now, there's still a lack in such automatic intelligent systems.

One probable reason for this lacking is that video is subject to different interpretation and description which can vary according to systems operators needs and applications (Pavlidis, 1992), (Kunt, 1991), (Jain, 1991). Many video representation techniques addressed this problem by trying to develop a specific solution for each application. Others focus on solving complex situations by assuming a simple environment, for example, without object occlusion, noise, or artefacts.

Consequently, advanced content-based video analysis has become a vastly active research field, and significant results have been reported for the last two decades (Hua, Lu, & Zhang, 2004), (Muller-Schneiders, Jager, Loos, & Niem, 2005), (Wactlar et al., 2001). However, the lack of precise and generic models for video content representation and the complexity of video processing algorithms make the development of fully automatic video semantic content description a challenging task. Actually, the complexity and diversity of video scenes makes hard to map the low-level features extracted automatically from video data, into high-level semantic concept. This challenge, which often referred as the semantic gap, is corresponding low-level spatio-temporal features that can be automatically extracted from video data with high-level semantic concepts. This, causes the existing systems and approach to be too non-flexible and cannot satisfy the need of video applications at the semantic level. So the use of domain knowledge is very necessary to enable higher level semantics in automatic parsing. This is where "Ontology" enters the scene.

Ontology is composed of a set of terms (vocabulary) and specifications about their meanings (properties, relationships). The most referenced definition of the notion of ontology is given by (Borst & Borst, 1997) as: "a formal specification of a shared conceptualization". It was used, in many fields, as a knowledge management and representation approach. For the expression of concepts and relations in ontology, several standard description languages have been defined, we mention: Resource Description Framework (RDF) (*RDF*, 2004), Web Ontology Language (OWL) (*OWL*, 2004) and, for multimedia, the XML Schema in MPEG.

Ontology is a way to represent formally the knowledge. On the top of that, it is not qualified by the vocabulary but the conceptualizations that the vocabulary terms are intended to deliver. Thus, no change is conceptually made when translating the terms from one language to. In addition, ontology is a mean for the experts of different domains to communicate together, to share their experience and accumulate knowledge.

Many important efforts, based on ontologies, have been done in the field of video analysis, in general, and video surveillance in particular. In the state of the art, we present some of these works.

In the last section of this chapter we present our ontology named "Video-Surveillance-Description Ontology". It is a new generic approach for video surveillance description, and designed to be used as a generalist high-level layer for video analysis, principally in a video-surveillance system. We considered the temporal dimension of the video, using appropriate features. Our proposed ontology introduces six main classes; one of which is a representation for generic scene types, divided into fifteen subtypes according to the number of moving objects before and after the interaction. This ontology will be based on in the next chapters to fulfil an automatic textual description of video surveillance, focusing mainly on interactions between objects.

II.2. Related Works

II.2.1. Ontology benefits and requirements

Ontology is a way to reduce the semantic gap in video analysis between low level features and the needed output. They present powerful mechanisms for organizing, structuring, and sharing knowledge. The reasoning, flexibility, share-ability, and representation make these models suitable to surveillance domains. For instance, ontology of video understanding will enable different experts to communicate and exchange their point of view about functionality or an expected output result.

We can already see that having a domain-based ontology is important to reach our objectives. As Korpipää, et al. mentioned, some of the basic requirements that hold also for our approach when designing our ontology are, the simplicity, the flexibility, the extensibility, the genericity, and the expressiveness (Korpipää, Jani, Kela, Malm, & others, 2003).

Despite the great advancements in the last decade, the complexity and the quantity of possible complex activities (Naeem & Bigham, 2007), the importance of the semantics associated with a behaviour (L. Chen & Nugent, 2009) and the interaction of several objects in the same environment (Cook, Augusto, & Jakkula, 2009), (Singla, Cook, & Schmitter-Edgecombe, 2010), among others, make creating a suitable ontology, based on understanding of human behaviour, a challenging task.

II.2.2. Previous works on video analysis using ontologies

The state of the art of both approaches for video analysis, the data-driven approaches and the knowledge-based approaches, mentioned in the state of the art section I.1, reports many researches based on ontologies, (Rodríguez, Cuéllar, Lilius, & Calvo-Flores, 2014).

Nevertheless, working on ontologies trespasses the video surveillance domain to a wider one which the video in general.

II.2.2.A. Ontology on contextual information and context-aware

A significant amount of researches on ontology has been done for the structural representation and recognition of contextual information (Rodríguez et al., 2014), and activities and interactions. Important context-aware ontologies have been proposed, like, CONON (CONtext ONtology) (Haas, 1995), the Pervasive Information Visualization Ontology (PIVOn) (Herrera, Herrera-Viedma, & Martínez, 2000), the Context Aggregation and REasoning (CARE) middleware (Herrera & Martinez, 2001), and the fuzzy ontology (B. Wang, Liang, Qian, & Dang, 2015). A wide range of factors are used to classify these previous work in human motion analysis and video understanding, such as: model-based vs. non-model based, human-object interactions and group activities, action and activity recognition and classification, complex activities recognition and behaviour understanding, and video description, etc.

Also, a number of surveys have reviewed the use of ontologies for context modelling, user context and human Behaviour (Rodríguez et al., 2014), (Villalonga et al., 2016).

II.2.2.B. Ontology in the domain of video surveillance

For video structured description, ontologies are highly recommended (Z. Xu et al., 2014). In the domain of video surveillance, various approaches of ontologies and algorithms were used to address different stages of the problem. (Vezzani & Cucchiara, 2010).

Video surveillance has its own set of most significant entities, terms, hierarchies, and relations. Due to the huge set of possible cases combined with the flexibility of description, the definition of unique video surveillance ontology is very ambitious and probably unfeasible. Nonetheless, a set of actions, events and entities can be selected due to their importance. The surveillance community has made some proposals for action, event, human activity and behaviour ontologies. Some shared concepts can be found among the following ontologies; also some ideas intersect with our proposed ones:

Video Surveillance Online Repository (VISOR) (Vezzani & Cucchiara, 2010) is a platform for annotating, and retrieving surveillance videos, which used as a support tool for different projects. It contains a large set of multimedia data and corresponding annotations. VISOR provides a list of video surveillance concepts used in the Visor system. The main concept of dividing between context and content is shared between many ontologies, including ours.

In (Ly, Truong, & Nguyen, 2016) a behaviour ontology is proposed, mainly based on set of scene model related by set of time relation. The set of scene model contains set of object model where low level data is specified, set of object relation, and set of object condition. Some of the concepts of the object Model intersect with ours.

More recent work can be found in (Alonso, Leal, Escalante, & Succar, n.d.), The authors present ViVA ontology, which is based on (Kazi Tani, Lablack, Ghomari, & Bilasco, 2015), (SanMiguel, Martinez, & Garcia, 2009) and ("VISOR," 2017). ViVA ontology proposes three main classes Content, Context and System. VIVA was designed with owl format and using Protégé. Protégé is a suitable tool for ontology presentation, which we decided to use it for our approach, in the interpretation of my graphical representation. Also, concepts concerning place, weather, location, and object may meet our same objectives, as follow; some of those influence our ontology.

Other interesting ontologies can be found in the Appendix VIII.1.

II.3. Proposed Ontology

The varied nature of objects participating to a scene, the variety of scene types or contexts, and the complex nature of the object behaviours, actions and interactions and in the execution, requires an abstract level of information to reduce the size of the description scope. This work presents an ontology-based method that combines low-level primitives of objects basic features, like size, colour, locations, speed and others, that should allow to intelligently deriving more meaningful high-level information.

In order to realize the knowledge-based and automatic generic description of video surveillance introduced in the previous section, the knowledge for video analysis is abstracted. Among many distinctive characteristics for this ontology, we mention that it:

1- Focuses mainly on the objects interactions, nonetheless it is expendable.
2- Presents detailed propositions about the interaction, from a methodologic and systematic approach.
3- Is not directed by the results of the automatic analysis, and there is no pre-assumption or condition which restricts this ontology.
4- Targets mainly the level of generic and abstract description, but it can be applied to any scene type or context.
5- Shall be convenient to describe real interactions during incidents as they appear in CCTV control rooms' reports.
6- Focuses on new concepts concerning mediation, action at a distant and close interaction, deformable and non-deformable objects, and others.

Our proposed ontology, named "Video-Surveillance-Scene-Description Ontology" or "VSSD Ontology" mainly describes the concepts that relate video, objects, and actions. VSSD ontology has been designed to be used as a generalist high-level layer for a video analysis, principally in video-surveillance system. VSSD ontology proposes six main classes: Context, Object, Video, Activity, Scene and Descriptor (see Figure II-1).

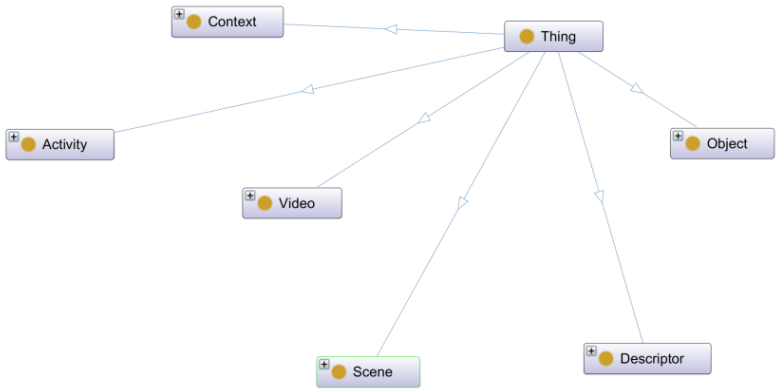


Figure II-1: VSSD ontology's six main classes: Object, Video, Context, Activity, Scene and Descriptor.

II.3.1. Context

This class contains all the elements that provide information about the real context, see Figure II-2. For example: the GPS coordinates, the place where the action happens which can have two types: (Indoor: Bank, School, etc.; Outdoor: circulation, garden, Parking, etc.); the environment (weather, altitude, temperature, pressure, lighting, humidity, noise) and the time class, one of the most important classes that drive all other class.

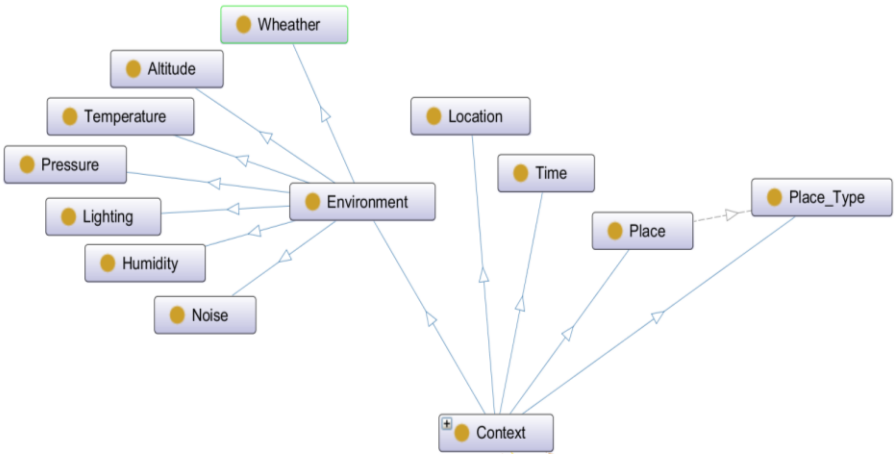


Figure II-2: The Context class.

II.3.2. Object

The **Object class** represents instances of humans, animals, plants, machines and all other inert objects, see Figure II-3. This class can represents all what exist in an environment. One of the most important features that can rule the way that an object can do the action, interaction or reaction is its deformability. The deformability criteria is mainly deduced from the object shape and motion, and is based on the degree of deformation (Kambhamettu C., Goldgof D. B., Terzopoulos D., Huang T. S., 1994). When focusing on an area of interest, the first thing to distinguish, if an object appears far/deep in the frame, is its deformability. Non-deformable objects actions or reactions during an interaction are easy to detect, analyse, understand and maybe predict. When deformable parts of an object, move freely in unpredicted way, the prediction becomes more difficult even for a human brain.

We chose to group all objects in two general sub-classes, deformable and non-deformable objects. Object deformability dilemma is considered in chapter III, and in section IV.3.2 object classification. For example, humans and animals are ''deformable''. Plants, machines and inert objects can be deformable or non-deformable, e.g. a tree is considered as deformable when each of the branches can move differently than the others or the trunk. A machine in this ontology indicates the machines controlled by intelligence (humans or artificial intelligence) like a car or a robot, in the opposite of an inert object like a box. We may find some of those objects, in other ontologies, under the name of agent.

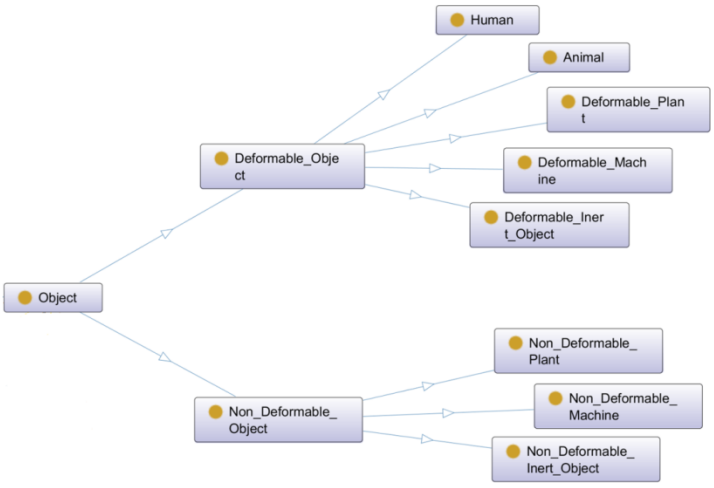


Figure II-3: The Object class.

II.3.3. Video

In visual surveillance systems, the cameras are mainly fixed. As the same object may appear several times in the same video, each appearance will be considered as an instance in Video_Object class. So, the Video_Object class is a subclass of video and object classes, see Figure II-4. This instance is delimited from the first moment of that appearance to the last one.

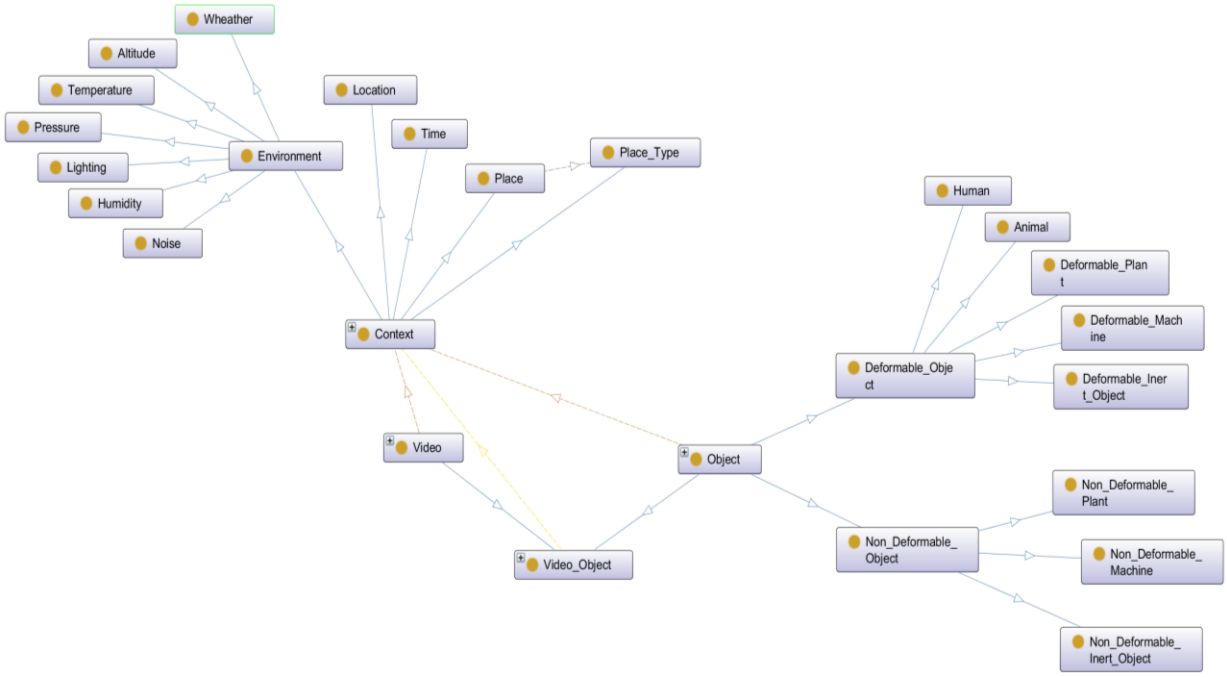


Figure II-4: The Video, object, video_object, and context classes.

A **sub_object** is mainly used for deformable objects, for example for articulated segments of human and animal bodies, or parts of machines, etc. A video object can have several sub_Objects. As we may have many **states** for each appearance (instance of video_object class), each of the states describes the object/video_object state. Similarly many of the states can be taken for each of the sub_objects to create a sub_object_state, see Figure II-5. The number of states depends on the time of appearance, time of disappearance, and the suitable frame difference that we would take. In plus, for each state, the video object state can have many **features (attributes)** like shape, surface, displacement, speed, trajectory and many others.

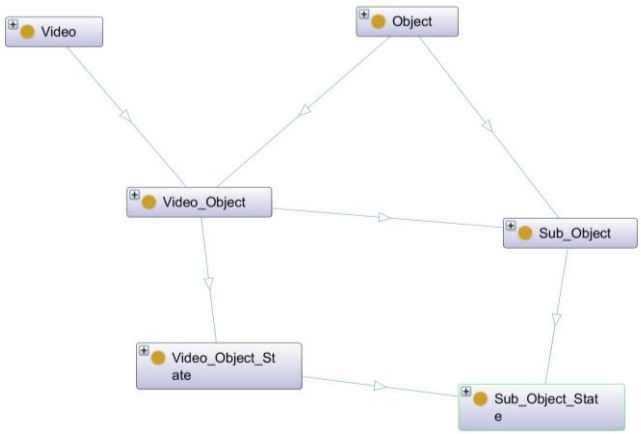


Figure II-5: The Sub_object, Video_Object_state, and Sub_object_state classes.

II.3.4. Activity and Action

Different taxonomies are used for describing an action. We can find, among others, the terms operation, gesture, action, event, activity, and behaviour. So far, there is not a unique standard ontological definition of those notions or concepts. Many can be found in different articles (Herath, Harandi, & Porikli, 2017), (Ranasinghe, Al Machot, & Mayr, 2016),(Morris & Trivedi, 2008), (Lavee, Rivlin, & Rudzsky, 2009), (Kaptelinin, 2013).

Usually, the literature names what human is doing and the way it is doing it human behaviour or human activity interchangeably (Ros, Cuéllar, Delgado, & Vila, 2013) (Remagnino, Foresti, & Ellis, 2005), (Rashidi & Cook, 2009). These activity/behaviour terms correspond to a sequence of human actions. However, most of these authors agree to define human action as the simplest unit in the human activity. As new approaches are being developed (L. Chen & Nugent, 2009), new levels appear in the system. For that, a difference should be made between the terms human behaviour, events and activity to differentiate between the concepts of what a human is doing in the environment (activity), and the purpose or meaning it could have (behaviour). An Event is the occurrence of an activity in a particular place during a particular time interval. The Behaviour is a description of activities and events within a specific context.

In our ontology we embed three hierarchical layers: activity, action-interaction and operation.

An **activity**, according to (Blunden, 1978), is the units of life. It is purposeful and developing interaction between actors ("subjects") and the world ("objects") (Kaptelinin, 2013). An activity is hierarchically structured into actions, see Figure II-6. For more complex scenes, activities may be, sequential, or concurrent according to performing time.

The second layer is the **Action**. The action is based on conscious processes concentrating to fulfil a goal or its sub-goals. In the philosophy of action (Wilson & Shpall, 2016), an action is defined as intentional, purposive, conscious and subjectively meaningful activity. For example, pushing a person is an action, while catching a cold is not considered a one.

In case of two or many objects, an action begins when one of those objects has the intent to perform an action, even while approaching. This action ends when the objects retreat. They may approach again to begin another action.

Another important concept is the mediation. The main distinctive features of humans, such as language, culture and society, the production and use of advanced tools, etc., all involve mediation; here we note the mediation of information as the most important one among interactions. They represent different aspects of the same phenomenon, that is, the emergence of a complex system of structures and objects, both immaterial and material which serve as mediating means embedded in the interaction between human beings and the world and shaping the interaction. In cultural-historical psychology, mediation is, arguably, the most important concept of all; it serves as the cornerstone of the activity theory as a whole (Vygotsky & Cole, 1978).

An example for the mediation is a human shooting another object (human, animal…). In this case the bullet can be considered as the mediation. We may equally well consider the linguistic interaction as a transmission of information, for example saying "Hello". In the opposite, when two humans are boxing, two animals are fighting, or when two animals are following each other, there is no mediation between the two objects or unmediated action.

We can consider that implicit information helps both objects to coordinate their interactions.

In the case of one, two or many objects, and where the action/interaction is unmediated, or at least not well noticed as visual mediation by the application, we distinguish between two action types:

a- There is no physical contact: then we consider "action at a distance" or "far action/interaction", for example: when two objects are running together, or when two humans are saluting each other, etc.
b- There is physical contact: then we consider "close action/interaction", for example, when two humans are fighting, or hand shaking, etc.

An Action is a series of **operations** done by an object on nobody, object, or many objects. The operations are considered the lower-level units implementation of the action. Accordingly the **Interaction States** can document the state of interaction at a related moment (existence, type and aggressiveness).
We present the relations between components and action. But those relations can be the same for activity and operation, or for the interaction state. We mention that:

- An object or video_object or sub_object can have an action/interaction, and an action is done by an object or video_object or sub_object.
- A video contains an action, and an action is viewed in a video.
- A video_object_state or a sub_object_state is a part of an action, and an action can have instance a video_object_state or a sub_object_state.

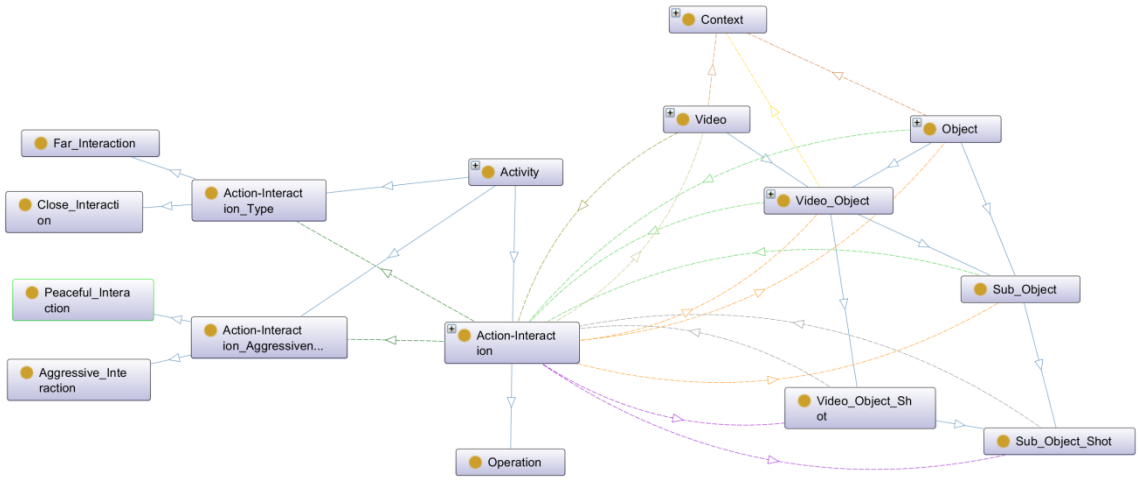


Figure II-6: The Activity, Action-Interaction, and Operation classes

II.3.5. Scene

To define a methodologic and systematic approach to describe the video scene especially the interaction between video objects in video surveillance, we identify fifteen types according to the number of moving objects and to their characteristics (features) before and after the action.

1. **0 Object (Scene without any moving objects):** when no object is moving in the scene.
2. **1 Object → 0a (Single object stops, no interaction with the environment):** when a single object is moving in the scene at some moment it stops. Examples: car parks, etc.

3. **1 Object → 0b (Single object stops, interaction with the environment):** when a single object is moving in the scene, without any interaction with another moving object, at some moment it stops, after mainly changing and interacting with the environment (background). Examples: car hits a store causing it to stop, etc.
4. **1 Object → 1a (Single object, no interaction with the environment):** when a single object is moving in the scene without any interaction with another object or without changing anything in the environment (background). Examples: human walking, or doing sports, car passing, etc.
5. **1 Object → 1b (Single object, interaction with the environment):** when a single object is moving in the scene without any interaction with another moving object but mainly changing and interacting with the environment (background). Examples: person switching on the lights, person is smoking, car switching on the lamps, crashing an ATM machine etc.
6. **1 Object → 1c (Single object, interaction with the inert objects of the environment):** when a single object is moving in the scene without any interaction with another moving object but changing and interacting with the inert objects of the environment (taking or leaving an inert object); and by that it changes its characteristics either gaining (good influence) or losing (bad influence) some. Examples: person or animal handling an inert object like box, person wears or removes his vast or hat, etc.
7. **1 Object → 1d (moving object trigger an inert object and stops):** when a single moving object in the scene, at a given moment, performs an action with another inert object, hence the object stops and makes the inert object to move. Examples: one ball hit another fixed ball and stops, one moving car hits another car hence it stops and makes the other car to move, etc.
8. **1 Object → 2a (moving object trigger an inert object):** when a single moving object in the scene, at a given moment, performs an action with another inert object and makes it to move. Examples: one ball hit another fixed ball, person is opening a door, one moving car hits an inert object (like another car) and makes it to move, etc.
9. **1 Object → 2b (moving object divides into 2):** when a single moving object in the scene, at a given moment, divides into 2 objects. Examples: person jumps out from a car, person removes his vast or hat, etc.
10. **2 Objects → 1a (moving object stops another moving object):** when there are two moving objects in the scene and, at a given moment, one object do an action and stops the other object. Examples: a moving car hits a moving person, etc.
11. **2 Objects → 1b (2 moving objects merge into 1):** when there are two moving objects in the scene that, at a given moment, merge into one single object. Examples: person jumps into a moving car, a person jumps on a moving skateboard, a person picks up and wears a hat, etc.
12. **2 Objects → 0 (2 moving object stops after interaction):** when there are two moving objects in the scene that, at a given moment, interact and stop moving. Examples: two cars make an accident; two objects collide and stop, etc.
13. **2 objects → 2a (2 moving objects without interaction):** when there are two moving objects in the scene without any interaction between them. Examples: two cars passing near each other, two humans passing by without any far or close interaction, human and animal co-appear in a scene without any kind of interaction, etc.
14. **2 Objects → 2b (2 moving objects with interaction):** when there are two moving objects in the scene, at a given moment, they interact, and then continue. Examples: two cars are passing near each other trying to avoid collision, two humans follow each other, two humans walking together, animal walking near a human, two humans salute each other,

two humans waving to each other, two humans seeing and walking toward each other, animal is enclosing on a human, two animals fighting, two humans boxing, etc.

15. **Many Objects → Many Objects (Group of moving objects with interaction):** when there are many moving objects in the scene, interacting together at a given moment, and continuing after. We do not consider here many objects in the scene so that the interaction can be divided in couples. This category it meant to describe scenes with a crowd. Anyway, this category may be divided into many other ones, but as it is not our field of interest, we preferred to keep it as one category. Examples: group fighting, or cheering, etc.

These fifteen types are mainly focused on scenes with 0, 1, and 2 objects in the scenes. For more than two objects in the scene, we put all of them in one class for later reconsideration. We must notice that a scene can also be a mixture of many of these types.

Concerning the **Scene_Sub_Type**, we may introduce more detailed interaction categories, such as: At distance or physical, Aggressive or Peaceful.

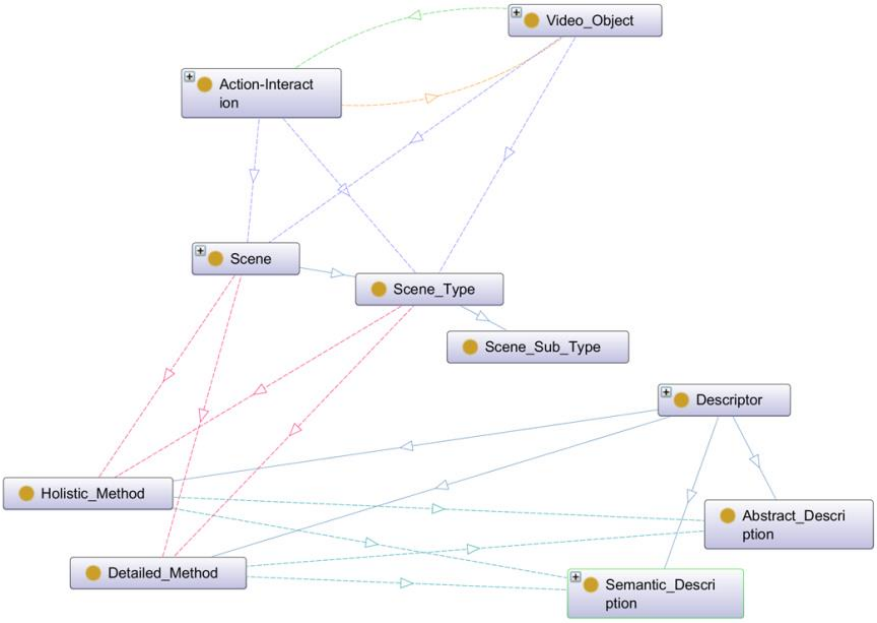


Figure II-7: The Scene, and Descriptor classes

II.3.6. Description

This class is intended to describe the whole scene from objects to action/interaction and context, according to the scene type and sub_type. It contains two main sub_classes: Abstract_Description, Semantic_Description, see Figure II-7. Those descriptions of a scene can be done using two methods:

1- Holistic method: this method takes the whole scene as one single closed box. It does not require for example the localization of body parts, the object or the action identification; the most important is what happens. Using this method, we consider all the possible combinations of actions/interactions in order to recognize, later, which one is the closest to this scene action. It is considered that the actuator actuated and action as a single box.

2- Detailed method: it is the study of each element of the scene, where the identification of each object, sub-object, action, operation, element apart is required.

Then, the scene description, according to the scene type and sub-type and the method used, can be a generic abstract (context free) or a much more semantic text where the context has a big influence.

In Figure II-8 we present the abstract description used in this study, see chapter IV. To have a semantic description one can add, simply, on this abstract description the information taken from the context, like location, time, place, and place, etc. For example:

"At frame 201: "Deformable" object "1" enters the scene, in "C" spot, on the "Left Middle" of the "Outside" area of the camera field of view, heading "Up Left", having respectively "regular" shape, "small" surface, and "slow" speed".	*"On Saturday 10/11/2018, at 11:35:22, a person "1" enters the scene, in the intersection "Verdun-Dunant" (33.890540, 35.484180), on the right of Verdun street, heading south, having respectively small body, and slow speed".*
Example of abstract description	Example for semantic description

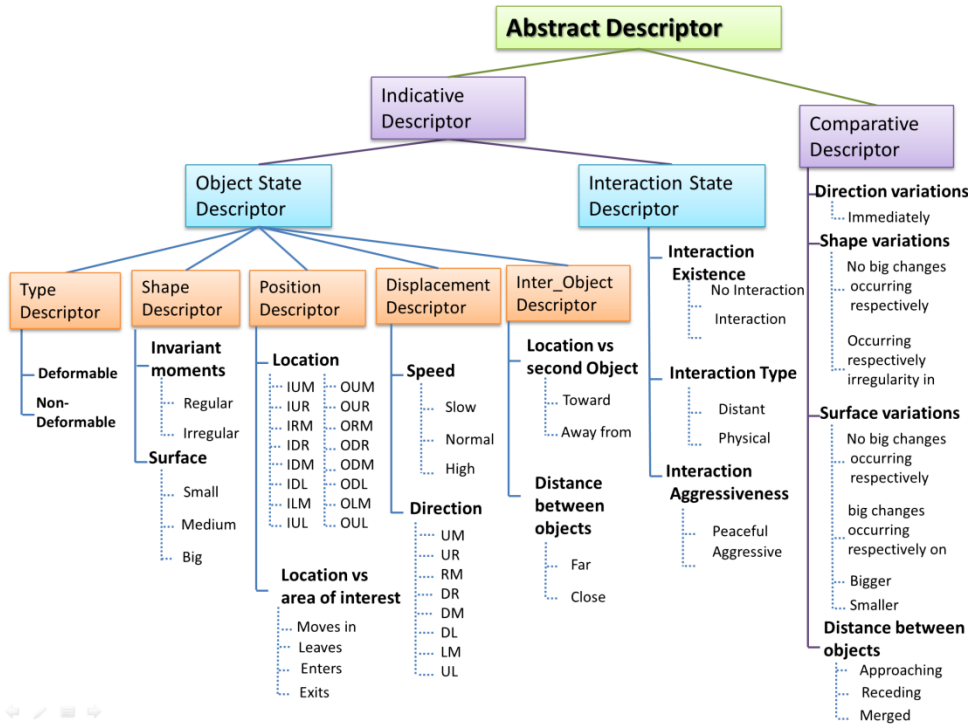


Figure II-8: Abstract description, having in the location and direction: U (Up), M (Middle), D (Down), R (Right), L (Left), I (inside), and O (outside).

Finally, in Figure II-9 we present all mentioned components of the "Video-Surveillance-Description Ontology".

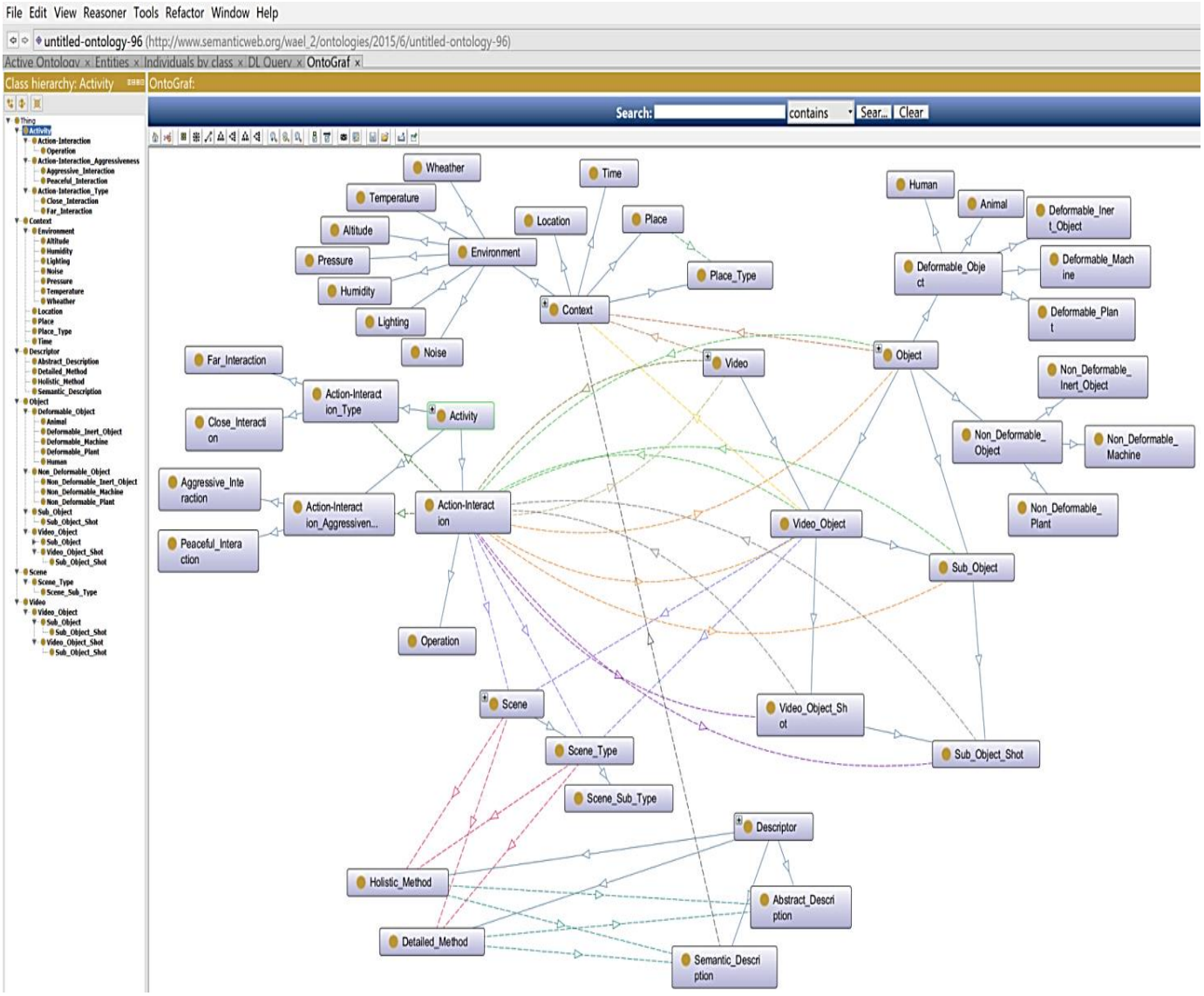


Figure II-9: The proposed "Video-Surveillance-Description Ontology".

II.4. Conclusion

In this chapter, we presented our proposed generic ontology for video description, mainly for video surveillance, taking into consideration some shared concepts as context, object, sub_object, activities, etc. Also, it presents some entities with new concepts like deformable/non-deformable object, fifteen scene types, close/far interaction, aggressiveness of interaction, etc. This ontology will be based on, in the next chapters, to fulfil an automatic textual description of video surveillance, focusing on interactions between two objects.

III. Classifying deformable and non-deformable video objects

III.1. Introduction

For the purpose of semantic video analysis and understanding, it is especially important to recognize and study the video content—i.e., the background, objects, actions, and their movements—to better understand their meaning. Recent research focuses on object movement and its meaning. Accordingly, object properties and characteristics are of considerable importance. One property that significantly drives and influences the objects movements and actions is the object deformability. Object deformability is an important property to qualify the actioner and the actionee, it also gives the main clues to well understand the action. From surveillance point of view, non-deformable objects reactions, during an interaction with another object, are easy to detect, analyse, understand and maybe predict. While deformable parts of an object, make the analysis more difficult even for a human brain. As the deformability criterion of an object is one of the most important high-level features, we found it crucial to differentiate between the two classes.

In many research works, object deformability is a mandatory prior piece of information, for further interpretation, which is not actually automatically extracted, instead it is assumed.

A deformable object is an object that, when in motion, can undergo shape deformations, for example, a walking man, or a running animal. A non-deformable object, by contrast, has a rigid shape, for example, a passing car, an opening door. We define temporal motion as a fragment of an object motion for a small number of successive frames. "Non-rigid motion" is standardly used to refer to all articulated, elastic, and fluid motion, denoted here "deformable motion". Likewise, rigid motion is denoted as "non-deformable motion". Importantly, deformable objects can have both deformable and non-deformable motion, whereas non-deformable objects are restricted to non-deformable motion.

The deformable / non-deformable nature can hardly be established by a learning approach given the difficulty of producing the data necessary for learning. On the other hand, from a visual point of view, the definition of the concept is relatively well defined. So we propose a heuristic approach expressing a physical model.

This chapter presents a new fully automated method for classifying deformable and non-deformable objects. It analyses the object's movements (object motion), differentiates deformable from non-deformable motion, and infers from this whether the moving object is deformable or non-deformable. Our classification method is effective without having any prior information about the environment, the shape of the object, or its displacement, and it does not depend on pre-assumptions. Our method aims mainly to deal with video-surveillance content where there is only one moving object in the scene. But applying object detection or segmentation algorithm, as done in the chapter IV, this method can easily be extended and applied on scenes having several objects.

As stated above, we study object deformability by analysing its motion. Thus, a motion-estimation technique is used to estimate motion between frames. Geometric transformations (viz., Fundamental matrix and Homography) are pursued to determine

whether each of the observed motions corresponds to a transformation. Hence, it is indispensable that we investigate, in Section III.2, the background and related works in both motion and object deformability, in addition to some motion-estimation techniques and geometric transformations. In Section III.3, we explain our approach. We then present, in Section III.4, the experiments we have done in order to validate and evaluate our method.

III.2. Background and related works

III.2.1. Background

The world seen by humans when moving appears stable, rigid, and three-dimensional (3D). This impression is probably the result of the fact that retinal images change over time (Hogervorst, Kappers, & Koenderink, 1997). A fundamental ability of the human visual system is its capacity to interpret motion in space. The visual system is capable of extracting useful information about the 3D structure from these retinal changes. This process is usually called structure-from-motion (SFM). However the ease with which humans detect motion and navigate around objects, and the difficulties in duplicating these capabilities in machines, have led to major challenges for computer engineers and scientists in understanding vision in humans and machines (Aggarwal & Nandhakumar, 1988).

Human vision is privy to many sources of depth information that do not depend merely on stereovision. These sources of information include motion parallax, shape from shading, and textural information. Parallel to this, studies that work with photography in general (i.e., both video and images) have proposed methods for extracting useful information about objects from images and frames. We distinguish works in the following directions:

- translation and/or rotation movement of a rigid body, (Tsai & Huang, 1984).
- projection: affine or orthographic, or perspective, (Del Bue, Lladó, & Agapito, 2007a).
- dimensional approaches: 3D (Structure From Motion SFM, parametric…), or 2D (Optical flow, change detection…), (Zang, Doerschner, & Schrater, 2009).
- extraction of 2D object features; points, corners, lines, edges, conic arcs, features correspondences, or the optical flow, (Stoll, Volz, & Bruhn, 2013).
- appearance of the object in multi-frame, (Hogervorst et al., 1997).
- types of view: monocular or stereoscopic or multiple view images (R. Hartley & Zisserman, 2003a).

Good reviews and plenty of explications of the available methods for estimating the 3D structure and motion from sequences of monocular and stereoscopic images can be found in (Aggarwal & Nandhakumar, 1988). Similarly, (Huang & Netravali, 2009), provides an excellent review for exploiting the consistency by using the multi-frame analysis and studying the object motion and structure from feature correspondences.

For a long time, studies proposing motion-based approaches to motion analysis have been largely restricted to the study of non-deformable object motion, or they were obliged to assume it. However, in the real world, deformable object motion is far more common.

Recently, the studies on analysing articulated motion, particularly human motion, has been inspired by a tremendous number of applications, and this analysis can be generally categorized as: (1) model-based approaches (J. Wang, Liu, Wu, & Yuan, 2014) and

(2) methods that do not require *a priori* shape models (H. Wang, Kläser, Schmid, & Liu, 2013). In the latter approach mostly useful for motion tracking when dealing with an unknown object where no *a priori* knowledge about the motion or the object's shape is available. The major difficulty to this type of approach is to establish feature correspondence. Consequently, to get around the problem, researchers either they impose constraints on the object's behaviour, or they focus on high-level processing supposing that matching is known *a priori*. Model-based approaches have the benefit of knowing, in priori, the approximate shape of the object, simplify this problem. However, these methods are not applicable when information about the object's shape is unavailable.

Despite this, there has been a lot of work on non-deformable objects and object deformability, and an increasing number of studies on deformable objects, especially with regard to simulating or segmenting articulated objects. However, most of these works assume the deformability (or non-deformability) of the objects, and relatively little research concerns the automatic discrimination of deformable and non-deformable moving objects. Some of this research is mentioned in the following subsection (i.e., Subsection III.2.2). In addition, because our object classification technique is based on motion classification, we will briefly address motion-estimation techniques—viz., homography and fundamental matrices—given its pertinence to this study.

III.2.2. Works related to object deformability

L. Wixson and A. Selinger (Wixson & Selinger, 1998) used a reference image pre-obtained from the video sequence for classifying moving objects as rigid or non-rigid based on the similarity of their appearance over multiple frames. They took as hypothesis that the appearance of the rigid objects under viewing conditions similar to orthographic projection changes much more slowly than that of most of the non-rigid living ones. It should be mentioned here that feature correspondences are not used with this method. According to the authors, the results are preliminary, and the method requires further testing and quantification; and additional work is needed to mitigate fluctuations resulting from occlusions that occur when the object moves behind a structure and for dealing with small object movements. In addition, they use relatively few number of experiments compared to other studies.

A. J. Lipton (Alan J Lipton & others, 1999) used the residual flow to analyze the rigidity and periodicity of moving objects. His work is based on the assumption that rigid objects present little residual flow, whereas non-rigid moving objects display higher average residual flow. However, this method cannot be applied to slowly moving objects nor to any revolving objects.

R. Cutler and L. S. Davis (Ross Cutler & Davis, 2000) proposed a method based on the temporal self-similarity of a moving object. Their approach suggests that, when an object displays periodic motion, its self-similarity measure shows periodic motion. They use periodicity to categorize moving objects. But their technique assumes that each object can be properly segmented from the background. However, this assumption does not always hold true.

J. Yan and M. Pollefeys (Yan & Pollefeys, 2006) concentrated on a factorization method based on motion segmentation in trajectory data. Factorization-based methods find an initial segmentation by thresholding the entries of a similarity matrix built from the factorization of the matrix of data points. According to E. Elhamifar and V. Vidal (Elhamifar & Vidal, 2009), it is likely that such factorization-based methods, in general, are correct

provided that the subspaces are independent, but they fail when this assumption is violated. Moreover, these methods are sensitive to noise. Otherwise, a spectral-clustering method, such as the one used by J. Yan and M. Pollefeys, can be used to deal with the issues already mentioned by using local information around each point to establish similarity between pairs of points. The objective in J. Yan and M. Pollefeys was to segment a wide range of motion, including independent, rigid, non-rigid, articulated, degenerate, non-degenerate. The data is then segmented by applying spectral clustering to this similarity matrix. According to E. Elhamifar and V. Rene, such methods are less effective at dealing with points near the intersection of two subspaces, because the neighbourhood of a point can contain points from different subspaces. This issue can be resolved with multi-way similarities that capture the curvature of a collection of points within an affine subspace. However, the complexity of building a multi-way similarity grows exponentially with the number of subspaces and their dimensions.

A. Del Bue et al. (Del Bue, Lladó, & Agapito, 2007b) evaluated a method that uses a trajectory to automatically segment a set of rigid and non-rigid moving points within a deformable object, given a set of 2D image measurements. They noticed a higher misclassification ratio with weak perspective effects, and a greater proportion of non-rigid points. Furthermore, points that are rigid for only a part of the sequence may go undetected. In addition, their proposal was subject to a relatively few experiments.

D. Zang et al. (Zang et al., 2009) used the optical flow to infer the object's rigidity and reflectance. They used the optical flow exclusively to detect rigid object motion for both specular and diffuse reflective surfaces. However, in order to derive the relationship between optic flow and rigid-object motion, they assumed that both the viewer and the environment were distant from the object, approximated by orthographic viewing and illumination parameterized by the direction on a sphere. Further, their results are also based on relatively few simulation examples and experiments.

Feng et al. (Feng, Won, Jeong, & Jeong, 2015) proposed an image matching method to match rigid object image and non-rigid object image by utilizing the same feature.

To the best of our knowledge, no previous research has proposed a method with the following characteristics: full automation in discerning deformable and non-deformable objects; complete generality and applicability to any type of object in a video (i.e., general-perspective projection for the general motion of a general object); a method that does not rely on conditions, assumptions, or additional information about the object in advance; one that takes into account the fact that deformable objects sometimes behave as non-deformable objects; and a method that benefits from temporal consistency. Our approach is the first one to join all those points together.

In this study, the discrimination of rigid from non-rigid motion is studied, to farther infer the rigidity or none of the object.

III.2.3. Motion estimation

Motion estimation, in a video sequence, is to determine the motion's vectors that describe the change from one frame to its adjacent one. As the motion is in 3D scenes, and as the images' frames are its projection onto a 2D plane, so finding the true motion is an ill-posed problem, so it called the apparent motion. The motion vectors may relate to the entire image or to specific parts, such as pixels, or even rectangular blocks, to build the motion field. In dense motion fields, each point is assigned a vector consistent of the motion direction, velocity, and the distance from an observer to the image location.

In video sequences, motion is a key source of information. Estimating the motion field is a useful starting point for solving several issues pertaining to motion analysis. Efficient and accurate motion estimation is essential to image-sequence analysis, motion analysis, computer vision, and video communication, and this information is fundamental to video understanding and object tracking.

The most common methods for estimating the motion field can be categorized into pixel-based methods (or "direct" methods) and feature-based methods (or “indirect” methods) (Dufaux & Moscheni, 1995):

Direct Methods:

- Pixel-recursive algorithms
- Transform-domain approaches
- Optical flow (Barron, Fleet, & Beauchemin, 1994), (Beauchemin & Barron, 1995).
  - Differential techniques.
  - Phase-correlation methods.
  - Frequency-domain methods.
  - Block-matching methods (Khammar, 2012), (Love & Kamath, 2006).
- Indirect Methods*:*
  - Feature-correspondence methods (Farin & de With, 2005), (Torr & Zisserman, 2000).

For more general comprehensive and comparative techniques the reader is referred to Appendix VIII.2.

III.2.4. Projective transformation and Epipolar Geometry

For better understanding how the transformations (the homograghy and the fundamental) can serve our objectives in this chapter, we found it indispensable to clarify some points:

III.2.4.A. Homography (projective transformation)

Homography is conceptually related to collineation, projectivity, and planar-projective transformation. It is an invertible transformation from a projective space (for example, the real projective plane).

It is considered to be a general transformation between the world and the image plane after imaging with a perspective camera (R. Hartley & Zisserman, 2003b). Homography also describes the transformation from one plane to another (i.e., a mapping from $P_2 \rightarrow P_2$). For example, the projection of points of a plane into an image plane can be described with homography.

Thus, for a set of point correspondences $\{X_i \leftrightarrow X_i'\}$ in two images, if all the points X_i are coplanar, then X_i and X_i' are related by a non-singular 3×3 homography matrix, such that:

$$X_i' = H.X_i \qquad \textit{(eq. III-1)}$$

H can be represented with homogeneous coordinates, as a non-singular linear homogeneous transformation.

Various algorithms have been proposed to estimate homography. Some use point correspondences, while others use lines, lines and points, conics, curves, discrete contours, or the planar texture.

In general, estimation algorithms can be classified according to (Criminisi, Reid, &

Zisserman, 1999) as: non-linear geometric (non-homogeneous) estimations, linear non-homogeneous estimations, and linear homogeneous estimations. For more information concerning these methods, the reader is invited to see (Anubhav Agarwal, Jawahar, & Narayanan, 2005) and (Dubrofsky, 2009):

NB: In real images, the position of the points $\mathrm{x_i}$ and $\mathrm{x'_i}$ is perturbed by noise. Image measurement errors will occur in both images and the estimation of H might not be perfectly accurate. For this reason, the image of $\mathrm{x_i}$ in the first image is mapped by H to the point $H.\mathrm{x_i}$ in the second image; and it is not necessarily equal to $\mathrm{x'_i}$: $\mathrm{x'_i} \neq H.\mathrm{x_i}$; vice versa : $\mathrm{x_i} \neq H^{-1}.\mathrm{x'_i}$.

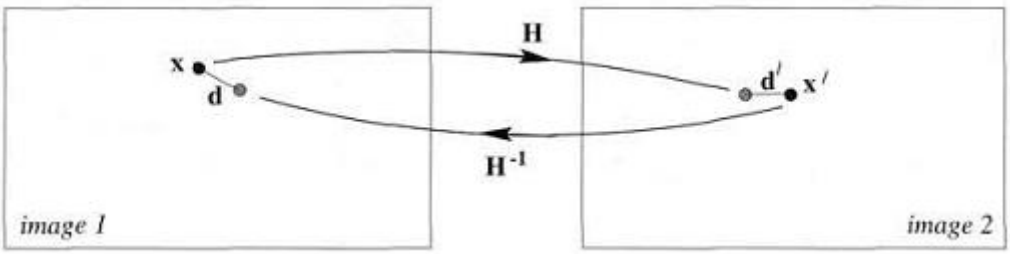


Figure III-1: Symmetric Transfer Error

If we suppose that $d(\mathrm{x'_i}, H.\mathrm{x_i})$ is the Euclidean image distance in the second image between the measured point $\mathrm{x'_i}$ and the point $H.\mathrm{x_i}$, and $d(\mathrm{x_i}, H^{-1}.\mathrm{x'_i})$ is the distance in the first image. Then, the error for a couple of corresponding points $\mathrm{x_i} \leftrightarrow \mathrm{x'_i}$can be measured in both images by a simple method called *Symmetric Transfer Error* (Figure III-1):

$$Er_i^H = d\left(x'_{i,} H.x_i\right)^2 + d\left(x_{i,} H^{-1}.x'_i\right)^2 \qquad \textit{(eq. III-2)}$$

Further, for all the corresponding points $\mathrm{x_i} \leftrightarrow \mathrm{x'_i}$ / $i = 1 \dots N$:

$$Er_H = \sum_{i=1}^{N}\left(d\left(x'_{i,} H.x_i\right)^2 + d\left(x_{i,} H^{-1}.x'_i\right)^2\right) \qquad \textit{(eq. III-3)}$$

Other error measurements can be used. These errors have been identified in the literature.

III.2.4.B. Fundamental matrix

For the most general case of a 3D non-deformable object moving in a 3D world, the set of point correspondences $\{X_i \leftrightarrow X_i'\}$ in two perspective-projection images are related by a Fundamental matrix, such that:

$$X'^T_i.F.X_i = 0 \;/\; i = 1 \dots N \qquad \textit{(eq. III-4)}$$

The Fundamental matrix *F* has Rank 2, and *det(F)=0*. It also has seven degrees of freedom, and it can map each point in an image to its corresponding point in the other image.

Several methods for estimating the Fundamental matrix have been studied. Some methods are linear, whereas others are not. A list of these methods would include seven-point algorithms, eight-point algorithm, methods based on minimizing the geometric-reprojection error (with the so-called Gold Standard method), minimizing the first-order geometric error (i.e., the Sampson distance and the symmetric epipolar distance), Levenberg-Marquardt optimization, an iterative linearized method, and others. For more information, the reader is referred to the references: (R. Hartley & Zisserman, 2003b), (Quan-Tuan Luong & Faugeras, 1996), and (Quang-Tuan Luong, Deriche, Faugeras, &

Papadopoulo, 1993).

NB: Before estimating the fundamental matrix, we should mention that in real images, similarly to the note mentioned above, the position of the points $\mathrm{x_i}$ and $\mathrm{x'_i}$ is perturbed by noise, and so, image measurement errors will occur in both the images, for that the epipolar constraint $\mathrm{x'_i}^{\mathrm{T}}.\mathrm{F}.\mathrm{x_i} = 0$/ i=1, …, N is not fully satisfied, then:

$$x'_i{}^T . F . x_i = \varepsilon \neq 0 \ / \ i = 1, \dots, N \ / \ \varepsilon = \textit{algebraic error.} \qquad \textit{(eq. III-5)}$$

Then the points $\mathrm{x_i}$ and $\mathrm{x'_i}$ do not necessarily lies on the epipolar lines l' and l (see the Figure III-2).

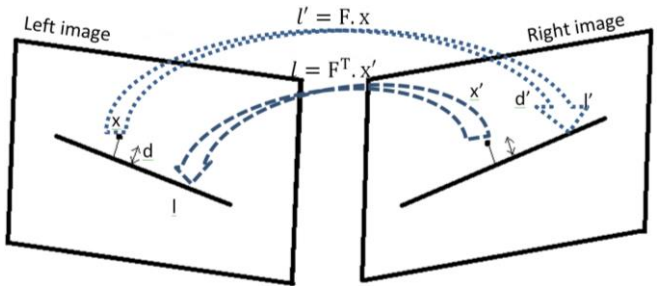


Figure III-2: Symmetric Epipolar Distance

However, rather than searching for the algebraic error ε, a geometric error can be often measured on image planes. This leads to the definition of epipolar distance (error), which is, in the right image, is the perpendicular distance from the point $\mathrm{x'_i}$ to the epipolar line $\mathrm{l'} = \mathrm{F}.\mathrm{x_i}$, and is written as $\left(\mathrm{x'_{i,}}\ \mathrm{F}.\mathrm{x_i}\right)$. In the same manner, in the left image it will be $\mathrm{d}\left(\mathrm{x_{i,}}\ \mathrm{F}^{\mathrm{T}}.\mathrm{x'_i}\right)$.

In General, the epipolar distance is computed for both images to avoid any bias in any computation using the epipolar distance, and that is what's called *Symmetric Epipolar Distance*:

For a couple of corresponding points $\mathrm{x_i} \leftrightarrow \mathrm{x'_i}$:

$$Er_i^F = d\left(x'_{i,}\ F . x_i\right)^2 + d\left(x_{i,}\ F^T . x'_i\right)^2 \qquad \textit{(eq. III-6)}$$

For all the corresponding points $\mathrm{x_i} \leftrightarrow \mathrm{x'_i} \ / \ i = 1 \dots N$:

$$Er_{ep} = \sum_{i=1}^{N} \left(d\left(x'_{i,}\ F . x_i\right)^2 + d\left(x_{i,}\ F^T . x'_i\right)^2\right) \qquad \textit{(eq. III-7)}$$

Also, other error measurements are being used, that can be found in the literature.

III.3. Proposed approach

In the real world, a general moving object has displacement, for example, from position A_{3D} to position B_{3D}. Its features correspond at both positions, as do the points along its surface. This displacement can be represented by 3D motion vectors $\{ \vec{V}_i, i: 0 \; to \; n \; \dots \}$. In a video, using a general projective camera, this object is projected on image planes (of different positions A_{2D}, B_{2D}, C_{2D} …) where each image plane (frame) is the projection of a position in 3D space at different time. Subsequently, 3D motion vectors $\vec{V}_i$ are projected to 2D motion vectors $\vec{v}_i$ from frame position A_{2D} to frame position B_{2D}, where each vector represents the displacement of a pixel from one image to another. This gives the corresponding points $x_i \leftrightarrow x_i'$, where x_i and x_i' are the two extremities of the vector $\vec{v}_i$.

The main questions here are:

1. First, how to detect object motion?
2. Then, how to find displacements of object points from frame position to another (2D motion vectors)?

Answers can be found in the motion estimation sub-section III.3.1.

When a static camera is used, the backgrounds in the frames are static. Consequently, background estimated motion in the frames will be a null vector. Moreover, because there is only one moving object in the scene, the motion-estimation will point out the object movement represented by motion vectors.

This process begins by deciding, for each temporal motion, whether the displacement between time t_1 and t_2 is deformable or not. In the case of non-deformable object motion, there will be a particular transformation to map x_i to its corresponding x'_i. This leads us to epipolar geometry and the Fundamental matrix—and, in special cases, to the Homography matrix.

Later, we will attempt to calculate this transformation. If found with a correct mapping, the temporal displacement (motion) is classified as non-deformable, else, it is considered as deformable. However, in each of the above cases, the object can be either. Thus, we studied the temporal consistency of the displacements to determine whether the object is deformable or non-deformable.

In summary, first, we detect object movements and estimate the motions vectors in the scene, using the optical flow as a motion-estimation method (explained in detail in III.3.1, below). The output from this step will be motions vectors belonging to the moving object, false vectors detected outside the moving object, falsely estimated vectors inside the moving object, and unusable motion vectors. Then, we filter these motions vectors (as explained in III.3.2, below). This step removes false, wrongly estimated, and unusable motion vectors. Only the true positive vectors belonging to the moving object remain. Next, we search for the transformation (Fundamental/Homographic matrix), if there is one, which satisfies these movements (as discussed in Subsection III.3.3). The output from this is the estimated transformation H/F. Subsequently, we determine whether the transformation correctly maps the two sets of corresponding points. By reference to this, the decision is made about the detected temporal motion as to whether it is deformable (as detailed below, in III.3.4). Finally, from the sequence of the temporal deformability of movements, we can infer the deformability of the moving object (for which, see Subsection III.3.5). This step will ultimately classify the object moving through the scene as deformable or non-deformable. We explain the sequence of procedures and the proposed algorithms in

Subsection III.3.6. Thus, in short, the proposed approach follows the five steps represented in Figure III-3:

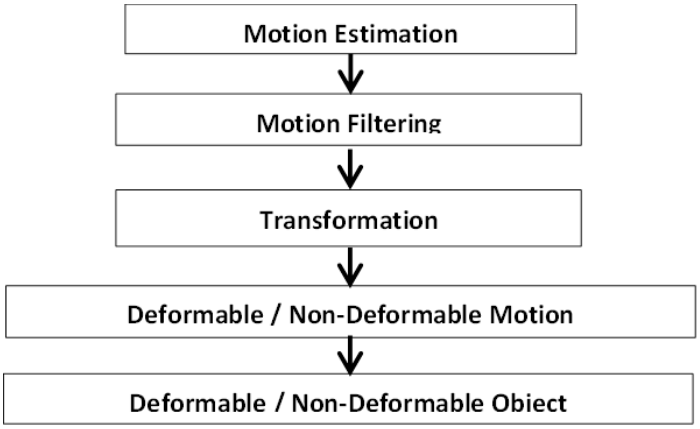


Figure III-3: Flow chart for the proposed method

III.3.1. Motion estimation

This study involves considering the projected object points on two image plans. The transformation, if any, that correctly maps the corresponding points must then be found. To this end, a reliable method is needed—one that can produce a very dense, accurate (to the extent of using sub-pixels), and regular field of vectors representing the pixel displacements of a moving object. Moreover, the capability to track each moving object pixel through frames is required. This must be combined with the ability to estimate any kind of movement, even slow movement or object rotation.

Many, mentioned in the related works, approaches for motion estimation (Optical flow approaches, feature-based approach, block matching…) were well examined and tested.

The most competitive methods, useful for this study, are the optical flow, the block matching and the feature correspondence. Conceptually, the optical flow field is a set of condensed feature matches, having one match for every image pixel. Conversely, one can view feature-correspondence methods as optical flow computation at a few selected locations, with a high probability that the optical flow will be correctly estimated. Two major differences between feature correspondences and optical flow may be identified (Fakih & Zelek, 2008):

- Feature correspondences have a higher signal to noise ratio.
- The number of reliable feature correspondences is lower than the number of optical flow values.

The problems related to methods based on the optical flow, comprising the dense sub-pixel block matching method, are mainly the computation time and the unreliability when estimating fast (i.e., large) motion. The problems with a feature-correspondence method concern the non-dense (i.e., the lack of density) and non-periodic (i.e., irregular) field of motion vectors.

Considering the optical-flow method, the time-consumption problem can be improved by parallelization when applicable. Moreover, the reliability problem in estimating large motion can be solved by introducing a pyramidal implementation, which allows for faster (i.e., larger) motion tracking.

On the other hand, the lack of density and periodicity in the motion vectors that result from using feature-correspondence methods can be overcome by constructing a dense velocity field from a sparse-correspondence-point velocity field. However, this solution is considerably difficult to apply.

In addition, for further application and interpretation of an action, it is not possible to sacrifice the availability of dense and regular information in order to avoid missing any part of the object's body. In fact, missing such an articulation leads to a misclassification of the object. Thus, every articulation of the body must be segmented to obtain further information concerning the action. Working with a dense and regular field will be beneficial when checking the deformability to determine the percentage or degree of the object's deformability. Finally, a dense and regular field may help with segmenting objects in the scene.

To achieve our main goals with best results, when no prior information about the content of the scene is available—and with a minimum number of hypotheses, assumptions, and constraints—an optical-flow method is adopted. In fact, such a method better suits this type of study, in spite of the complexity and the computational time required. Furthermore, optical-flow methods and algorithms are widely parallelizable, and they can take advantage of advances in processing technology and parallelizing systems. Moreover, the optical-flow method uses a pyramidal implementation that allows for faster motion tracking. In addition, this method deals with some remaining problems listed.

The optical-flow approach is a well-known concept that has been exploited for several years, with many techniques and a variety of methods (Barron et al., 1994).

One of the most interesting methods of working with the optical flow is the Lucas-Kanade method (Lucas, Kanade, & others, 1981). However, experiments with the Lucas-Kanade algorithm reveal that it is unsuitable for large displacements caused by the approximation when omitting higher-order terms (i.e., higher than the first terms in the optical-flow equation) (Bruhn, Weickert, & Schnörr, 2005), (Wedel & Cremers, 2011). Thus, improvements (e.g., the pyramidal approach (Burt & Adelson, 1983)) have been made to the Lucas-Kanade algorithm. Marzat (Marzat, 2008) presented a pyramidal implementation of the Lucas-Kanade method with regularized least squares[1]. To ameliorate the results, Marzat used several optimization techniques: he implemented a pyramidal approach (i.e., a multi-resolution approach) and in plus an iterative and temporal refinement. The reader is referred to (Marzat, 2008) and (Dumortier, 2009), to read more about the pyramidal representation and its advantages after implementing Lucas-Kanade method, and Marzat's pyramidal method.

Marzat (Marzat, 2008) and Dumortier, 2009 (Dumortier, 2009) conducted many comparative tests, focusing on differential techniques (viz., the Lucas-Kanade algorithm, the Horn-Schunck algorithm, and block-matching approaches). This is related to the fact that other techniques do not appear dense, nor do they use excessive filtering or many parameters. According to Marzat, his algorithm is more accurate than the Lucas-Kanade and block-matching algorithms. On one hand, as we saw, the Lucas-Kanade algorithm is unsuitable for large disparities. On the other hand, a block-matching algorithm using typical techniques, such as those explained in (Khammar, 2012), cannot give sub-pixel-wise information without

[1] It is an estimation to linearize the least squares, because the calculations with least squares risk producing an absurd estimation. So the least squares: $J_{LK}=\sum_{\Omega}[\nabla I.\vec{\omega}+I_t]^2$ became: $J_{LKL2}=\sum_{\Omega}[\nabla I.\vec{\omega}+I_t]^2+\alpha|\omega|^2$, with α adjustable, representing the regularity of the solution.

further processing. Marzat solved these problems by using scale pyramids which reduces the image resolution. The disparities gained here are then used in the higher resolution images.

Moreover, we did several comparison tests with optical-flow algorithms to identify the method that best suits our type of work.

The corpus mentioned in Section III.4 was used. We compared Marzat's algorithm with the following algorithms, found in the Computer Vision System Toolbox in Matlab: the Lucas-Kanade algorithm, the Horn-Schunck algorithm, and block matching. We determined that the solution proposed by Marzat best suits better this type of studies. Indeed, the first two algorithms (viz., Lucas-Kanade and Horn-Schunck) are decidedly unsuitable for large disparities (Horn & Schunck, 1981), (Meinhardt-Llopis, Sánchez Pérez, & Kondermann, 2013), (Bradski & Kaehler, 2008), (Michael, 1992), and the block-matching algorithm does not provide sub-pixel-wise information without further processing.

To summarize Marzat's algorithm:

- It offers more precision (i.e., sub-pixel estimation) for the motion vectors
- It does not require much filtering
- It detects both slow and fast motion
- It is more coherent and consistent
- It is completely parallelizable

Thus, Marzat's optical-flow method meets the objectives of the present study, and it was implemented in this work.

Estimating motion with Marzat's algorithm produces motion vectors belonging to the moving object, false vectors detected outside the moving object, falsely estimated vectors of the motion inside the moving object, and unusable motion vectors. An example is shown in Figure III-4. Hence estimating motion with Marzat's algorithm requires filtering to ameliorate its results and to remove unreliable motion vectors. This process is described in the following Subsection, III.3.2.

It is well known that the object-boundary motion might not always be consistent with the object's 3D motion. However, we consider this effect marginal, and it is already well filtered by Marzat's algorithm (with a smoothness effect as a result of the pyramidal approach). In addition, for the remaining unreliable object-boundary motion, we must use thresholds to determine the inconsistency that will be taken into consideration when classifying the object as deformable or non-deformable.

Figure III-4: Scene with a person (Frame 25): Marzat's algorithm applied for estimating motion.

III.3.2. Motion filtering

The purpose of this step is to eliminate all unreliable motion vectors data that have not been already filtered by Marzat's algorithm. The false-positive appearance of vector movements in uniform areas is the first case that can be detected. The second case appears when the detected vectors are parallel to the local texture (see Figure III-9), meaning that any estimation of those vectors will be erroneous. This is a typical limitation, and one that is common to all optical-flow estimation tools. The Lucas-Kanade algorithm can identify such cases when motion vectors are estimated with the help of a tensor-structure matrix of "weak rank 2" (i.e. when at least one of its eigenvalues is close to zero). In addition, especially small vectors are insignificant to further interpretation. Thus, and to avoid further critical errors in processing, it is indispensable to filter all such vectors, even if, in doing so, there is a risk of losing some true-positive vectors. For this type of work, it is better to have fewer reliable vectors than many that are unreliable. All the remaining vectors should belong to the moving object, and they should be reliable and regularly dispersed over the parts of the object. To achieve this, after detecting and estimating motion vectors with Marzat's algorithm for each pixel in the desired frame, three simple filters are used: the *small-vectors filter, uniformity filter,* and *texture filter*.

III.3.2.A. Small-vectors filter

All insignificant vectors with a very small abscissa and ordinate (<0.5) are eliminated. Indeed, these vectors are insignificant in searching for a transformation, and they can decrease the reliability of the calculated transformation. Moreover, those vectors are basically considered as unimportant motion (e.g., the motion of tree leaves), noise, or poor detections.

III.3.2.B. Uniformity filter

Where there are uniform areas (i.e., when the variation of the intensity is small) in the frame, Marzat's algorithm detects false-motion vectors. Thus, we check each of the remaining motion vectors in the desired frame after the application of the small-vectors filter, if a vector exists in a uniform area, it will be eliminated.

III.3.2.C. Texture Filter

Marzat's method uses the Lucas-Kanade approach, which is based on motion-vector estimation according to gradient calculations. That can generate false vectors estimation, especially on edges where vectors appear parallel with local texture. In this filtering, these vectors must be eliminated. Thus, for each of the motion vectors remaining after small vectors filtering and uniformity filtering, in desired frame, we find intensity's variations in the vector surrounding block, in the direction and orientation of this motion's vector. If the average of intensity's variations (differences) is low, so the motion's vector is on the same direction and orientation as the local texture; that means it is not reliable and it will be eliminated.

Applying the 3 filters on Figure III-5-c and Figure III-9-c give accordingly as results Figure III-8, and Figure III-10.

a

b

c

Figure III-5: Walking Scene: Frames references (Figure III-5-a and Figure III-5-b), and result of Marzat's algorithm applied give Figure III-5-c (without filtering).

Figure III-6: Walking Scene: Uniformity Filtering result (regular size) of Figure III-5-c; we can notice that the groups of false vectors on the left and near the boy are deleted.

Figure III-7: Walking Scene: Texture Filtering (zoomed size) of Figure III-6; we can notice that the groups of false vectors near the left foot of the boy are deleted.

a

b

c

Figure III-8: Highway 4 scene: Frames references (Figure III-8-a and Figure III-8-b), and result of Marzat's algorithm applied give Figure III-8-c (without filtering).

Figure III-9: Highway 4 scene: Uniformity Filtering result (zoomed size) of Figure III-8-c; we can notice that the false vectors around the car are deleted.

Figure III-10: Highway 4 scene: Texture Filtering result (zoomed size) of Figure III-9; we can notice that the false vectors near right doors of the car are deleted

Figure III-11:Scene with a person (Frame 25): Marzat's algorithm after filtering.

III.3.3. Transformation

The output from the motion-filtering step can be considered as two sets of corresponding points. That is, it is the perspective projections from two positions taken with a general projective camera[2] of a general moving object.

The final step in this study is to determine whether the object is deformable. It is necessary to identify deformable motion (i.e., displacements) among non-deformable motion. Thus, it is essential to begin with a kinematics theory of non-deformable bodies. For general 3D non-deformable static bodies, a well-known transformation exists between two corresponding features taken from two different camera positions at two different times. This case is equivalent to the case of one static camera taking two images of a 3D non-deformable moving body at two different times. Thus, when a non-deformable object is observed in two perspective-camera views, its feature correspondences satisfy an epipolar constraint for a general non-deformable body. The transformation that can map the correspondence of the points is called the Fundamental matrix. The study of the Fundamental matrix is part of epipolar geometry.

In order to find out whether the temporal displacement of the object is deformable, the deformability constraint of a non-deformable body motion will be identified. This can be done by finding a Fundamental matrix that is able to correctly map the set of points X_i of the object from one image plan to its corresponding X'_i in the other image. If this Fundamental matrix can be found, the displacement of the object is non-deformable; if not, then the displacement is deformable.

As stated before, when a non-deformable object is observed in two perspective-camera views, its feature correspondences satisfy an epipolar constraint for a general non-deformable-body. However, it can also be satisfied by an planar projective transformation constraint (i.e., with 2D homography) in several cases, such as for planar objects (or objects that are assumed to be planar objects), certain objects in special cases (e.g., distant or small objects), planar motion (i.e., pure translation or rotation in the image plan when the rotation axis is an orthogonal image plan), or when using a 2D camera (also known as planar camera) (orthographic). In such cases it is clear that the planar motion is a degenerate non-deformable-body motion. The Homography matrix can be successful at replacing the

[2] The case of a general projective camera as uncalibrated camera is the case of this study, seeking more generality.

Fundamental matrix in these cases, and it can be considered a simplification of the problem. It can be used for a lot of applications: camera fixed in mobile vehicle, mobile vehicle in urban or building environment and robots in movements....

The homography matrices were also tested, by a simple replacement of the fundamentals one.

With two sets of corresponding points, the Fundamental and Homography matrices can always be estimated with existing estimation methods. The problem reverts from finding a Fundamental (or Homography) matrix to determining whether the estimated Fundamental (or Homography) matrix correctly maps the corresponding points.

III.3.3.A. Homography (Projective transformation)

Various algorithms were proposed for homography estimation. In the present work, deformable motions and objects should be distinguished from non-deformable ones. To this aim, an estimation algorithm will be used to estimate the homography that relate the corresponding points of the objects, either if the object is deformable or is not-deformable; and results will be compared. Thus, a comparison method with the same estimating algorithm will be employed. This will attenuate the result errors effects.

Thus, in this study, we used the well-known DLT[3] algorithm coupled with normalization[4] by Hartley and Zisserman (R. Hartley & Zisserman, 2003b) for estimating the homography . The DLT estimates the Homography matrix, given two sets of corresponding points. The normalized DLT (NDLT) algorithm is a linear homogeneous solution based on minimizing a suitable cost function to numerically solve the linear equations of the Homography.

The solution proposed by the NDLT is the method of least squares using Singular Value Decomposition (SVD) (Abdel-Aziz & Karara, 1971).

Nb: As a code for NDLT, we used the Matlab function *vgg_H_from_x_lin* (Zisserman et al., 2012).

III.3.3.B. Fundamental Matrix

General projective camera, which is as uncalibrated camera, is studied in this work seeking more generality. Two perspective views (two images) are considered: right and left. Beside this, set of points in one image and its correspondences in the other are also represented. The aim in this part is to find the transformation that can map each of the points in one image to its correspondence, in the other image.

In this study, the normalized 8-point algorithm (N8PA) is used to estimate the Fundamental matrix, because it provides adequate results and because it is quick and easy to implement. The 8-point algorithm was first introduced by H. Christopher Longuet-Higgins (Longuet-Higgins, 1981), and then coupled with the normalization by Hartley and Zisserman (R. Hartley & Zisserman, 2003b).

The N8PA estimates the Fundamental matrix, given two sets of corresponding points. The starting equation is different from the one used with the homography but is solved using the same main steps (minimizing a suitable cost function to numerically solve the linear equations and least squares using Singular Value Decomposition) (R. Hartley & Zisserman, 2003b).

Nb: As a code for N8PA, we used the Matlab function *fundmatrix* (Kovesi, n.d.).

[3] The Direct Linear Transform (DLT) algorithm was introduced by Abdel-Aziz and Karara (Abdel-Aziz & Karara, 1971).

[4] The normalization step was introduced by Hartley (R. I. Hartley, 1997).

III.3.4. Deformable and Non-Deformable Motions

After calculating the optical flow between two frames using Marzat's algorithm—and after filtering the motion fields and estimating the Fundamental matrix *F* and the Homography matrix *H* that relate the corresponding points ($x_i \leftrightarrow x'_i$ / *i= 1… N*) covering the object in the two image plans in all cases (i.e., for both deformable and non-deformable objects)—the motion deformability can be investigated. Through this investigation, it is possible to determine whether the projected displacement of the object from position A_{3D} to position B_{3D} (between times t_1 and t_2) is a deformable or non-deformable displacement.

The investigation uses the property of non-deformable body motion discussed above (in Subsection III.3.3). In the case of non-deformable motion, its feature correspondences can satisfy a Fundamental matrix (in general cases) or a Planar-Projective transformation (in special cases). Thus, *F* (or in special cases, *H*) can be checked. A correct mapping of the corresponding points means that the motion is non-deformable, according to the matrix *F* (or *H*) that is found. Otherwise, the motion is deformable. As a consequence, the problem now is to determine whether the transformation (*F* or *H*) is, in fact, a correct mapping of the corresponding points.

Ideally, the transformation *F* (or *H*) is a perfect mapping of the corresponding points, and *F* (or *H*) can correctly map all *N* points x_i to x_i' and vice versa:

- For H: x'_i should coincide with $H.x_i$for all the N points x_i ($x'_i = H.x_i \;/\; i = 1 \dots N$) and x_i should coincide with $H^{-1}.x'_i$ for all the N points x'_i ($x_i = H^{-1}.x'_i \;/\; i = 1 \dots N$).
- For F: the case is a little bit different: x'_i should be on the epipolar line l'_i corresponding to x_i for all the N points x_i ($l'_i = F.x_i \;/\; i = 1 \dots N$) and x_i on the epipolar line l_i corresponding to x'_i for all the N points x'_i ($l_i = F^T.x'_i \;/\; i = 1 \dots N$).

However, because of real images errors, the position of the points x_i and x_i' is disturbed by noise. Image-measurement errors will occur in both images and the estimation of *F* and *H* will not be perfectly accurate. In such cases, two issues are taken into consideration:

- *The error margin when mapping the corresponding points* ($x_i \leftrightarrow x'_i$): establishing the acceptable error in mapping x_i to x_i' and vice versa. For that, we calculate the error for each two corresponding points ($x_i \leftrightarrow x'_i$:
  - For *H*: by the Symmetric Transfer Error, described in Subsection III.3.3, or by the mean distance (error):

$$Er'^{H}_{i} = (d(x'_i, H.x_i) + d(x_i, H^{-1}.x'_i))/2 \quad \textit{(eq. III-8)}$$

  - For *F*: by the Symmetric Epipolar Distance (error), described in Subsection III.3.3, or by the mean distance (error):

$$Er'^{F}_{i} = (d(x'_i, F.x_i) + d(x_i, F^T.x'_i))/2 \quad \textit{(eq. III-9)}$$

  Finally, for two corresponding points ($x_i \leftrightarrow x'_i$:
  - For *H*: if $Er^H_i \leq \gamma_H$ (respectively, $Er'^H_i \leq \gamma'_H$) then H correctly maps the couple $x_i \leftrightarrow x'_i$, and it does not if $Er^H_i > \gamma_H$ (respectively, $Er'^H_i > \gamma'_H$) for the Symmetric Transfer Error (respectively, the mean distance).
  - For *F*: if $Er^F_i \leq \gamma_F$ (respectively, $Er'^F_i \leq \gamma'_F$) then F correctly maps the couple $x_i \leftrightarrow x'_i$, and it does not if $Er^F_i > \gamma_F$ (respectively, $Er'^F_i > \gamma'_F$), for the Symmetric Epipolar Distance (respectively, the mean distance).

where γ_H, γ'_H and γ_F, γ'_F are the *mapping error thresholds* for *H* and *F*, respectively. γ_H, γ'_H and γ_F, γ'_F are later calculated in order to apply the thresholds to all types of objects and movements.

- *The error margin in the percentage of correctly mapped points:* establishing the acceptable percentage of points that are not correctly mapped, even though both sets of corresponding points are considered to be correctly mapped, in general.

As mentioned above, an error will occur when estimating *F* or *H* in mapping the corresponding points. Thus, it is not the case that all the *N* points in the two sets of corresponding points will be correctly mapped with *H* or *F*. To consider the two sets of corresponding points $x_i \leftrightarrow x'_i$ / *i = 1…N* as correctly mapped, it will be sufficient if the percentage of the number of correctly mapped points is more than a certain threshold: $\delta_F = (N'_F.100 / N)$ for *F*, or $\delta_H = (N'_H.100 / N)$ for *H; with* N'_F <=N and N'_H <=N are the number of correctly mapped points corresponding to F and H.

Moreover, δ_F and δ_H should be generalized as much as possible so that they can be applied for all types of objects and movements. This must be accomplished in such a way that, whatever the object is, and for any temporal movement between the two frames im_{n-1} and im_n, the two sets *2N* of filtered and corresponding points can be found. Then, *F* (and, respectively, *H*) can be estimated; subsequently, the percentage of correctly mapped corresponding points (p_F for *F,* and p_H for *H*) can be found. Finally, if $p_F >= \delta_F$ ($p_H >= \delta_H$), then the two sets of corresponding points are correctly mapped by F or H, and the transformation represents a correct mapping of the two sets of corresponding points between im_{n-1} and im_n. As a result, the temporal movement of the object is classified as non-deformable motion. Else, it is deformable. In other words, p_F (p_H) can be seen as the non-deformability percentage of the temporal movement of an object for *F (H)*; subsequently, δ_F (δ_H) is the *motion non-deformability threshold* for *F (H)*. The values δ_F and δ_H are calculated as per the method described in Section III.4.

Because δ_F and δ_H should be generalized as much as possible so that they can be applied to all types of objects (deformable, non-deformable, small, medium, and large, with texture, smooth, etc.) and movements (slow, medium, fast, small, large, in all directions, etc.), the *F* or *H* motion non-deformability thresholds δ_F and δ_H should be investigated as to whether they can be affected by the following two parameters:

- *Number of motion vectors:* the number of couples among the corresponding points. Several tests were conducted with different scenes and scenarios, by taking several sets of random vectors in detected objects to determine whether the number of motion vectors needed to calculate the Fundamental *F* or the Homography *H* can seriously affect δ_F and δ_H. Moreover, the correlation was calculated.
- *Average Length of the Motion Field (*ALMF*):* the average motion-vector length. Several tests were conducted by taking the same moving object and similar motion with different vector lengths, resulting in different average lengths.

Based on tests, it was clear that the number of motions vectors does not seriously affect the motion non-deformability thresholds δ_F and δ_H for *F* and *H,* respectively. This conclusion is evident because only a few vectors (four vectors for *H*, and eight vectors for *F*) are needed to define and represent the true temporal displacement of the object. Thus, the density of the motion field can be reduced in order to diminish the time required for filtering. This can be done without losing the regular dispersity needed to cover the entire object, and thus

without losing any important information about the object.

Concerning the length of the motion, initially in the experiments, the mapping error threshold for *H* (respectively, *F*) is fixed regardless of the motion length, and different lengths of motion can significantly affect δ_F and δ_H in such a way that the smallest average length of the motion field will have the highest motion non-deformability threshold for *H* (respectively, *F*) (see Section III.4) to maximize the results (i.e., to minimize the errors in discriminating between non-deformable and deformable).

Therefore, as the generality is pertinent to solving the problem of deformable and non-deformable motion with different types of videos, different objects, and objects moving at different speeds with different motion-field lengths. Thus, a normalization step is added to normalize the length of the motion field after motion filtering and before calculating the transformations. For that, all motion vectors are normalized to an average motion-field length equal to *n* (*n=1, 2, 3, 4 … round (*original average length*)*).

The Fundamental F_n and the Homography H_n were calculated for each of normalized set of vectors corresponding to normalization level *n*. Therefore, instead of finding the motion non-deformability threshold δ_F for *F* (and, respectively, δ_H for *H*), a set of thresholds δ_F^n (respectively, δ_H^n) must be found: one for each normalization level *n*: $\delta_{\mathrm{F}}^{\mathrm{n}} = \{\delta_{\mathrm{F}}^{1}, \delta_{\mathrm{F}}^{2}, \delta_{\mathrm{F}}^{3} \dots \delta_{\mathrm{F}}^{\mathrm{i}} \dots \delta_{\mathrm{F}}^{\mathrm{n}}\}$(respectively, $\delta_{\mathrm{H}}^{\mathrm{n}} = \{\delta_{\mathrm{H}}^{1}, \delta_{\mathrm{H}}^{2}, \delta_{\mathrm{H}}^{3} \dots \delta_{\mathrm{H}}^{\mathrm{i}} \dots \delta_{\mathrm{H}}^{\mathrm{n}}\}$).

N.B.: δ_F^n and δ_H^n are used with the Symmetric Epipolar Distance and the Symmetric Transfer Error, respectively, but when using the mean distance, we must find $\delta'^{\mathrm{n}}_{\mathrm{F}} = \{\delta'^{1}_{\mathrm{F}}, \delta'^{2}_{\mathrm{F}}, \delta'^{3}_{\mathrm{F}} \dots \delta'^{\mathrm{i}}_{\mathrm{F}} \dots \delta'^{\mathrm{n}}_{\mathrm{F}}\}$(respectively, $\delta'^{\mathrm{n}}_{\mathrm{H}} = \{\delta'^{1}_{\mathrm{H}}, \delta'^{2}_{\mathrm{H}}, \delta'^{3}_{\mathrm{H}} \dots \delta'^{\mathrm{i}}_{\mathrm{H}} \dots \delta'^{\mathrm{n}}_{\mathrm{H}}\}$).

In summary, in seeking to maximize the study's results by finding the ultimate motion non-deformability thresholds for discriminating deformable and non-deformable motion, two essential parameters had to be considered:

- The normalization level.
- The mapping error threshold.

For each normalization level n, the mapping error threshold (γ_F^n, γ'^n_F for *F,* and γ_H^n, γ'^n_H for *H*) must be found in a way to lead to the ultimate motion non-deformability threshold (δ_F^n, δ'^n_F for *F,* and δ_H^n, δ'^n_H for *H*).

It should be noted here that, for small object movement, deformable motion can be confused with non-deformable motion in the real world. Furthermore, the length of the motion vectors and the difference in length among motion vectors are very small. Thus, the Fundamental matrix *F*, the Homography *H,* and the motion non-deformability thresholds are unreliable, which will raise the percentage of errors when discriminating between deformable and non-deformable motion. Moreover, by having especially long movement vectors, errors in estimating the motion vectors (with the optical flow) and in estimating *F* and *H* will be duplicates, and the motion non-deformability thresholds will be unreliable, which will again increase the percentage of errors.

Following the experiments, and in order to obtain reliable results, the Average Length of Motions Field ALMF should fall between seven and ten. For that, the motion vectors inputted during the third step (viz., transformation) should have an average length of between seven and ten. By changing (i.e., by eloigning or approaching) the input-compared frame im_i (i.e., the frame compared with the current frame im_n) and repeating (i.e. reiterating) the first and the second steps (viz., motion estimation and motion filtering), the desired average length of the motion field can be obtained. For example: instead of

comparing frame X_n with frame X_{n-1}, we can compare it with frame X_{n-2} or X_{n-3}, or it can be compared with another frame until a suitable result is found.

Consequently, after finding the filtered motion-vectors field, the field is normalized, and for each normalization *n,* the Fundamental matrix F_n is estimated (and, respectively, the Homography matrix H_n) to relate the normalized corresponding points ($\bar{x}_i^n \leftrightarrow \bar{x}'^n_i$ /i= 1… N/ n=1, 2, 3…). Then, for any normalization level, the percentage of correctly mapped points (p_F^n for *F*, and p_H^n for *H*) is calculated using the mapping error threshold that is already known (γ_F^n for *F*, and γ_H^n for *H*). Then, p_F^n (respectively, p_H^n) is compared with the motion non-deformability threshold that is already known (δ_F^n for *F* and δ_H^n for *H*). If $p_F^n >= \delta_F^n$, then the motion (i.e., temporal movement under testing) is non-deformable regarding *F*, and if $p_H^n >= \delta_H^n$, then the motion is non-deformable regarding *H*; otherwise, when $p_F^n < \delta_F^n$ (or respectively, when $p_H^n < \delta_H^n$) the motion is deformable.

In the section III.4, we describe several experiments and an intriguing method of searching for thresholds using a new type of graph—called the "Best Maximum – Acceptable Minimum Graph" (see Section 4). For each normalization (*1* to *10*), each transformation (Fundamental and Homography), and each type of error (mean distance, Symmetric Transfer Error, or the Symmetric Epipolar Distance), the ultimate couple mapping error threshold (γ_F^n, γ'^n_F for *F* and γ_H^n, γ'^n_H for *H*) and the motion non-deformability threshold (δ_F^n, δ'^n_F for *F* and δ_H^n, δ'^n_H for *H*) are found in a way that maximizes the success (the *percentage of success*) of the algorithm. The ultimate thresholds are shown the Table III-1. Furthermore, we describe the processes in the algorithm (Subsection III.3.6), and some experiments and new methods of searching for ultimate thresholds in Section III.4.

Table III-1: Table of ultimate thresholds: (a, b): a is the mapping error threshold, and b is the motion non-deformability threshold; below these thresholds is the corresponding percentage of success (%).

Transformation		Normalization	1	2	3	4	5	6	7	8	9	10
H	Mean distance	$(\gamma'^n_H, \delta'^n_H)$:	(1.4, 87.8)	(2.4, 84.07)	(3.8, 85.66)	(5, 85.28)	(6.4, 85.84)	(7.4, 84.65)	(9, 85.29)	(10.8, 86.06)	(13, 86.69)	(14, 85.11)
		%S:	73.42	73.57	73.85	74.32	74.29	74.23	75.03	72.87	71.53	73.9
F	Mean distance	$(\gamma'^n_F, \delta'^n_F)$:	(0.6, 83.36)	(1, 79.13)	(1.4, 76.92)	(1.8, 76.04)	(2.2, 75.52)	(2.8, 76.65)	(3.8, 80.91)	(4.8, 82.56)	(5, 81.06)	(6, 90.76)
		%S:	81.56	82.16	82	82.05	82.04	82.32	82.93	82.68	80.98	80.51
H	Symmetric Transfer Error	(γ^n_H, δ^n_H):	(3, 84.53)	(12.6, 85.19)	(28.4, 85.3)	(50, 85.28)	(78, 85.16)	(120, 85.43)	(160, 85.14)	(230, 85.8)	(345, 86.89)	(350, 83.66)
		%S:	73.86	74.48	74.62	74.32	74.51	74.23	75.33	72.75	71.63	74.15
F	Symmetric Epipolar Distance	(γ^n_F, δ^n_F):	(0.6, 81.31)	(2.2, 80.16)	(3.8, 76.6)	(6.6, 76.25)	(15, 81.31)	(25, 83.08)	(35, 83.02)	(47, 82.61)	(51, 81.26)	(72, 81.18)
		%S:	81.8	82.58	82.09	82.41	82.79	82.92	83.12	82.64	81.45	82.44

III.3.5. Deformable and Non-Deformable Objects

Until now, object motion has been classified independently for each frame (i.e., according to the movement or displacement in the frame), but frame-by-frame detection results in many classification errors. Moreover, whether the temporal displacement is

deformable or non-deformable does not necessarily indicate that *all* its motion will be of that sort. It is not the case that objects deformability can be inferred from a single motion exclusively. For these reasons, the entire series of the object's apparent motion should be considered.

When an object appears in frames, the series of its temporal motion (Motions[5] : X_i, X_{i+1}, X_{i+2} ... X_j, ... X_n) will be studied and classified as deformable or non-deformable motion (like $D_i, D_{i+1}, ND_{i+2} \cdots, D_j \cdots, ND_n$).

Two criteria should be taken into consideration:

- *Errors in classifying the temporal motion:* the temporal motion can be misclassified owing to errors in the motion-estimation algorithm and transformation-estimation errors caused by image noise, etc.
- *A deformable object can have non-deformable motion:* A deformable object can be mistaken for a non-deformable object if it acts (i.e., appears) as a non-deformable object for an extended period. This occurs when the articulations of a deformable object are hidden at the time of motion or have the same displacements as an entire body, A good example could be when only the upper-body of a person (deformable object) appears and his hands moves as the same way as his upper-body (he's moving as one block). This can happen in one frame, or it can be consistent (e.g., a deformable object can be in motion for a long time with its articulations hidden); but this does not mean that the object is non-deformable. It should be noted that the opposite case is never true, because non-deformable objects cannot have deformable motion.

As a result, several cases must be considered:

- In the series of motion classifications, if a motion is classified as **deformable motion**, then—if this is said of a **deformable object—**the classification is correct and should be left unchanged (i.e., it should not be corrected), but if **deformable motion** is said of a **non-deformable object,** then the classification is an error and should be corrected to **non-deformable motion**.
- In the series of motion classifications, if a motion is classified as **non-deformable motion**, then—if this is said of a **non-deformable object**—the classification is correct and should left unchanged (i.e., it should not be corrected), but if **non-deformable motion** is said of a **deformable object,** then two cases appear:
  - Either, the classification is an error and should be corrected to deformable.
  - Or, the classification is correct and:
    - Either, the motion classification should be left unchanged, so that it remains **non-deformable motion**, taking into consideration that this is a case of *consistent* non-deformable motion from a **deformable object**.
    - Or, the motion classification should be changed to **deformable motion**, even though this is known to be untrue from a temporal point of view. It is, however, true from a general point of view for classifying the object per se as deformable, because this is the case of *inconsistent* non-deformable motion (in a one or two frames) of a **deformable object**.

[5] The motion (i.e., the temporary motion) is denoted according to the frame of its motion vectors and the destination frame. For example, the displacement of the object from frame X_k (the suitable corresponding frame of X_j for the study) to frame X_j ($X_k \rightarrow X_j$) is called motion X_j.

Therefore, for all of these reasons:

- At first, the temporal consistency will be studied when the motion is classified for all series of movements to correct classification errors, and to exclude the inconsistent non-deformable movements in a deformable object.
- Second, object deformability will be inferred from the corrected (persistent) series of motion classifications, taking into consideration the consistent non-deformable motions of a deformable object.

The temporal information can be used to study the consistency of the motion-classification results and will lead to an improvement in the reliability of decisions over time.

In this study, we used the temporal-consistency algorithm proposed by Jaffré and Joly (Jaffré & Joly, 2005). Their goal was to improve the results obtained for an object detector operating independently on each frame of a video document; the results for the object detector are "smoothed" along the time dimension using a temporal window.

In order to reduce false detections (i.e., false alarms and incorrect detections), Jaffré and Joly (Jaffré & Joly, 2005) propose exploiting the persistence properties of objects in a video sequence. They consider a temporal window of *N* frames centered on the subject frame. For this given frame, they count the number of occurrences of each object in the previous *N/2* and the subsequent *N/2* frames. Then, only the objects whose number of appearances is above a threshold (N_2) are validated. A probabilistic approach is used for a theoretical computation of the thresholds *N* and N_2.

To validate all the correct detections, and reject all the false alarms, they search N and N_2 so as to maximize:

$$\begin{aligned}\arg\max_{N,N_2} P[(X \geq N_2) \cap (Y < N_2)] &= \arg\max_{N,N_2} P(X \geq N_2)\, P(Y < N_2) \\ &= \arg\max_{N,N_2} \left(\sum_{i=N_2}^{N} P(X=i)\right)\left(\sum_{i=0}^{N_2-1} P(Y=i)\right) \\ &= \arg\max_{N,N_2} \left(\sum_{i=N_2}^{N} C_N^i p_d^i q_d^{N-i}\right)\left(\sum_{i=0}^{N_2-1} C_N^i p_f^i q_f^{N-i}\right)\end{aligned} \qquad \textit{(eq. III-10)}$$

Having X =Number of correct detections in N frames / Y =Number of false alarms in N frames.
p_d is the probability of success/ $q_d = 1 - p_d$ is the probability of failure.
p_f is the probability to have a false alarm in a frame/ $q_f = 1 - p_f$.
Or they proposed to find N and N_2 by maximizing recall and precision:

$$\begin{aligned}&\arg\max \alpha \times recall + (1-\alpha) \times precision \\ &= \arg\max_{N,N_2} \alpha P(X \geq N_2) + (1-\alpha)\frac{P(X \geq N_2)}{1 + P(X \geq N_2) - P(Y < N_2)}\end{aligned} \qquad \textit{(eq. III-11)}$$

Having:

$$Recall = \frac{number\ of\ correct\ detected\ objects}{number\ of\ objects\ to\ find} \qquad \textit{(eq. III-12)}$$

$$Precision = \frac{number\ of\ correct\ detected\ objects}{number\ of\ detected\ objects} \qquad \textit{(eq. III-13)}$$

But they are lead to the same results of the first equation.
They proposed a numerical resolution for the maximization of the two expressions.

The aim in our study, however, is not object detection or face localization in a shot, but rather classification. Strictly speaking, then, in our study there are no misdetections, but only correct or false classifications. Misclassifications (i.e., false classifications) will merely be a failure in the classification. As a result, the algorithm in this work was subject to a few modifications: e.g., X and Y used in the maximization equation are assumed to be independent in (Jaffré & Joly, 2005), which is not the case in our study. Therefore the equation becomes:

$$\begin{aligned} & \underset{N,N_2}{Arg\ max}\, P[(Y < N_2)\ \cap (X \geq N_2)\ ] \\ = & \underset{N,N_2}{arg\ max}\, P(X \geq N_2)\, P((Y < N_2)/(X \geq N2)) \\ = & \underset{N,N_2}{arg\ max}\, P(X \geq N_2)\, P((Y < N_2)/(N - Y \geq N2)) \\ = & \underset{N,N_2}{arg\ max}\, P(X \geq N_2)\, P((Y < N_2)/(Y \leq N - N2)) \end{aligned} \qquad \textit{(eq. III-14)}$$

where *X* is the number of correct classifications in *N* frames, and *Y* is the number of false classifications in *N* frames. In (Jaffré & Joly, 2005), p_d is the probability of success, $q_d = 1 - p_d$ is the probability of failure, p_f is the probability of a false alarm in a frame, and $q_f = 1 - p_f$; in our case (without misdetection), $p_f = q_d = q$, and $q_f = p_d = p$.
NB: while the problem has been change to a classification, and, X and Y are not completely independent, in addition, Y=N-X, it seems enough to maximize only the X term without the Y term ($\arg\max_{N,N_2} P[(X \geq N_2)\ ]$). But the need of both of the terms (X and Y) in the maximization equation is proved in the Appendix VIII.3.
Also, as reminder, P(A/B) it's the conditional probability, sometimes denoted $P_B(A)$, it's the probability of A knowing that B have occurred.

Now, $P(X \geq N_2) = \sum_{i=N_2}^{N} C_N^i p^i q^{N-i}$ and $P((Y < N_2)/(Y \leq N - N_2))$ will generate two cases: (N_2<= N-N_2 => N_2<=N/2) or (N_2>N-N_2 => N_2>N/2):

- If N_2<=N/2, then $P((Y < N_2)/(Y \leq N - N_2)) = \sum_{i=0}^{N_2-1} C_N^i q^i p^{N-i}$.

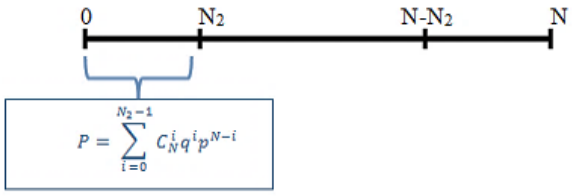


Figure III-12: The case N_2<=N/2.

- If N_2>N/2, then $P((Y < N_2)/(Y \leq N - N_2)) = \sum_{i=0}^{N-N_2} C_N^i q^i p^{N-i}$, because the $P((Y < N_2)/(Y \leq N - N_2))$ when N-N_2 <Y<N_2 is equal to *0*.

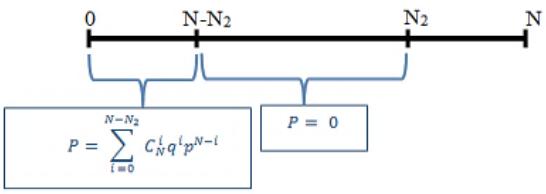


Figure III-13: The case N_2>N/2

Finally, the maximization equation will be:

$$\underset{N,N_2}{Arg\ max}\, P[(X \geq N_2)] \begin{cases} \underset{N,N_2}{arg\ max}\left(\sum_{i=N_2}^{N} C_N^i p^i q^{N-i}\right)\left(\sum_{i=0}^{N_2-1} C_N^i q^i p^{N-i}\right) \text{ / If } N2 \leq N/2 \\ \underset{N,N_2}{arg\ max}\left(\sum_{i=N_2}^{N} C_N^i p^i q^{N-i}\right)\left(\sum_{i=0}^{N-N_2} C_N^i q^i p^{N-i}\right) \text{ / If } N2 > N/2 \end{cases} \quad \textit{(eq. III-15)}$$

To find the optimal values for *N* and N_2 that suit our aim, the numerical resolution proposed in (Jaffré & Joly, 2005) was used to maximize the expression, having a probability of success *p = 82.58* and a probability of failure *q= 100-82.58= 17.42*. We selected in Table III-1 the case of the Fundamental matrix as the transformation, the Symmetric Epipolar Distance as the distance measure, normalization level two, the mapping error threshold γ_F^2 *=2.2*, and the motion non-deformability threshold δ_F^2 *= 80.16*.

However, the set of solutions was a plateau (see Appendix VIII.3), and a solution was found that can be generic to several applications. Thus, the couple (*N, N_2*) taken is: (*N, N_2*) = (*11, 6*), where, $P[(Y < N_2) \cap (X \geq N_2)]$ is maximized to *0.988454*).

Then, to study the temporal consistency of a series of motion classifications, for each classification of motion deformability, we consider a window of 11 classifications (5 to the left, from before; 1 under consideration; and 5 to the right, from after). There are then four possible cases. If the **desired classification** (i.e., the classification result, under consideration, that we are studying in terms of its consistency) is **deformable** (or, respectively, **non-deformable**), the number of deformable-motion classifications (Nb) between the 11 motion classifications is counted:

- If *nb >= 6*, the classification remains **deformable** (respectively, **non-deformable**).
- If *nb < 6*, the classification should be changed to **non-deformable** (respectively, **deformable**).

When this temporal-consistency algorithm was applied once to the corpus (on 24 series from different videos), the *percentage of success* for the entire algorithm increased by more than 6%, resulting in an 89% accuracy rate (see the examples in Table III-2, below).

Table III-2: Results from the temporal consistency-amelioration testing on 75 different videos (2141 frames), taking the Fundamental matrix, the Symmetric Epipolar distance, and the normalization level 2, where γ_F^2 =2.2 and δ_F^2 = 80.16.

	% of true classification	% of false classification as deformable	% of false classification as non-deformable
Before temporal consistency	**82.58**	**9.06**	**8.36**
After temporal consistency	**89.025**	**6.025**	**4.95**

The outputs from the temporal-consistency algorithm are the corrected and smoothed series of motion classifications. However, in case there are still isolated classifications after applying the temporal-consistency algorithm, this algorithm can be reiterated as needed, until the final output is completely smooth and unchangeable (i.e., stable). When temporal-consistency was applied, it increased the percentage of success by more than 6%, see the examples in Table III-2 above. When applied a second time, the *percentage of success* (percentage of true classification) increased to more than 91.8%. By

taking into account working with that big variety of videos and those very hard scenes taken (critical scenes, see section III.4), and knowing that the 8.2% of errors were the errors of Marzat's algorithm cumulated with the errors of filtering, the errors of estimating the transformation and the errors of classification of motion deformability, this percentage is considered as a good percentage relatively.

Nb: Supposing that a 3D object seen by a camera can be considered a planar object, the Homography matrix can be used, but in such cases, the *percentage of success* will be 75% without temporal consistency, and 80% with temporal consistency.

The final step in classifying the object is simple. We classify the object as deformable or not by looking on the motion-classification series of its appearance. If all the persistent motion classifications are non-deformable, then the object is non-deformable. Yet, the existence of one sub-series of deformable classifications is sufficient for the object to be classified as deformable.

III.3.6. Proposed Algorithms

Having provided a thorough explanation of each step, we turn now to the process sequences used to determine deformable and non-deformable motions and objects, as detailed in the following algorithms:

Deformable and non-deformable motion algorithm

Step 1: Estimate motion between two frames using Marzat's algorithm (Subsection III.3.1). This generates a motion field.

Step 2: Filter the motion field using three filters (Subsection III.3.2).

Step 3: If the average length of the motion field (ALMF) is *not* between 7 and 10 (Subsection III.3.4), Steps 1 and 2 should be repeated after changing (by eloigning or approaching) the input frame that is being compared with the current frame until the average length of the motion field falls between 7 and 10.

Step 4: Normalize the motion field to obtain normalized corresponding points (Subsection III.3.4).

Step 5: Estimate the transformations—i.e., the Fundamental matrix F_n (respectively, the Homography H_n) corresponding to each normalization level (Subsection III.3.3).

Step 6: Calculate the percentage of correctly mapped points (p_F^n for F respectively p_H^n for H) corresponding to each normalization level (Subsection III.3.4).

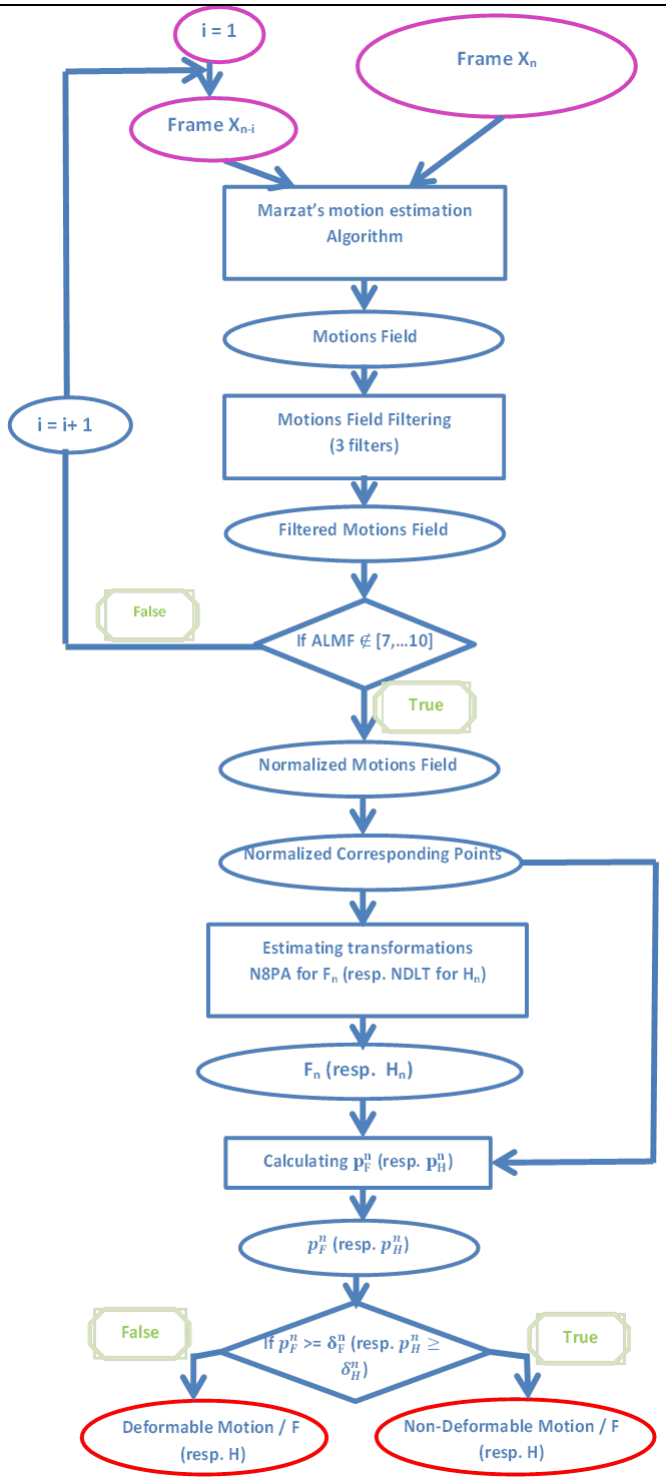
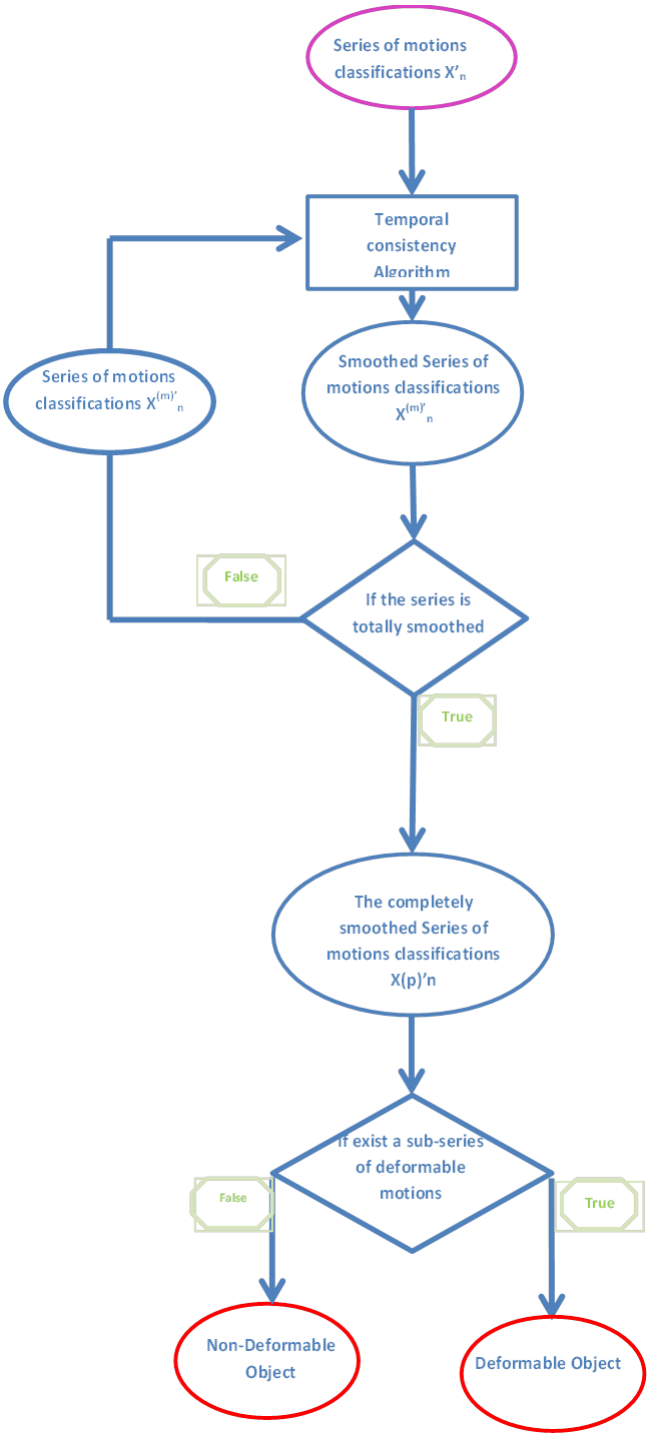
Step 7: If $p_F^n \geq \delta_F^n$ (resp. $p_H^n \geq \delta_H^n$) then the motion is non-deformable, given that δ_F^n and δ_H^n are the mapping error thresholds for F_n and H_n corresponding to each normalization level (Subsection III.3.4). Otherwise, if $p_F^n < \delta_F^n$ (resp. $p_H^n < \delta_H^n$), then the motion is deformable.

Deformable and non-deformable object algorithm (Subsection III.3.5)

Step 1: Apply the temporal-consistency algorithm to the series object-motion classifications. This results in the application of the deformable and non-deformable motion algorithm to all moving objects appearing in the scene.

Step 2: If the smoothed motion-classification series results from Step 1 are not totally smooth, then Step 1 is repeated on this new series, until we obtain a final smoothed and unchangeable (i.e., stable) series.

Step 3: If a sub-series of deformable motion exists in the final smoothed and stable motion-classification series, then the object is deformable. If not, the object is non-deformable.

Deformable/non-Deformable Motion Algorithm	*Deformable/non-Deformable Object Algorithm*
	

III.4. Experiments

This section describes the experiments conducted in order to determine the thresholds (viz., the mapping-error threshold and the motion non-deformability threshold) for each normalization level, distance type, and transformation type. Thresholds are essential for discriminating between deformable and non-deformable motion, and they affect the percentage of success corresponding to each case.

Because we could not find any real public dataset particularly dedicated to the study of deformable and non-deformable object classification with which to compare our proposed method, we created our own dataset (Youssef, 2015). By filming some videos, and collecting others from real video-surveillance cameras. In addition, we did not find, in any related research mentioned in Section III.2, any source code that could be used to test our dataset for the purpose of comparison. We should also mention that, because many of the videos used in our experiments were taken from real police video-surveillance cameras, they were not diffusible.

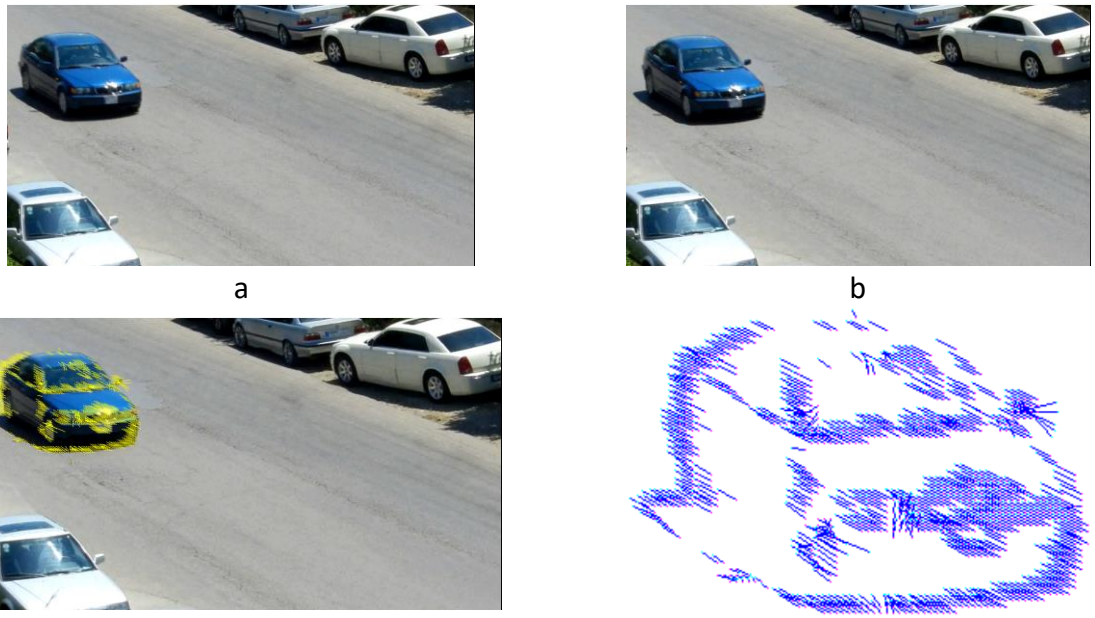
We tested our approach on 75 colour videos containing 30 different scene types with more than 2,100 tested frames. All videos were taken using a static camera.

A large variety of scenes was taken into consideration:
- Many types of scenes.
- Deformable and non-deformable objects.
- Different resolutions.
- Different kind of actions (running, fighting, rolling, crashing…).
- Different speed of action (slow or fast).
- Different luminosities (indoor and outdoor).
- Different distances (close and far scenes).

The objects in these videos had diverse properties. They differed in:
- Nature: there were adults, children, cars, doors, chairs, maps, boxes, bicycles, cats, and dogs.
- Size relative to the screen: from small to large.
- Distance from the camera: from close to far.
- Depth: there were 2D-like objects (e.g., maps) and 3D objects (e.g., cars).
- Motion speed: from slow (e.g., a person walking) to very fast (e.g., a vehicle on the highway).
- Nature of movement: translation, rotation, forward/backward, and skewed.
- Lighting conditions: from low lighting to balanced and well-contrasted scenes.

Thus, this dataset is considerably diverse, and this makes it ideal for our purposes, insofar as the motion in these videos is especially difficult to classify.
Here, in Figure III-14, Figure III-15, and Figure III-16 we describe three typical examples: The "Highway 2" scene, the "Walking" scene and "Bomb2" scene.

A. Scene from “Highway 2”: a non-deformable object

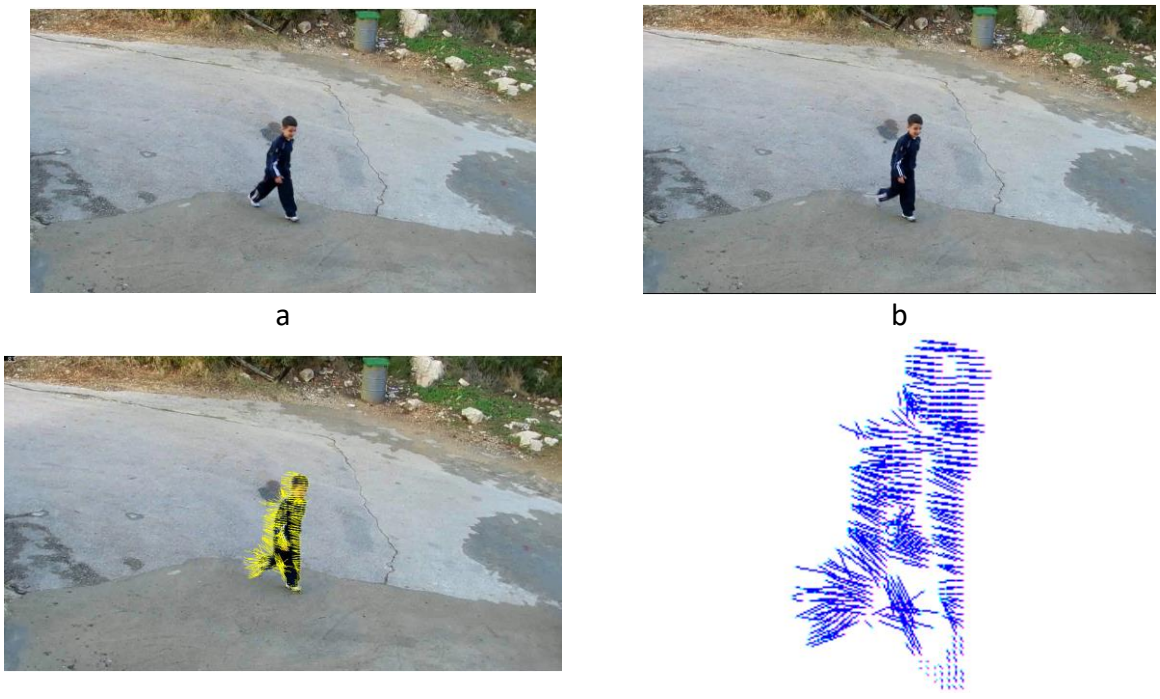


a b

c d

Figure III-14: Scene from “Highway 2”: (a) Frame 97, (b) Frame 96, (c) Motion vectors, (d) Zoomed motion vectors (755×2 corresponding points).

B. Scene from “Walking”: a deformable object with deformable motion

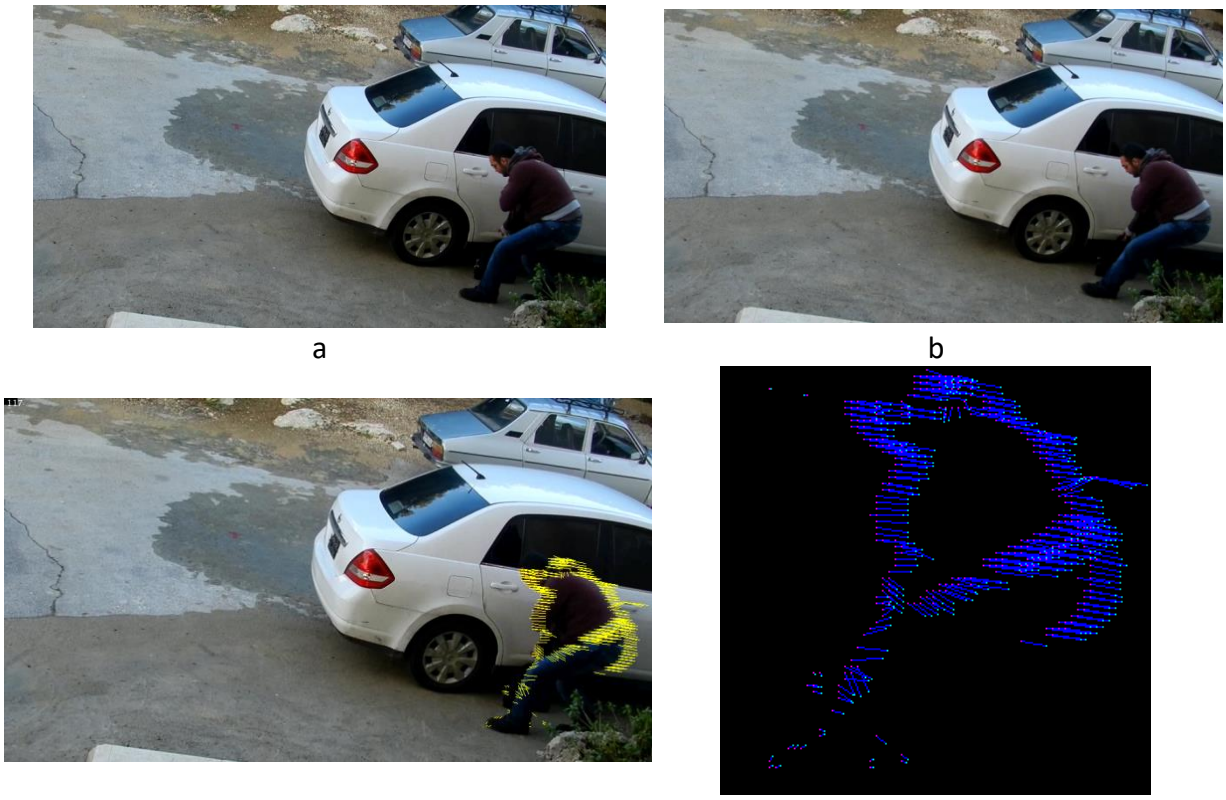


a b

c d

Figure III-15: Scene from “Walking”: (a) Frame 83, (b) Frame 80, (c) Motion vectors, (d) Zoomed motion vectors (365×2 corresponding points).

C. Scene from “Bomb2”: a deformable object with a non-deformable motion



a b c d

Figure III-16: Scene from “Bomb2”: (a) Frame 117, (b) Frame 114, (c) Motion vectors, (d) Zoomed motion vectors (365×2 corresponding points).

Thresholds for the motion-deformability experiments

The thresholds used for deformable and non-deformable motion discrimination must be generic. Therefore, experiments were done on several different objects (i.e., both deformable and non-deformable objects). On a large variety of videos from the corpus mentioned above, and a large number of different objects (deformable and non-deformable) were chosen for tests.

For each of those objects, all the series of its temporal motions (Motions: X_i, X_{i+1}, X_{i+2} ... X_j, ... X_m) were studied when this object appears in m frames. For each frame and its corresponding one, the optical flow was calculated using Marzat’s algorithm at first, after that the motion fields was filtered, then motion vectors were normalized, and for each normalization n (from 1 to round (ALMF) or 10), transformations (F_n / H_n) that relates normalized corresponding points ($\bar{x}_i^n \leftrightarrow \bar{x}'^n_i$ / i= 1… p (p is points number)) covering the object in the two image plans were estimated, (Fundamental matrix F by the Normalized 8-Point Algorithm (N8PA) and the Homography matrix H by the Normalized Direct Linear Transform (NDLT)).

For the Normalized 8-Point Algorithm (N8PA), among several possibilities, we tested 2 functions, the first was the Matlab library function *estimateFundamentalMatrix*, and the second was the Matlab function *fundmatrix* (Kovesi, n.d.). Results were close, but we preferred to work with *fundmatrix* (Zisserman et al., 2012).

For the Normalized Direct Linear Transform (NDLT)), 2 functions were tested, the first was the Matlab function *vgg_H_from_x_lin* (Zisserman et al., 2012), the second was the function *homest2d* of Sasikanth (Sasikanth & Kroon, 2010). Results were close, but we

preferred to work with *vgg_H_from_x_lin*.

After obtaining the normalized corresponding points and the corresponding transformations (F_n / H_n) for each normalization level, two thresholds are needed:

- ***The mapping error thresholds*** corresponding to each normalization level n (γ_F^n, γ'^n_F for F_n and γ_H^n, γ'^n_H for H_n) for both distance types. The thresholds must be general for all videos, scenes, objects, and motions. These same thresholds are then used to calculate the percentage of correctly mapped points (p_F^n for F_n, and p_H^n for H_n).
- ***The motion non-deformability thresholds*** for each normalization level n (δ_F^n, δ'^n_F for F_n, and δ_H^n , δ'^n_H for H_n) for both distance types. Again, the thresholds must be general. These thresholds are then used to classify any object motion as deformable or not.

We then calculate the percentage of success for any type of transformation (*F* or *H*), with any type of distance, for each *mapping error threshold,* and for each *motion non-deformability threshold.*

In the following subsection, for better understanding, we explain how to derive the motion non-deformability thresholds when the **mapping error thresholds are fixed** (i.e., when they are equal to one, for example). Then thresholds are improved when we apply the **ultimate thresholds** with variable **mapping error thresholds.**

A- Mapping Error Thresholds Fixed to 1:

Let the *mapping error thresholds* $\gamma_F^n = \gamma'^n_F = \gamma_H^n = \gamma'^n_H = 1$, where n is the normalization level (n =1…10). For each object motion in the scene, the percentage of correctly mapped points (p_F^n, p'^n_F for *F,* and p_H^n, p'^n_H for *H*) is calculated. Let m be the number of motions tested for any given object appearance in the scene, and let M_m denote the set of all these motions: $M_m = \{X_1, \dots X_i \dots X_m\}$.

For each X_i we have p_F^n, p'^n_F , p_H^n, and p'^n_H for each normalization level n, and these are presented as $p_F^n i$, $p'^n_F i$, $p_H^n i$, and $\mathrm{p'^n_H i}$. The set M_m contains deformable and non-deformable motion. Each motion is classified manually, as deformable or not, by reference to its movement between the two corresponding frames, and in a critical and rigorous way. For example, if only a small part of a human body (e.g., a part of a hand) is moving in a manner different from the body, regardless of whether this motion was correctly estimated with the Marzat optical flow, the object is considered deformable.

Let D_r denote the sub-set of the set M_m, with all the manually classified deformable motions. Similarly, sub-set ND_s contains all the manually classified non-deformable motions, where r and s are the cardinalities for D_r and ND_s, respectively, such that $r + s = m$.

The *motion non-deformability thresholds* (δ_F^n, δ'^n_F, δ_H^n , δ'^n_H) differentiate between these two sub-sets. Ideally, all motions in ND_s must have $p_F^n i \geq \delta_F^n$ (respectively, $p'^n_F i \geq \delta'^n_F$) or $p_H^n i \geq \delta_H^n$ (respectively, $p'^n_H i \geq \delta'^n_H$). Similarly, all motions in D_r must have $p_F^n i < \delta_F^n$ (respectively, $p'^n_F i < \delta'^n_F$) or $p_H^n i < \delta_H^n$ (respectively, $p'^n_H i < \delta'^n_H$). When the *motion non-deformability thresholds* (δ_F^n, δ'^n_F, δ_H^n , δ'^n_H) are found, they can be used for any kind of video, object, or motion.

However, when working with real images, we cannot find the *motion non-deformability thresholds* that completely split the two sub-sets D_r and ND_s. Therefore, we settle for thresholds that conform to the following two conditions concurrently (see Figure III-17):

- The biggest number (or percentage) of motions in ND_s have their own percentages of correctly mapped points ($p_F^n i$, $p'^n_F i$, $p_H^n i$ or $p'^n_H i$), above the corresponding threshold.

Likewise, the minimum number of motions in ND_s have their own percentages of correctly mapped points, below this same corresponding threshold. In other words, we retain the "best maximum" of non-deformable motions above the threshold, and an "acceptable minimum" of non-deformable motions below the threshold.

- The biggest number (or percentage) of motions in D_r have their own percentages of correctly mapped points ($p_F^n i$, $p'^n_F i$, $p_H^n i$ or $p'^n_H i$), below the corresponding threshold. Likewise, the minimum number of motions in D_r have their own percentages of correctly mapped points, above the corresponding threshold. In other words, we keep the "best maximum" of deformable motions below the threshold, and an "acceptable minimum" of deformable motions above the threshold.

Figure III-17 shows the ideal threshold (the yellow vertical line), where all non-deformable motions (in ND_s) have their respective percentages of correctly mapped points above the threshold, and all deformable motions (in D_r) have their respective percentages of correctly mapped points below this threshold. While for the discovered threshold (the violet oblique line), the maximum non-deformable motions (in ND_s) have their percentages of correctly mapped points above the threshold, and the maximum deformable motions (in D_r) have their percentages of correctly mapped points below this threshold.

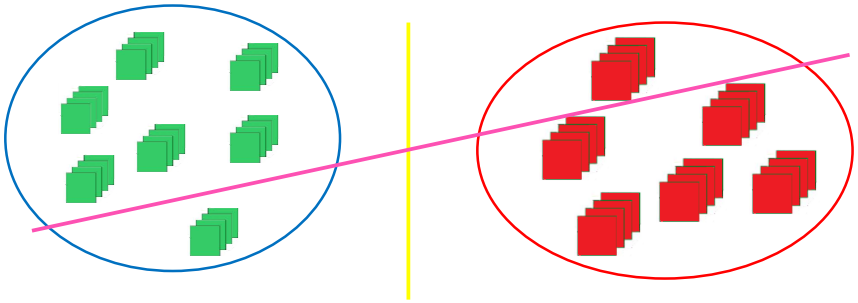


Figure III-17: Ideal threshold (the yellow vertical line) and the discovered one (the violet oblique line).

For this purpose, a new kind of graph is used. We call it the "Best Maximum – Acceptable Minimum Graph". For any given normalization level *n*, transformation, and distance type, a graph for the motion non-deformability threshold (δ_F^n, δ'^n_F, δ_H^n and δ'^n_H) is obtained as follows:

- The sub-set ND_s of the non-deformable motions is sorted in descending order, according to the percentage of correctly mapped points for each frame in the sub-set.
- On the other hand, the sub-set D_r of the deformable motions is sorted in ascending order, according to the percentage of correctly mapped points for each frame in the sub-set.
- A percentage is given for each element in the two sub-sets, representing its placement within the sub-set. This value is called the "***Placement Percentage.***" For example, the 5^{th} element in ND_s will be given the percentage $(5 \times 100)/s$, and the 5^{th} element in D_r will have the percentage $(5 \times 100)/r$.
- A graph is constructed such that:
  - The *X* axis represents the Placement Percentage.
  - The *Y* axis represents percentage of correctly mapped points.
  - Series *1* represents the reverse sorted ND_s elements (i.e., frames) in blue.
  - Series *0* represents the sorted D_r elements (i.e., frames) in red.

Figure III-18 shows the graph for a normalization level of *2,* with the Fundamental matrix F_2, using the mean distance, and with a *mapping error threshold* $\gamma'^2_F = 1$.

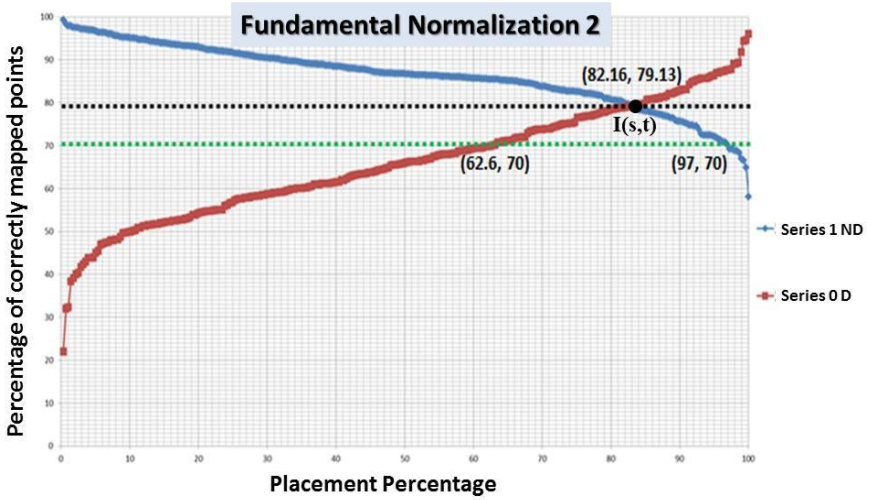


Figure III-18: Graph for F_2 with a mapping error threshold $\gamma'^2_F = 1$, y=70, intersecting with Series 0 at 62.6, and Series 1 at 97.

Note that for any point *(x, y)* on the curve:

- The *abscissa (x)* represents the Placement Percentage of this point according to its own series. For example, if *x = 60* on Series 1, 60% of the non-deformable motions are above this point. If *x = 60* on Series 0, however, *60%* of the deformable motions are below it.
- The *ordinate (y)* represents the percentage of correctly mapped points for this motion. Thus, if *y = 60*, *60%* of the point correspondences in the motion field are correctly mapped, given that the Fundamental F_2 is found.

Here the requested threshold *t* is a value of *y=t*, where:

((maximum of points in Series 1 are above line y=t) ⋂ (maximum of points in Series 0 are below line y=t)).

With this type of graph, the intersection of the two curves represents the best existing solution, where the maximum number of non-deformable motions (in ND_s) have their percentages of correctly mapped points above this coincidence point, and the maximum number of deformable motions (in D_r) have their percentages of correctly mapped points below this coincidence point. Let I(s,t) be the intersection point, with t denoting the requested threshold.

For example, in Figure III-18, if we settle for *y= 70*, rather than the intersection point, we know that *97%* of non-deformable motions are above this threshold, and consequently well classified. However, only *62.6%* of deformable motions are below this threshold, meaning that only 62.6% are well classified. Alternatively, if we take *y=t=79.13* (the ordinate of the intersection point), then *82.16%* of deformable and *83%* of non-deformable motions are well classified, and this is the optimal percentage.

Let I(s,t) be the intersection point, with t denoting the requested threshold. Notice that the abscissa, *s*, for the point of intersection *I(s,t)* represents, in this case, the ***percentage of success*** for the entire algorithm, insofar as the number of deformable motions and the number of non-deformable motions that are tested are approximately the same. Moreover, the Placement Percentage is the same for both series *ND* and *D*.

In summary, the intersection points in the graph indicate the best threshold for their normalization level, according to *F* or *H* with the usable distance. Accordingly, we calculate the *motion non-deformability thresholds* δ'^{n}_{F} and δ'^{n}_{H} for each normalization level (see following graphs for n=1..3).

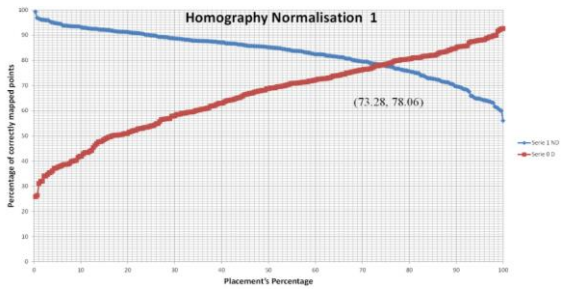


Figure III-19: Graph of the H_1 with $\boldsymbol{\gamma'^{1}_{H} = 1}$

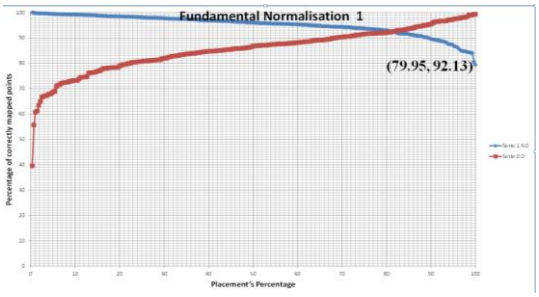


Figure III-20: Graph of the F_1 with $\boldsymbol{\gamma'^{1}_{F} = 1}$

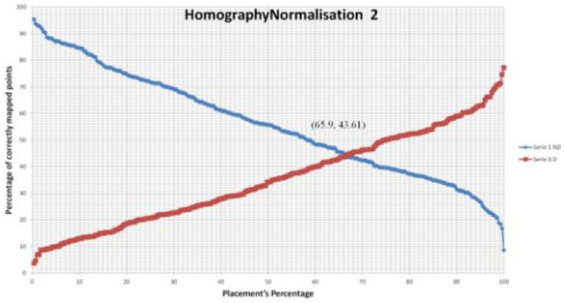


Figure III-21: Graph of the H_2 with $\boldsymbol{\gamma'^{2}_{H} = 1}$

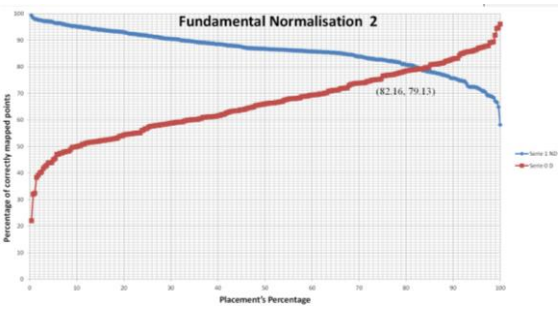


Figure III-22: Graph of the F_2 with $\boldsymbol{\gamma'^{2}_{F} = 1}$

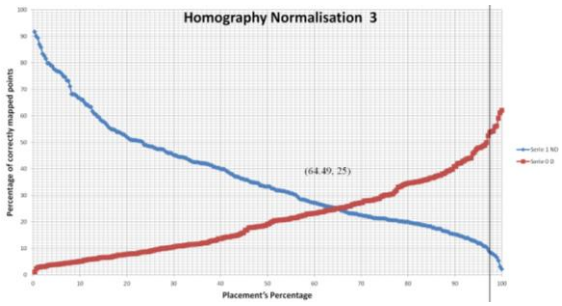


Figure III-23: Graph of the H_3 with $\boldsymbol{\gamma'^{3}_{H} = 1}$

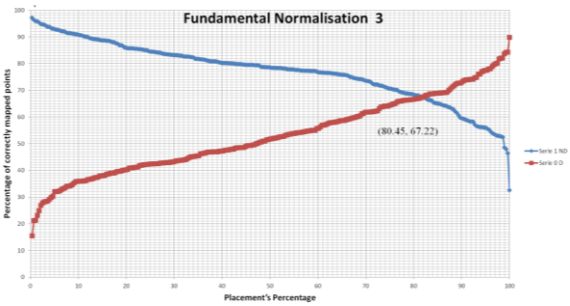


Figure III-24: Graph of the F_3 with $\boldsymbol{\gamma'^{3}_{F} = 1}$

The temporal motion non-deformability thresholds for each normalization when the mapping error threshold is fixed to 1 ($\gamma'^{n}_{F} = 1$ and $\gamma'^{n}_{H} = 1$) are summarized in the Table III-3.

Table III-3: Table of temporal motion non-deformability thresholds for each normalization when the mapping error threshold is fixed to 1 ($\boldsymbol{\gamma'^{n}_{F} = 1}$ and $\boldsymbol{\gamma'^{n}_{H} = 1}$); (a, b): a is the percentage of success (%S), and b is the motion non-deformability threshold.

Transformation \ Normalization			1	2	3	4	5	6	7	8	9	10
H	mean distance	$(\%\boldsymbol{S}, \boldsymbol{\delta'^{n}_{H}})$:	**(73.28, 78.06)**	**(65.9, 43.61)**	**(64.49, 25)**	**(62.15, 15)**	**(61.46, 9.88)**	**(62.2, 6.74)**	**(61.36, 4.8)**	**(58.13, 3.4)**	**(58.63, 2.65)**	**(59.34, 1.94)**
F	mean distance	$(\%\boldsymbol{S}, \boldsymbol{\delta'^{n}_{F}})$:	**(79.95, 92.49)**	**(82.16, 79.13)**	**(80.45, 67.22)**	**(79.23, 57.87)**	**(77.57, 49.82)**	**(74.86, 42.16)**	**(72.95, 36.62)**	**(72.2, 31.47)**	**(71.08, 27.3)**	**(73.52, 23.21)**

B- Ultimate Thresholds

As explained in the previous subsection, for each normalization level n, the intersection points along the curve represent the best *motion non-deformability thresholds* when the *mapping error thresholds* are fixed to *1*. However, when the *mapping error thresholds* are variable, the percentage of correctly mapped points changes for each motion in $M_m\ (ND_s\ , D_r)$. Moreover, for all motion, the curves of variations change along with the intersection points.

For a given normalization level n, a transformation type, and a distance type, each *mapping error threshold* value leads to a different graph, and thus to a new intersection point. The best intersection point is the one with the highest abscissa (i.e., the highest percentage of success).

The aim is to find the *mapping error threshold* γ that can give the ultimate *motion non-deformability threshold* that maximizes the *percentage of success*. When the *mapping error thresholds* are variable, the search for the ultimate *motion non-deformability thresholds* can be done by finding the best intersection points by reference to the "Best Maximum – Acceptable Minimum" graph. To do this, we studied the variation of the curves' intersection points, according to the variation in the *mapping error thresholds* (γ_F^n, γ'^n_F for F_n and γ_H^n, γ'^n_H for H_n).

For each normalization level n, each type of distance measure, and for *F* and *H*, we generated a graph of the variation in the curves with intersection points according to the variations in the *mapping error thresholds*. For example, for the transformation *F*, normalization *3*, the mean distance, and a *mapping error threshold* of γ'^3_F, varying between *0.2* and *10* with intervals of 0.2 ($\gamma'^3_F = 0.2{:}\ 0.2{:}\ 10$), the graph will take the following form:

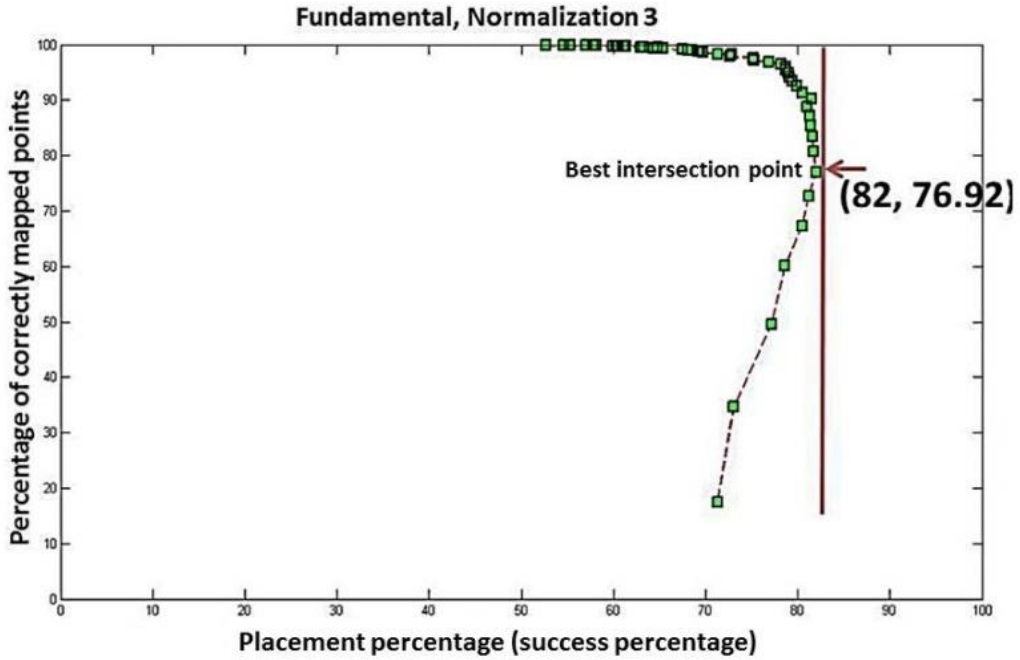


Figure III-25: Graph of the variation of curves with intersection points according to variable mapping error thresholds, for F, normalization 3, mean distance, and a mapping error threshold of $\gamma'^3_F = 0.2{:}\ 0.2 : 10$.

In Figure III-25, it is clear that the *mapping error threshold* $\gamma'^3_F = 1.4$ is the threshold that maximizes the *percentage of success* to *82%,* which corresponds to the ultimate *motion non-deformability threshold* of *76.92 %.* Furthermore, for each normalization level n, we

calculate the ultimate corresponding couple (*mapping error threshold* and *motion non-deformability threshold*) that maximizes the *percentage of success* for F_n and H_n, using different distance measurements. The values from this calculation are found in Table III-1.

The thresholds in Table 3, and the ultimate thresholds in Table III-1, confirm that when the *mapping error threshold* is fixed, the best *motion non-deformability thresholds* will have decreasing values proportional to the normalization level (see Table III-3). However, if the *mapping error threshold* is variable in the appropriate way, the ultimate *motion non-deformability thresholds* will have approximately the same value, regardless of the normalization level or the distance type used (see Table III-1). This ensures high stability and reliability with regard to our algorithm.

On the basis of our experiments we recommend using the Fundamental matrix as the transformation, the Symmetric Epipolar Distance, normalization level *2*, the *mapping error threshold* γ_F^2 *=2.2*, and the *motion non-deformability threshold* δ_F^2 = ***80.16*** (note: this is not to suggest that other thresholds are undesirable**)**. As an example, when we compared our results for the two scenes above (viz., "Highway 2" where p_F^2 *= 94.1722*, "Walking" where p_F^2 *= 68.7671,* "Bomb2" where p_F^2 *= 95.2681, and* "Big Map 3" where p_F^2 *= 98.4701*) with the corresponding threshold (δ_F^2 = ***80.16***), we can easily infer the deformability of each corresponding motion. The scene from "Highway 2" was classified as non-deformable motion, the scene from "Walking" was classified as deformable motion, the scene from "Bomb2" was classified as non-deformable motion, and the scene from "Big Map 3" was classified as non-deformable motion —and all classifications were correct.

Several different examples are described in Appendix VIII.3.

III.5. Conclusion

In this chapter, we proposed a threshold-based decision-making system aimed at determining whether an object or its motion corresponds to a non-deformable model using geometric projection modelling. The method relies on estimating parameters of a standard geometric transformation, which could be considered as a model of non-deformable object motion. The accuracy of this transformation in representing the object motion is then analysed to infer the actual deformability (or non-deformability) of the object.

We improved the results using temporal consistency, reaching a relatively high rate of precision (approximately *92%*). Such a precision rate is largely sufficient to address new topics where knowledge about object deformability is an input.

This study provides the video surveillance research a rigorous and precise algorithm, which can be major feature when classifying the interaction between scenes objects. Also it is an important characteristic for the objects to be described at the final textual output.

IV. Description of video surveillance scenes

IV.1. Introduction

Most existing video surveillance systems provide the infrastructure to mainly capture video images, transmit, store, and distribute them, while the task of threat detection and analysis is left to human operators. Detecting an incident in a live source or searching the video archives for a specific one almost completely relies on scarce and costly human resources.

Every second counts. The fastest the operator detects the incident; the best the damage is minimized. Incident detection counts on the capabilities of a human operator, to observe, to analyse (detect and identify) **moving objects**, and to understand their **actions and interactions** within the **field-of-view** (FOV) of the cameras.

In the management of surveillance control rooms, the cameras per monitor ratio or the number of CCTV screens per operator is an important factor. Most of the police forces, in surveillance field, suffer from human resources deficiency. Especially when having a big number of cameras. Consequently, a limited number of operators are responsible to constantly monitor a large area, by observing a single monitor showing multiple streams simultaneously or sequentially, and this is the case of most of CCTV control rooms. However vigilant the operators are, monitoring process suffers from the huge amount of information, which leads to inattention due to fatigue, interruptions and distractions, and physical limits. Police operators cannot keep continuous surveillance effectively. Unfortunately, in such manual system, many incidents are miss-detected. As a result, surveillance videos are often used in passive monitoring or as evidence for post-incidents investigations. These miss-detections of important events can be dangerous in critical surveillance tasks such as public places, sensitive locations, airport, and border control surveillance.

Beside this, accessing video data storage is very limited and far away from efficiency when the analysts are working on post-incident investigation. Those video analysts need specific location, specific time and specific incident type and description. Most of the time, at least one of those three is not available or, let ‘us say, not accurate. For this reason, it may take a very long time for a human to detect it. Then tracking the involved objects (persons and vehicles) and analysing them is another part of the problem. The analyst should fetch all surrounding cameras to trace each one of those objects, and hopefully uncover all the necessary information about them. The information may be object identification, person description, vehicle description and plate number, etc. In some cases, it may take months to analyse one incident.

To overcome these limitations of traditional surveillance methods, the computer vision and artificial intelligence communities are seeking to develop automated systems for the real-time monitoring and archives investigation of contents understanding like vehicles, people, other objects, actions and interactions.

In a surveillance control room, especially when observing a dynamic scene like public places, motion is the daily basis. As mentioned, motion information stands out as the most important cue to identify the dynamic content of videos. Extraction and analysis of motion

information in videos are crucial in automated surveillance video systems. And as daily motion is the regular thing, the most important part of observation or analysis is to focus on non-linear motion like the one we may observe during an interaction between objects in the scene, mainly humans, vehicles, or any kind of moving objects, either deformable or non-deformable. The scene type and environment can be very divers. Moreover, many objects can exist in the scene, and many types of interaction could occur, as mentioned in the ontology.

For most human, describing what is happening in a video is an easy task. For computers, extracting, analysing, and understanding what is happening from video pixels and generating a description is still a very complex problem. A relatively wide panel of works on many fields concerning video description in general and video surveillance in particularly has been published (see the state of the art in chapter I). To simplify the problem, some researches added more assumptions to significantly improve the results but limiting so their applicability in the real world. Most of the researches have specific limitations. They are designed for particular sets of objects, and actions in a specific context. They often lack of a generic multimodal framework to achieve system robustness in multiple contexts, object types and actions performed.

The state of the art for videos surveillance description was thoroughly discussed in chapter I, where we showed that the behaviour understanding and sentences generation approach is the most suitable for our video surveillance description system. Indeed, this approach takes into consideration the need for extracting important behaviour understanding features, and the need for generic expressive structural description. For that reason, in the next section, we share an overview of some of the researches on automated video surveillance, where the interest is to focus on the semantic content features which can be useful for video description. Those features are mainly involved in the behaviour understanding and automated visual surveillance fields.

Then, we present our proposed approach, which is a generic Video Surveillance Scene Description (VSSD), with main focus on interactions between objects, designed to meet the needs of dynamically changing conditions like objects, interaction and context.

IV.2. State of the art

Working on video content analysis and understanding, it is not a field for video surveillance application only, but it trespasses that for many other applications and domains. There is a big need in variety of applications and domains not restricted to surveillance applications, we mention: video indexing (commonly based on text or other) for content-based video annotation / retrieval, human-computer interfaces, computer games, animation and special effects, video editing, analysis of sport athletics, healthcare systems, interactive application and environment (automated houses and cars…), video segmentation, analysis of human conditions (e.g., athletic performance…), etc.

A lot of studies mainly focused on surveillance applications, like person identification, person or car tracking, crowd flux analysis and statistics or congestion analysis (Feris, Datta, Pankanti, & Sun, 2013), anomaly detection and alarming (Neves, Narducci, Barra, & Proença, 2016), access control, interactive surveillance using multiple cameras (tracking objects…), people counting (Hou & Pang, 2011), behaviour analysis (Pantic, Pentland, Nijholt, & Huang, 2007) (T. Ko, 2008) and action recognition (Neves et al., 2016).

Behaviour analysis and semantic content extraction contains analysis and recognition of motion, actions and interactions between objects. It is, for visual surveillance, one of the most advanced and complex research in image processing, computer vision, and artificial intelligence. The studies in this area focus on the advancement of visual analysis techniques in order to extract the semantic information about regularity or abnormality of the scenes objects behaviours (e.g., human & vehicle).

An automated visual surveillance system which can understand and learn behaviour from observed activities in a video sequence requires a reliable integration of image processing techniques and artificial intelligence techniques (Jan, 2004) (T. Xiang & Gong, 2008).

Our main goal is to obtain a meaningful semantic content which can be used for description of what is happening in a monitored area, either in live mode to take appropriate action based on that interpretation, or in offline mode like storing the video sequence with the textual description to provide an easy intelligent access. The description may vary according to the need, context, objects, and intended actions.

For extracting useful content features, many means were used depending on the output specific goal, we mention: object detection and segmentation, object tracking, trajectory analysis, action analysis, activity classification and recognition, and others.

Huge amount on "Behaviour analysis and understanding" studies and surveys are not restricted to only video surveillance systems; Many excellent surveys like (W. Hu, Tan, Wang, & Maybank, 2004) (Vishwakarma & Agrawal, 2013).

(Taha, H. Zayed, E. Khalifa, & M. El-Horbaty, 2014), (Teddy Ko, 2011), (T. Ko, 2008), (Liang Wang, Hu, & Tan, 2003), and (Kumar & Mittal, 2007) discuss the general framework and the general architecture of a video understanding system exploiting behaviour analysis.

More recently, since the evolution of many neural network techniques, many of these techniques and algorithms were used in many content extraction fields. Some content extractions were largely improved and achieved satisfied results, other still not mature. As can be clearly predicted, the introduction of these machine learning techniques for this field of research is very promising.

Next, we present the state of the art of some related subjects, concerning object(s) tracking, trajectory analysis, action analysis and recognition, and textual description templates. Also the reader can refer to the Appendix VIII.5 for more related works concerning object detection, object segmentation, object classification and video action analysis. Finally, before presenting our proposed approach, we highlight some of the complexities that face most works when dealing with video surveillance.

IV.2.1. Object tracking and Multi-object tracking

IV.2.1.A. Object tracking

The task is to track moving objects through frame sequences. Object tracking is the process of locating, over time, a moving object. This can be difficult in some cases; depending on the angle, distance and the object speed. Most studies use matching techniques to make sure that the same blob is being tracked in each subsequent frame. Different techniques can be mainly divided into seven main categories, according to (Morris & Trivedi, 2008), which are: region-based tracking, contour-based tracking, feature-based

tracking, model-based tracking, hybrid tracking, optical flow-based tracking, prediction-based techniques.

In another approach, (Neves et al., 2016) distinguish in general, tracking approaches regarding to the tracking technique adopted (Bayesian, Kernel Filter Model / Shape, by-Detection) and the type of information (motion, appearance, and shape) used to model target objects, usually denoted as target representation.

To track moving objects, deep neural networks, especially convolutional neural networks (CNN) were recently proposed. Some promising results are shown in (H. Li, Li, & Porikli, 2016) and (Nam & Han, 2016). Wang et al., in (Lijun Wang, Ouyang, Wang, & Lu, 2015), proposed to create an object tracker by online selecting the most significant hierarchical features from an ImageNet pre-trained CNN.

Zhai et al. (Zhai, Chen, Mori, & Roshtkhari, 2019) used a Bayesian classifier as a loss layer in CNN tracker.

As for Nam et al. (Nam & Han, 2016), they trained a multi-domain CNN, for tracking objects, using learning generic representations.

Comprehensive surveys for conventional object tracking can be found in (Morris & Trivedi, 2008), (Teddy Ko, 2011), and (W. Hu et al., 2004).

A comparison of methods based on deep learning, but mainly focused on visual tracking, has been presented in (P. Li, Wang, Wang, & Lu, 2018). According to Li and al., their comparison shows that, using deep convolutional neural network for tracking, could improve significantly the performance.

IV.2.1.B. Multi-object tracking

To solve the problem of multi-object tracking, one can plan to use the object tracking algorithms, in multiple instances; however this approach requires an additional data association module, as, for example, in the Multiple Hypothesis Tracking (Reid, 1979), or the Joint Probabilistic Data Association Filter (Fortmann, Bar-Shalom, & Scheffe, 1983), or the appearance similarity (Breitenstein, Reichlin, Leibe, Koller-Meier, & Gool, 2009), or the prediction-based tracking (Particle filter, Kalman filter). Our selected approach ("Motion-Based Multiple Object Tracking - MATLAB & Simulink," n.d.), after detecting moving objects in each frame, uses kalman filter to predict the track's location, for associating the detections corresponding to the same object over time.

Other techniques were proposed, like ObjectTracker, Deform PF-MT, PWP3D, Globally-Optimal Greedy Algorithms, Continuous Energy Minimization for Multi-Target Tracking, Two-Granularity Tracking, GMCP-Tracker, Urban Tracker, BPF, Tracking Interacting Objects, Learning to Track, and many others.

A comparative work of many of the above algorithms is presented in the experimental section IV.4. However, it is important to mention that this comparison was made at an early stage of this thesis. Therefore, some recent studies, based on deep learning, were not included in this comparison, but will be considered in our future work.

Recently in (Ankush Agarwal & Suryavanshi, 2017), the authors propose a multiple object tracking by using a region based convolutional neural network (RCNN) for object detection and by creating a regression network for generic object tracking.

In (Milan, Rezatofighi, Dick, Reid, & Schindler, 2017) the authors used a recurrent neural network and LSTM, to perform target state prediction, state updates, and data association.

Another work using deep neural network for Multi-object tracking can be found in (Gaidon, Wang, Cabon, & Vig, 2016).

IV.2.2. Trajectory analysis

The trajectory of motion is important for the analysis of a video, and can be widely applied to many domains, such as indexing and extracting a video (W. Hu et al., 2007), (W. Hu et al., 2004), video scene segmentation, video semantic analysis. The analysis of the trajectories can help the recognition of the events, actions or interactions between objects. It is an intermediate level between the low and the high level of analysis. However, the direct modelling of spatiotemporal variations of trajectories is complex because of their non-linearity.

Movement trajectories provide rich information about the spatio-temporal activity of an object. Each trajectory records not only the coordinates of points (the position sequence of the tracked target) and the local directions (direction of the object at each position) on the image trajectories (Bashir, Khokhar, & Schonfeld, 2007), (Buzan, Sclaroff, & Kollios, 2004), (Chan, Hoogs, Schmiederer, & Petersen, 2004), but also speed and acceleration (Hongeng, Nevatia, & Bremond, 2004), (Xiaogang Wang, Tieu, & Grimson, 2006).

An enormous work on the understanding of behaviours, events and actions has been conducted on the basis of trajectory analysis. The majority of these efforts in the field of visual surveillance are focused on similarity and clustering of trajectories (Anjum & Cavallaro, 2008), (Kataoka et al., 2013), (Xiaogang Wang et al., 2006); detection of abnormal trajectories (Kataoka et al., 2013), (Dimitrios Makris & Ellis, 2002), (D. Makris & Ellis, 2005) ; detection and classification of events (Z. Zhang, Huang, Tan, & Wang, 2007), (Piciarelli, Micheloni, & Foresti, 2008), (Hervieu, Bouthemy, & Cadre, 2008); and scene modelling (Points of Interest (POI) where interesting events happen (entry / exit, stop), activity paths (PA), junctions, roads) (Xiaogang Wang et al., 2006), (D. Makris & Ellis, 2005), (Black, Ellis, & Makris, 2004), and (Sangho Park & Trivedi, 2007), where, later discussed in IV.3.3, a similar outputs were presented with our approach.

A recent work (Dogra, Ahmed, & Bhaskar, 2016) proposed a method using a finite state machine to analyse the trajectory and the instantaneous velocity to detect what they named it “event(s)-of-interest”, means when an interesting variations occur. These events of interest used to help in summarizing the scenes.

In our approach we used a similar concept to detect important variation, but not only in velocity and trajectory, but also directions, surface, Hu moments, and deformability, to trigger the description at such moments.

IV.2.3. Action and Activity classification and recognition

Another important area of research today is Action and activity classification and recognition. Its goal is to automatically analyse ongoing activities from an unknown video. It includes the analysis and the recognition of patterns to infer higher level description of objects actions and interactions. Also, it is the process of recognizing the actions to know and understand what is happening in a given context (Loy, 2010).

In video various types of activities, it differs according to many varieties and complexity, which make difficult to analyse, classify and recognize these activities. To overcome these difficulties and improve their system, most of the researches in this field apply some restrictions, concerning scene type (Oliver, Rosario, & Pentland, 2000), (Nevatia, Zhao, & Hongeng, 2003), on object type (Liang Wang et al., 2003), (Moeslund, Hilton, & Krüger, 2006), (Ivanov, Stauffer, Bobick, & Grimson, 1999), action type (A. Kojima, Tamura, & Fukunaga, 2002), (Aggarwal, 2004), scenario (H. Li, Tang, Wu, Zhang, & Lin, 2010), (J. Wu, Osuntogun, Choudhury, Philipose, & Rehg, 2007), (A. Gupta & Davis, 2007), (Ryoo & Aggarwal, 2007a).

Video monitoring or analysing suspicious activities can be conceptually divided in to four categories (Vishwakarma & Agrawal, 2013), (Aggarwal & Ryoo, 2011):

1- Gestures: they are the elementary movements of peoples' articulations; also, they are the atomic components describing the overall motion. This action is simple and is performed in a short time, such as: moving a leg, turning a head, etc.
2- Actions: they are single person activities where multiple gestures (atomic actions) compose it in a temporal sequence, such as: walking, and jumping, etc.
3- Interactions: they are inter-object activities that involve two or more objects (human, animal, object, etc.). For example, One to one interaction like human running together, animal chasing a human, two human are fighting (Taj & Cavallaro, 2010), (Zen, Lepri, Ricci, & Lanz, 2010); interaction between many objects (Coppola, Cosar, Faria, & Bellotto, 2017), (Candamo, Shreve, Goldgof, Sapper, & Kasturi, 2010), (Sangho Park & Aggarwal, 2006), (Sangho Park & Aggarwal, 2003), person – vehicle interaction (S. Park & Trivedi, 2007), Human and inert objects like human leaving a bag (Aggarwal & Ryoo, 2011), (Ryoo & Aggarwal, 2007a), (Moore, Essa, & Hayes, 1999), (A. Gupta & Davis, 2007), (Peursum, West, & Venkatesh, 2005), (Ferrando, Gera, Massa, & Regazzoni, 2006), etc.
4- Crowd activities: they are the activities performed by groups of multiple objects (S. Pellegrini, Ess, Schindler, & Gool, 2009), (Cristani et al., 2011), (Stefano Pellegrini, Ess, & Van Gool, 2010), (Cui, Liu, Gao, & Metaxas, 2011), (Szczodrak et al., 2011), (Cho & Kang, 2012), (Ke, Sukthankar, & Hebert, 2007). For example: a protest, a group of wild animals, etc.

Taking advantage from these four categories, we present in the chapter II a deeper perspective for conceptually categorizing the type of video surveillance scenes into 15 categories.

Human activity recognition approaches is mainly divided into two categories. (1) The traditional representation-based approach based on the feature detectors and descriptors, e.g. trajectory, and Scale-Invariant Feature Transform (SIFT), then for action recognition, a generic trainable classifier is applied; (2) Learning-based representation approach, which is a recently developed approach with capability of learning features automatically and directly from the images and frames. In this approach no need for feature detectors and descriptors.

A- Traditional human activity recognition approaches: or approaches based on hand-crafted local features, was classified by (Aggarwal & Ryoo, 2011) into two main categories: Non-hierarchical and hierarchical approach.
   1- The non-hierarchical approach or single layer approach recognizes the human activity, based directly on image sequences, by matching the activity with already known ones. This approach mainly used for simple and short activities such as

periodic activities and primitive action (jumping, running, waving, etc.). It is divided into two sub-classes: sequential approach and space-time approach.

2- The hierarchical approach is usually used for complex human activities such as multi-object activities, human-object interactions and group activities. It represents these by describing them in terms of simpler activities. It can be classified into three categories: syntactic approaches, statistical approaches, and description-based approach.

B- Learning-Based Action Representation Approach

On other hand, this recent approach is based on the last progress on learning field. It has capability to learn the feature automatically from the raw data (frames). A new concept end-to-end learning is introduced, means the transformation from pixel level to action classification.

(Sargano, Angelov, & Habib, 2017) divided these approaches into two categories: non-deep learning-based approaches and deep learning-based approaches:

1- Non-Deep Learning-Based Approaches:

These approaches are based on two main approaches: dictionary learning where the representative vectors, called code words (codebook), learned from the large number of samples, and genetic programming where features are automatically learned the spatiotemporal motion features for action recognition.

2- Deep Learning-Based Approaches:

(Deng & Yu, 2014) have classified the deep learning models into three categories: a- generative/unsupervised models (like Deep Belief Networks (DBNs), Restricted Boltzmann Machines (RBMs), Deep Boltzmann machines (DBMs), and regularized auto-encoders), b- discriminative/Supervised models (like Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs), and Recurrent Neural Networks (RNNs)), and c- Hybrid models.

Interesting works is being handled newly in this field (Pham, Khoudour, Crouzil, Zegers, & Velastin, 2019). Also, important survey can be seen in (Asadi-Aghbolaghi et al., 2017).

But so far, as mentioned before, a lot of work with the learning models has been done on images and classification in images, where those algorithms have achieved very good results. In videos, some work dealt with gestures, actions and group activities, and promising results were found. However, fully data-driven deep models, referred to as “black-box”, have also some limitations in videos (Sargano et al., 2017), where the performance of the learning-based methods solely is still not up to the mark. This is mainly due to the unavailability of huge datasets for action recognition unlike in the object recognition where huge dataset exists.

For action and activity classification and recognition, till now, either in both approaches (traditional and learning), not very distinctive results have been done on the level of interaction.

Some studies suggest that unsupervised learning it is going to be far more important in the long run. Since the human and animal learning is mostly unsupervised.

In our work we focus mainly on the interactions. Our intention is to detect the existence of interaction between two objects, and to classify it according to its types. As using learning-based approaches for videos at the stage of features extraction is still not satisfying and does not take into consideration the temporal nature of interactions, we choose to extract meaningful feature appropriate for our classification. Nerveless, we used for our

classifications the learning-based approaches in both faces, Non-Deep and Deep learning, see section experiments IV.4.5.

IV.2.4. Textual description templates

In the state of the art of the behaviour understanding and sentences generation approach, used for our video surveillance description system, shows that many sentence generations are based on handcrafted structured templates. These templates differ from work to another according to needs. Some of these templates are presented in Table IV-1:

Table IV-1: Templates used for video surveillance textual description

Reference	Template
(Nishida & Takamatsu, 1982), (Nishida, Takamatsu, Tani, & Doi, 1988)	A/the (**agent**) (verb) (**object**) (**location**)
(Ivanov et al., 1999)	(**Agent**) (verb) (frame range)
(Gerber et al., 2002)	(Interval of validity)! (verb) phrase(subject, object)
(Jiangung Lou et al., 2002)	(**The Obj**) (Action) in (The **place name) [at (high/low/middle) speed**]
(Carles Fernández et al., 2008), (C. Fernández et al., 2011)	Verb (**Agent**, **direction, location**)
(Fernández Tena, 2010)	(**Frame number**)! (**Agent**), **(location in frame (x,y)), (direction), (velocity), (action)**
(Khan, Lei Zhang, & Gotoh, 2011)	(subject (S)) performs (action (A)) on (**object (O)**)
(Barbu et al., 2012)	(subject)(verb) // (subject)(verb) (**object**) // (subject)(verb) (Complement) // (subject) [adverb](verb)(**object**) [Person pose]
(Ahmed et al., 2019)	A (Color) (heavy, medium, light) (Car, Bus, Cycle, Bike, vehicle) (**moving, moving high-speed, moving low-speed**) from (**Region Name) toward (Region Name**) // A Person (walking, running, loitering) from **(Region Name) toward (Region Name)**

Many concepts and terms (highlighted in bold) in these templates are found interesting for our system, we mention the location, frame range, variety of speed, and the direction. Nerveless, our structured templates is more developed, see sub-section IV.3.8.

IV.2.5. Complexity of the video

A lot of success has driven the image processing field to converge toward the learning models. While in videos field in general, and specifically in video surveillance field, still did not achieved what is needed and expected. As working on videos is a very complex problem due to the diversity of scenes, we mention:

- **The context place type and scene location:** the place where the surveillance can occur. This place can handle two main types: Indoor and outdoor. Indoor places are like Airports, Banks, schools, governmental buildings, metro stations etc. Outdoor places are like City streets, Circulation, Gardens, Parking, Squares, etc.).
- **The Environment**: as it is subject to weather conditions, reflections, and irregular lighting changes according to daylight.
- **The number of objects**: as the scene can handle many objects (crowd or groups), two objects, one object with partial or complete occlusion or deformation.
- **The object types:** as the object can be a human, animal, plant, machine, inert object, and in extension any element composing an urban or a rural landscape.
- **The variety of actions and interactions:** it is an open field of gestures, actions and interactions, in each activity, viewed from different view angles, at different scales.

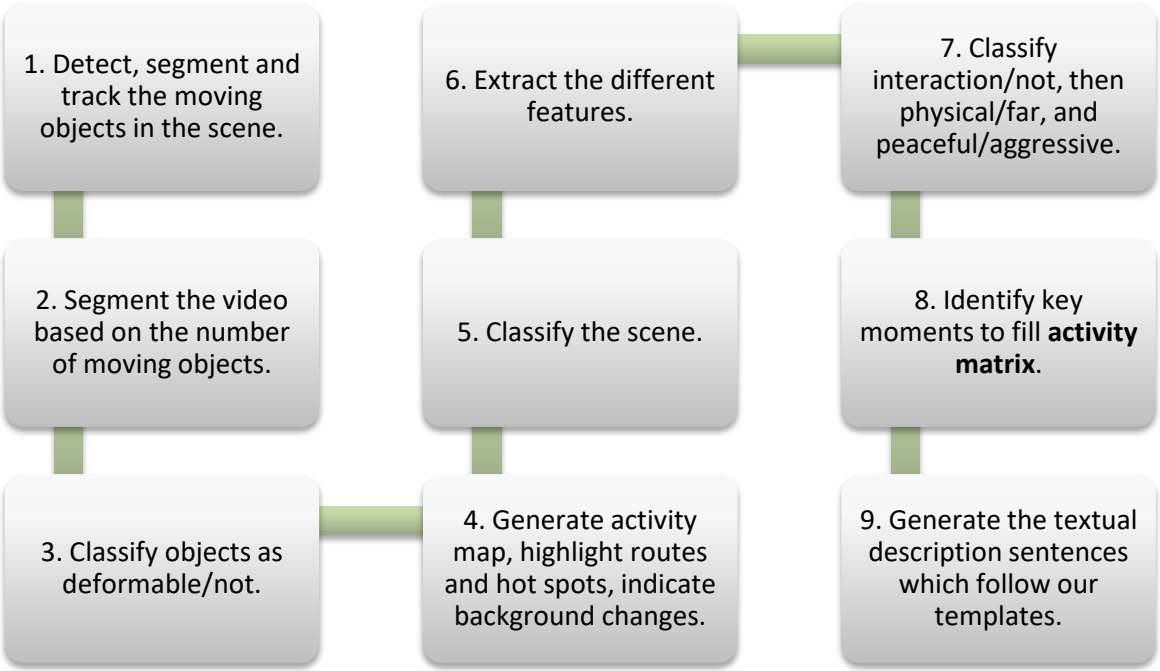
Significant research and advancement in solving these difficulties have been achieved. As seen in the literature, some research, to simplify the problems, added more assumptions. This may have improved the results significantly but it limited and restricted its applicability in real world. The algorithms developed therefore were designed for a particular type of objects, a specific context, and some particular actions.

IV.3. Proposed Approach

An efficient way to encounter the difficulties of varieties of scenes, object types and actions performed, may be by proposing a generic non-contextual approach. This chapter proposes a new approach for a generic context-independent textual description of the scenes in real-world surveillance video. We propose new representations for the sentences based on well-structured templates, which can be applied to generate incidents scenes description similar to ones used in police investigators reports. Our approach is based on the ontology described in chapter II, which combines low-level primitives of objects basic features, to allow deriving more meaningful high-level information; and this made it more methodologic, and easily expendable. This general approach is not restricted to a context, object or interaction type; in the contrary, it can be applied to any of the scene types at the level of application. In plus, we introduce additional new concepts, concerning mediation and action at a distance, and a new manner of segmenting video and categorizing the scenes, in sub-sections IV.3.4 and IV.3.6. This approach is named Video Surveillance Scene Description approach or VSSD.

The key idea behind our approach is to leverage meaningful content features from the scene for better understanding and an appropriate scene description. These features are well selected, for the real cases and real need of police operators and investigators. They are considered useful in generating alerts, enquiring the video surveillance footage, and inferring textual sentences.

Our main focus was on the characteristics and behaviours of objects, and the interactions between them, mainly when having two objects in the scene. To accomplish that, many phases, listed below, were made:



- First, we detect, segment and track the moving objects in the scene.
- Second, we segment the video based on the number of objects moving in the scenes.
- Third, objects are classified as deformable and non-deformable.

- Fourth, we generate an activity map, highlighting the routes and hot spots in the FOV, and indicate background changes.
- Fifth, we classify the scene.
- Sixth, we extract the different features.
- Seventh, we detect the existence of interaction or not by classification, then we classify whether the interaction is physical or far, and finally whether the interaction is peaceful or aggressive.
- Eighth, we identify the key relevant moments, to fill an **activity matrix**.
- Finally, we generate the textual description sentences which follow our templates.

However, each of the phases listed above, can have as input one or many phases' results; also, its output can feed one or many phases. The proposed approach diagram that can show these relations is shown in Figure IV-1.

The presented description of the video surveillance scenes VSSD, take into consideration the main needed information when an incident occurs. In real cases, when an incident occurs, five main points are mainly needed for a scene description, also known as the five Ws (Who, what, where, when and why (is replaced by how)):

1-	**WHO**	Who is the concerning object, which the description should focus on (especially, when there are many objects in the scene)
2-	**WHAT**	What the objects are doing in the scene: movements, actions, activities, etc.
3-	**WHERE**	In which location, in the case of video, in which position in the frame.
4-	**WHEN**	At what trench of time the incident occurred
5-	**HOW**	How did the objects (the who) perform their doings (the what). The circumstances and the way the person perform the crime is one of the most important clues for the decree (judgment), for example: "A car speeds up toward a person and hits the person", is very different than: "A car slow down, change direction away from the person and hit the person", because the second may indicate the intention of the car driver to avoid the person.

For adaptable description output, we left the verbosity frequency at many levels of scene description to be controlled, according to the preference, by the user.
The presented approach kept its modularity of analysis tools, allowing improving the produced description at any time. The semantic content can be easily increased by adding information about the context. Next, we present a detailed explanation of each of the stages.

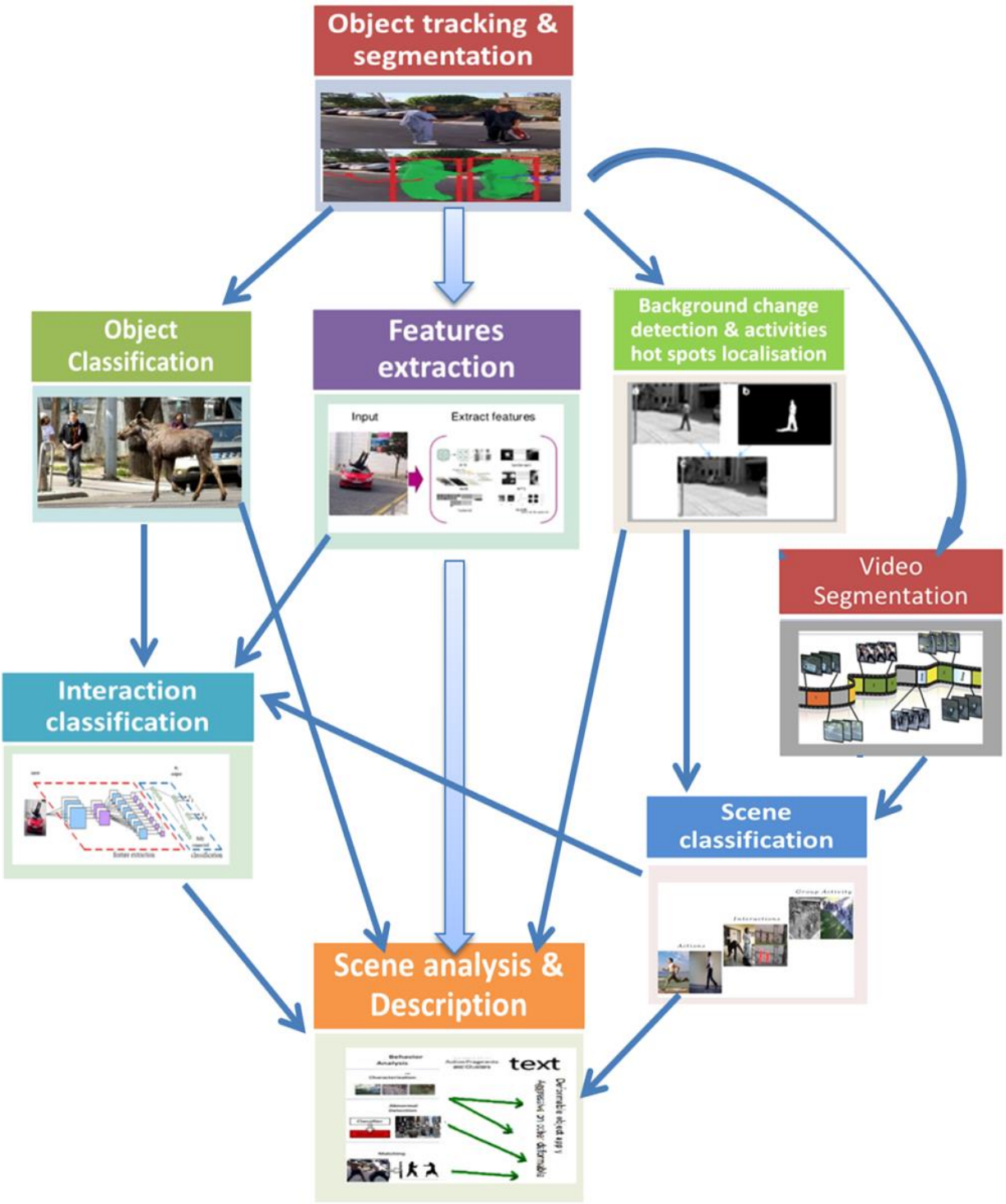


Figure IV-1: Proposed approach diagram, where each phase connected to an arrow starting point feeds the phase connected to the terminal point of the corresponding arrow.

IV.3.1. Segmentation and tracking algorithm

The objective of video object tracking is to follow a targeted object through the sequence of video frames. The segmentation, in turn, classifies the pixels of a video image, separating the foreground from the background. Tracking and segmentation are very important operations, since all the following operations (features extraction and interaction classification) rely on them. Bugs and bad decisions, in these two operations will affect the quality of the final results.

Many efforts have been done in those two fields by the research community, and a lot of problems remain, especially in the case of occlusion. Most of the segmentation and tracking algorithm will be confused between the two objects in this case.

Tracking and segmentation algorithms are not the main core of this thesis. They are considered as being only a tool. As many works have been done in this field, one can assume that we have a tracking and segmentation algorithm which provides a good measurement for further processing in the approach. But in our approach, we wished to deal with realistic results in order to measure their impact on video surveillance automation systems. For that purpose, we searched available tracking and segmentation algorithms, seeking for a one that can provide us with acceptable results for further interpretation. We mainly focused on the available algorithms, able to perform multi-target/object tracking and segmentation of any object types, as we need to track and segment two objects in the scene at the same time. After a comparison of more than 20 algorithms, we found few algorithms able to deliver all of the mentioned conditions. For more detail, the reader is referred to the experimental section IV.4. We chose the algorithm for segmentation and multi-object tracking provided by Matlab, called “Motion-Based Multiple Object Tracking”. This algorithm is based on two main steps, according to (“Multiple Object Tracking Matlab,” n.d.):

1- **Detecting moving objects in each frame:** by using a background subtraction algorithm based on Gaussian Mixture Models. For noise filtering, morphological operations are applied to the resulting foreground mask. Then, using blob analysis the algorithm detects groups of connected pixels, which are probably corresponding to moving objects.
2- **Tracking the moving objects from frame to frame:** by, first, assigning detections to tracks using “**Kalman filter”** for motion estimation and prediction. Then, initializing new tracks based on unassigned detections, confirming and updating existing assigned tracks, coasting existing unassigned tracks, and finally deleting unassigned tracks, for a too long time.

We add to the algorithm the ability to draw the trajectories of the objects centroids. Samples of results of applying the algorithm on the Fight_RunAway2 scene from CAVIAR Video Sequence (“CAVIAR,” 2004) are shown in Figure IV-2.

It is important to mention that the selected “Motion-Based Multiple Object Tracking” algorithm may not be the best one, but it is the simplest, most available, suitable and easy to use one. We can easily replace it whenever we find a better one. We chose this "plug and play" principle for the sake of flexibility.

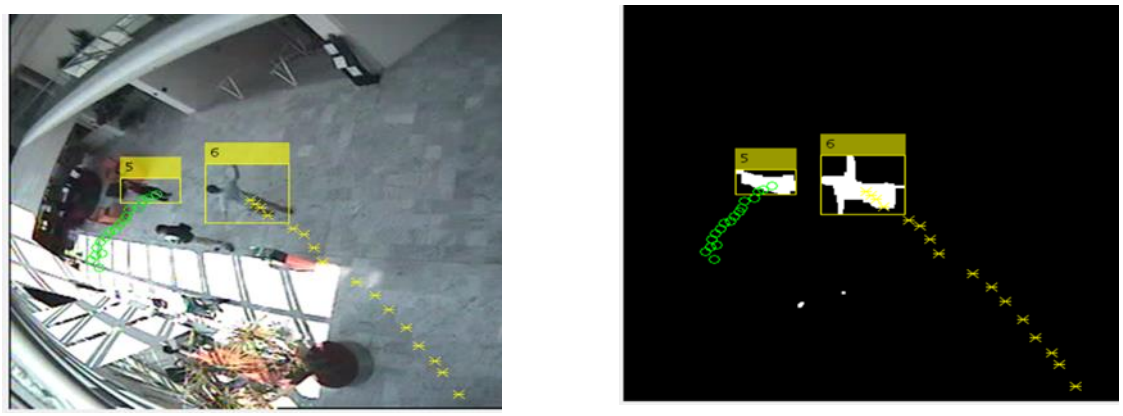


Figure IV-2: Objects movements and trajectories, Fight_RunAway2 scene from CAVIAR Video Sequence ("CAVIAR," 2004).

IV.3.2. Object classification

In our ontology, we differentiated between two general sub-classes of an object: deformable and non-deformable objects. When an observer in a surveillance control room is focusing on an area of interest, the first thing to distinguish, if an object appears in the frame, is its deformability. As mentioned, from surveillance point of view, non-deformable objects actions during an interaction are easy to detect, analyse, understand and maybe predict. When deformable parts of an object move freely in unpredicted way, the analysis becomes more difficult, even for a human brain.

Our main goal is to have an abstract description, which can be applied in any scene type such as circulation, city surveillance, elder house, hospital, borders, zoo, etc. The object can be a human, an animal, a machine or even a plant. We could not find, for our knowledge, a generic algorithm that can segment any object type in any scene, under any circumstances (occlusion, lighting, perspective…), in its semantic sub-components with a very high accuracy. Therefore, we did not try to go deeper into analysis of sub-object segments in the presence of interacting deformable objects.

Recently, for classifying the objects extracted from the scenes after segmentation, many existing algorithms with deep neural network (YOLO v3, Mask R-CNN) can perform good results, especially when it is restricted to a particular scene type having a known number of objects types. As our description is meant to stay on abstract general level, we did not consider applying any of those algorithms. But, in a later stage at the level of application, adding a classification tool to our algorithm it would be very favourable, especially when it is contextual-based where the scene and objects types are well known, which make it simple to learn such algorithms and to have very accurate results. A worthy push for such algorithm, if it has as an input for classification the object is deformable or not.

Consequently, as a first step of analysis and classification of object interactions, we can easily check whether the blobs encasing each one of the objects, resulting from the tracking algorithm, refer to deformable or non-deformable object, following the method mentioned in chapter III. Nerveless, for the reason of having good assessment for the interaction classifications approach and not cumulating, with the mentioned approach, the errors produced by the object classification algorithm, we added the manual classification of the objects, at the level of experimentation, as an input for the system.

IV.3.3. Background change detection, and activity hot spots localisation

After segmentation and tracking, we propose a new simple and efficient way to highlight first the background change detection, and second the routes and activity hot spots localisation by generating an activity map and background model.

As we want to maintain the modularity of our approach, we chose to not use the background model generated by these pre-processing tools, so we build our own model. To have a model for background can help to detect all potential changes made by objects in the background, for the goal to differentiate between the many scene types, as mentioned in sub-section IV.3.6.

To estimate the background model, our method, uncommonly, is non-supervised method. Apart from that, the main target from highlighting the routes and activities spots is to help understanding the scene by indicating the most used areas in the Field of View (FOV), such as the main routes used by the moving objects or the spots for activities like their interactions. Following that the capability of indicating important clues in the trajectories of the moving objects when shifting from an area to another, triggering by that a description moment (see sub-section IV.3.8).

To this end we present a new activity map M_a. It is a matrix representing the temporary cumulative occurrence (till the current frame), where each value represents the number of times this pixel has been taken into consideration through the analysed sequence.

This activity map is initialized according to that definition:

$$M_a(x,y,0) = \begin{cases} 0 & If\ I(x,y,0)\ belongs\ to\ an\ object\ in\ frame\ 0 \\ 1 & Otherwise \end{cases} \quad \textit{(eq. IV-1)}$$

And the matrix of temporal activity is defined as:

$$M_{ta}(x,y,t) = \begin{cases} 0 & If\ I(x,y,t)\ belongs\ to\ an\ object\ in\ frame\ t \\ 1 & Otherwise \end{cases} \quad \textit{(eq. IV-2)}$$

Then for each frame of the sequence, we update the value of the activity map as follow:

$$M_a(x,y,t) = M_a(x,y,t-1) + M_{ta}(x,y,t). \quad \textit{(eq. IV-3)}$$

Based on this activity map, we calculate the background model consequently we detect the background change and we calculate the activity map following the routes and activity hot spots localisation.

IV.3.3.A. Changed Background detection

Even for one camera, many background models may be needed, especially when having long time of surveillance videos. This may be due to light changes, especially in outdoor scenes sunlight changes, or background inert objects displacement when adding or taking objects for long time from the background.

To detect all the changes applied on the background, we calculate a temporary background model based on the activity map presented in (eq. IV-4).

This temporary background model is a cumulative background (till the current frame at the moment), where all pixels in the past frames are taken into consideration except the one belonging to objects. The concept is to compare the current frame (all pixels not

belonging to an object) with the temporary background to detect if there is a big change occurring.

This temporary background matrix M_{tb} *is* initialised that way:

$$M_{tb}(x, y, 0) = I(x, y, 0) \,.\, M_a(x, y, 0) \qquad \textit{(eq. IV-4)}$$

Where "." is a scalar multiplication.

Then for each frame of the sequence, we update the value of the matrix as follow:

$$M_{tb}(x, y, t) = (M_{tb}(x, y, t-1).M_a(x, y, t-1) + I(x, y, t).M_{ta}(x, y, t))/\, M_a(x, y, t) \qquad \textit{(eq. IV-5)}$$

NB: $M_a(x, y, t)$ may contain zeros, and so, all the corresponding values in $M_{tb}(x, y, t)$ were not calculated but replaced by zeros.

Finally, once the whole sequence of N frames has been processed, the background model is generated as being the average image without moving objects, and equal to $M_{tb}(x, y, N)$. An example of the background model can be seen in Figure IV-3.

To determine when a background has changed and the moment when a new background model is required, we compare the temporary background model, each time, with the current frame to determine if a percentage p_p of pixels changed significantly their values. The algorithm fills out a "background changes" vector, referring to those frames.

$$V_{bc}(t) = \begin{cases} 1 & if\ p_p\ of\ I(x, y, t).M_{ta}(x, y, t)\ are\ different\ from\ M_{tb}(x, y, t).M_{ta}(x, y, t)) \\ 0 & Otherwise \end{cases} \qquad \textit{(eq. IV-6)}$$

Example of a background changes vector:

([0] [0] [0] [0] [0] [0] [0] [0] [0] [0] **[1]** [0] [0] **[1]** [0] [0] [0] [0] [0] **[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]** ….).

All the backgrounds detected as changed ones are not taken into consideration when calculating the temporary background model. As miss classified background change can occur, we use again the temporal-consistency algorithm proposed by (Jaffré & Joly, 2005) to correct this artefact. And the example above after filtering will be:

([0] [0] [0] [0] [0] [0] [0] [0] [0] [0] **[0]** [0] [0] **[0]** [0] [0] [0] [0] [0] **[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]** ….).

The change in the background can be temporary or permanent relatively. Many reasons and actions could be behind that. To indicate which case it is, a threshold (changed sequence cs_b) should be set designating for how much time (number of frames) the backgrounds remain detected as changed in the filtered vector before considering the change as permanent.

If the background change is temporary, this may imply a further investigation to indicate the reason and the location of the change. The detection of temporary change may help to classify the scene, as done later in the sub-section IV.3.6.

If a permanent change were detected, this indicates a new sequence of frames began. Subsequently a new background model is required, and all the above calculation of the activity map M_a and M_{ta} and the background temporary matrix $M_{tb}$and V_{bc} should be reset.

NB: Determining the above-mentioned thresholds p_p and the cs_b is very verbose and depends on too many factors; we mention the scene type, the FOV, and the user need. For

example, if the user needs to detect small object left or taken, like a bag, in an indoor airport hall scene, this p_p should be set to very small value, later cs_b can be set to several seconds only. On the contrary, in a traffic scene, detecting if a car is parked on the road, this may be satisfied with bigger p_p threshold, later cs_b can be set to several minutes.

IV.3.3.B. Routes and activity hot spots localisation

The routes and activity hot spots may extremely differ between one location and another and one sequence to another, especially when having non-constrained interactions in the real videos.

To localise and present the routes and activity hot spots on one map, the activity map, for the whole sequence of N frames, a simple calculation can be done. Once the whole sequence has been processed, we generate the activity map M_{am} as follow:

$$M_{am} = M_a(x, y, N)/N$$

For each of the scene frames where its background was detected as changed are not taken into consideration when calculating the activity map M_{am}. A permanent change detection of the background implies a new calculation of the activity map M_{am}.

In Figure IV-3, where in the activity maps we can see several degrees of grey. The darkest grey represents the smaller value, which means more objects pass by these points.

Figure IV-3: Example: background Model (left image), activity map (right image), part of video Fight_OneManDown ("CAVIAR," 2004).

In the activity map, the low values represent flows of moving objects in the FOV. To find routes and regions of activities, we use a simple method in three steps:

1- **Pixels quantification:** on the activity map, we did a linear quantification of pixel intensities into four levels, see Figure IV-4 to determine location types, from high to low intensity: marginal areas, regular routes, and hot spots (highly frequented locations). .
2- **Morphological filters:** Opening and closing operators are a$_{pp}$lied on the area generated by the previous quantification process to produce a smoother map, see Figure IV-4.
3- **Applying on background model:** finally, to visualize the routes and hot spots, we simply project the results on the background model. The marginal areas, the routes and the activity hot spots are shown in different colours in the following examples.

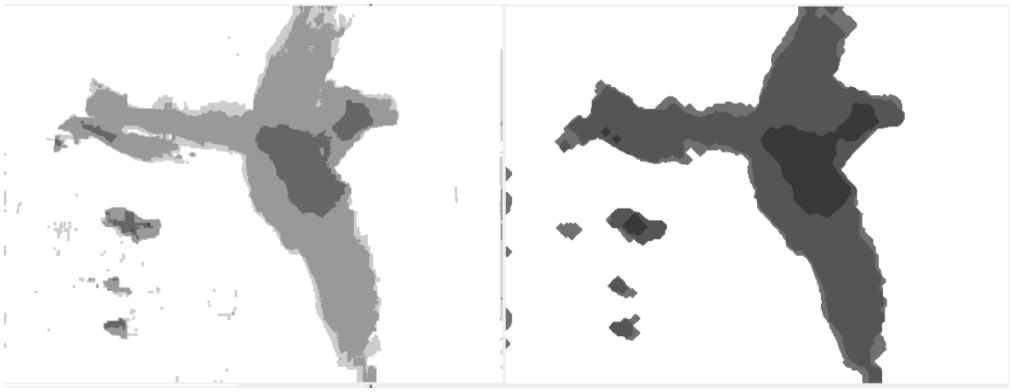


Figure IV-4: Pixels quantification of activity map (left image), and morphological filters result (right image), part of video Fight_OneManDown ("CAVIAR," 2004).

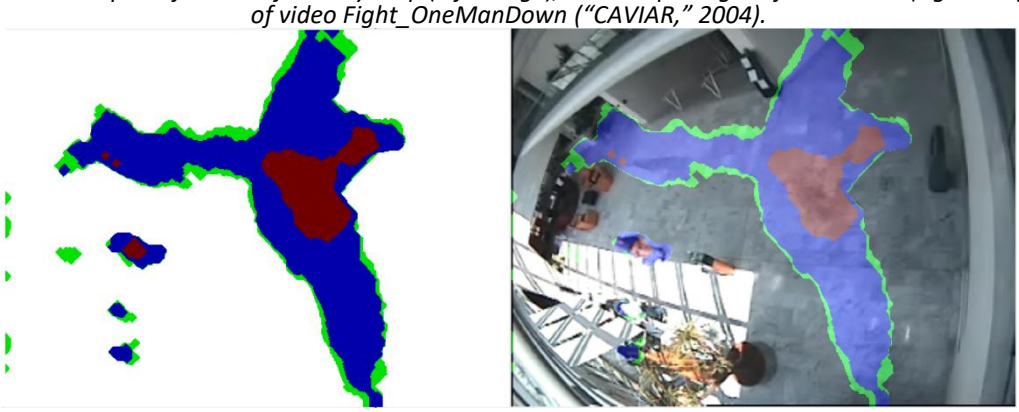


Figure IV-5: Coloured morphological filters result (left image), and projected to background model (right image), part of video Fight_OneManDown.

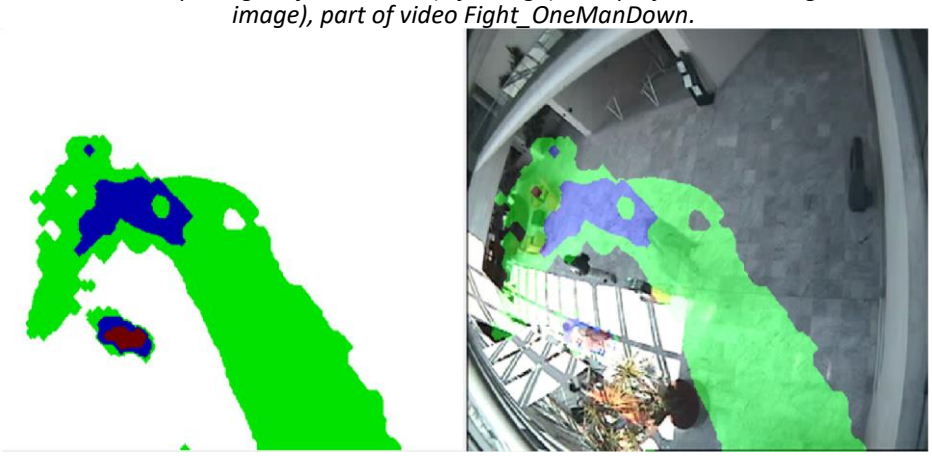


Figure IV-6: Coloured morphological filters result (left image), and projected to background model (right image) of scene "Fight_RunAway2".

Figure IV-6 shows the routes in green and the activity hot spots in blue and red, of scene "Fight_RunAway2", taken from the database "Caviar" ("CAVIAR," 2004). We notice in Figure IV-5, Figure IV-6 and many other examples, not shown here, that the red areas are mainly the road intersections, road turns and objects meeting areas.

The result of the above steps populates a scene model which can be used in the description phase, sub-section IV.3.8 or used for detecting anomalies.

IV.3.4. Video Segmentation

After segmenting and tracking video objects, we perform simple automatic segmentation of the video into sub-scenes. These sub-scenes take into account the number of objects. Four types of sub-scene are taken into consideration, see in Figure IV-7:

1- No objects.
2- One object.
3- Two objects.
4- Many objects.

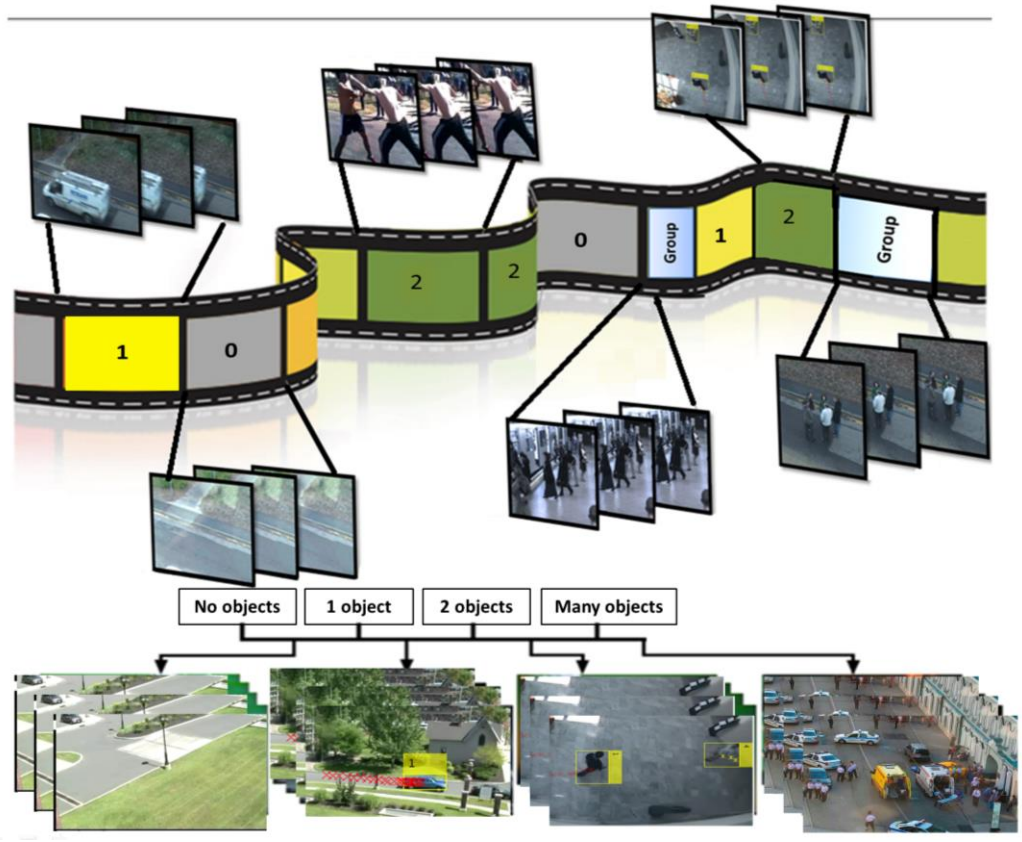


Figure IV-7: Simple video segmentation according to the number of objects in the scene

The actions can be performed by one object (self-action or interactions with the background), two objects (interacting with each other) or many objects (many interacting objects or crowd activity), and in each case many types exist.

Having one person in the scene performing gestures, or interactions with the background, is the subject of many researches, and some had already good results. This is also the case when many objects in the scene are performing group activity. Many interesting works exist in the field of crowd detection and analysis. The crowd analysis is an interesting field; it can help in managing the flow. However, this field focuses mainly on the crowd as one entity to study the flow, the number of persons, and others; and do not focus on incidents and interactions. In this thesis, since our high interest is to detect and analyse incidents in public places, we will focus mainly on sub-scenes corresponding to two interacting objects.

IV.3.5. Feature extraction

Selecting and extracting the right features has an important role in improving both the efficiency and accuracy, and can guide the algorithm to powerful results. Automatic extraction of features directly from images and videos to feed machine learning algorithms, as an end-to-end solution exist. As known so far, the learning models and algorithms are recent efforts and have achieved promising results on gestures, actions and group activities

recognition. However, few researches have been done on objects interactions. The obvious reason is the complexity of the task even for a human brain. It is challenging to analyse and understand an interaction in some cases; uncountable types of interactions can happen, and each can occur in many different ways. Even the reaction of an object in the interaction varies from case to case and may be unpredictable.

An improvement in the domain of object interaction analysis and understanding can be done, but it is still a challenging task, and a recruit to traditional feature extraction may be the best solution. In plus, the learning-based approach for videos is still not satisfying and does not take into consideration the temporal nature of interactions.

In our approach, we join the meaningful features extracted from objects, listed in this section, with the power of deep learning as it will be shown in experiments described later. We tried to extract the realistic features observed by human being when two objects are interacting. Later we extend the set of input features, using operators that generate combined features. These simple and combined features are intended to capture some higher-level ones about objects displacements and/or interactions to study a potential correlation. They are, for example, the relative distance between two objects, the Hu moments differences, angles of displacements, and the speed of the movement of an object, etc.

In two objects scenes, the duration of the shortest interaction considered were of one second length. Five types of features were extracted, spatial, temporal, inter-objects, inter-frames and trajectory features:

A. **Object spatial Features:** after the segmentation, in **each of the scene frames**, the features, shown in Table VIII-13 of appendix VIII.6, are extracted from the object 1 (obj1) state and object 2 (obj2) state. Objects spatial features are mainly concerning dimensions (width, height, surface, perimeter), position (x,y), shape (bbox, intensity, RGB, hu), and type (deformable/non-deformable).

B. **Object temporal Features:** after extracting the spatial features, the following features are extracted. Those features designate the variations occurring between the past frame (f-n) and current one (f) for object 1 (obj1) and object 2 (obj2) each one separately. Objects temporal features are mainly concerning variations in objects dimensions (width, height, surface, perimeter), displacement (distance, speed, angle), and shape (bbox, intensity, RGB, hu). More detailed explanations are presented in Table VIII-14 in the appendix VIII.6.

C. **Inter-Objects Features:** after extracting the spatial features, the following features are extracted. Those features designate the difference between features of the object 1 (obj1) and those of the object 2 (obj2). Objects inter-objects features are mainly concerning variations between the two objects dimensions (width, height, surface, perimeter), displacements (distance, speed, angle), and shapes (bbox, intensity, RGB, hu). More detailed explanations are presented in Table VIII-15 in the appendix VIII.6.

D. **Inter-Frames Features:** We take a window of M frames. In this window, and **for each frame**, we extract (see Figure VIII-15):
   - The spatial features of object one and object two
   - The temporal features of object one and object two
   - The inter-objects features for object one and object two

For almost each of the features, we extract the derivative and second derivative, if it can be applied. Then for **each of the features (f)**, **its derivative (df)** and **its second derivative (ddf)**, we find seven global **inter-frames features** (see Table VIII-16 in the appendix VI which are the minimum, the first, the last, the middle, the average, the median and the STD, normalized by the maximum value).

E. **Trajectory Features:** The last set of features is related to trajectory. For that, we compute the trajectory of object one and object two centroids through window frames, and the trajectory of the middle points between object one and object two centroids through window frames. For each of those three trajectories, we apply three smoothing filters: the average filter, the first order prediction filter and the second order prediction filter. For information on these filters the reader may refer to the appendix VIII.6.
Finally, for each of those three original trajectories (object one, object two, and middle trajectories of objects), we generate theses new filtered trajectories. So, applying that, we obtain nine trajectories. At last, for each of those new trajectories we calculate two features, **the standard deviation (of the distance between the filtered position and the one corresponding to the centroid),** and **the largest distance, rendering so some information about the smoothness or the chaotic aspect of the trajectories.**

Finally, for each window, we have one set of features containing the inter-frames features and the trajectory features, see Figure IV-8. In all we have for each window 2498 features (divided between 926 inter-frames of spatial features, 1008 inter-frames of temporal features, 546 inter-frames of inter-objects features and 18 trajectory features).

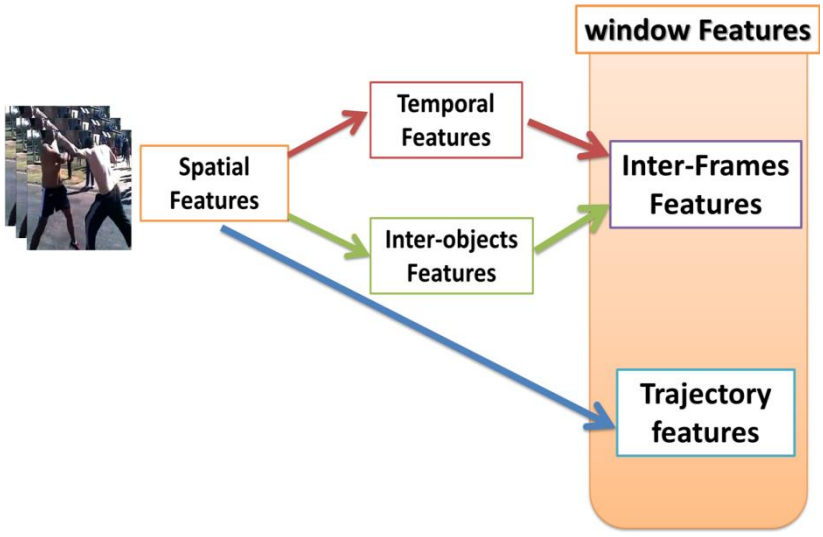


Figure IV-8: Features extracted from a window of N frames. Each set of features at an arrow starting point feeds the set of features at its terminal point.

These extracted features are well chosen to be meaningful so they can be used for many reasons:

- **Classifying different interactions types:** they are suitable to detect the existence of interaction between the objects, and to classify if the interaction is far or close, aggressive or peaceful.
- **Generating alerts:** they contain very important features to trigger the alerts when needed in real scenes.

- **Querying intelligently the archives:** they contain very important features and most used in real cases to count on when investigating intelligently the archives.
- **Generating the textual description:** they are suitable to fill the structural templates useful for the real cases.

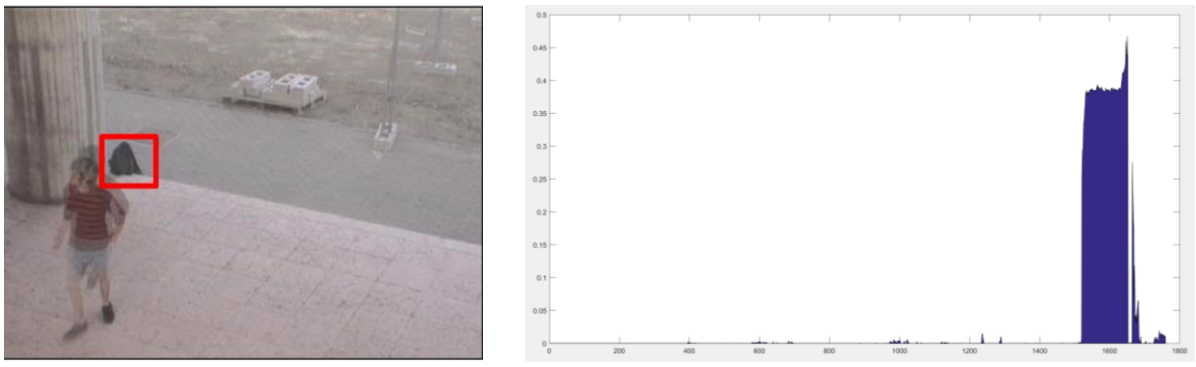
IV.3.6. Scene type classification

In our ontology, we proposed to classify scenes type into 15 types according to the number of objects and the type of action and interactions performed. In the Table IV-2, we present a simple way to discriminate between 13 of 15 types according to background changes, the number of objects and objects characteristics (features) changes before and after performing the action. Let us consider here that a scene is a temporal window of a predefined length centred on a location where a change occurs (number of objects, background …).

For the number of objects, before and after the action, we use the results of the video segmentation (see sub-section IV.3.4). Concerning the background change, the results are taken from the background change detection, as explained in sub-section IV.3.3.

For features changes of moving objects, we mainly focus on two characteristics, Hu moments (Hu_objs_diff) and surface (S1_objs_diff), mentioned in sub-section IV.3.5. The following table summarizes the expected evolutions of those features in different cases.

For example, the classification of the type "1 Object → 1c" corresponds to an object left in the scene. Figure IV-9-a shows a scene where a person leaves a bag in the FOV and another person takes it. The right side Figure IV-9-b presents the graph of the comparison between each of the scene frames and the corresponding temporary background model, see sub-section IV.3.3. It easily indicates critical background changes near the frame 1500 (the time the bag was left in the FOV), and back to normal near the frame 1700 (the time the bag was picked up).



a b

Figure IV-9: Example scene left bag (left figure a), on the right side: X axis is the frame's number, Y axis is the percentage of changed pixels between $I(x,y,t).M_{ta}(x,y,t)$ *and* $M_{tb}(x,y,t).M_{ta}(x,y,t)$.

Table IV-2: Scene type classification into 15 classes according to number of objects before and after the action, background changes and object features of the moving objects

Features to test / Type of scene		Number of objects before action	Number of objects after action	Background changes detection	Features changes of moving objects
1	**0 object**	**NA**	**NA**	**NA**	NA
2	**1 Object → 0a**	**1**	**0**	**No**	
3	**1 Object → 0b**	**1**	**0**	**Yes**	
4	**1 Object → 1a**	**1**	**1**	**No**	
5	**1 Object → 1b**	**1**	**1**	**Yes**	No Hu moments (Hu_objs_diff) changes
6	**1 Object → 1c**	**1**	**1**	**Yes**	Hu moments (Hu_objs_diff) changes, and surface (S1_objs_diff) changes In plus, for Left object case: Hu moments or surface are smaller. For taken object case: Hu moments or surface are bigger
7	**1 Object → 1c**	**1**	**1**	**Yes**	All the moving object features indicate different object before and after
8	**1 Object → 2a**	**1**	**2**	**Yes**	
9	**1 Object → 2b**	**1**	**2**	**No**	
10	**2 Objects → 1a**	**2**	**1**	**Yes**	
11	**2 Objects → 1b**	**2**	**1**	**No**	
12	**2 Objects → 0**	**2**	**0**	**Yes**	
13	**2 Objects → 2a**	**Need more processing, see section IV.3.7**			
14	**2 Objects → 2b**				
15	**Many Objects**	**Many**	**Many**		

IV.3.7. Interaction classification

In a surveillance control room, when observing a dynamic scene like public places, motion is the daily basis. And as seen in sub-section IV.3.6, various scene types with objects performing actions can be present. But the most important part of observation or analysis for public place surveillance is to focus on irregular actions like unusual interaction between objects in the scene, mainly humans, vehicles, or any kind of moving objects, either deformable or non-deformable. In fact, two objects interacting can be of high interest when analysing incidents in public places.

And so, after detecting the existence of two objects in the scene, deeper analysis and classification is done in this section in order to decide, mainly, whether there is interaction or no. This situation has been labelled as 2 Objects → 2a and 2 Objects → 2b in our ontology. There is a lack of scientific studies targeting this special case.

There are many points here to be considered. First, many types of objects exist. Second, the interaction can be distant, physical or both at different consecutive times. We consider it distant, when there is no physical touch between the objects and, on the contrary, as a close one. Third, unlimited types of interactions can occur.

Working in a CCTV control room, especially when dealing with security, every second counts. The sooner the operator detects the incident, the quicker potential damages can be minimized. Most of the interactions start as “distant”. For example: saluting before shaking hands, heading to a fight, shouting before hitting, avoiding accident then fighting, etc.

For that, it is important to detect the moment a distant interaction starts. Popping up cameras when distant interaction starts before a fight or accident, can give more time to the operator to focus with PTZ (movable camera) on what is important before losing it, for example, the plate number of a car or the description of the offender before he runs away, etc. In many incidents, the difference between revealing the truth or not by the operator is measured in seconds.

To deal with these problems, especially the three mentioned points, we took the following strategies. The first point concerning objects types, as mentioned, we choose to classify the objects in two main categories, deformable and non-deformable according to chapter III, which is the most visual important feature that can affect the way an object can perform the act. Later, more features are needed for the deformable objects concerning the sub-objects, and further sub-categories can be classified for deformable and non-deformable objects, see chapter II. For the second point, after creating a sliding window and extracting objects and interactions features from that window, many classification algorithms were tested, mentioned in section IV.4, to classify the existence of interaction between the two objects inside the window. We use a neural-based deep model algorithm as the classification algorithm. The first part of the model learns inter-frames and trajectory features. Then, the second step consists in using these learned features to train the algorithm in order to classify the entire window sequence. Later, for the entire scene, after having classified each window as showing an existing interaction or not, then the same concept used in chapter II for temporal-consistency analysis proposed by (Jaffré & Joly, 2005) can be applied to correct the wrong classified windows.

After classifying the scene where there is an interaction between the two objects, then third point is aiming at going deeper into more layers on analysing this interaction. We add more sub-categories like: aggressive, non-aggressive, bad influence, good influence, etc.

In this study, we choose to classify the interactions, when exists, into objective-likely classification as distant or physical, and more subjective classification as aggressive or non-aggressive. Using the selected features, mentioned in section IV.3.5, the objective is to detect and classify potential interaction between two objects in a temporal window, and later to classify if it is distant or physical interaction, aggressive or peaceful.

To perform these tasks, then, after the dataset scenes selection, see sub-section IV.4.1, and feature extraction, see sub-section IV.4.3, the datasets were prepared and pre-processed, see sub-section IV.4.4. After that, many tests on different machine learning algorithms (classical algorithms or Neural networks) were made, see section experiments IV.4.5.

Consequently, we choose different multi-layered Deep Neural Network DNN. For implementation, simple Feedforward networks called Pattern recognition networks in MATLAB and Simulink environment (“Pattern recognition network - MATLAB,” n.d.) was learned. In order to achieve the desired outputs, several tests were made after altering this model to handle deeper architecture (by adding more layers) and by determining the best parameters that maximise the results such as the activation function, the training method, etc.

IV.3.7.A. Interaction vs non-interaction classification

Our objective here is to detect if an interaction exists between the two objects in the selected scenes. For each of the windows, the algorithm extracts the features and feed the DNN to attribute the value 1 or 0 to the window according to the existence of an interaction or not between the two objects, see Figure IV-10.

After tests, the chosen algorithm is the seven-layer pattern recognition network, a feedforward network composed of fully connected layers, six hidden layers and one output layer. Each hidden layer contains 586 neurons. The six hidden layers' activation function is the Log-sigmoid transfer function and a softmax transfer function in the output layer. Other parameters are shown in the experiments section IV.4. As result of training this algorithm and testing it with approximate of 14902 records and 2305 features, takin from 285 scenes, the accuracy of the test set achieved 87.5% after 325 epochs.

IV.3.7.B. Distant vs physical interaction classification

Most of the interactions in the real word start from a distance (saluting, fight …), detecting if an interaction is distant (far) or physical (close) have big importance. But working on distant interaction, is not an easy case, and as for our knowledge, we couldn't find any research taken into consideration this case.

For that, if interaction is detected between two objects in the windows, here, we classify if each of those interactions in the selected scene is distant or physical. After tests, on many machine learning algorithms, the chosen algorithm is the four-layer Pattern recognition network of fully connected layers, three hidden layers and one output layer. Each hidden layer contains 426 neurons. The three hidden layers' activation function is the Log-sigmoid transfer function and a softmax transfer function in the output layer. Other parameters are shown in experiments section IV.4. As result of training this algorithm and testing it with approximate of 7079 records and 2303 features, the accuracy of the test set achieved 93.7% after 102 epochs.

IV.3.7.C. Aggressive vs Non-Aggressive interaction classification

Detecting an interaction if it is aggressive or peaceful is very important especially in the domain of public safety and security. Moreover, detecting distant aggressive interaction can alert the CCTV control rooms' observers at early stages, giving them precious time to act. As mentioned, in an incident, an offender may leave the scene in seconds; if the observers were alerted at the right time they may use the movable cameras to catch the plate number before he leaves the crime scene.

For that, if interaction is detected between two objects in the windows, here, we classify if each of those interactions in the selected scene is aggressive or non-aggressive. After similar tests as above, the chosen algorithm is the four-layer Pattern recognition network of fully connected layers, three hidden layers (with Log-sigmoid) and one output layer (using softmax). Each hidden layer contains 546 neurons. Other parameters are shown in experiments section IV.4. As result of training this algorithm and testing it with approximate of 6703 records and 2303 features, the accuracy of the test set achieved 93.8% after 88 epochs.

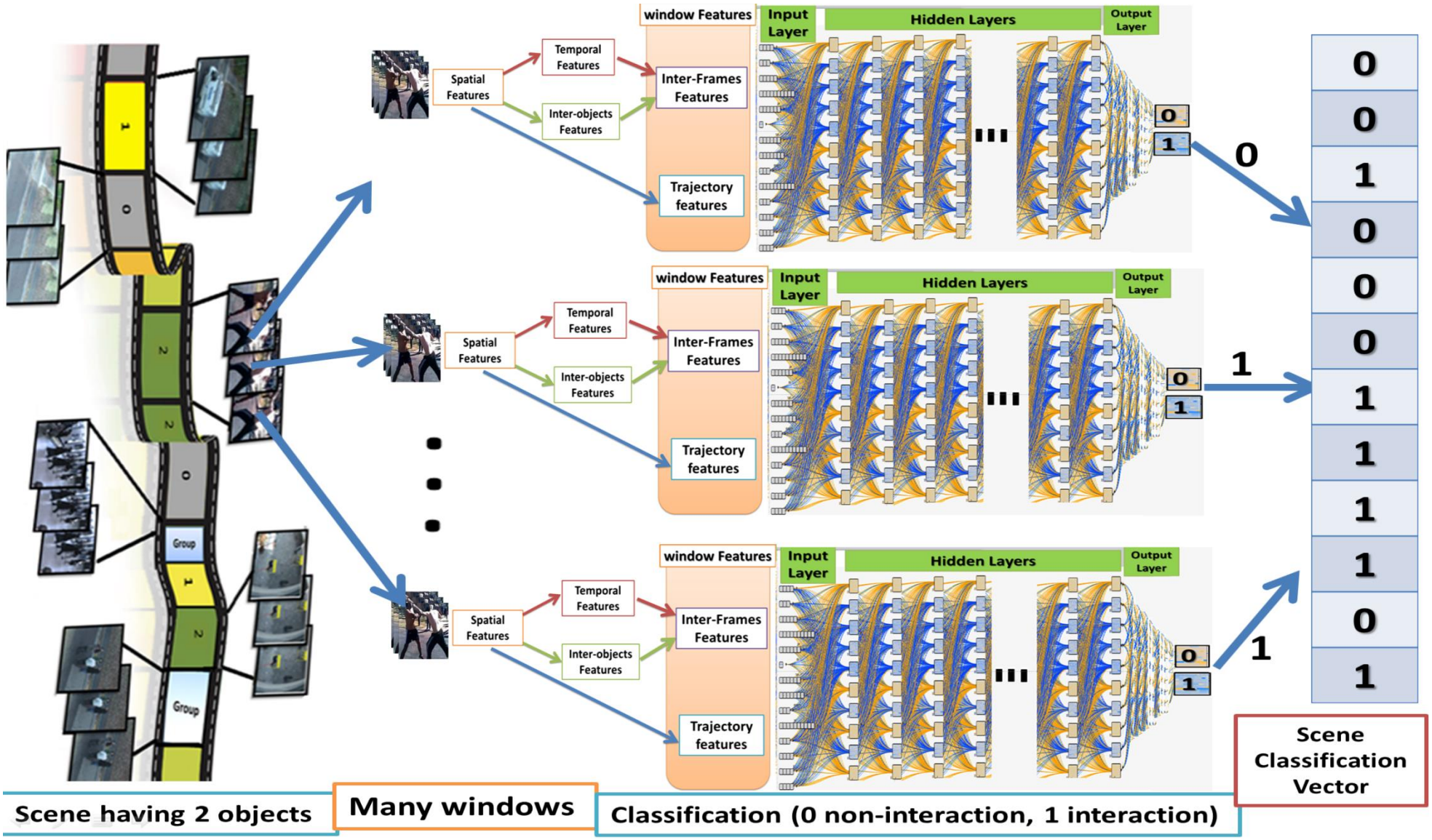


Figure IV-10: Extracting the scene classification vector indicating the existing or not of interaction.

Finally, after detecting for each window the existence of interaction (or not) between two objects in the selected scenes; also, whenever an interaction is detected, if it is distant or physical, and if it is aggressive or peaceful; all the results of each type of classification in the scene is collected and concatenated in 3 sequential classification vectors, see Figure IV-10. As the output vectors may contain wrong decisions, for that, we applied the method proposed by (Jaffré & Joly, 2005) to correct these marginal wrong decisions. After this post-processing step, the resulting vector will be more coherent and presents sequences of 0's and 1's indicating (the 1's) the time when the interaction starts and when it finishes.

IV.3.8. Scene analysis and description

"At frame 392: Deformable object 2 moves, in F spot, on the right middle of the inside area of the camera field of view, heading immediately down right, toward the object 1, no big change occurring respectively on its shape, and big changes occurring respectively on its surface having now bigger one, and having respectively slight increasing of its Speed.

"The two objects are respectively receding; a physical peaceful interaction occurs between them."

Example of scene description and of object interaction description.

Having such a description based on activity analysis of a scene can be very beneficial for video surveillance in live mode, for example, the system, with some key text description like "aggressive", can raise alerts, and help in the post processing of this situation for dispatching patrols, etc. Also, specific information of object type, shape, displacement, location, and/or direction, interaction type, can be searchable using textual queries, and this could help investigators to solve many problems and generate textual reports. In AppendixVIII.8, a sample of police report is shown.

When dealing with real incident cases, a big need appears is to have an answer quickly on the five mentioned "W"s (Who, what, where, when, and why), plus the How. Answering those can set the right frame to scene description. For that, in our method we describe the scenes, in an abstract way and no matter what the context is. We mainly focus on scenes having two objects, but the same method can be used for different types of scenes. In each scene, we want to describe textually what is happening in many sentences localised along a temporal dimension.

So, from the extracted features we analyse in the scene:

- What is related to each of the two moving objects: on many states, at relevant moments and positions in the frames, like type, shape, and displacement.
- What is related to the two moving objects together: on many states, at relevant moments and positions in the frames, like inter-objects distance, position and direction.
- What is related to the objects interaction if exists: before, during (distant and physical) and after, and if the interaction is aggressive or not.

Then, differently than the works presented in (Dogra et al., 2016) where the authors extract key frames only from the trajectory, we extract **key moments** from those several characteristics, where the object/interaction corresponding states, show irregularity to trigger the description. Triggered by these key moments, a **scene activity characteristics matrix** of corresponding characteristics is filled.

Finally, we apply logical rules on the scene activity characteristics matrix to generate textual description sentences by filling new proposed structured templates.

IV.3.8.A. Scene key moments

We chose several characteristics representing the behaviour of the objects in different aspect. These characteristics are then represented using a graph-based pattern discovery, from which we extract key moments, to generate the proposed description. For better understanding these key moments, we show some of the key frames, for the scene "LeftBox" from the database ("CAVIAR," 2004), between the Figure IV-24 and Figure IV-29. These characteristics are:

1- Object characteristics:

a. The object type: as mentioned in chapter III, a so-called deformable object type can change its behaviour between non-deformable and deformable, while a non-deformable object is not able to do so. These changing moments are considered key moments.

b. Shape related: in the scene, many shape features are relevant to trigger the scene description. For each of the two objects, we analyse the next two features, searching for big variations on their values as key moments:
   i. Invariant moments or "Hu moments" are shape features used to generate the temporal feature "Hu_objs_diff" from Appendix VIII.6 Brutal changes of Hu moments indicate a big change or irregularity in the shape of the object. See Figure IV-11.
   ii. The object surface is associated with the temporal feature "S1_objs_diff" from Appendix VIII.6 Here again, brutal changes of this feature may indicate that something is occurring on the object surface. See Figure IV-12.

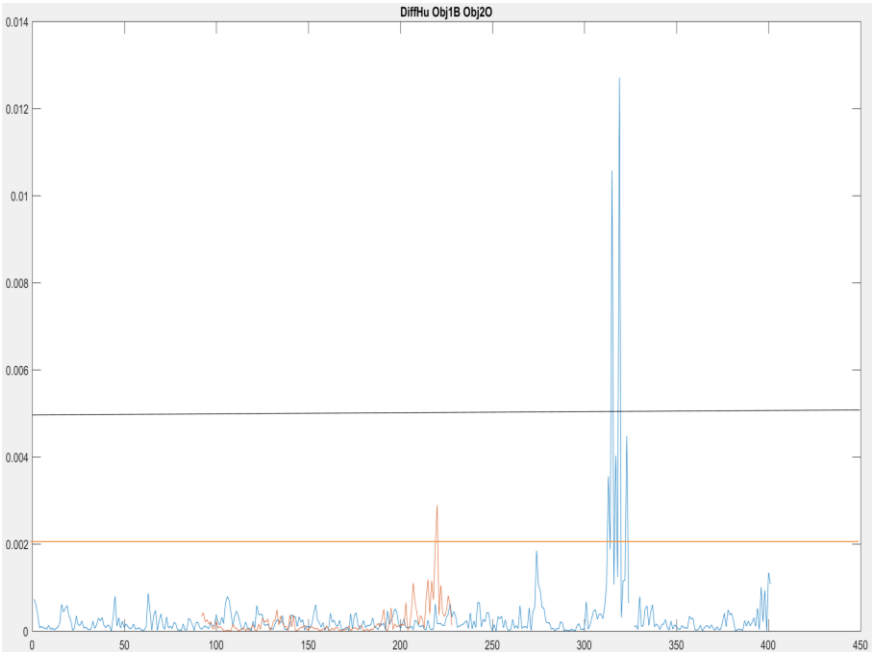


Figure IV-11: Objects Hu moment variations in the scene "LeftBox" from the database ("CAVIAR," 2004), it shows a brutal variations between frames 300 and 350, this is due to false detection.

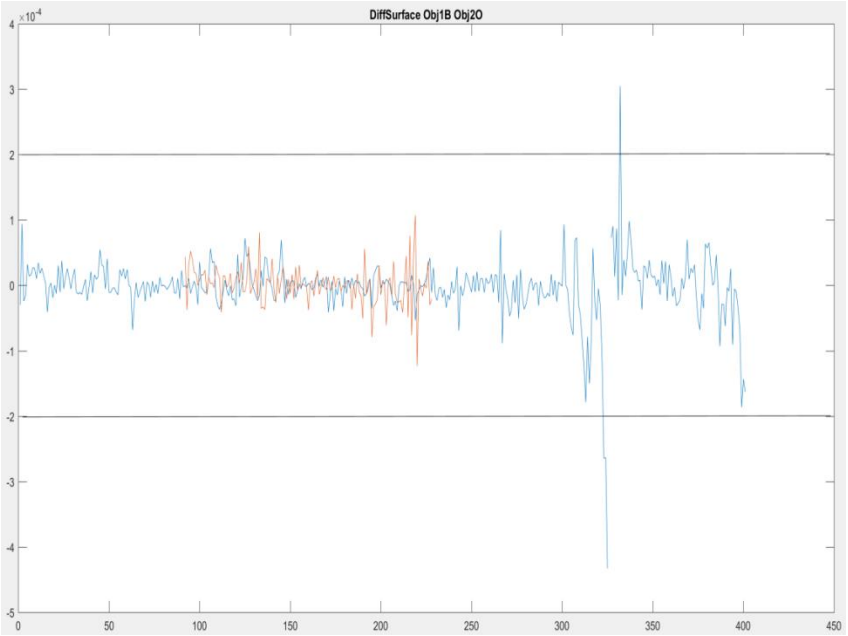


Figure IV-12: Objects surfaces variations in the scene "LeftBox", same false detection of the object 1 influence its surface.

c. Position related:

i. Position in the frame: indicating when an object moves from one area to another in the FOV can be considered a key moment. We quantized the position space in the frame into sixteen possible values taken according to the field of view (Figure IV-13): Up Middle (UM), Down Middle (DM), Right Middle (RM), Left Middle (LM), Up Left (UL), Up Right (UR), Down Left (DL), Down Right (DR). Also, the field of view is divided according to the centre into two areas: inside (I), and outside (O).

ii. Position to an area of interest: important moments to describe the objects states are when they enter or exit areas of interests (routes and activity hot spots) located in the scene frame, seen in section IV.IV3.3 For example, in the whole scene "*LeftBox*", taken from the database "Caviar" ("CAVIAR," 2004), in Figure IV-14 and Figure IV-15. we identify different areas according to the regions intensities taken from pixels quantification and morphological filters of the activity map, and projecting the results on the background model, all seen in section IV.3.3 These areas represent the marginal areas (higher intensity region in white colour), regular routes (in green colour), and 2 levels activity hot spots (in blue colour and in red colour for lower intensity region). Then we label with a letter each area from the highest region intensity to lowest one, except the marginal areas. Here we have 3 areas: A, B, and C.

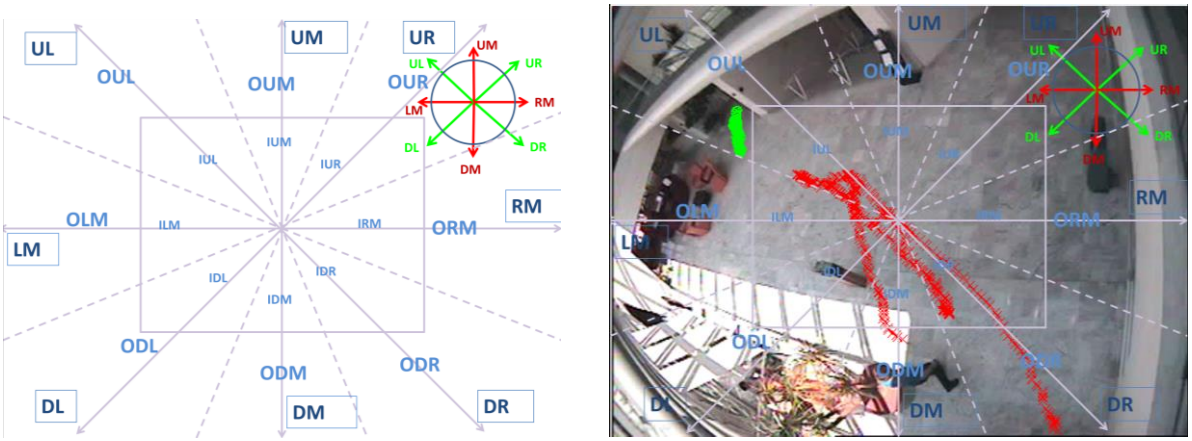


Figure IV-13: Left image shows the eight directions (in red and green) and the sixteen areas (dashed lines are the sectors borders). Right image shows trajectories (object1 trajectory in red, and object2 trajectory in green.

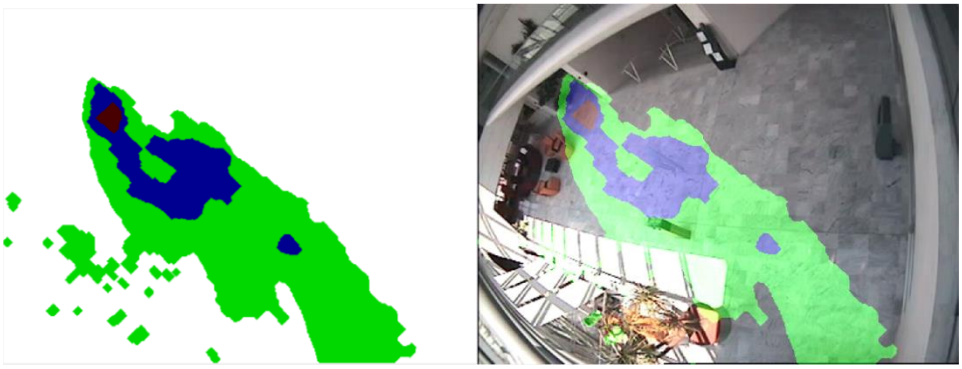


Figure IV-14: Figure showing the routes in green and the 2 levels of activity hot spots in blue and red, of scene "LeftBox.

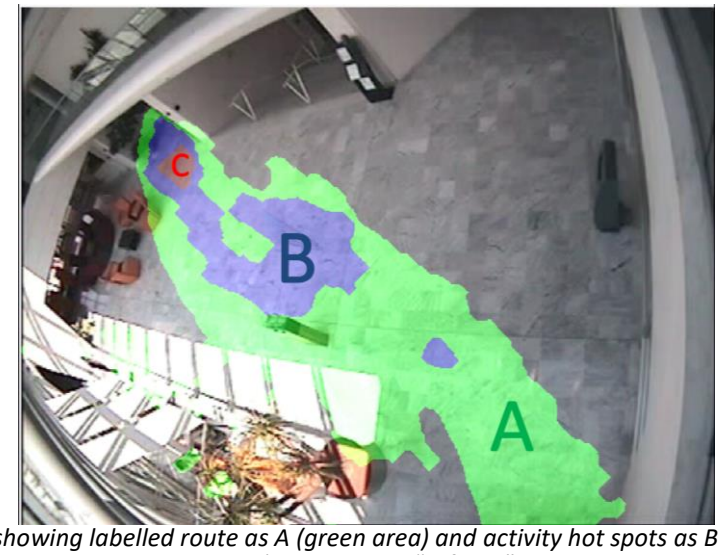


Figure IV-15: Figure showing labelled route as A (green area) and activity hot spots as B (bleu area) and C (red area) in the scene "LeftBox".

d. Displacement related: we choose two features to be analysed:
   i. The object speed variations (Figure IV-16): where the most important is when the object applies a big brutal change in its acceleration between state and another; we use for this the feature the temporal feature Speed to calculate the acceleration.
   ii. The object direction variations (Figure IV-17): we quantized the direction space into eight possible values (Figure IV-13) to observe brutal changes of that feature (from at least 2 quantization steps). We use for this the feature the temporal feature Angle.

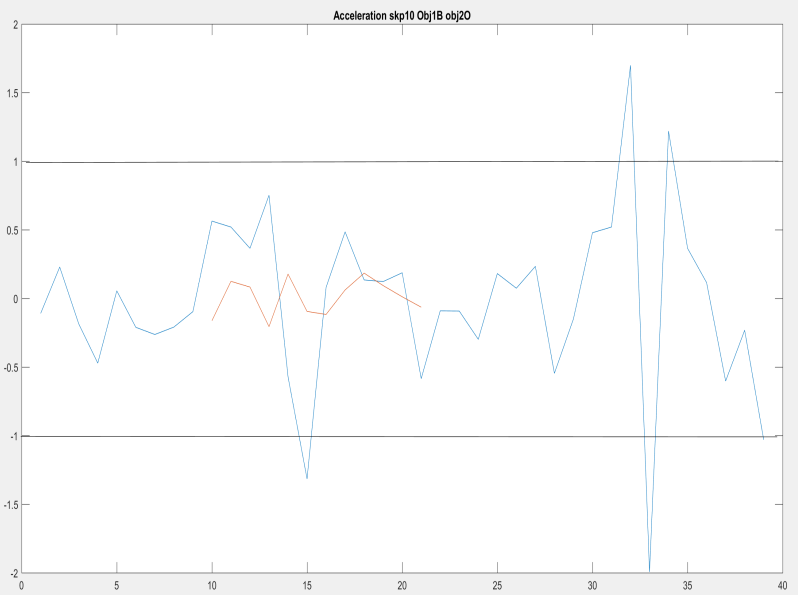


Figure IV-16: Objects speed variations (skipping 10 frames between 2 positions used for computations) in the scene "LeftBox".

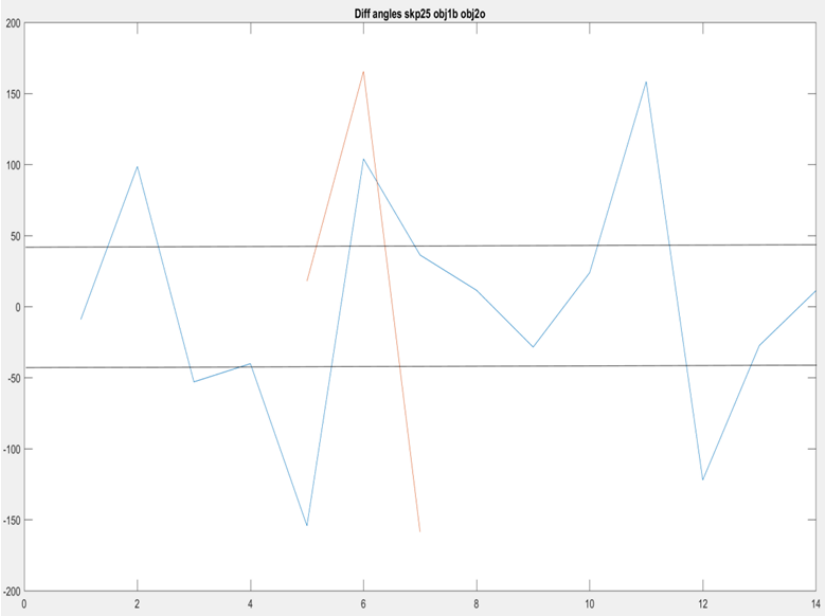


Figure IV-17: Objects angles variations (skipping 25 frames between 2 positions used for computations) in the scene "LeftBox ".

2- **Inter_Object characteristics:** in the scene, many inter_objects features are relevant to trigger the scene description. We chose only two of them:
   a. Distance between the objects: Its local minima or maxima can help to decide when generating a description. See Figure IV-18.
   b. Position and direction compared to the other object: here, key moments are indicated in two cases: first, when an object starts and ends going "toward" the other object; second, when that object starts and ends getting "away from" that other object. Object 1 is considered heading toward the object2, when its vector direction is pointing close to the position of the object 2; otherwise, when this vector is in opposite direction, object 1 is considered getting away from the object2.

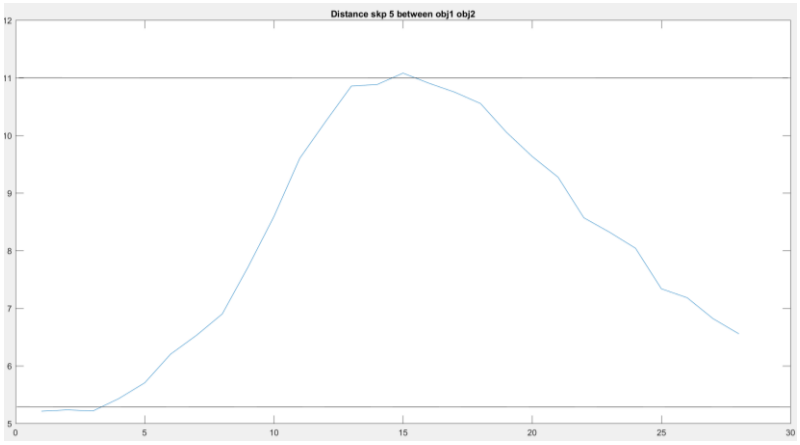


Figure IV-18: Two Objects distances (skipping 5 frames between 2 positions used for computations) in the scene "LeftBox".

3- **The interaction features:** for these features, we take the results of the states' values analysis in section IV.3.7 to indicate the variance in three: first, interaction existence classification variations are shown in Figure IV-19, Figure IV-20, and Figure IV-21; second, distant vs physical interaction classification variations (Figure IV-22); and third, aggressive vs peaceful interaction classification variations (Figure IV-22).

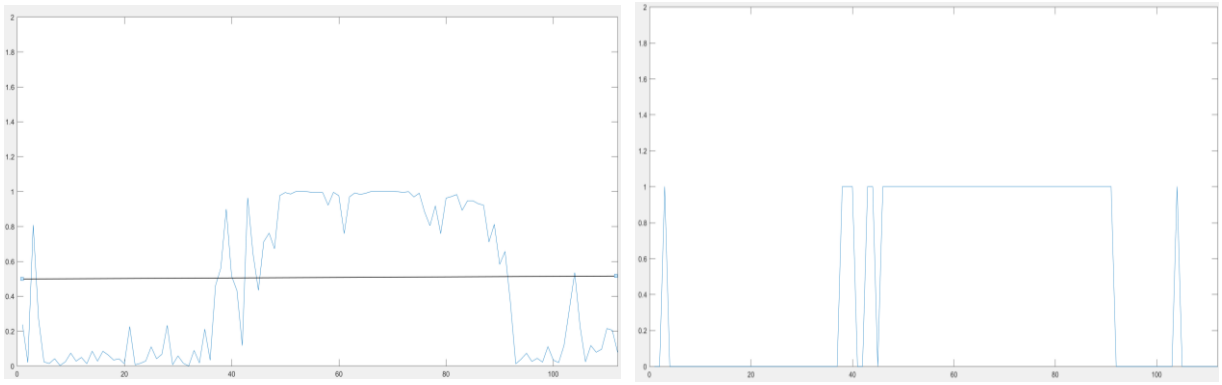


Figure IV-19: Interaction existence classification results for the scene "LeftBox", (0 no–interaction, 1 interaction): a- classification direct results, b- results after quantification (>=0.5) without filtering.

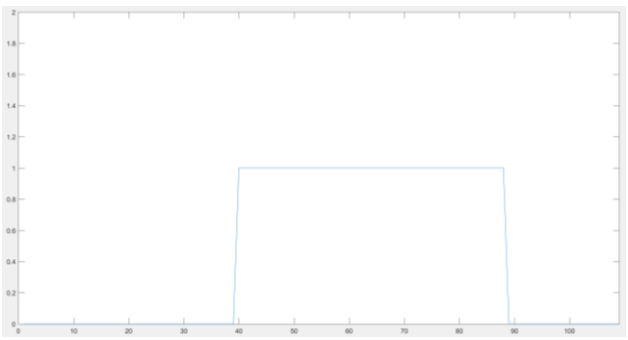


Figure IV-20: Interaction existence classification for the scene "LeftBox" after filtering using (Jaffré & Joly, 2005), shows an interaction began to appear at window starting at frame 40.

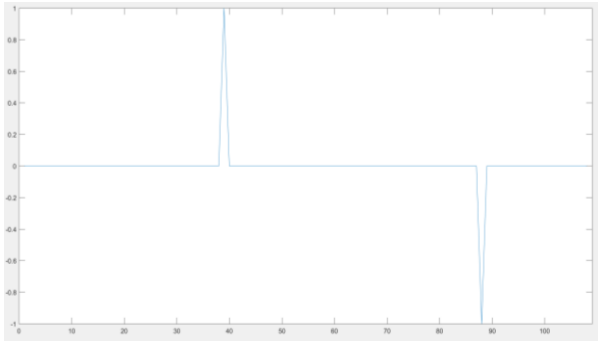


Figure IV-21: Interaction existence classification variance for the scene "LeftBox" after filtering

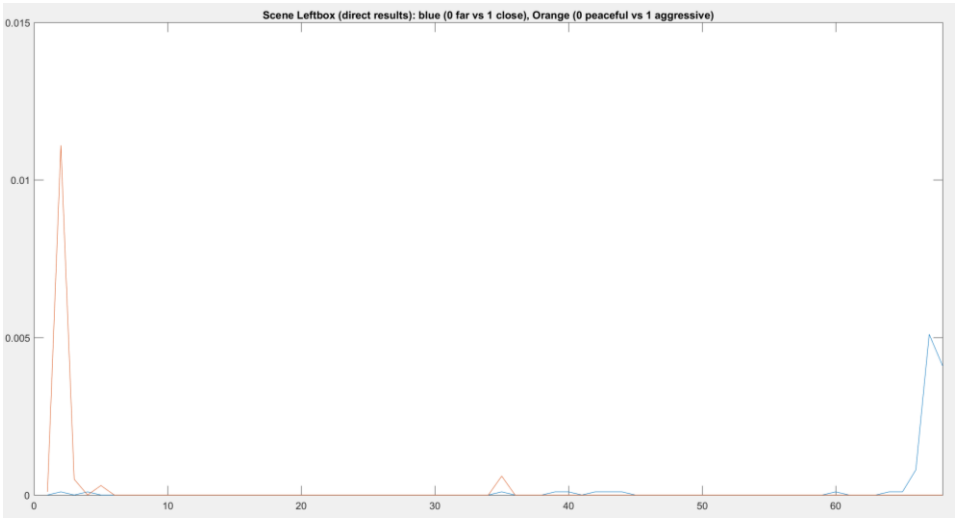


Figure IV-22: Classification direct results: blue line (physical (1) or distant (0)), orange line (peaceful (0) or aggressive (1)). Obviously all values are very close to 0 (distant and peaceful interaction).

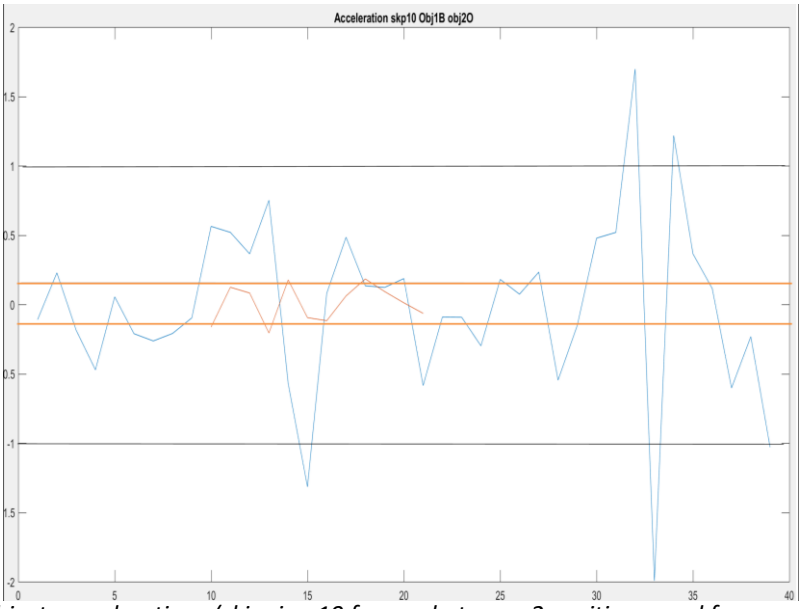


Figure IV-23: Objects accelerations (skipping 10 frames between 2 positions used for computations) in the scene "LeftBox": different threshold should be chosen for each of the objects.

After analysing all those characteristics, key moments should be extracted according to important variations showing irregularity. The importance of the variations differs from scene type to another and from user needs to another. For example, the object acceleration thresholds depend directly on the scene type, were the velocity of an outdoor circulation scene differs dramatically from an indoor one. Even though in the same indoor type scene, these thresholds differ according to the limitations and conditions set by the responsible on the monitoring system. Another example, the acceleration variations graph presented in the Figure IV-23, belonging to the "LeftBox" scene having a large Depth Of Field (DOF), the user should take that into consideration and select different thresholds to each person, or the key moments of one of them will not be presented in the description. Moreover, while some end users want to keep text generation to the minimum, others want to generate text for every frame.

Taking all of that into consideration, and as we want to keep our system generic and context-free, we kept the verbosity of the description density to be controlled by the user according to the scene type. And so, those thresholds can be fixed manually, using a percentage scale or regulator for example, by the end user in order to de determine the amount of text generated by each feature. Different thresholds indicate different number of key moments for each characteristic. Triggered by these key moments, a scene activity characteristics matrix of corresponding characteristics is filled.

In the "LeftBox" scene analysis, as shown in figures Figure IV-11, Figure IV-12, Figure IV-16, Figure IV-17, and Figure IV-18 we chose the thresholds in a way to not exceed more than 5 key moments for each characteristic.

IV.3.8.B. Scene activity characteristics matrix

For each video (VIDEO ID) and each scene (activity ID), which begins from birth frame to death frame, and according to the key moments ($k_m \in \{k_1, k_2, \dots k_n\}$) extracted, the corresponding characteristics were generated and filled in a vector V(k_m), as follow:

$$V(k_m) = \{Frame_{nbr}(k_m), Obj1_{char}(k_m), Obj2_{char}(k_m), InterObj_{char}(k_m), Interaction_{char}(k_m)\} \textit{ (eq. IV-7)}$$

Where $\mathrm{ObjI_{char}(k_m)}$ = {type, position to an area of interest, position in frame, direction, position vs other object, shape_hu, shape_surface, speed} characteristics at key moment k_m, for each object (I=1,2); and $\mathrm{InterObj_{char}(k_m)}$ = {distance between objects} characteristic at key moment k_m; and $\mathrm{Interaction_{char}(k_m)}$ ={Interaction existence, Interaction type (Distant (far) vs Physical (close)), Interaction aggressiveness (aggressive vs peaceful)} characteristics at key moment k_m.

Finally, all the V(k_m) produce the scene activity characteristics matrix M_{sac}, following the equation:

$$M_{sac}(ACTIVITY_{ID}, VIDEO_{ID}) = \{V(k_1), V(k_2) \dots, V(k_m), \dots V(k_n)\} \textit{ (eq. IV-8)}$$

This matrix can be easily used to build sophisticated scenarios models based on context characteristics thresholds, which can be set to generate wanted alerts. Also, this

matrix can archive needed characteristics, and make them quickly and easily available using simple queries for later retrieval. An example of M_{sac} can be seen in Table IV-3.

IV.3.8.C. Scene description

For the scene description, we use a template-based method to generate texts to represent objects interactions and activities at the extracted key moments in the scene. For that, we propose new ontology-based (see sub-section II.3.6) generic structured templates containing main information reported by the police in case incident description.

Mainly, we propose two types of templates:

1- Object characteristics templates: for each of the objects, three templates were introduced:

a. At the key moment when an object enters the scene, the description follows a template, called "Object entrance template":
{"Type" object "ID" enters the scene, {from "Area of Interest" spot | outside areas of interests}, on the "Frame Area Symbol" of the {"Inside" | "Outside"} area of the camera field of view, heading "Direction", [{"Toward" | "Away from"} the object "ID",] having respectively {"Regular" | "Irregular"} shape, {"Small" | "Medium" | "Big"} Surface, and {"Slow" | "Normal" | "High"} speed}.

NB: In these templates, words between quotes denote value, clauses between curly brackets indicate many clause templates to decide among them according to rules, and clauses between brackets indicate that the clause may not be present in the output sentence (to be decided according to rules).

b. At each of the key moments of an object (other than its entrance or exit), a proposed "Object characteristics template" is structured as follow:
{"Type" object "ID" {moves, in "Area of Interest" spot, | leaves "Area of Interest" spot, and moves, | leaves "Area of Interest" spot, and moves, in "Area of Interest" spot, | moves,} on the "Frame Area Symbol" of the {"Inside" | "Outside"} area of the camera field of view, heading [immediately] "Direction", [{"Toward" | "Away from"} the object "ID",] {"No big changes occurring respectively on" | "Occurring respectively irregularity in"} its shape, and {"No big changes occurring respectively on" | "big changes occurring respectively on"} its Surface [having now {"Smaller" | "Bigger"} one], and having respectively [considerable | slight] {"Increasing" of its | "Decreasing" of its | "Stable"} speed}.

A graph representation of this template is presented in Appendix VIII.7 Figure VIII-16.

c. For an object exiting, a proposed "Object exit template" is structured as follow:

{"Type" object "ID" [leaves "Area of Interest" spot] and exits the scene, {from "Area of Interest" spot | outside areas of interests}, on the "Frame Area Symbol" of the {"Inside" | "Outside"} area of the camera field of view, heading [immediately] "Direction", [{"Toward" | "Away from"} the object "ID",] {"No big changes occurring respectively on" | "Occurring respectively irregularity in"} its shape, and {"No big changes occurring respectively on" | "big changes occurring respectively on"} its

Surface [having now {"Smaller" | "Bigger"} one], and having respectively [considerable | slight] {"Increasing" of its | "Decreasing" of its | "Stable"} speed}.

2- Inter-objects characteristics templates: two templates were introduced:

a- When the second object enters the scene, at this key moment the description is activated and follows "Inter-Objects entrance template":
{The two objects are respectively {"far" | "close"}, {no interaction | a {"distant" | "physical"} {"aggressive" | "peaceful"} interaction} occurs between them}.

b- At each of the key moments a proposed "Inter-Objects characteristics template" is structured as follow:

{The two objects are respectively {"approaching" | "receding" | "merged"}, {no interaction | a {"distant" | "physical"} {"aggressive" | "peaceful"} interaction} occurs between them}.

When an object first entrance occurs, we use indicative description (big surface, slow speed, regular shape, etc) of its state in the entrance template (see section II.3.6). Later, at a moment k_m during the scene, we use comparative description (bigger surface, increasing the speed, etc) of the current moment k_m state with the last state moment k_{m-1}.

To establish the mapping between the scene activity characteristics matrix M_{sac} and the new proposed structured templates, simple logical threshold-based rules, founded on the ontology shown in chapter II, were used. Examples of these rules are shown as follow:

- For the position to an area of interest: we introduced three rules for the "Object characteristics template":

a) For an object l at a key moment $\mathrm{k_m} \in \{\mathrm{k_1}, \mathrm{k_2}, \ldots \mathrm{k_n}\}$, in a given scene,

$$\text{if } \exists \text{ Å} \in \text{"area of interest"} \;/\; C(obj_l(k_m)) \in \text{Å}$$
$$\Longrightarrow add\ to\ description\ \{\text{moves, in "Å" spot}\}.$$

$C(obj_l)$ represents the centroid position of object $\mathrm{l} \in \{1,2\}$.

This rule indicates that if the object's centroid belongs to an area of interest, we map the area symbol into the template.

b) For an object l at a key moment k_m , in a given scene,

$$\text{if } \exists \text{ Å} \in \text{"area of interest"} \;/\; C(obj_l(k_m)) \notin \text{Å} \text{ and } C(obj_l(k_{m-1})) \in \text{Å}$$
$$\Longrightarrow add\ to\ description\ \{leaves\ \text{"Å"}\ spot, and\ moves\ \}.$$

This rule indicates that if the current object's centroid does not belong to any of the areas of interest, but its position, at the past key moment, was belonging to any area, we map the symbol of the area into the template as, *{leaves "area of interest" spot, and moves}*.

c) For an object l at a key moment k_m , in a given scene,

$$\text{if } \forall \text{ Å} \in \text{"area of interest"} \;/\; C(obj_l(k_m)) \notin \text{Å} \text{ and } C(obj_l(k_{m-1})) \notin \text{Å}$$
$$\Longrightarrow add\ to\ description\ \{moves\}.$$

This rule indicates that if the current and last object centroids not belonging to any of the areas of interest we map *{moves}* into the template.

- For the position and direction compared to the other object: two rules were applied for each of the objects:

  As example, for an object 1, at a key moment k_m, in a given scene. Let's α be the angle between the vector direction of the object 1 and the vector direction formed by the centroid of object 1 and the centroid of object2:

$$\alpha = \left(\overrightarrow{\left(C\left(obj_1(k_{m-1})\right), C\left(obj_1(k_m)\right)\right)}, \overrightarrow{\left(C\left(obj_1(k_m)\right), C\left(obj_2(k_m)\right)\right)}\right)$$

  a) if $|\alpha| \leq 45^{\circ} \Longrightarrow$ *add to description* {"*toward*" *the object* "2"}.

  This rule indicates that if the absolute value of α is smaller than 45 degrees, than "object1" is heading toward the "object2".

  b) if $|\alpha| \geq 135^{\circ} \Longrightarrow$ *add to description* {"*away from*" *the object* "2"}

  This rule indicates that if the absolute value of α is bigger than 135 degrees, than "object1" is considered going away from the "object2".

Finally, after mapping the scene activity characteristics matrix M_{sac} into the structured templates, the system can generate mainly two kinds of scene textual description, a full description and a short one. The full description describes, for each key moments, all the features values (see Table IV-4), whereas the short one describes only features associated to the most important variations (showing the irregularity) at each moment (see Table IV-5). From the short description, two objects life cycle descriptions can be exported.

An example of describing the scene "LeftBox", taken from the database "Caviar" ("CAVIAR," 2004), containing pictures, key moments, matrixes, full and short description can be seen on the next pages. For another example, the reader can refer to the experiments sub-section IV.4.6.

The scene LeftBox shows a person enters the scene, changes its direction and speed many times, and interacts with another person at distance, see below figures.

Figure IV-24: Objects in the scene "LeftBox" at the frame 101: showing the trajectory of object 1, when the object 2 first enters the scene.

Figure IV-25: Objects distant interaction in the scene "LeftBox" at the frame 145: showing the first distant peaceful interaction between objects 1 and 2.

Figure IV-26: Objects after interaction in the scene "LeftBox", at the frame 190.

Figure IV-27: Object 2 exiting the scene "LeftBox", at the frame 238.

Figure IV-28: Object 1 in the scene "LeftBox" is miss detected.

Figure IV-29: Object 1 exiting the scene "LeftBox".

Next, in *Table IV-3, we present the scene activity characteristics matrix Msac showing objects and interaction states for the scene "LeftBox", all the red marked values are key moments corresponding characteristics where irregularity exists. Object 1 shows key moments with its direction when it is near the frame 101, 110, 185, and 230, and with its speed near the frame 160. Object 2 shows keys activities, with its direction when it is near the frame 185, and with its speed near the frame 145, 160 and 230, and with its shape near the frame 230. The interaction between the 2 objects begins near the frame 145 and ends near the frame 190.*

In Table IV-4, the full description of the scene "LeftBox", results of mapping the M_{sac} into the proposed templates, is shown. Where, at each key moment, it reflects the corresponding irregularity shown above, and describes the full state of the objects and interaction. To be noticed as example, near the frame 145 the two objects start distant peaceful interaction, which ends near the frame 190.

In Table IV-5, we show the short description of the scene "LeftBox", which describes at each moment only the corresponding irregularity. To be noticed as example, near the frame 145 the two objects start distant peaceful interaction, which ends near the frame 190. Also, the big variations on object 1 directions are when he is near the frame 101, 110, 185, and 230.

Table IV-3: The scene activity characteristics matrix M_{sac} showing objects and interaction states for the scene "LeftBox".

Death Frame: 450	Characteristics type	Object 1												Object 2												Inter-objects		Interaction		
Birth Frame: 1	Key moments (Frames)	Deformability (0/1)	Position to an area of interest	Position in the frame	Direction (25)	Interpretation of the Direction	Position and direction / second Object	Interpretation Position / second Object (Toward, Away)	Shape (Hu)	Interpretation Shape (Hu)	Shape (Surface)	Interpretation Shape (Surface) (C: change -+)	Speed (10)	Speed (Sp:speeding, Sl:slowing down, st:stable)	Deformability (0/1)	Position to an area of interest	Position in the frame	Direction (25)	Interpretation of the Direction	Position and direction / second Object	Interpretation Position / second Object (Toward, Away)	Shape (Hu)	Interpretation Shape (Hu) (I: Irregularity)	Shape (Surface)	Interpretation Shape (Surface) >2 (C: change -+)	Speed (10)	Speed (Sp:speeding, Sl:slowing down, St: stable)	Distance between objects	Interpretation Distance between objects (Approach, Recede, Merge)	Existence (E/-)	Type (D: distant, P: Physical)	Aggressiveness (A, P)
177	11	1	A	ODM	104	UM			0.003		0.4522		1.265																			
	101	1	B	IUL	0	RM	-	-	0.006	-	0.8541	-	0.563	sl	1	B	OUL	-73.3	DM	-	-	0.0039		0.1862		0.26		5.21		-		
ACTIVITY NUMBER	110	1	B	IUL	-48.27	DR	159	A	0.006	-	0.784	-	0.563	st	1	B	OUL	-73.3	DM	-66	-	0.0031	-	0.2729	-	0.26	st	5.35	R	-		
	145	1	A	IDM	-56.3	DR	162	A	0.003	-	0.9724	-	1.628	sp	1	B	OUL	-79.69	DM	-49	-	0.0027	-	0.4396	-	0.097	sl	8.59	R	E	D	P
	160	1	B	IDR	-56.3	DR	159	A	0.004	-	1.3232	-	0.312	sl	1	C	OUL	-79.69	DM	-22	T	0.0023	-	0.4513	-	0.274	sp	11.22	R	E	D	P
	185	1	B	IDR	90	UM	-54	-	0.004	-	1.1868	-	0.877	sp	1	C	OUL	108.435	UM	-73	-	0.0022	-	0.4673	-	0.123	sl	10.553	A	E	D	P
48	190	1	B	IDR	90	UM	-17	T	0.004	-	1.04	-	1.011	sp	1	C	OUL	108.435	UM	-145	A	0.0025	-	0.4738	-	0.123	st	10.056	A	-		
	230	1	B	ILM	138.532	UL	-7	T	0.004	-	0.8862	-	0.644	sl	1	B	OUL	95.1944	UM	132	-	0.0065	I	0.19	-	0.411	sp	6.818	A	-		
VIDEO ID	238	1	B	ILM	138.532	UL	-9	T	0.003	-	1	-	0.551	sl	1	B	OUL	95.1944	UM	129	-	0.0074	-	0.1776	-	0.347	sl	6.4	A	-		
	300 ... 350								0.012		0.0003		1.698																			
	412	1	A	ODR	-69.07	DM			0.005	-	0.1893	-	0.578	sp																		

Table IV-4: The full description of the scene "LeftBox", results of mapping the M_{sac} into the proposed templates.

Info	Frame #	Desc target	Textual description
450	11	Object 1	"*Deformable* " object "1" enters the scene, *from " A " spot,* on the "*Down Middle* " of the "*Outside* " area of the camera field of view, heading "*Up Middle* ", having respectively "*Regular* " shape, "*Small* " surface, and "*High* " speed.
Death Frame	101	Object 1	"*Deformable* " object "1" leaves, "A" spot, and moves, in "B" spot, on the "*Up Left* " of the "*Inside* " area of the camera field of view, heading immediately "*Right Middle* ", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "decreasing" of its speed.
		Object 2	"*Deformable* " object "2" enters the scene, *from " B " spot,* on the "Up Left" of the "*Outside* " area of the camera field of view, heading "*Down Middle* ", having respectively "*regular* " shape, "*small* " surface, and "*low* " speed.
		Object 1 & Object 2	The two objects are respectively "Far", No Interaction occurs between them.
1	110	Object 1	"*Deformable* " object "1" moves, in "B" spot, on the "*Up Left* " of the "*Inside* " area of the camera field of view, heading immediately "Down *Right* ", "Away from" the object "2", "No big changes occurring respectively on" its shape , and "No big changes occurring respectively on" its surface, and having respectively "stable" speed.
		Object 2	"*Deformable* " object "2" moves, in "B" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Down Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its Surface, and having respectively "stable" Speed.
		Object 1 & Object 2	The two objects are respectively "Receding", no Interaction occurs between them.
Birth Frame	145	Object 1	"*Deformable* " object "1" leaves, "B" spot, and moves, in "A" spot, on the "*Down Middle* " of the "*Inside* " area of the camera field of view, heading "Down *Right* ", "Away from" the object "2", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "increasing" of its speed.
		Object 2	"*Deformable* " object "2" moves, in "B" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Down Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively considerable "decreasing" of its speed.
		Object 1 & Object 2	The two objects are respectively "Receding", A "Distant" "Peaceful" Interaction occurs between them.
	160	Object 1	"*Deformable* " object "1" leaves, "A" spot, and moves, in "B" spot, on the "*Down Right* " of the "*Inside* " area of the camera field of view, heading "Down *Right* ", "Away from" the object "2", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively considerable "increasing" of its Speed.
		Object 2	"*Deformable* " object "2" leaves, "B" spot, and moves, in "C" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Down Middle", "Toward" the object "1", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively considerable "increasing" of its Speed.
		Object 1 & Object 2	The two objects are respectively "Receding", A "Distant" "Peaceful" Interaction occurs between them.
177	185	Object 1	"*Deformable* " object "1" moves, in "B" spot, on the "*Down Right* " of the "*Inside* " area of the camera field of view, heading immediately "Up Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "increasing" of its speed.
		Object 2	"*Deformable* " object "2" moves, in "C" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading immediately "Up Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "decreasing" of its speed.
		Object 1 & Object 2	The two objects are respectively "Approaching", A "Distant" "Peaceful" Interaction occurs between them.
ACTIVITY NUMBER	190	Object 1	"*Deformable* " object "1" moves, in "B" spot, on the "*Down Right* " of the "*Inside* " area of the camera field of view, heading "Up Middle", "Toward" the object "2", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "increasing" of its speed.
		Object 2	"*Deformable* " object "2" moves, in "C" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Up Middle", "Away from" the object "1", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively "stable" speed.
		Object 1 & Object 2	The two objects are respectively "Approaching", No Interaction occurs between them.
48	230	Object 1	"*Deformable* " object "1" moves, in "B" spot, on the "*Left Middle* " of the "*Inside* " area of the camera field of view, heading immediately "Up Left", "Toward" the object "2", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "decreasing" of its speed.
		Object 2	"*Deformable* " object "2" leaves, "C" spot, and moves, in "B" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Up Middle", "Occurring respectively irregularity in" its shape, and "No big changes occuring respectively on" its surface, and having respectively considerable "increasing" of its speed.
		Object 1 & Object 2	The two objects are respectively "Approaching", No Interaction occurs between them.
VIDEO ID	238	Object 1	"*Deformable* " object "1" moves, in "B" spot, on the "*Left Middle* " of the "*Inside* " area of the camera field of view, heading "Up Left", "Toward" the object "2", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having slight "decreasing" of its speed.
		Object 2	"*Deformable* " object "2" exits the scene, from "B" spot, on the "*Up Left* " of the "*Outside* " area of the camera field of view, heading "Up Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "decreasing" of its speed.
		Object 1 & Object 2	The two objects are respectively "Approaching", No Interaction occurs between them.
	412	Object 1	"*Deformable* " object "1" leaves "B" spot and exits the scene, from "A" spot, on the "*Down Right* " of the "*outside* " area of the camera field of view, heading immediately "Down Middle", "No big changes occurring respectively on" its shape, and "No big changes occurring respectively on" its surface, and having respectively slight "increasing" of its speed.

Table IV-5: The short description of the scene "LeftBox", describe at each moment only the corresponding irregularity.

Info	Frame #	Desc target	Textual description
450	11	Object 1	"*Deformable*" object "1" enters the scene, *from "A" spot,* on the "*Down Middle*" of the "*Outside*" area of the camera field of view, heading "*Up Middle*", having respectively "*Regular*" shape, "*Small*" surface, and "*High*" speed.
Death Frame	101	Object 1	"*Deformable*" object "1" leaves, "A" spot, and moves, in "B" spot, on the "Up Left" of the "Inside" area of the camera field of view, heading immediately "*Right Middle*".
		Object 2	"*Deformable*" object "2" enters the scene, *from "B" spot,* on the "Up Left" of the "*Outside*" area of the camera field of view, heading "*Down Middle*", having respectively "*regular*" shape, "*small*" surface, and "*low*" speed.
		Object 1 & Object 2	The two objects are respectively "Far".
	110	Object 1	"*Deformable*" object "1" heading immediately "Down *Right*", "Away from" the object "2".
1	145	Object 1	"*Deformable*" object "1" leaves, "B" spot, and moves, in "A" spot, on the "Down Middle" of the "Inside" area of the camera field of view, "Away from" the object "2".
Birth Frame		Object 2	"*Deformable*" object "2" having respectively considerable "decreasing" of its speed.
		Object 1 & Object 2	"Start" a "Distant" "Peaceful" Interaction.
177	160	Object 1	"*Deformable*" object "1" leaves, "A" spot, and moves, in "B" spot, on the "Down Right" of the "Inside" area of the camera field of view, "Away from" the object "2", and having respectively considerable "decreasing" of its speed.
		Object 2	"*Deformable*" object "2" leaves, "B" spot, and moves, in "C" spot, "Toward" the object "1", and having respectively considerable "increasing" of its Speed.
		Object 1 & Object 2	The two objects are respectively "Receding".
ACTIVITY NUMBER	185	Object 1	"*Deformable*" object "1" heading immediately "Up Middle".
		Object 2	"*Deformable*" object "2" heading immediately "Up Middle".
		Object 1 & Object 2	The two objects are respectively "Approaching".
	190	Object 1	"*Deformable*" object "1" moves, "Toward" the object "2".
		Object 2	"*Deformable*" object "2" moves, "Away from" the object "1".
		Object 1 & Object 2	No more Interaction occurs between them.
48	230	Object 1	"*Deformable*" object "1" moves, on the "Left Middle" of the "Inside" area of the camera field of view, heading immediately "Up Left", "Toward" the object "2".
		Object 2	"*Deformable*" object "2" leaves, "C" spot, and moves, in "B" spot, "Occurring respectively irregularity in" its shape, and having respectively considerable "increasing" of its speed.
VIDEO ID	238	Object 1	"*Deformable*" object "1" moves, "Toward" the object "2".
		Object 2	"*Deformable*" object "2" exits the scene, from "B" spot.
	412	Object 1	"*Deformable*" object "1" leaves "B" spot and exits the scene, from "A" spot, on the "*Down Right*" of the "*outside*" area of the camera field of view, heading immediately "Down Middle".

Extracted from the short description described above, object 1 and object 2 life cycles can be described as below:

- Object1 life cycle description:

- Near the frame 11, deformable object 1 enters the scene, from A spot, on the down middle of the outside area of the camera field of view, heading up middle, having respectively regular shape, small surface, and high speed.
- Near the frame 101, deformable object 1 leaves, A spot, and moves, in B spot, on the up left of the inside area of the camera field of view, heading immediately right middle.
- Near the frame 110, deformable object 1 heading immediately down right, away from the object 2.
- Near the frame 145, deformable object 1 leaves, B spot, and moves, in A spot, on the down middle of the inside area of the camera field of view, away from the object 2, and starts a distant peaceful interaction with object 2.
- Near the frame 160, deformable object 1 leaves, A spot, and moves, in B spot, on the down right of the inside area of the camera field of view, away from the object 2, and having respectively considerable decreasing of its speed.
- Near the frame 185, deformable object 1 heading immediately up middle.
- Near the frame 190, deformable object 1 moves, toward the object 2, and no more interaction occurs with object 2.
- Near the frame 230, deformable object 1 moves, on the left middle of the inside area of the camera field of view, heading immediately up left, toward the object 2.
- Near the frame 238, deformable object 1 moves toward the object 2.
- Near the frame 412, deformable object 1 leaves B spot and exits the scene, from A spot, on the down right of the outside area of the camera field of view, heading immediately down middle.

- Object2 life cycle description:

- Near the frame 101, deformable object 2 enters the scene, from B spot, on the up left of the outside area of the camera field of view, heading down middle, having respectively regular shape, small surface, and low speed.
- Near the frame 145, deformable object 2 having respectively considerable decreasing of its speed, start a distant peaceful interaction with object 1.
- Near the frame 160, deformable object 2 leaves, B spot, and moves, in C spot, toward the object 1, and having respectively considerable increasing of its speed.
- Near the frame 185, deformable object 2 heading immediately up middle.
- Near the frame 190, deformable object 2 moves, away from the object 1, no more interaction occurs with object 1.
- Near the frame 230, deformable object 2 leaves, C spot, and moves, in B spot, occurring respectively irregularity in its shape, and having respectively considerable increasing of its speed.
- Near the frame 238, deformable object 2 exits the scene, from B spot.

IV.4. Experiments and results

As the proposed method consists in many stages, we made experiments to evaluate each step of the following process:

- Selection of the datasets.
- Object tracking and segmentation.
- Features extraction on each window from scenes having two objects, and data pre-processing.
- Classifications using three deep neural network algorithms:
  1) The first algorithm was learned and tested to identify, the existing of the interactions in those scenes.
  2) The second algorithm was learned and tested to classify, the interactions when exists, if they are distant or physical interaction in those scenes.
  3) The third algorithm was learned and tested to classify, the interactions when exists, if they are aggressive or peaceful interaction in those scenes.
- As an example scene, the scene activity characteristics matrix M_{sac} was calculated, and then mapped into templates to generate textual descriptions.

In the following we address each of these steps.

IV.4.1. Datasets selection

Deep NNs are requiring a huge amount of training data. However, what we seek for in the scene is rare. I fact, we want annotated video datasets with descriptions about far, physical, aggressive or peaceful interactions between only two objects. For that, a huge effort was made to gather publicly available datasets concerning videos objects actions and interactions (see Table IV-6), including some diversity about object types and area of application. Then, from all these datasets, 323 scenes of 1903 seconds, the ones having only two objects were extracted and manually annotated, as for, existence of interaction or not, physical or far interaction, peaceful or aggressive interaction. Some extracted scenes were not taken into consideration as the tracking and segmentation algorithm missed to detect the two objects. Consequently, no acceptable features could be extracted for further interpretation. After that, the sixth and seventh stages, mentioned in section IV.3., were applied including features extractions from sliding windows of frames, see sub-section IV.4.3, then, the data was prepared and pre-processed, including the dataset increasing by reversing the footage and other methods, see sub-section IV.4.4. Finally, 285 scenes were studied.

Next, according the extracted scenes and the annotations, the three classifiers were trained and tested. Later, a temporal-consistency analysis was applied, to filter the 3 sequential classification vectors, which were used for description.

Nb: for more information about the videos existing in these datasets, the reader can refer to (Chaquet, Carmona, & Fernández-Caballero, 2013), (“CV Datasets,” n.d.), (“YACVID,” 2018), (Borges, Conci, & Cavallaro, 2013).

Table IV-6: Tables of datasets used							
#	**Name**	**Reference**	**Description**	**Main area of application**	**Number of scenes extracted**	**Total duration of scenes extracted (per sec)**	**Total duration of scenes taken (per sec)**
1	BEHAVE Interactions Test Case	(Blunsden & Fisher, 2010)	This dataset comprises various scenarios of people interacting. It covers 10 types of interactions: in group, approach, walk together, split, ignore, following, chase, fight, run together, and meet.	Event detection, modelling crowd, multi-objects activities recognition, visual tracking	51	589	327
2	CAVIAR: Context Aware Vision using Image-based Active Recognition	("CAVIAR," 2004)	This dataset contains many scenes type: Walking, fainting, Leaving bags behind, groups meeting, walking together and splitting up, two people fighting.	Interaction analysis, activity recognition, trajectory clustering of, motion segmentation, tracking	18	128	101
3	"EPFL" data set: Multi-camera Pedestrian Videos	("EPFL," n.d.)	This dataset contains four sequences: 1-Laboratory sequences (4 videos inside), Campus sequences (2 videos outside), Terrace sequences (outside), Passageway sequence (underground train station).	people detection and tracking	29	248	125
4	UT-Interaction dataset	(Ryoo et al., 2010)	This dataset contains 20 video sequences (of 1 minute each), showing 6 classes of human interactions: shake-hands, point, hug, push, kick and punch.	Interaction analysis, Complex activities recognition, Action recognition.	13	424	138
5	Activity modelling and abnormality detection dataset	(Varadarajan & Odobez, 2009)	This dataset consists of a 45 minutes long video of a junction controlled by traffic lights.	multi-object tracking	6	26	12
6	Advanced Video and Signal based Surveillance	(*i-LIDS*, 2007)	This dataset consists of 3 sequences (abandoned baggage) and 4 sequences (parked vehicle) of approximately 20 minutes duration.	abandoned baggage detection, parked vehicle detection	11	125	24
7	Audio-visual people dataset	(*Audiovisual people dataset*, n.d.)	This dataset consists of three sequences with a video camera and two microphones.	people detection tracking	5	106	72
8	VISOR Video surveillance online repository	("VISOR," 2017)	This dataset sequence presents 15 people walking on square, interacting and involving in different groups.	Repository, interaction analysis,	4	38	28
9	VIRAT Video Dataset	(Oh et al., 2011)	This dataset consists of many real outdoor scenes including 23 event types distributed among numbers of instances throughout 29 hours of video.	Activity analysis, tracking, human-vehicle interaction recognition, action recognition	13	79	39
10	Collective Activity Dataset	(Choi & Savarese, 2012)	This dataset consists of many sets of images concerning mainly of people 'Crossing', 'Waiting', 'Queuing', 'Walking', 'Running', and 'Shaking hands'.	Multi-Target Tracking, people activity	10	140	23

Table IV-7: Table listing 10 of the tested algorithms for objects segmentation and tracking										
#	**Algorithm**	**Reference**	**Tracking**	**Multi-object**	**Segmentation**	**Object type**	**Free**	**Notes**	**Implementation**	**Subjective quality**
1	Tracking Interacting Objects	(X. Wang, Türetken, Fleuret, & Fua, 2016)	Yes	yes	no	ANY	upon request ("Tracking Interacting Objects – CVLAB," n.d.)	Based on network-flow Mixed Integer Program	Moderate	++
2	Globally-Optimal Greedy Algorithms	(Pirsiavash, Ramanan, & Fowlkes, 2011)	yes	yes	no	Pedestrian	Yes, (Pirsiavash, n.d.)	Based on flow network	Very Hard	++
3	Continuous Energy Minimization for Multi-Target Tracking	(Milan, Roth, & Schindler, 2014)	yes	yes	no	Pedestrian	Yes, (Milan, 2014)	Based on continuous energy minimization	Moderate	+
4	Discrete-Continuous Energy Minimization for Multi-Target Tracking	(Andriyenko, Schindler, & Roth, 2012)	yes	yes	no	Pedestrian	Yes, (Milan, 2012)	Based on Discrete-Continuous energy minimization	Hard	+
5	Two-Granularity Tracking	(Fragkiadaki, Zhang, Zhang, & Shi, 2012a)	yes	yes	yes	Pedestrian	Yes, (Fragkiadaki, Zhang, Zhang, & Shi, 2012b)	Based on Mediating Trajectory and Detection Graphs	Hard	+
6	GMCP-Tracker	(Roshan Zamir, Dehghan, & Shah, 2012)	yes	yes	no	Pedestrian	Yes, (Roshan Zamir et al., 2012)	Global Multi-object Tracking Using Generalized Minimum Clique Graphs Based, on shifting the approximation from the temporal domain to the object domain	Moderate	+
7	Urban Tracker	(Jean-Philippe Jodoin, Bilodeau, & Saunier, 2014)	yes	yes	no	ANY	Yes, (J.-P. Jodoin, Bilodeau, & Saunier, 2013)	Multiple Object Tracking in Urban Mixed Traffic, based on background subtraction to detect moving objects	Moderate	+
8	Moving-Target-tracking-with-opencV		yes	yes	yes	Any	Yes, (Son, 2015/2019)	Based on Background subtraction and Kalman Filter	Moderate	+++
9	Online Multi-Object Tracking by Decision Making	(Y. Xiang, Alahi, & Savarese, 2015a)	yes	yes	no	Any	Yes, (Y. Xiang, Alahi, & Savarese, 2015b)	Based on Markov decision processes	Moderate	+
10	Motion-Based Multiple Object Tracking		yes	yes	yes	Any	Yes, ("Motion-Based Multiple Object Tracking - MATLAB & Simulink," n.d.)	Based on moving objects detection by background subtraction algorithm; then, on Kalman filter for tracking and predicting the moving objects from frame to frame.	Easy	+++

IV.4.2. Segmentation and tracking algorithms comparison

Many efforts have been done in this field by the research community. We searched for available multi-target/object tracking and segmentation algorithms, seeking for a simple one, easy to install, that can provide us with acceptable results for further interpretation. After an exhaustive search for available algorithm in the domain of segmentation and tracking, more than 20 algorithms were found. We list in the Table IV-7 above 10 of the tested algorithms and their ability to fulfil the properties we are looking for.

Some published algorithms were not available as functional code; others deliver the tracking data without the segmentation results (Pirsiavash, n.d.) or the segmentation results without the tracking ones; others did not handle the propriety of multi-object tracking. On the other hand, we found codes compliant with our conditions, but only adapted for humans (Jiang, Rodner, & Denzler, 2012), (Benfold & Reid, 2011), (Choi & Savarese, 2012), (Possegger, Mauthner, Roth, & Bischof, 2014), (Milan, 2012), (Milan, 2014), (Fragkiadaki et al., 2012b), (Roshan Zamir et al., 2012).

After all, we choose the algorithm for segmentation and multi-object tracking provided by Matlab ("Motion-Based Multiple Object Tracking - MATLAB & Simulink," n.d.), as it is the most suitable in our case. It is multi-object tracker with object segmentation, which is easy to implement, and have acceptable results.

IV.4.3. Data preparation and pre-processing

We searched scenes having only two moving objects taken from fixed camera in all the above-mentioned datasets. Some of those scenes contain interaction between the two objects and others do not. As not many works were done in the case of distant interaction, and its aggressiveness, we did not find in these datasets a large diversity of scenes. Finally, after creating sliding window, for our experiments we have chosen the size to have 25 frames, we extracted 2498 features from each window. Five types of features were mined, spatial, temporal, inter-objects, inter-frames and trajectory features. We found in all those datasets around 6029 windows, from which only 2208 windows of interaction. Between the 2208 interactions windows, we were able to differentiate 5962 distant interactions, and only 67 physical interactions.

Despite that the used segmentation and tracking algorithm ("Motion-Based Multiple Object Tracking") deals with the common occlusion problem by predicting and correcting its location using "Kalman filter", the results still suffer from this problem when two objects physically interact, because interaction may occur for a long time. After a given number of frames (8 frames in this case), one of the objects is lost. Both objects are labelled as being one. After that the object finishes interacting, the algorithm detects and labels one of the objects as being a different object than before the interaction. Hence, in the case of physical interaction or when there is close distance between objects, the windows automatically selected and showing two different objects are in a limited number. Consequently, we implement the whole procedure for all the dataset after reversing the video to increase the number of available samples for learning (especially for the case the physical interaction), and to catch the last moments of interaction between the objects.

Increasing artificially datasets is a typical approach, especially when the number of positive examples is too small. Here the reversed footage they were judged and annotated

as it were played in the regular temporal order. The reversed video does not change the classification statement of interaction, if it exists, neither for distant and physical interaction, nor for aggressive and peaceful interaction.

Also, the features were trimmed to optimize the results of classifications. As in some cases, some values cannot be computed; we removed some of the features with missing values. Mainly those features are first and second derivative of a characteristic like Hu moments, or distance between objects, etc. As results, we kept only 2305 features.

Finally, we were able to have around 11200 windows taken from 285 scenes with duration of scenes taken and tested are shown in Table IV-6. From these 11200 windows only 3955 are representing interaction. Between the 3955 interactions windows, we were able to differentiate 3692 distant interactions, and 263 physical interactions, 438 aggressive and 3517 peaceful interaction windows.

Despite of the size of collected multiple datasets; the global amount of data remains insufficient for sharp evaluation benchmark. After preparing the input dataset for the three classification algorithms, we suffered from the imbalance between the negative and positive inputs for the three classifications. For better classification results, we choose to duplicate the number of positive results.

At the end, for each of the three classification algorithms the balanced dataset input ends up to be like the following in the Table IV-8:

Table IV-8: Balanced dataset input characteristics.

	Input records	**Features per record**	**Input records classifications**	
Interaction and non-interaction classification algorithm	14902	2305	Interaction	Non-interaction
			7657	7245
Distant and physical interaction classification algorithm	7079	2303	Physical	Distant
			3640	3439
Aggressive and peaceful interaction classification algorithm	6703	2303	Aggressive	Peaceful
			3439	3264

IV.4.4. Classification training and results

According to the three corresponding balanced datasets, mentioned above, several machine learning algorithms were tested to determine, the existence of interactions, and if they are distant or physical, and if they are aggressive or peaceful.

For each of the three classifications, in the MATLAB and Simulink environment, we used “Classification Learner app” (“Classification Learner App - MATLAB & Simulink,” n.d.) from the “Statistics and Machine Learning Toolbox” to train and test different models of several classical machine learning algorithms, and “Deep Learning Toolbox” (“Deep Learning Toolbox Matlab,” n.d.) to implement, train and test a multi-layered Deep Neural Network DNN, as follow:

- For the classical machine learning algorithms experimentations: using the Classification Learner app, we performed automated training to search for the best classification

model type, including decision trees (simple, medium, and complex), discriminant analysis (linear and quadratic), logistic regression, support vector machines (linear, quadratic, cubic, fine Gaussian, medium Gaussian, and coarse Gaussian), nearest neighbors (fine KNN, medium KNN, coarse KNN, cosine KNN, cubic KNN, and weighted KNN), and ensemble classifiers (boosted trees, bagged trees, subspace discriminant, subspace KNN, and RUSBoost trees). Then, for the classification models showing best results, many trainings were performed after fine tuning the corresponding parameters to maximize the results. Finally, for each of the three classifications, one classification model, showing ultimate results, were selected.

- For the Deep Learning Neural Networks experimentations: a multi-layered Deep Neural Network DNN was implemented for each classification. Simple Feedforward fully connected networks called Pattern recognition networks ("Pattern recognition network - MATLAB," n.d.) in MATLAB and Simulink environment were trained by back propagation of error. A standard network for pattern recognition is a two-layer feedforward network, with a sigmoid transfer function in the hidden layer, and a softmax transfer function in the output layer. In order to achieve the desired outputs, several tests were made after altering this model to handle deeper architecture (by adding more layers) and by determining the best architecture (number of layers, number of neurons) and the best parameters which maximise the results such as:
  1) The activation function (Log-sigmoid 'logsig', positive linear function 'poslin' which is similar to 'ReLu', and hyperbolic tangent sigmoid function 'tansig')
  2) The training method (Scaled conjugate gradient backpropagation 'trainscg', Levenberg-Marquardt backpropagation 'trainlm', Gradient descent with momentum and adaptive learning rate backpropagation 'traingdx', Gradient descent with momentum backpropagation 'traingdm', Gradient descent with adaptive learning rate backpropagation 'traingda', Conjugate gradient backpropagation with Polak-Ribiére updates 'traincgp', Conjugate gradient backpropagation with Fletcher-Reeves updates 'traincgf', Conjugate gradient backpropagation with Powell-Beale restarts 'traincgb', BFGS quasi-Newton backpropagation 'trainbfg', Resilient backpropagation 'trainrp', and One-step secant backpropagation 'trainoss').
  3) Lambda
  4) Sigma.

Finally, for each of the three classifications, one classification DNN model, showing ultimate results, were selected.

In the following, we present, for each of the three classifications after experiments and testing, the best results using a classical machine learning algorithm and the best results using a DNN:

1- Interaction vs non-interaction classification:
For the classical machine learning algorithm classification, we used as input dataset 887 non-overlapping windows records (224 interactions and 663 non-interactions). The algorithm that shows best results after many tests was "AdaBoost decision Tree" from the ensemble methods with 82% of accuracy, having maximum number of splits (tree depth) set to 30 and the number of learners set to 50, see Figure IV-30.
For the DNN, we used the corresponding balanced data mentioned in the Table IV-8 above (14902 input records having each 2305 features, from which 7657 classified as interaction and 7245 classified as non-interaction), and we chose completely different

scenes for each of the training, validation and test sets. The chosen algorithm, as mentioned in sub-section IV.3.7.A, is the seven-layer Pattern recognition network, a feedforward network composed of fully connected layers, six hidden layers and one output layer. Each hidden layer contains 586 neurons. As a method for network learning we used the "trainscg", which is an implementation of the scaled conjugate gradient backpropagation method, the six hidden layers activation function is the transfer function Log-sigmoid "logsig" and a softmax transfer function in the output layer. Other parameters are shown in Table IV-9. As result of training this algorithm and testing it with 14902 records and 2305 features, having a training set of approximate 80% and a validation and test sets approximate 10 % each one, the accuracy of the test set achieved 87.5% after 325 epochs, see the confusion matrix at Figure IV-31.

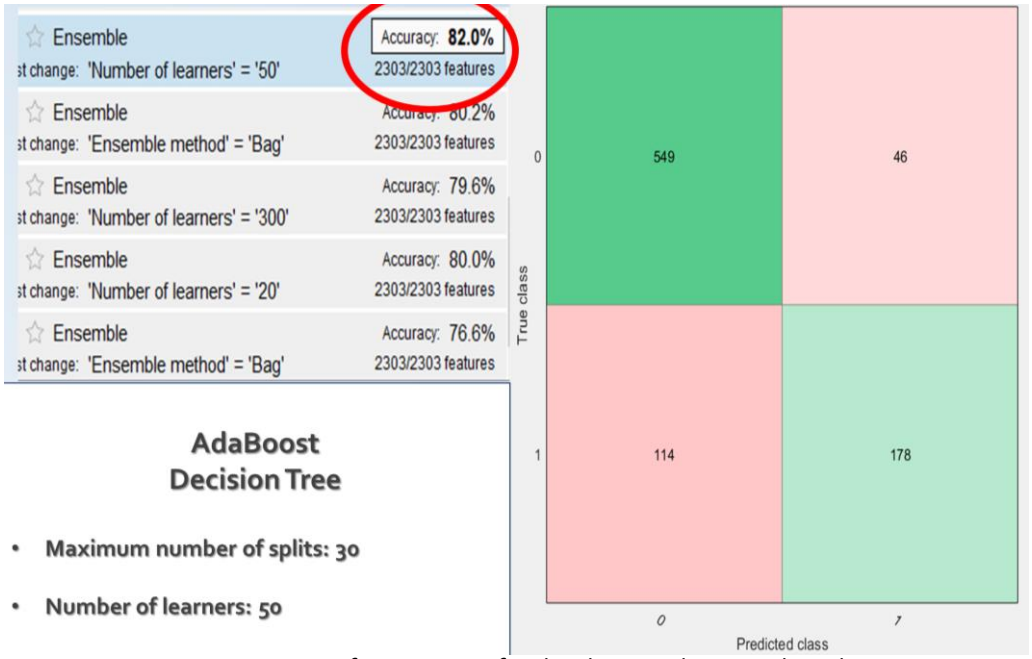


Figure IV-30: Confusion matrix for the chosen AdaBoost algorithm

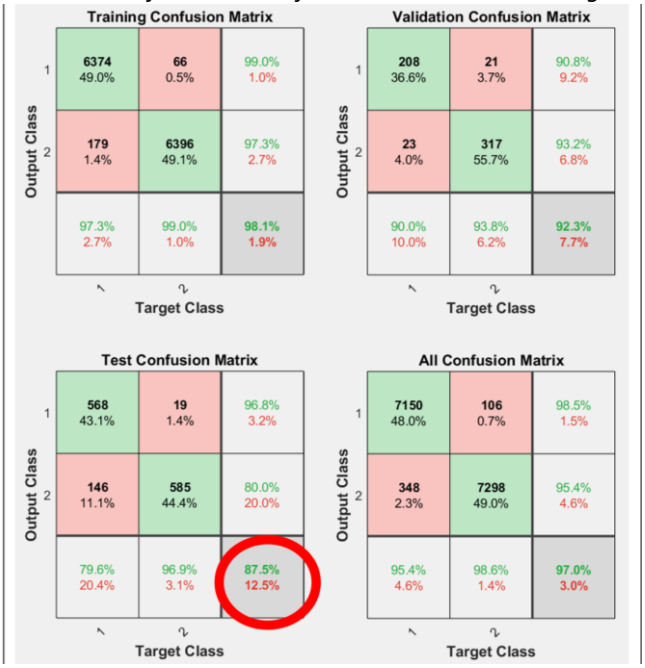


Figure IV-31: Confusion matrix for the chosen DNN algorithm

2- Distant and physical interaction classification:

For the classical machine learning algorithm classification, we used as input dataset 239 non-overlapping windows records (13 physical and 226 distant). The algorithm that shows best results after many tests was “Bag Decision Tree” from the ensemble methods with 89.5% of accuracy, having maximum number of splits (tree depth) set to 30 and the number of learners set to 30.

For the DNN, we used the corresponding balanced data mentioned above (7079 input records having each 2303 features, from which 3640 classified as physical interaction and 3439 classified as distant interaction), and we chose completely different scenes for each of the training, validation and test sets. The chosen algorithm is the four-layer Pattern recognition network, a feedforward network composed of fully connected layers, three hidden layers and one output layer. Each hidden layer contains 426 neurons. As a method for network learning we used the "trainscg", which is an implementation of the scaled conjugate gradient backpropagation method, the three hidden layers activation function is the transfer function Log-sigmoid "logsig" and a softmax transfer function in the output layer. Other parameters are shown in Table IV-9. As result of training this algorithm and testing it with 7079 records and 2303 features, having a training set of approximate 70% and a validation and test sets approximate 15 % each one, the accuracy of the test set achieved 93.7% after 102 epochs.

3- Aggressive and peaceful interaction classification:

For the classical machine learning algorithm classification, we used as input dataset 239 non-overlapping windows records (24 aggressive and 215 peaceful). The algorithm that shows best results after many tests was “Bag Decision Tree” from the ensemble methods with 87.9% of accuracy, having maximum number of splits (tree depth) set to 20 and the number of learners set to 30.

For the DNN, we used the corresponding balanced data mentioned above (6703 input records having each 2303 features, from which 3439 classified as aggressive and 3264 classified as peaceful), and we chose completely different scenes for each of the training, validation and test sets. The chosen algorithm is the four-layer Pattern recognition network, a feedforward network composed of fully connected layers, three hidden layers and one output layer. Each hidden layer contains 546 neurons. As a method for network learning we used the "trainscg", which is an implementation of the scaled conjugate gradient backpropagation method, the three hidden layers activation function is the function Log-sigmoid "logsig" transfer function and a softmax transfer function in the output layer. Other parameters are shown in Table IV-9. As result of training this algorithm and testing it with 6703 records and 2303 features, having a training set of approximate 70% and a validation and test sets approximate 15 % each one, the accuracy of the test set achieved 93.8% after 88 epochs.

Table IV-9: Used parameters for the three classification DNN algorithms

	chosen algorithm	Number of layers	Number of neurons in each hidden layer	Training function	Sigma σ	Lambda λ	Regularization	Activation function
Interaction or no interaction	Pattern recognition network: feedforward network composed of fully connected layers	7 (6 hidden, 1 output)	586	Trainscg	**5.0** **e**^(−**7**)	**5.0** **e**^(−5)	0.5	Logsig
Distant or physical interaction		4 (3 hidden, 1 output)	426	Trainscg	**5.0** **e**^(−**7**)	**5.0** **e**^(−7)	0.5	Logsig
Aggressive or peaceful interaction		4 (3 hidden, 1 output)	546	Trainscg	**5.0** **e**^(−**7**)	**5.0** **e**^(−7)	0.5	Logsig

These experimental results show that the classic machine learning algorithms (AdaBoost, and Bag Trees) results are quite comparable to the DNNs results, with a small advantage to the latter ones.

IV.4.5. Scenes description results

In addition to the scene described in section IV.3.8, where it is mainly focused on different variation of object characteristics and peaceful interaction between the two objects; using the same methodologies, we present here the description another scene named “Fight_RunAway2” taken from the database “CAVIAR” (“CAVIAR,” 2004). This scene shows a fight between two persons, where the main focus is the detection of distant aggressive interaction, and to show the effect of the tracking and segmentation algorithm false detection.

To better understand the scene, important moments are shown in the following figures:

Figure IV-32: Objects distant interaction in the scene “Fight_RunAway2”: frame 279 showing the trajectories of each object, and the distant aggressive interaction between objects 1 (showing as 5) and 2 (showing as 6).

Figure IV-33: Objects physical interaction in the scene “Fight_RunAway2”: frame 321 showing the physical aggressive interaction between objects 1 and 2. The two object are here detected as being one (object number 2 (showing as 6).).

Figure IV-34: Objects after interaction in the scene “Fight_RunAway2: frame 468 showing no more interaction between the object 1 (as 6) and object 2 (as 8).

Figure IV-35: Objects exiting in the scene “Fight_RunAway2”: frame 488 showing the object 1 (as 6) exiting the scene.

The graph-based pattern discovery was implemented and the new scene activity characteristics matrix was filled to highlight appropriate key moments for description phase. As mentioned, we kept the verbosity of the description density to be controlled by the user; next we select thresholds in a way to show only the major irregularities. These characteristics are:

1- Object characteristics:

a. Shape related:

i. Invariant moments or "Hu moments": In Figure IV-36, a simple threshold to 2 lead to localize seven peaks. Five of them are related with object 1 entering or leaving a sunny area, and where the object 1 is very far in the field of view. At that location, object 1 is miss-detected. The other two points indicate the two moments when the two objects are approaching physically (the algorithm detects them as one big object) and when they separate after the physical interaction.

ii. The object surface: In Figure IV-37, putting threshold to 2 indicates three peaks. One of them corresponds to object 1 entering the sunny area. The other two peaks indicate the two moments when the two objects are approaching and separating after the physical interaction.

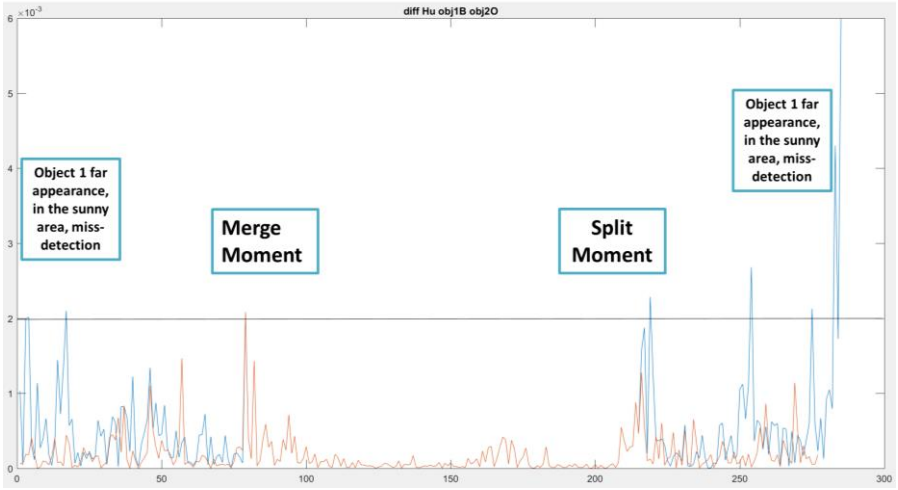


Figure IV-36: Variations of Hu moments in scene "Fight_RunAway2".

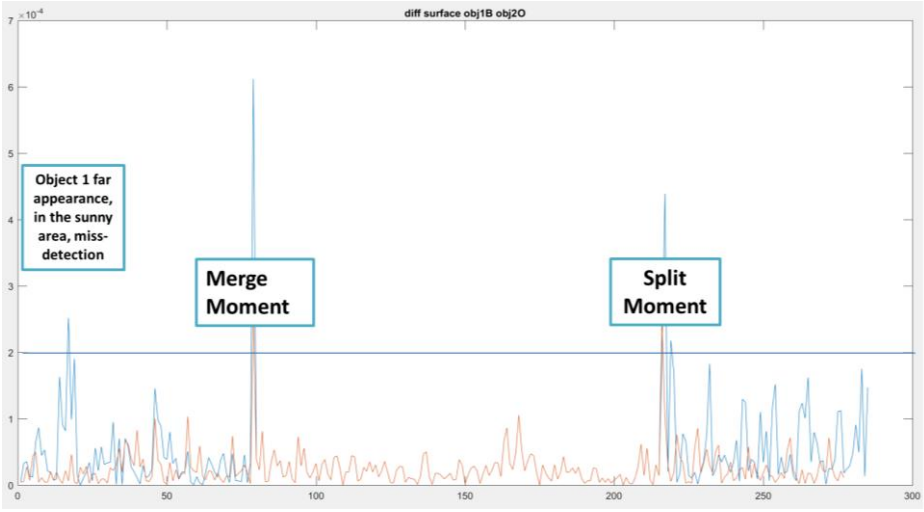


Figure IV-37: Variations of Objects surfaces in the scene "Fight_RunAway2".

b. Position related:

i. Position in the frame: Figure IV-38 shows objects trajectories in the scene "Fight_RunAway2", where the eight directions are taken according to the field of view, the dashed lines indicates the quantized sectors borders of each direction.

ii. Position to an area of interest: In Figure IV-39.

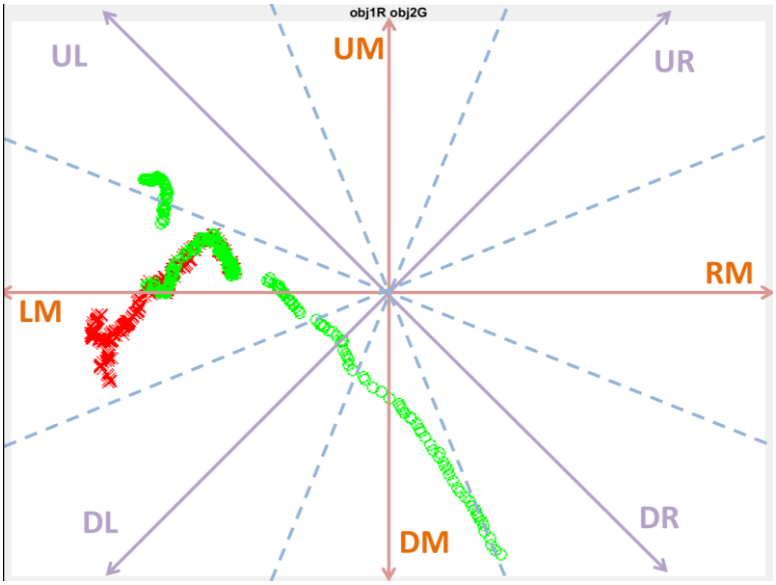


Figure IV-38: Objects trajectories in the scene "Fight_RunAway2": the red trajectory is the object1 centroid displacement and the green trajectory is for the object2.

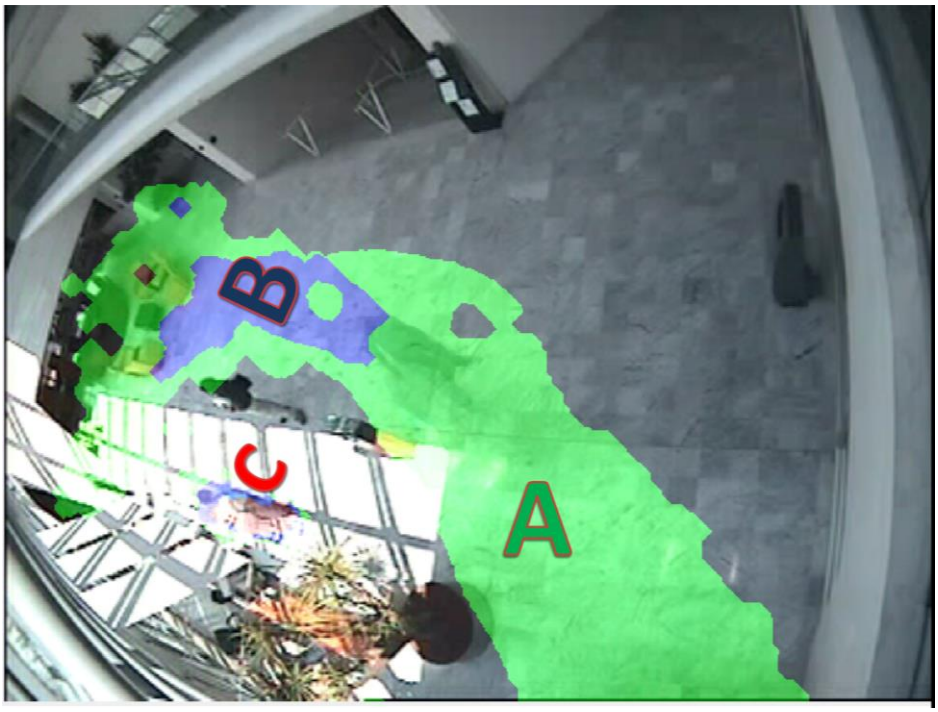


Figure IV-39: Figure showing, in the scene "Fight_RunAway2", the areas of interests (routes in green and the activity hot spots in blue and red).

c. Displacement related:

i. The object speed variations: Figure IV-40 shows objects accelerations (skipping 5 frames between 2 positions used for computations) in the scene "Fight_RunAway2", where for each of the two objects the accelerations variations show two moments (when the two objects approaches and separates after the physical interaction and the algorithm detect them as one big object). Also variations for object 2 show big changes in the speed around the frame 240.

ii. The object direction variations: Figure IV-41 shows objects angles variations (skipping 10 frames between 2 positions used for computations) in the scene "Fight_RunAway2", where, with a threshold fixed to 45 degrees this indicates six big changes. Two changes (1 and 3) indicate the two moments when the two objects approaches and separates after the physical interaction; Summits 2 indicate a big change in direction when objects are fighting. Two changes in blue (4 and 5) indicate the two moments (frames 243 and 462) when object 1 perform big change in its direction, and one change in orange (6) indicates when object 2 perform big change in its direction.

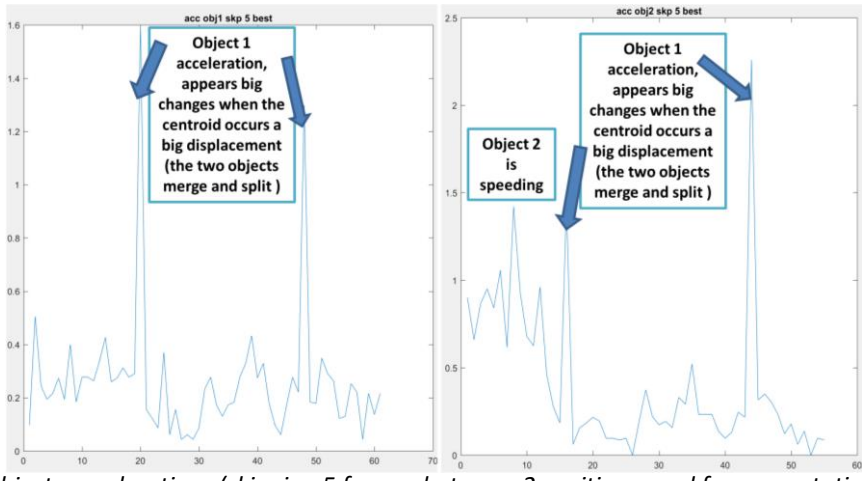


Figure IV-40: Objects accelerations (skipping 5 frames between 2 positions used for computations) in the scene "Fight_RunAway2" for each of the two objects.

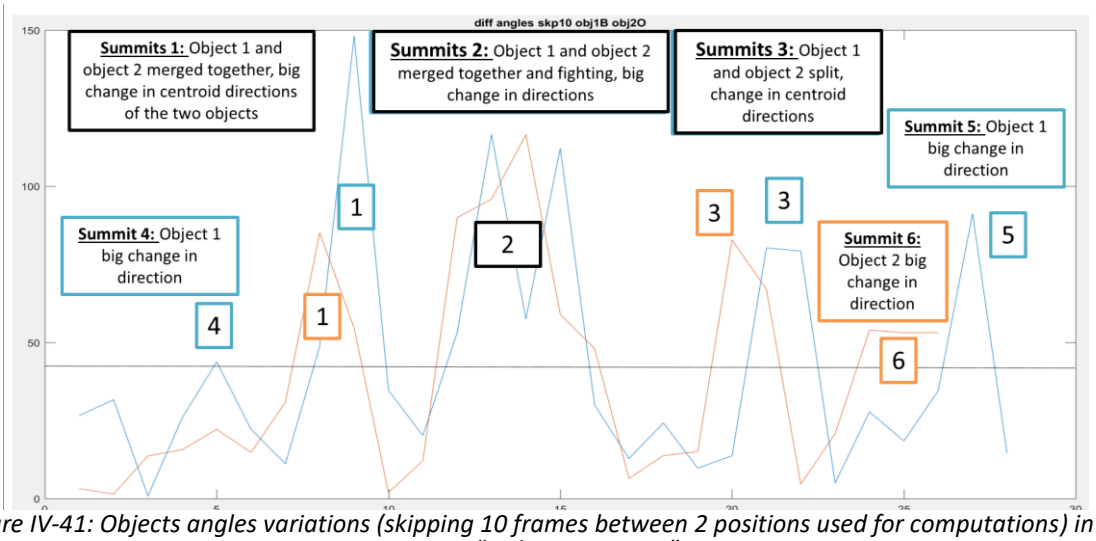


Figure IV-41: Objects angles variations (skipping 10 frames between 2 positions used for computations) in the scene "Fight_RunAway2".

2- Inter_Object characteristics:

a. Distance between the objects: In Figure IV-42.

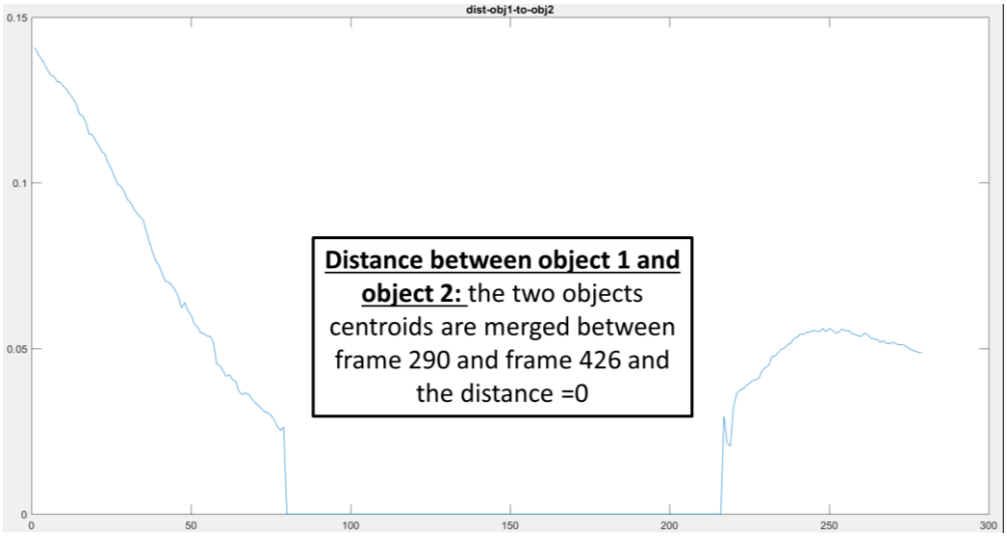


Figure IV-42: Two Objects distances in the scene "Fight_RunAway2": the local minimum occurs when the two objects approaches physically. The algorithm detects them as being one single object. The distance is then 0.

3- The interaction features: In this example, and because of the occlusion caused by the physical approach of the two objects, there is a false detection, by the used tracking and segmentation algorithm, between the frame 290 and the frame 426. Thus, the algorithm detects them as one big object. As a result, the system was not able to detect the physical aggressive interaction that starts at frame 321 and ends at frame 397. And so, only the start of distant aggressive interaction was detected at frame 279, and its end at frame 467.

From the extracted key moments (km), the corresponding characteristics were generated and filled in the vector $V(k_m)$, to produce the scene activity characteristics matrix M_{sac}. Finally, the short description is shown in Table IV-10.

To be noticed in the description, the two objects after entering the scene, start approaching toward each other and decreasing their speeds. Then, near the frame 279 the two objects start distant aggressive interaction.

Between the frames 290 and 320, one of the objects was occluded by the other, and then a physical interaction starts near the frame 321 till the frame 397, followed by another occlusion till the frame 426. Consequently, as the system only pre-processes, for classifications, the interaction between two objects, and as the two objects were false detected by the tracking and segmentation algorithm as being one, the system was unable to generates description between the frame 291 and the frame 425.

Later in the description, one can easily note the distant peaceful interaction between the frames 426 and 468, while the two persons were rolling away, before exiting the scene.

Table IV-10: The short description of the scene "Fight_RunAway2".

Info	Frame #	Desc target	Textual description
551	193	Object 1	"*Deformable* " object "1" enters the scene, *from " A " spot,* on the "*Left Middle* " of the "*Outside* " area of the camera field of view, heading "*Up Left* ", having respectively "*regular* " shape, "*small* " surface, and "*slow* " speed.
	196	Object 1	"Deformable" object "1" Occurring respectively irregularity in its shape.
Death Frame	202	Object 1	"Deformable"object "1" having respectively slight "Increasing" of its speed.
		Object 2	"Deformable" object "2" enters the scene, *from " A " spot,* on the "*Down Right* " of the "*Outside* " area of the camera field of view, heading "*Up Left* ", having respectively "*regular* " shape, "*medium* " surface, and "*Normal* " speed.
		Object 1 & Object 2	The two objects are respectively "far".
	210	Object 1	"Deformable"object "1" heading "*Up Right* ", "Occurring respectively irregularity in" its shape , and having respectively slight "decreasing" of its speed.
		Object 2	"Deformable" object "2" moves, on the "*Down Middle* " of the "*Outside* " area of the camera field of view.
		Object 1 & Object 2	The two objects are respectively "Approaching".
	232	Object 1	"Deformable" object "1", leaves "A" spot, having respectively slight "increasing" of its speed.
		Object 2	"Deformable" object "2", moves, on the "*Down Middle* " of the "*Inside* " area of the camera field of view, "Toward" the object "1", and having respectively considerable "Increasing" of its speed.
	243	Object 1	"Deformable" object "1" moves, in "B" spot, heading immediately "*Up Middle* ", and having respectively slight "decreasing" of its speed.
		Object 2	"Deformable" object "2" moves on the "*Down Left* " of the "*Inside* " area of the camera field of view, "Toward" the object "1", and having respectively considerable "Increasing" of its speed.
1		Object 2	"Deformable" object "2" leaves "A" spot, and moves "Toward" the object "1", "big changes occuring respectively on" its Surface having now "Smaller" one, and having respectively slight "decreasing" of its Speed.
	263	Object 1	"Deformable" object "1" heading "*Up Right* ", having respectively slight "increasing" of its Speed.
		Object 2	"Deformable" object "2" moves, in "B" spot, on the "*Left Middle* " of the "*Inside* " area of the camera field of view, and having respectively slight "decreasing" of its Speed.
Birth Frame	279	Object 1	"Deformable" object "1" having respectively slight "decreasing" of its Speed.
		Object 2	"Deformable" object "2" heading "*Left middle* " , "Toward" the object "1", having respectively considerable "decreasing" of its Speed.
		Object 1 & Object 2	"Start" a "Distant" "Agressive" Interaction.
	290	Object 1	"Deformable" object "1" heading immediatly "*Right Middle* ", "Occurring respectively irregularity in " its shape , and "big changes occuring respectively on" its Surface having now "Bigger" one, and having respectively considerable "decreasing" of its Speed.
		Object 2	"Deformable" object "2" heading immediatly "*Up Middle* " , "Toward" the object "1", "Occurring respectively irregularity in " its shape, and "big changes occuring respectively on" its Surface having now "Bigger" one, and having respectively considerable "increasing" of its Speed.
153	426	Object 1	"Deformable" object "1", leaves "B" spot, heading immediatly "*Down Left* ", "Occurring respectively irregularity in" its shape, and "big changes occuring respectively on" its Surface having now "Smaller" one, and having respectively considerable "increasing" of its Speed.
		Object 2	"Deformable" object "2", leaves "B" spot, heading immediatly "*Up Middle* ", "Occurring respectively irregularity in" its shape, and "big changes occuring respectively on" its Surface having now "Smaller" one, and having respectively considerable "increasing" of its Speed.
		Object 1 & Object 2	The two objects are respectively "Receding", A "Distant" "Peaceful" Interaction occurs between them.
ACTIVITY NUMBER	429	Object 1	"Deformable" object "1" moves, in "A" spot, "Away from" the object "2", having respectively considerable "decreasing" of its Speed.
		Object 2	"Deformable" object "2" moves, in "A" spot, on the "Up *Left* " of the "*Outside* " area of the camera field of view, "Away from" the object "1".
	447	Object 1	"Deformable" object "1" moves, "Away from" the object "2", "Occurring respectively irregularity in " its shape, and having respectively slight "increasing" of its Speed.
		Object 2	"Deformable" object "2" moves, "Away from" the object "1", having respectively slight "decreasing" of its Speed.
	452	Object 1	"Deformable" object "1" moves "Away from" the object "2", having respectively slight "decreasing" of its Speed.
42		Object 2	"Deformable" object "2" heading immediatly "*Up Left* ", "Away from" the object "1", having respectively slight "decreasing" of its Speed.
	462	Object 1	"Deformable" object "1" heading immediatly "*Left Middle* ", having respectively slight "increasing" of its Speed.
		Object 2	"Deformable" object "2" moves "Away from" the object "1", having respectively slight "decreasing" of its Speed.
VIDEO ID	468	Object 1	"Deformable" object "1" having respectively slight "decreasing" of its Speed.
		Object 2	"Deformable" object "2" heading "*Down Left* ", having respectively slight "increasing" of its Speed.
		Object 1 & Object 2	The two objects are respectively "Approaching", no more interaction occurs between them.
	476	Object 1	"Deformable" object "1" heading "*Up Middle* ", "Toward" the object "2", "Occurring respectively irregularity in" its shape, having respectively slight "decreasing" of its Speed.
		Object 2	"Deformable" object "2" heading "*Up Left* ", "Toward" the object "1", having respectively slight "decreasing" of its Speed.
	488	Object 1	"Deformable" object "1" exits the scene, from "A" spot, heading "Toward" the object "2", "Occurring respectively irregularity in " its shape.
	491	Object 2	"Deformable" object "2" exits the scene, from "A" spot.

IV.5. Discussion

After running all the tests on our system, some difficulties show up, especially with the segmentation and tracking algorithm results where it suffered from one major traditional issue and another marginal one:

1- The major issue: is the traditional occlusion when two moving objects are physically close. Using Kalman filter in the tracking algorithm to estimate the location when an object is occluded worked well on occlusion with an inert object but not with a moving one. In latter case the occluded object location and boundary box were estimated for some number of frames (the default is 8 frames), while the foreground object is miss-detected by detecting both objects as one.
   This problem was a major one because its results affect all the system, as seen in the example in section IV.4.6. Consequently, the physical interaction in a scene was detected for only 8 frames per scene. And then, analysing and describing the interaction had no more effect until the two objects separate.
2- The marginal issue: is the false segmentation of the object when it is moving in a complex background (illumination, and high texture). This issue can trigger a description declaring a big change in the object Hu moments or surface, as the example seen in section description IV.3.8. This can be over passed at the level of thresholding.

For these particular problems caused by the tracking and segmentation algorithm, as our system is flexible, with a simple “plug and play” this algorithm can be replaced. Having lately good results with detecting objects using deep learning, like YOLOv2 and Mask R-CNN, a good plan could be by testing these algorithms and applying a pre-processing stage to extract the objects segmentation, then if one of them deliver better results and satisfy all the conditions, our selected algorithm can be replaced.

The detection of interaction existence tests show 12.5% of false detection, where 11% are due to false detection of non-interactions as interactions. However, detecting at which frame a far interaction starts it is very delicate, even for human brain, where sometimes it seems more subjective. Two experts can identify, according to their perspectives, the start of far interaction at different frames.

While the classification of aggressiveness is more subjective with hardly acceptable 6% of false classification, differentiating between physical and distant interaction should be accurate. Nonetheless, the 6 % of false classifications between distant and physical are mainly due to the false detection, by the tracking and segmentation algorithm, of the two objects when approaching as being one even before the physical contact.

Despite of the size of collected multiple datasets, the global amount of extracted data remains insufficient for sharper evaluation benchmark, and this affect directly the results of the classification algorithms. This needs to be improved by testing this system on real footage of surveillance, were having more data can improve dramatically the classification accuracy.

Moreover, it is difficult to evaluate video description, some systems typically perform an automatic quantitative evaluation of their descriptive sentences using machine translation and image captioning metrics such as BLEU (Papineni, Roukos, Ward, & Zhu,

2002), ROUGE (Lin, 2004), METEOR (Banerjee & Lavie, 2005), SPICE (Anderson, Fernando, Johnson, & Gould, 2016) and other metrics.
But assessing how good the semantic representation of visual content is, it is not a straightforward task. A video can be correctly well described in a variety of sentences. Similarly, if many persons where asked to describe the same scene, they can provide different descriptions, each one from his own perspective. This indicates that video description is also subjective and uncertain. Beside this, in many practical cases, human activities are too complex to be described with short, simple sentences, in only one way. For that, as the video description templates and models are abstractions of the natural video description processes, they had to focus only on the relevant and prominent components, thus models can be diverse and uncertainty rises. (Song et al., 2017).
And so, evaluating video description is a hard task, either automatically generated or manually, because there is no absolutely correct answer, and no absolute standards to measure systems outputs. For this end, for correct evaluations, many systems provide a human expert assessment instead of the automated ones (Awad et al., 2018), (Graham et al., 2018), (Graham et al., 2018), (Aafaq et al., 2018), (Ahmed et al., 2019).
For our approach, we found this stage irrelevant as our description output will be according to the structured templates, reflecting our reports needs as experts in the practical field.

On the other hand, our approach provides many contributions, we mention:

- In our approach, the input features are dedicated to the interaction classification process, where many of other methods do not export appropriate features from the videos.
- Our approach took into consideration the diversity of scenes, context, objects type, actions and interactions, where no conditions and restrictions were applied.
- Our approach is benefitting from machine learning and DNN, in its experimental phases.
- Our approach implements the original new classification idea of distant vs physical interaction, while other works focus only on the physical one. Moreover, detecting distant aggressive interaction can alert the surveillance control rooms' observers at early stages, giving them precious time to act.
- The experimental results show, in our classification fields, that the classic machine learning algorithms (AdaBoost, and Bag Trees) can produce quite comparable results to the DNNs, with a small advantage to the latter ones.
- The used features for description can be extended to add object colour and other features, and at the same time be used for querying the data, which is a great need in CCTV systems. All the above-mentioned extracted features are expressed and stored in the database, under the scene activity characteristics matrix M_{sac}, as a high-level symbolic description of the object's activity. This metadata contains information driven by time for each of the detected objects and interactions including: trajectory and routes taken through the field of view, time of interaction, speed and its variation, shape and its changes, directions and its variations, deformability and interaction with other objects, aggressiveness or not for its interaction, and others. This information is attached to each object detected by the system. Such intermediate information may be very helpful to the end user, especially the operators as it can be set up to generate alerts, and the investigators when searching the archive for an incident with specific description. We tried to surround, in those extracted features, most of the queries used in practice to search for an incident. For example: we can search for a car coming from

the north east, heading to the south west, speeding and had an accident, by searching in M_{sac} for big deformable object coming from the upper right of the outside area of the camera FOV, heading down left, toward the other object, having respectively considerable increasing in its speed, and both objects are respectively approaching.

- Our approach present a new ontology-based generic non-contextual way for description using well-structured generic templates, compared to the ones found in the state-of-the-art like (Ahmed et al., 2019), see sections IV.2.5 and IV.3.8.
- In our approach, we are not competing with the state-of-the-art video description frameworks. Our video description approach is a well-designed framework which focuses, mainly, on examination of interaction analysis, understanding and description for video surveillance system. It is, only, one more step forward, toward an advanced level of intelligent surveillance system.
- In our approach, the form of the textual description is controlled by predefined well-structured templates, not textual descriptions that are training the system. The verbosity of the description can be tuned at any time by the end user.
- The new structured templates containing main information reported by the police in real case incident description. Consequently, the output textual description can be generated automatically as draft reports to be based on.

IV.6. Conclusion

In this chapter, we presented a new generic non-contextual approach, based on the ontology seen in chapter II, for description video surveillance scenes. A new set of features, appropriate for videos, was extracted from the scenes after applying an off-shelf tracking and segmentation algorithm. A new classification of scenes types was proposed, based on background changes, the number of objects and variations of object characteristics. Interactions types classifications, based on DNN, take role. A graph-based pattern discovery was implemented and new matrix of scene activity characteristics was introduced to highlight appropriate key moments for description phase. Finally, new rule-based templates were proposed to structure the textual descriptions of the scenes. The verbosity of these descriptions can be controlled by the user.

V. Surveillance systems - Between theory and practice

V.1. Introduction

The law enforcement agencies are in constant search for effective public safety and security strategies to help deal with criminal and terrorist acts. While many believe that the adoption of “community policing” strategy has led to greater efficiency and effectiveness in policing, law enforcement agencies are interested in using new and effective tools that can enhance these community policing efforts, particularly in public places.

Among the most modern public safety and law enforcement tools adopted is the use of surveillance systems, in particular, to combat crime and terrorist acts in public places. Law enforcement agencies believe that community policing, which embodies a combination of proactive crime prevention and community involvement with more traditional policing functions, can be used; since monitoring public places can promote problem solving strategies, assist in arrests and investigations, and increase the fear of criminals from the possibility of arrest. In addition, it may be considered that camera surveillance systems in public places may also have positive effects, such as increasing users' sense of public safety and improve partnership between the community and the Police against crime.

For more than three decades, camera surveillance systems have been used in many countries including Britain, France, Spain, the United States, Monaco and others. After September 11, 2001 attacks, the use of these systems became more common in order to deter future terrorist attacks. This incident highlighted the contribution of large surveillance systems to crime detection and prevention, leading to further improvements and developments in these systems.

The use of such a system in the field of safety and security varies from system to another according to the system goals and context. For example, it can be used in public places for crime fighting and order maintenance, or it can be used in critical facilities like airports or seaports and for security reason and threats detection, or on border for intrusions detection or in metro stations for security and safety, etc. Also, the use is not restricted to security, but it can trespass it for many other fields as nursing like for elderlies or enfant, management like for traffic or industries or sports fields, statistics like in sports or merchandise, etc.

Due to the huge number of videos produced from the surveillance systems cameras, two main problems face the surveillance control room management and operators, first is the shortage of active monitoring associated with the need of automatic alerts, and the second is the difficulties of investigating the archives.

As mentioned, enormous work has been done on the analysis and understanding of videos in general, and the surveillance systems in particular; several significant solutions from the research and market fields have been proposed. However, the lack of generic and precise models for video content representation, and the high complexity and diversity of video scenes, make building of fully automated intelligent video analysis and description a challenging task. Additionally to these difficulties facing these automated systems, and because of the diversity of end users needed outputs, the application domain still shows a big gap between what is needed by the end users, what is produced in the research field

and what is delivered by the companies as video analytics tools. By end user, we mean law enforcement personnel, security officer, or any surveillance system operator.

The work presented here was developed with “Beirut CCTV control room”. This last chapter of contribution is aiming at confronting our experience in the field of academic research and our expectations in the field of actual management of video surveillance systems. The main intent of it is to highlights the existing gaps, and to give scientists, engineers and managers alike, a general understanding, from the theoretical and practical perspectives, of the available solutions and practical needs, involved with the surveillance system. To do so, we take some distance with the precise scope of our previous contributions to enlarge our field of view to all the main aspects of contributions and expectations in this area before concluding this thesis.

In the following section V.2, we present a quick surveillance system overview about where and why it is used, as well as the existing video analytics in the market. Later, in section V.3, we explain how surveillance systems can be used to fight the crime. In the section V.4, we highlight some existing gaps between practice and research field as well as our propositions on how to reduce some of these gaps.

V.2. Surveillance system overview

Visual surveillance systems are one of the most important surveillance systems used and the most effective weapon to combat crime and terrorism in public places.

These systems are now a dominant feature of institutional security systems and are used in multiple locations and areas, and for different purposes.
Places where surveillance systems are used by the public or private sector are:

- Public places: streets and intersections, especially in city centres, squares and parks, shopping centres, museums and public libraries.
- Public utilities like the airports and seaports, military and security facilities, prisons and detections, and transportation (trains, tramways, metro, buses, cars, planes, etc.).
- Inside and surrounding buildings, public and private institutions, industrial buildings, environmental places and embassies.
- Inside and surrounding sports venues, health centres, hospitals, pharmacies, hotels, banks, ATMs, restaurants, schools and universities.

Camera surveillance systems can be used in almost all aspects of life and are an important tool for management, security and law enforcement. These systems can therefore be used in vast fields, and for different purposes as needed. We mention:

- Monitoring and managing public and private establishments.
- Traffic management.
- Control and management of crowds.
- In the scientific field, military and space research.
- For public safety, security and law enforcement: where the purposes and objectives are to fighting crimes and terrorist acts, protection of property and individuals, assistance the law enforcement agencies in decision-making operations (street demonstrations, celebrations, official VIP movements, emergencies, etc.), risk management in fire, natural disaster and crime situations, security monitoring of sensitive places, collecting public information, and laws enforcement (such as traffic law).

Many commercial system providers exists which offer surveillance video analytics solutions for residential, commercial, and law enforcement agencies. Next we mention, most known deliveries from different video analytics tools existing in the market:

- Motion detection
- Automatic number plate recognition ANPR system, also known as License Plate Recognition (LPR).
- Facial and iris recognition
- People-counting
- Abandoned or Removed Object Detection
- Intrusion or Trespassing, also known as Perimeter Protection
- Vandalism and camera tampering detection
- Detecting and counting crowds
- Loitering
- Object tracking: where they provide, according to this, stationary vehicle, direction detection, wrong way detection, illegal turn.
- Vehicle type recognition
- Colour detection
- Pedestrian detection, on highway.
- Gait analysis
- Congestion and accident detection
- Smoke/fire detection

Also, some providers offer investigation tools where you can search the database according to some features, mainly speed, colour, and type (pedestrians or car).
To our knowledge there are no frameworks allowing to evaluate and to compare technologies integrated by the providers in their systems. Furthermore, there is a total lack of certification for such precise software pieces that are dealing with citizen's security. Therefore, there is a necessity for an independent evaluation of such capabilities. The most used and reliable ones, comparing to others, are the ANPR, and facial recognition.

V.3. Surveillance systems for fighting crime and terrorist acts

Anti-social behaviour in public places has significant costs on the society, the economy, and the lives of citizens. For example, in Beirut, from the beginning of year 2019 till the end of March, more than 970 crimes are occurring in public places risking citizens' lives, from which, for example, 175 are snatching. Another example, in UK, from 2007 till now, graffiti and vandalism alone cost to the British government around £ 3.4 billion pounds sterling a year.

From this perspective, all countries are searching for a valuable tool in the field of public safety and security management, and in the protection of persons and property, especially in the fight against crime and terrorist acts by preventing them. Many means, beside surveillance systems, are used; we mention streets lights, or false cameras (we mean by these cameras covers). After experience, law enforcement agencies are adopting the

visual surveillance systems, and pushing to more improvement to answer all their needs and goals.

All cameras in any surveillance system are sending their signals to be stored in archive. Therefore, two methods of system usage can be mainly distinguished:

1- Archive search: by the 'investigators', and because of the high technicality and difficulty, investigators must be well trained, to harness the technology and use it efficiently.
2- Active Monitoring: means the interactive surveillance by observers to try to detect incidents and monitor them during their occurrence. Detecting an incident during its occurrence is a very difficult task, especially when the operator has to deal with a large number of cameras and a wide geographical area.
In both cases, the video analytics can play a key role in optimizing the system usage.

How surveillance systems fight the crimes? Surveillance systems lead to a major shift in the fight against crime and terrorist acts, along two main ways: indirectly and directly. All this may call for a change in the organization of the police and its way of policing in the face of crime:

V.3.1. The indirect effect

Surveillance systems have indirect, non-negligible effects on the fight against crime and terrorist acts. These effects fall into two main points: the first is deterrent crime prevention, and the second is the collection of security information and the production of intelligence.

V.3.2. The direct effect

Surveillance systems can fight the crime and the terrorist acts directly in three stages:

1. Before the event: suspected incidents or events can be detected, by observers during interactive monitoring or by video analytics, and can be prevented. This is called proactive prevention. This is the case, for example, when an ANPR system detects and triggers the alert when a wanted car pass by a gantry, or when a face recognition system alerts the observer about a suspected person passing in the view field of a camera, etc.
2. During the event: events are followed by observers, video analytics, as well as live incidents reported by citizen to police emergency rooms. Here, the problem is to locate quickly the incident, help the police to assess the situation and appropriately intervene, and prevent the situation from escalating. Video analytics can help the operator to quickly locate the incident by searching some features, tracking cars or persons, counting people in the crowds, collecting information about involved people (faces, descriptions…) and cars (models, plate number, routes…).
3. After the event: it is possible to return to previous events with the aim of investigating an incident by the relevant police units and arresting and convicting the criminals. Video analytics can help investigating the archive with appropriate and advanced features, which save dramatically the time wasted to find the incident, and recognize the involved people and cars and their habits, trajectories and behaviour patterns, hot spots and other critical information.

In all the above points, direct or indirect, surveillance systems can be a game changer in public security and other fields when using appropriate intelligent video analytics. But the main problem remains where the end users do not get what they need. In the next section, we focus on this point and we propose some practical needs and improvements to video analytics.

NB: It is worth to mention here that, many contradictory opinions exists about the use of the surveillance systems and its risk on privacy and liberty, where the fear from the "Big Brother" is increasing. In plus, many claims about the non-effectiveness of using surveillance systems remain as it is only displacing the crimes, adding to that, the allegations about the surveillance systems as not cost effective. These two mains points are the subject of many researches in this field; here it is not the right place to amplify the explanation about it. Shortly, after doing many researches about these two important points, and after practicing and manging such a system during three years, one cannot deny that, such a system can be limitary miss-used in the wrong hands. Hence, the firm regulations and procedures, to not allowing that from happening, take role; in the other hand, the installation of a surveillance system should not exceed the public places. Concerning the cost effectiveness point, one of the main strategies of fighting crimes is to displace them; also, there is a miss-conception, when measuring the crimes statistics before and after the installation of surveillance systems, which is not taking into consideration that, after installation of surveillance cameras, the detection of the crimes increase yet not the number of crimes. Moreover, one question is placed at the disposal of the reader judgement: if a surveillance system helped saving only one person life, and sure it can help many, how one can estimates the cost of this person's life?

V.4. Filling the gap between practice and theory

There is a big gap between the research field and the application domain. What is considered as done in the research field may be considered inefficient or insufficient for the end user. Let us take, as a quick example, the simple video motion detection which has been used to provide a way for alerting the operators or activating the recording systems. While this kind of tool is widely installed in commercially available surveillance systems, motion detection still has many drawbacks, as it can be falsely triggered by non-motion incidents such as those due to variations in the illumination or weather-related changes. These false alarms are a significant disturb for a police officer, costing him to be alerted and to prepare for an action, which is thus time and effort consuming.

While in the research field, video analytic studies address very advanced subjects, on the other hand, in the market, the production is sometimes still not at the needed level. This discrepancy could be explained as kind of miss understanding of the real needs to improve the use of the system, from three perspectives:

- From the end user point of view: the system managers or CCTV control room managers (for example the police) may not know how to express their needs in terms of video analytics and may have difficulties to estimate a priori its power and impact.
- The companies and industries are the link between the end users and research laboratories, and so, they focus on valuable technology transfer leaving on the side tools

which may generate a too low profit, even if these ones may be of interest in some specific cases.
- As a result, the research field, not always take into accounts the real needs, or real feedback, while it is spreading its efforts in many directions.

We should note here, that many efforts and conferences have been held trying to raise the awareness about theses gaps and to approach these mentioned fields perspectives (ICDP conf, 2017), (Zaman et al., 2017). Yet many works still have to be done.

According to the research done, and my experience as manager of a new CCTV system, some difficulties on both sides, in the research field and in the practice field, must be highlighted:

V.4.1. Main difficulties and propositions in the research field

- Need for strategic planning for problem solving

  One important difficulty every researcher suffers from is not knowing where and what to do in an already known field, especially if there are no normalized procedures, datasets, and results to refer to. Even if it's the researcher role to find where he or she is standing among the studies, but without a strategic planning, this could become quickly a waste of energy, talent and time. Having such a plan, the researchers will not focus their efforts on an already existing field with previously good results, conversely the dedication would be to start a new one or at least to continue where the previous finished.

- The non-availability of pretended good algorithms

  One of the most frustrating cases occurs when reading about an algorithm producing good results in some field, without having the capability, in most of the cases, to use it as it is without major efforts, because of the non-availability of a piece of runtime software or because its high cost, or if appears after testing the algorithm that the claims about the good results are overestimated. In the opposite case, having an algorithm with very good results, and available to the public, it could be a great contribution in the scientific field, for example, a tracking and segmentation algorithm.

- The great need of real datasets

  Working on video analytics, especially with AI oriented algorithms, raises a great need of huge databases of real video surveillance. Even though there is an access for a one, another need pops up to front, corresponding to the lack of dataset annotation and the lack of positive cases of significant incidents. With respect to the applicable regulations and procedures, the solution would be the continued cooperation between the laboratories and surveillance systems owners, and to take advantage from all marked incident scenes and annotation taken from their side.

- Tracking and segmentation problems

  In the video analytics field, even with the existence of more sophisticated methods and algorithm, there is still a great need to optimize indispensable objects segmentations and tracking algorithms, because it is the base of most of the objects features and the first step of many systems. Most of the algorithms drawbacks and inaccuracy are mainly due to two problems:

1) The shadow and illumination: it is hard to segment the object from its shadow especially when having a difficult background and changing illumination. Also, it makes it difficult to segment the sub-object parts.
2) The occlusion problem: we may consider two types of occlusions, one with a background object (and this is the easiest one), and the second with another moving object, where the occlusion may take long time when there is an interaction. In that case, the system loses the objects during the interaction and may identify it after as being another object, which leads to more post-processing steps (recognition and re-assignment).

V.4.2. Main difficulties and propositions in practice field

As already mentioned, there are two main problems facing the surveillance control room management, the shortage of active monitoring following the need of automatic alerts, and the difficulties investigating the archives:

- <u>The shortage of active monitoring</u>
  Actively monitoring a whole city with thousands of cameras is impossible, even when having a big number of operators. Moreover, even with low number of observed cameras, most of the incidents are logically miss-detected in like-wise manual system due to natural limitations from deploying indispensable solely human operators facing CCTV screens. These miss-detections could be caused by:
  - The limitation of human being capability: the human attention span is limited and tasks that require intensive sustained vigilance such as monitoring CCTV feeds should be covered in brief shifts of 20 minutes and maximum 30 minutes before resting for a while, and covering a limited number of cameras. Deficiency in human resources makes these conditions hard to fulfil. Therefore, long hours watching an excessive number of video screens is the real daily bases of many surveillance systems.
  - Boredom from monitoring is a real issue also, nobody take it into count.
  - Lack of a priori and readily accessible knowledge for what to look for
  - Distraction and interruptions by additional responsibilities such as other administrative tasks (Gill et al., 2005).

  Considering the above human limitations, an effective live monitoring therefore requires a large number of operators that will inevitably increases the cost. Confronted to this reality, plus the limited resources, the management tend usually to allocate little resources to the live monitoring and prefers to focus on the passive one. This choice tend to be the right one considering the small number of incidents that can be detected with human live monitoring compared to the other duties that could yield a better return.
  However, considering the great importance of live monitoring, using video analytics that can actively alert the operators on possible incidents becomes the best support to the human monitoring.

- The difficulties investigating the archives
  Searching hundreds of hours of video footages for an incident without having exact location and time is barely impossible. Especially when we are not sure if the incident will shows up inside the Field of View (FOV) or not. Sometimes, it is only some meters farer or some minutes later than the searched incident occurs. Most of the times, the incidents are not in the FOV, but we still need to search the surrounding areas for a possible passage of a motorcycle or a car or a person with some specific features. All the activities mentioned above are extremely time-consuming, and they are done manually. Having video analytics than can decrease dramatically the search time, by using some features, will be a great improvement.
  Despite much advancement in the field of automated surveillance, video analytics is still facing some challenges when it comes to real world conditions.

In the following section, we list some of the main problems and propose some improvements, on the level of systems and software, which could be very beneficial for the end users:

V.4.2.A. System improvements propositions

- Integration with other systems
  A Video Management Software (VMS) is not effective alone. Surveillance systems using a VMS would be more effective when it is integrating with other systems. Let us mention for example:
  1) VMS integrated with ANPR systems (including database of wanted cars), where the wanted (marked) car in the ANPR system, when triggered, can be highlighted by the VMS cameras.
  2) VMS integrated with face recognition systems (including database of wanted people), where the fixed and PTZ cameras can search for faces to be compared, and then highlight them if they appear as wanted persons.
  3) VMS integrated with a mapping system: surprisingly most of the VMS come without a mapping system, which is one of the most important and needed systems for crimes analysis or traffic management. A mapping system can handle all the geographic and context-based database of the area under surveillance, and by this mean the abstract description can be improved by combining the contextual information (place, location, popularity, weather, temperature, lighting …) to generate more semantic and meaningful descriptions.
  4) VMS integrated with the “Computer Aided Dispatch” (CAD) systems: where it can show the live location of moving cars (for example patrols).
  5) VMS integrated with “Internet of Things” (IOT) systems: where IOT allows to aggregate data from various sensors, which is a very vast domain.
  6) VMS integrated with Business Intelligence (BI) tool: where hot spots of all the incidents can be highlighted according to live analysis of these incidents, helping the operators to focus on the most possible places where and when incidents may occur.

  All the above mentioned systems can be handled by an appropriate mapping system where all pieces of information can be seen as embedded layers.

- The VMS should be more than management

  The VMS it is not only for the video management, but it can be considered and improved to be a collector of data, connector of devices and provider of analysis, among others.
  As it has been mentioned before, all the collected data from video analytics and descriptors will accumulate information to form big data that can be learn through clustering, and regression for knowledge modelling and event prediction. Hence, this should lead to new opportunities and new challenges. Data mining and AI can induct prediction and proactivity. For example, the system can latter reference the relation between a specific person or object or car and a specific location, and indicates its hot zones. Another example is when a target enters a scene, the learnt routes provide typical patterns of behaviour, and probabilities are assigned about possible exit points. Later, atypical behaviour can be identified with targets that do not use the established routes and will alarm the security personnel.
  In terms of architecture and design, in order to reduce the excessive bandwidth needs for transmission and the computational load of the central processors, there is a trend in video analytics practice to prefer distributed intelligence solutions, which consist in locating more video processing or intelligence analytics at the level of cameras (or sensor).

  Nowadays, with the great progress we witness in the domain of AI and DNN, in my opinion, the next-generation of intelligent video surveillance analytics systems will induct DNNs and more generally AI as far as we can identify that there is a great need.

  The importance of user-centred design or user-centric models is now widely recognised in video indexing and retrieval. But we should not forget the excessive bandwidth and the computational load. Both types (centralized and distributed) of systems should co-exist. Hence, those systems could take benefits from the advantages of both designs. Keeping some level of distribution, it can furnish more natural and flexible user interactions. Relevant feedback from the user is essential so that the system can perform better; this can optimize the learning of the system and its performance. Especially when, the limited amount of positive cases to train recognition algorithm adds difficulties in detecting these so-called rare events.

  When seeking for performance, there is no escape from coexistence of both systems, where the system should be learned locally, at level of each camera alone, and at a global level, fed with all outputs of cameras, to “see the whole picture (puzzle)” of the current state of an area under surveillance.

  The concept of learning from the data is widely appropriate for video surveillance, since it is possible to allow the system to learn by observing scenes activity over long periods of time, where the scene context and structure influences, directly or indirectly, the way that objects act. Therefore, specific type of events or incidents may be associated with specific regions. For instance, doors or gates oblige people to pass by; roads constrain vehicles to move along in a particular direction, etc.

  This strengthen our point that what is needed, is merging heuristic-based and learning approaches, where we can learn dedicated heuristic features for better understanding the area, the objects, and the incidents, seeking better performance and generic abstract description. Accordingly, each area seen by a camera has its own particularity

(routes used, speed level, density…), added to the whole city common characteristics (weather, demographic, geo-economic…).

- Real timeliness

  Useful video analytics algorithms for surveillance systems should be real time. However, very accurate and robust real time analysis needs intensive computing processing. And intensive computing processing needs lots of investment. Again, one possible solution would to make the accuracy easily adjustable by the users so it could match their computing capability when doing real time surveillance. Besides, this could also open new market for the software companies. Market that are currently inaccessible because of the investment barrier.

- Standardization for video analytics

  Since there are so many hardware types and video file format, the integrated systems with high diversity are facing hard time to exchange data and to view or access the video contents. This case will be more critical for video analytics, as high-level semantic descriptors are required to represent properties of objects, events, scene contents, and so forth.

V.4.2.B. Software improvements propositions

- False alerts or false negative

  In the research field, having a precision rate of 85% and above can be considered as good results for some tasks. However, in the practice, the reliability of intelligent video analytics is a paramount issue, since 1% of error can be catastrophic. For example, a false alert of 1% for 1 million cars entering Beirut city daily, means 10 000 records that need to be checked manually. Additionally, frequent false alarms induce mistrust in the operators, who quickly tend to ignore the system. Therefore, improving accuracy even by some percentage could have tremendous positive impact on the work of the end users (Velastin, 2009).

- Miss-detections or False Positive

  CCTV control room managers usually mistake miss-detection for false detection. The miss-detection is a very critical problem and most video analytics companies don't communicate very openly about it. For example, the ANPR, despite of being one of the most reliable systems, it still has its miss-detections rate even if it's very low. And Failure to detect, for example, a suspected or a stolen car going into the country to be used in a bombing, is not an acceptable error. Again, increasing precision in general will help. But for this particular problem, one possible solution is to make threshold that triggers the alert of the algorithm adjustable, in a simple way, by the end user. This way, the operator can, for critical cases like an imminent threat for example, lower the threshold (even if this might increase the number of false alerts). Then it will be up to the end user to find the right balance between the miss detection and false detection.

- Query parameters

  Due to the rapid increase of the number of cameras used in video surveillance and the huge amount of footage that can be produced, the challenge of video analytics is to extract meaningful information in order to produce high-level semantic resources. These resources can be later used to search for a specific content. Being able to query the video database for using combined parameters could help dramatically the operators looking for an incident and save a lot of time. The existing products on the market allow searches according to some important parameters, like the colour, direction and type of an object. But in practice, there is a wider range of needs, like the ones provided by this thesis. It would be good if the wide range of features involved in the algorithm can be taken as parameters for searching. This set can be enlarged to some recognition-based parameters. Let us mention:

  - Object type, like animals, bicycle, truck, motorcycle, etc.
  - Same object like the same person, car, scooter, text and letters, etc.
  - Shapes like box, logo, tattoos, wheel rim, hat, knife, umbrella, car lamps, etc.
  - Special marks, like car with damage in a special area, or broken light, etc.
  - Night special marks, like, coloured lamp light (blue, neon…), lamp shape, the distance of car lamps, flashers…
  - Many other parameters may be used also.

  It could be also very useful to have the ability to simulate the scene by designing a scenario, for the system to search for similar ones, including all possible parameters like objects, movements, and interactions. For example, a person with a box comes from a specific route…
  Moreover, it is very beneficial if the search results came with percentage of reliability, indicating the percentage of matching the searched criteria.
  NB: many of the above-mentioned parameters and features could be valuable also to be used for alerts.

- Tracking and recognition

  The system should be able to recognize and track, through cameras, even far ones:

  1) Wanted persons.
  2) Wanted cars.
  3) Any selected object.

  An idea is to assign high probability to the surrounding cameras to search in it. Also, when marking and selecting a car or person, it would be beneficial if the movable (PTZ) cameras can automatically seek for more description of the tracked object (car plate number, face…).

- Detection on some special cases

  It would be very helpful if the system can detect also the “non-existence” of a feature, for example:

  - For cars: detecting cars without plate numbers.
  - For motorcycles: detecting motorcycles without plate numbers and persons on motorcycles without casks
  - For people: person “without face” (whom faces are covered).

- Behaviour and body language analysis

  Detecting special behaviours and analysing body language locations could help preventing crimes. For instance, detecting the special behaviour of the thieves (face down, waiting, surfing …) could help prevent a robbery.

- Improve robustness

  A major challenge for real world scenes is the dynamic nature of real world conditions. Achieving robust algorithms is a challenge especially under illumination variation, weather conditions, under view changes, existence of multiple objects, occlusion, deformation, shadow, reflections, video noise, and moving background. Significant research and advances in solving these difficulties have been achieved, but user can profit still from lots of improvement.

- The night time surveillance

  The night time is one of the biggest problems which CCTV observers and investigators suffer from. Even when using Infra-Red cameras, a wanted car or person may be detected but no more details or description will be available. Moreover, during the sunset and the sunrise even the IR is not effective. What is needed is to lighten all the surveillance area with white lights, as it is better than yellow. Any enhancement that can be done at the software level, it would be of great benefit.

- The rain

  This is a common problem for all surveillance systems, when not only the water sticks on the lens but also dust, or more generally when the rain makes the vision impossible. For the water to slide down from the lens, some new lenses were introduced. Also, some systems use a special solution to spray the lenses. But the problem of the vision remains, and need more attention, maybe some kind of filtering at the level of software may improve the vision.

V.5. Conclusion

Visual surveillance systems supported by appropriate intelligent video analytics can be an effective weapon in the hands of the law enforcement agencies. In this chapter we overviewed the surveillance systems and their usage for fighting crimes. We highlighted the existing gaps between the research field, production companies and end users of surveillance systems. Many propositions, inspired by practical and theoretical experience in this field, were made to narrow this gap. These propositions encounter the research field, and the practical one, especially at the level of future intelligence video analytics development and integration with other systems. Some of these propositions are innovative yet simple to be applied, which can bring great benefits and optimize the use for surveillance systems operators for live monitoring, investigating, and analysing the crimes.

VI. General Conclusion

In this chapter, we first list our key contributions in the field of video surveillance analysis and description. Then, we will show how the questions mentioned in the introduction find an answer thanks to our VSSD approach. Finally, we indicate interesting directions for future research in this field.

VI.1. Key contributions

The key contributions of this thesis are the following ones:

- Our VSSD approach introduces a new heuristic way to discriminate between deformable and non-deformable objects in the scenes.
- Our VSSD approach presents a new generic flexible and extensible ontology for video surveillance scenes description.
- Our VSSD approach introduces new concepts concerning mediation and interaction at a distance, deformable and non-deformable objects, abstraction in description, and a new manner of categorizing the scenes.
- Our VSSD approach implements the original new classification idea of distant vs physical interaction, while other works focus only on the physical one. Moreover, detecting distant aggressive interaction can alert the surveillance control rooms' observers at early stages, giving them precious time to act.
- Our VSSD approach presents a set of new features dedicated to the interaction classification process. While many methods focus on exporting features using convolutional networks from the frames, in VSSD approach we took into consideration the distinctive propriety of videos which is the temporal relation through these frames, and we focused on selecting valuable features which can influence or get influenced by the interaction, like the object direction, shape, deformability, Hu moments, speed, etc. These video scenes important features can be used for generating alerts and intelligently investigating the archives from real case perspective.
- Our VSSD approach integrates the traditional methods of the features extraction along with machine learning and DNN of the interaction classifications. This integration allows our approach to benefit from the advantages of both methods, by giving more control when selecting the features and more results' accuracy when classifying.
- Our VSSD approach provides a novel direction to work in generic abstract domain. As mentioned, working with video surveillance is a complex problem due to the scenes diversity, like scene location, environment, object types, actions and interactions. To simplify the problems, some research added more assumptions, and this may have improved the results significantly but it limited and restricted its applicability in real world. To encounter this, our VSSD approach is kept generic, not restricted to any of the scene categories, and abstract, not semantic.
- VSSD can be easily extended and improved without major changes in the overall process.
- Our VSSD approach presents a new ontology-based and non-contextual way for video scene description using well-structured generic templates. The new structured templates contain the main information reported by the police in real case incident

descriptions. Consequently, the textual output description can be generated automatically as draft reports to be based on.

- Our VSSD approach allows the end user to tune the verbosity of many key description characteristics.
- Finally in chapter V, we developed several propositions on both sides, in the research field and in the practical field, driven by practical experience to reduce the existing gaps between the surveillance systems operators' needs from one side, the research field and the commercial (industry) field from the other side.

VI.2. In Summary

As a summary, we propose here to go back to the questions identified in the introduction of this thesis, and to answer them with collected elements taken from the previous chapters:

a) How should the video description system be built in order to be more efficient and useful for different surveillance systems?

In general, surveillance systems differ from a system to another mainly by the diversity of the scenes.

To build an efficient and useful description system that can suit different surveillance systems, the diversity of the scene types, the environment, the objects types, the actions, and the interactions should be taken into consideration. In our ontology-based approach, as seen in chapter II, we kept our approach generic, not restricted to any of the scene categories, and abstract, not semantic. Also, as seen in chapter IV section IV.3, our system preserves the modularity where each of the composing algorithms can be treated as independent module. The algorithm can be replaced or altered by adding more information as input or producing more outputs like features.

b) How to decide what visual information to extract from video?

As we know what we want to use the system for, what to describe, and what we need to search for in the video archives, this can help to decide what visual features to extract.

We want to use our system for intelligent search, generating alerts and extracting textual reports, also our system should be able to describe any objects abnormality and the interaction between two of them. For all of that, as seen in chapter IV section IV.3, our approach concentrated on extracting and analysing simple yet influencing visual information in videos surveillance which can feed these targets, we mention the object direction, shape, deformability, size, Hu moments, speed, position, trajectory, existing of interaction, interaction type, and interaction aggressiveness, etc. These characteristics are very useful for Beirut CCTV control room as search, alerts and reports, see chapter IV and Appendix VIII.8. Nevertheless, our system is not restricted to these characteristics where others can be added to the mentioned functionalities by simply adding these extra characteristics, from the 2305 features, to the scene activity characteristics matrix and the description.

c) What should a system describe? How to decide when generating a textual description along the time dimension? How much the description is practical and responds to the user needs?

The description produced by a system varies from a surveillance system to another according to the needs and goals of the corresponding control room management. It could be a full description of the footage (having, at each moment, a full description of all the features of the existing objects) or more simple description for only certain events or behaviours (E.g.: abnormal activities like law violations, and others). This situation may be highlighted by a change in speed, or shape, or trajectory…

As seen in chapter IV sub-section IV.3.8, our approach is able to produce three types of description: the full description, the short description and object life cycle description. For appropriate description, we presented a simple threshold-based method to detect all features' abnormalities.

As the system will be used by the observers, investigators and analysts, it should respond to their needs from one side, and from other side it should not submerge them with useless outputs. In our approach, for more practicality, the verbosity of the system outputs is adaptable and controlled by the end user (by selecting what features must be used as triggers for detecting abnormality and generating description, and by controlling the density of generated descriptions).

d) How to design a powerful sentence generation model? What an adequate textual/sentence representation contains? What is the best combination of different components of a representative sentence?

A sentence model for a video surveillance system is powerful when this output model can help to generate the real case incident reports done by the investigators, and interpreted by the analysts.

In real cases, as mentioned in chapter IV, when an incident occurs, five main points are mainly needed for a scene description, also known as the five Ws (Who, what, where, when and why (which is often replaced by how)).

As seen in chapter IV section IV.5, the video description is subjective and uncertain and there is no best combination of the representative sentence, only an adequate and preferable one. In our approach, we embedded all the mentioned pieces of information in the sentence description. Also, we used suitable structured templates; similar to the ones used for police reports, reflecting our reports needs as experts in the practical field.

VI.3. Conclusion and Perspectives

If properly implemented and used, visual surveillance systems, supported by intelligent video analytics, can become a very effective weapon in the hands of the law enforcement agencies. It can not only help arresting and convicting criminals in very fast and efficient way, but also (and most importantly) it can also prevent some crimes from happening.

This thesis looks fundamentally at the problem of describing important contents in videos surveillance scenes, based on a new generic context-free ontology, focusing on objects interactions. While analysing and understanding a wide variety of video scenes, our approach introduces new concepts and highlights important features for better classifications of video object interactions. These features, used as key parameters in video analytics tools, are much suitable for supporting surveillance systems operators by alerts and intelligent search. Moreover, our system outputs can support investigation reports, according to investigators needs, with many types of automatic textual descriptions based on new well-structured generic rule-based schemas or templates.

In this thesis, we did not pretend to build the ultimate highly intelligent surveillance analysis and description system. This research can be seen as a step forward toward this target. It can be taken as a set of propositions that could improve the existing video surveillance systems in many ways.

Many imperative works can be done in the future succeeding this thesis. This work can be extended by fulfilling the remaining concepts of the proposed ontology. As in this work, the objects categorization was sufficient as deformable and non-deformable, also the sub-objects and the interaction subtypes are not deeply investigated, so one can start to analyse the deeper level of these classifications. For objects categories a lower level of deformable and non-deformable object classifications can be reached, namely humans, animals, plants, machines and inert objects. Concerning sub-objects, the deformable object articulations and its movements can be the focus of many studies, especially its relation and correlation with the object interaction. Also, interactions subtypes can be investigated to add more layers to the classified ones (distant vs physical, and aggressive vs peaceful), for example the interaction has bad influence or good influence on an object, etc.

In plus, for having better results in tracking and segmentation which can have a huge positive influence on the system results, the current tracking and segmentation algorithm can be replaced by a more recent algorithm based on deep learning, potential YOLO v3 or Mask R-CNN.

Additionally, as applying this description approach on other scenes' types (according to our types seen in chapter IV) seems simple, an interesting future work will be to apply it on more complex scenes, showing interactions between more than two objects.

Moreover, the need for more data for learning the classifiers is still a big issue; especially real databases, diversified annotated, and classified. As the head of Beirut CCTV control room, I have the chance to manage the control room's databases and its annotations, about real incidents and events, in a suitable way for later tests, so I will take advantage of this rear opportunity to drive these tests and researches. This can be led to more improvement in both fields, research as automatic video analysis and practical as law enforcement applications.

Furthermore, to optimize the automation process of the entire approach, as well as to generate specific and practical scenes descriptions, integrating contextual information (mentioned in the chapter II) would make the description more relevant to the operators. As an example, areas and spots of the camera field can be named into more meaningful and contextual descriptions like, in front of the “name” market or intersection of “name 1” road with “name 2” road. A practical method to introduce this information is to combine the Video Management System (VMS) with a mapping system, as mentioned in chapter V. No training is mandatory to reach that level of description.

On top of that, as working on thousands of hours of videos surveillance footage, all the output data from our system, when applying, can form a big data. This big data or metadata collected and accumulated over a set of few months can be then learned through clustering, and regression for modelling and prediction of the objects behaviours and interactions.

From another perspective, driven by our personal experience as researcher and surveillance system manager, in order to encourage more cooperation, we highlighted the existing gaps between the research field, the video analytics companies and the needs of surveillance system end users. On a large scale, several possible future directions in the practical field and the research one were extensively discussed, keeping in mind that the surveillance systems are only tools that cannot operate alone and that should be integrated with other systems, within a larger public safety and security strategy. From these propositions, we mention the one about answering the need for flexibility when it comes to adjusting the algorithm thresholds allowing the end user to have more control based on his or her needs.

With the current advancement in the field of learning methods, many researches of different domains tend to shift their approaches to rely completely on learning strategies at all levels. We believe that it still early in the domain of video surveillance to rely completely on learning strategies. Nerveless in the future, new technologies and software supported by AI and more specially DNNs will change the face of surveillance systems. Until then, we believe that, for better intelligent video analytics products encountering all the needs of surveillance system management goals (live monitoring and crime solving by searching the archives), a hybrid combination of learning and traditional video understanding approaches shall still be considered. The proposed VSSD can be seen as an example of this combination. On the other hand, these coming changes need to be matched by the readiness acceptation of human operators and managers to meet the challenges ahead and to make the appropriate changes in security policy and planning. The users of these systems, from law enforcement agencies to criminal justice systems, will need to evolve and to adapt quickly to these new technical changes and opportunities.

VII. References

Aafaq, N., Mian, A., Liu, W., Gilani, S. Z., & Shah, M. (2018). Video Description: A Survey of Methods, Datasets and Evaluation Metrics. *ArXiv:1806.00186 [Cs]*. Retrieved from http://arxiv.org/abs/1806.00186

Abdel-Aziz, Y. I., & Karara, H. M. (1971). Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proceedings of the Symposium on Close-Range Photogrammetry (American Society of Photogrammetry), 1971*, 1–18.

Agarwal, Ankush, & Suryavanshi, S. (2017). *Real-Time* Multiple Object Tracking (MOT) for Autonomous Navigation*. 5.

Agarwal, Anubhav, Jawahar, C., & Narayanan, P. (2005). A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*.

Aggarwal, J. K. (2004). Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 12–12. https://doi.org/10.1109/CVPR.2004.434

Aggarwal, J. K., & Nandhakumar, N. (1988). On the computation of motion from sequences of images-A review. *Proceedings of the IEEE*, *76*(8), 917–935. https://doi.org/10.1109/5.5965

Aggarwal, J. K., & Ryoo, M. S. (2011). Human Activity Analysis: A Review. *ACM Comput. Surv.*, *43*(3), 16:1–16:43. https://doi.org/10.1145/1922649.1922653

Ahmed, Sk. A., Dogra, D. P., Kar, S., & Roy, P. P. (2019). Natural Language Description of Surveillance Events. In P. Chandra, D. Giri, F. Li, S. Kar, & D. K. Jana (Eds.), *Information Technology and Applied Mathematics* (Vol. 699, pp. 141–151). https://doi.org/10.1007/978-981-10-7590-2_10

Aishy, A. (2001). *Object and Event Extraction for Video Processing and Representation in On-Line Video Applications*. Thesis, Institut national de la recherche scientifique (INRS), Montreal, 20 decembre.

Akdemir, U., Turaga, P., & Chellappa, R. (2008). An ontology based approach for activity recognition from video. *Proceeding of the 16th ACM International Conference on Multimedia - MM '08*, 709. https://doi.org/10.1145/1459359.1459466

Alonso, M. A. P., Leal, P. H., Escalante, H. J., & Succar, E. S. (n.d.). *ViVA Project*.

Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2016). SPICE: Semantic Propositional Image Caption Evaluation. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 382–398). Springer International Publishing.

Andrei, Georgios, E., Daniel, H., Krystian, M., Siddharth, N., Caiming, X., & Yibiao, Z. (2018). Language and Vision Workshop - 2018. Retrieved April 5, 2019, from http://languageandvision.com/2018.html

Andriyenko, A., Schindler, K., & Roth, S. (2012). Discrete-continuous optimization for multi-target tracking. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1926–1933. https://doi.org/10.1109/CVPR.2012.6247893

Anjum, N., & Cavallaro, A. (2008). Multifeature Object Trajectory Clustering for Video Analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1555–1564. https://doi.org/10.1109/TCSVT.2008.2005603

Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H. J., Ponce-Lopez, V., Baro, X., … Escalera, S. (2017). A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences. *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 476–483. https://doi.org/10.1109/FG.2017.150

Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., … Blasi, S. (2018). *TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search*. 24.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *ArXiv:1511.00561 [Cs]*. Retrieved from http://arxiv.org/abs/1511.00561

Bai, L., Lao, S., Jones, G. J., & Smeaton, A. F. (2007). Video semantic content analysis based on ontology. *International Machine Vision and Image Processing Conference (IMVIP 2007)*, 117–124. IEEE.

Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72. Retrieved from https://www.aclweb.org/anthology/W05-0909

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., … Zhang, Z. (2012). Video In Sentences Out. *ArXiv:1204.2742 [Cs]*. Retrieved from http://arxiv.org/abs/1204.2742

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, *12*(1), 43–77. https://doi.org/10.1007/BF01420984

Bashir, F. I., Khokhar, A. A., & Schonfeld, D. (2007). Real-Time Motion Trajectory-Based Indexing and Retrieval of Video Sequences. *IEEE Transactions on Multimedia*, *9*(1), 58–65. https://doi.org/10.1109/TMM.2006.886346

Beauchemin, S. S., & Barron, J. L. (1995). The computation of optical flow. *ACM Computing*

Surveys (CSUR), *27*(3), 433–466.

Benfold, B., & Reid, I. (2011). Stable multi-target tracking in real-time surveillance video. *CVPR 2011*, 3457–3464. IEEE.

Black, J., Ellis, T., & Makris, D. (2004). A hierarchical database for visual surveillance applications. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 1571–1574. https://doi.org/10.1109/ICME.2004.1394548

Blunden, A. (1978). *Leontyev's Activity Theory and Social Theory*. 15.

Blunsden, S., & Fisher, R. B. (2010). *The BEHAVE video dataset: ground truthed video for multi-person behavior classification*. *4*, 1–12.

Borges, P. V. K., Conci, N., & Cavallaro, A. (2013). Video-Based Human Behavior Understanding: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(11), 1993–2008. https://doi.org/10.1109/TCSVT.2013.2270402

Borst, W. N., & Borst, W. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., & Gool, L. V. (2009). Robust tracking-by-detection using a detector confidence particle filter. *2009 IEEE 12th International Conference on Computer Vision*, 1515–1522. https://doi.org/10.1109/ICCV.2009.5459278

Brox, T., & Malik, J. (2010). Object Segmentation by Long Term Analysis of Point Trajectories. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (pp. 282–295). Springer Berlin Heidelberg.

Bruhn, A., Weickert, J., & Schnörr, C. (2005). Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *International Journal of Computer Vision*, *61*(3), 211–231. https://doi.org/10.1023/B:VISI.0000045324.43199.43

Burt, P., & Adelson, E. (1983). The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, *31*(4), 532–540. https://doi.org/10.1109/TCOM.1983.1095851

Buzan, D., Sclaroff, S., & Kollios, G. (2004). Extraction and clustering of motion trajectories in video. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, *2*, 521-524 Vol.2. https://doi.org/10.1109/ICPR.2004.1334287

Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., & Kasturi, R. (2010). Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *IEEE Transactions on Intelligent Transportation Systems*, *11*(1), 206–224. https://doi.org/10.1109/TITS.2009.2030963

Cavallaro, A., Steiger, O., & Ebrahimi, T. (2005). Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(4), 575–584. https://doi.org/10.1109/TCSVT.2005.844447

CAVIAR: Context Aware Vision using Image-based Active Recognition. (2004). Retrieved March 26, 2019, from https://homepages.inf.ed.ac.uk/rbf/CAVIAR/

Chan, M. T., Hoogs, A., Schmiederer, J., & Petersen, M. (2004). Detecting rare events in video using semantic primitives with HMM. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, *4*, 150-154 Vol.4. https://doi.org/10.1109/ICPR.2004.1333726

Chaquet, J. M., Carmona, E. J., & Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, *117*(6), 633–659. https://doi.org/10.1016/j.cviu.2013.01.013

Chen, B.-C., Chen, Y.-Y., & Chen, F. (2017). *Video to Text Summary: Joint Video Summarization and Captioning with Recurrent Neural Networks*. 14.

Chen, L., & Nugent, C. (2009). Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, *5*(4), 410–430.

Cho, S., & Kang, H. (2012). Integrated multiple behavior models for abnormal crowd behavior detection. *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, 113–116. https://doi.org/10.1109/SSIAI.2012.6202466

Choi, W., & Savarese, S. (2012). A Unified Framework for Multi-target Tracking and Collective Activity Recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer Vision – ECCV 2012* (Vol. 7575, pp. 215–230). https://doi.org/10.1007/978-3-642-33765-9_16

Cilla, R., Patricio, M. A., Berlanga, A., & Molina, J. M. (2014). Human action recognition with sparse classification and multiple-view learning. *Expert Systems*, *31*(4), 354–364. https://doi.org/10.1111/exsy.12040

Classification Learner App - MATLAB & Simulink. (n.d.). Retrieved April 9, 2019, from https://www.mathworks.com/help/stats/classification-learner-app.html?searchHighlight=Classification%20Learner%20app&s_tid=doc_srchtitle

Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., … Wixson, L. (2000). *A System for Video Surveillance and Monitoring*. 69.

Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, *5*(4), 277–298. https://doi.org/10.1016/j.pmcj.2009.04.001

Coppola, C., Cosar, S., Faria, D. R., & Bellotto, N. (2017). Automatic detection of human interactions from RGB-D data for social activity classification. *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-*

MAN), 871–876. https://doi.org/10.1109/ROMAN.2017.8172405

Criminisi, A., Reid, I., & Zisserman, A. (1999). A plane measuring device. *Image and Vision Computing*, *17*(8), 625–634. https://doi.org/10.1016/S0262-8856(98)00183-8

Cristani, M., Paggetti, G., Vinciarelli, A., Bazzani, L., Menegaz, G., & Murino, V. (2011). Towards Computational Proxemics: Inferring Social Relations from Interpersonal Distances. *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 290–297. https://doi.org/10.1109/PASSAT/SocialCom.2011.32

Cui, X., Liu, Q., Gao, M., & Metaxas, D. N. (2011). Abnormal Detection Using Interaction Energy Potentials. *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 3161–3167. https://doi.org/10.1109/CVPR.2011.5995558

Cutler, R., & Davis, L. S. (2000). Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 781–796. https://doi.org/10.1109/34.868681

Cutler, Ross, & Davis, L. S. (2000). Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 781–796.

CV Datasets on the web. (n.d.). Retrieved March 26, 2019, from http://www.cvpapers.com/datasets.html

Das, P., Xu, C., Doell, R. F., & Corso, J. J. (2013). A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2634–2641. https://doi.org/10.1109/CVPR.2013.340

Deep Learning Toolbox Matlab. (n.d.). Retrieved April 9, 2019, from https://www.mathworks.com/help/deeplearning/index.html

Del Bue, A., Lladó, X., & Agapito, L. (2007a). Segmentation of Rigid Motion from Non-rigid 2D Trajectories. In J. Martí, J. M. Benedí, A. M. Mendonça, & J. Serrat (Eds.), *Pattern Recognition and Image Analysis* (Vol. 4477, pp. 491–498). https://doi.org/10.1007/978-3-540-72847-4_63

Del Bue, A., Lladó, X., & Agapito, L. (2007b). Segmentation of rigid motion from non-rigid 2d trajectories. *Iberian Conference on Pattern Recognition and Image Analysis*, 491–498. Springer.

Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Found. Trends Signal Process.*, *7*(3–4), 197–387. https://doi.org/10.1561/2000000039

Dogra, D. P., Ahmed, A., & Bhaskar, H. (2016). Smart video summarization using mealy machine-based trajectory modelling for surveillance applications. *Multimedia Tools and Applications*, *75*(11), 6373–6401. https://doi.org/10.1007/s11042-015-2576-7

Donahue, Jeff, Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2017). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Trans. Pattern Anal. Mach. Intell.*, *39*(4), 677–691. https://doi.org/10.1109/TPAMI.2016.2599174

Donahue, Jeffrey, Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2625–2634.

Dubrofsky, E. (2009). *Homography Estimation*. PhD diss., University of British Columbia.

Dufaux, F., & Moscheni, F. (1995). Motion estimation techniques for digital TV: A review and a new contribution. *Proceedings of the IEEE*, *83*(6), 858–876.

Dumortier, Y. (2009). *Perception monoculaire de l'environnement pour les systèmes de transport intelligents* (Thesis, Paris, ENMP). Retrieved from http://www.theses.fr/2009ENMP1640

Ebrahimi, T. (1997). MPEG-4 video verification model: A video encoding/decoding algorithm based on content representation. *Signal Processing: Image Communication*, *9*(4), 367–384. https://doi.org/10.1016/S0923-5965(97)00028-3

Elhamifar, E., & Vidal, R. (2009). Sparse subspace clustering. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2790–2797. IEEE.

Fakih, A., & Zelek, J. (2008). Structure from Motion: Combining features correspondences and optical flow. *2008 19th International Conference on Pattern Recognition*, 1–4. https://doi.org/10.1109/ICPR.2008.4761007

Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every Picture Tells a Story: Generating Sentences from Images. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (pp. 15–29). Springer Berlin Heidelberg.

Farin, D., & de With, P. H. N. (2005, July 1). *Evaluation of a feature-based global-motion estimation system*. 59603X. https://doi.org/10.1117/12.632680

Farin, D., & With, P. H. N. de. (2005). Evaluation of a feature-based global-motion estimation system. *Visual Communications and Image Processing 2005*, *5960*, 59603X. https://doi.org/10.1117/12.632680

Feng, J., Won, I., Jeong, J., & Jeong, D. (2015). Rigid and non-rigid object image matching using deformable object image discrimination. *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 1–4. IEEE.

Feris, R., Datta, A., Pankanti, S., & Sun, M. (2013). Boosting object detection performance in crowded surveillance videos. *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, 427–432. https://doi.org/10.1109/WACV.2013.6475050

Fernández, C., Baiget, P., Roca, F. X., & Gonzàlez, J. (2011). Determining the best suited semantic events for cognitive surveillance. *Expert Systems with Applications*, *38*(4), 4068–4079. https://doi.org/10.1016/j.eswa.2010.09.070

Fernández, Carles, Baiget, P., Roca, X., & Gonzàlez, J. (2008). Interpretation of complex situations in a semantic-based surveillance framework. *Signal Processing: Image Communication*, *23*(7), 554–569. https://doi.org/10.1016/j.image.2008.04.015

Fernández Tena, C. (2010). *Understanding image sequences the role of ontologies in cognitive vision: a dissertation submitted by Carles Fernández Tena at Universitat Autònoma de Barcelona to fulfil the degree of Doctor en Informàtica, Bellaterra, april 2010.* Centre de Visió per Computador, Sardañola del Vallés.

Ferrando, S., Gera, G., Massa, M., & Regazzoni, C. (2006). A New Method for Real Time Abandoned Object Detection and Owner Tracking. *2006 International Conference on Image Processing*, 3329–3332. https://doi.org/10.1109/ICIP.2006.313137

Fortmann, T., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, *8*(3), 173–184. https://doi.org/10.1109/JOE.1983.1145560

Fragkiadaki, K., Zhang, W., Zhang, G., & Shi, J. (2012a). Two-Granularity Tracking: Mediating Trajectory and Detection Graphs for Tracking under Occlusions. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer Vision – ECCV 2012* (Vol. 7576, pp. 552–565). https://doi.org/10.1007/978-3-642-33715-4_40

Fragkiadaki, K., Zhang, W., Zhang, G., & Shi, J. (2012b). Two-Granularity Tracking: Mediating Trajectory and Detection Graphs for Tracking under Occlusions. https://doi.org/10.1007/978-3-642-33715-4_40

Francois, A. R. J., Nevatia, R., Hobbs, J., Bolles, R. C., & Smith, J. R. (2005). VERL: an ontology framework for representing and annotating video events. *IEEE MultiMedia*, *12*(4), 76–86. https://doi.org/10.1109/MMUL.2005.87

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD : Deconvolutional Single Shot Detector. *ArXiv:1701.06659 [Cs]*. Retrieved from http://arxiv.org/abs/1701.06659

Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. (2016). VirtualWorlds as Proxy for Multi-object Tracking Analysis. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4340–4349. https://doi.org/10.1109/CVPR.2016.470

Galteri, L., Seidenari, L., Bertini, M., & Bimbo, A. D. (2017). Spatio-Temporal Closed-Loop Object Detection. *IEEE Transactions on Image Processing*, *26*(3), 1253–1263. https://doi.org/10.1109/TIP.2017.2651367

Gandhamal, A., & Talbar, S. (2015). Evaluation of background subtraction algorithms for object extraction. *2015 International Conference on Pervasive Computing (ICPC)*, 1–6. https://doi.org/10.1109/PERVASIVE.2015.7087065

Gerber, R., Nagel, H.-H., & Schreiber, H. (2002). *Deriving Textual Descriptions of Road Traffic Queues from Video Sequences*. 5.

Gill, M., Spriggs, A., Allen, J., Hemming, M., Jessiman, P., Kara, D., … Swain, D. (2005). Control room operation: findings from control room observations. *Home Office Online Report*, *14*(05).

Girshick, R. (2015). Fast R-CNN. *ArXiv:1504.08083 [Cs]*. Retrieved from http://arxiv.org/abs/1504.08083

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. 580–587. Retrieved from http://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as Space-Time Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253. https://doi.org/10.1109/TPAMI.2007.70711

Graham, Y., Awad, G., & Smeaton, A. (2018). Evaluation of automatic video captioning using direct assessment. *PloS One*, *13*(9), e0202789.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., & Saenko, K. (2013). Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 2712–2719.

Gupta, A., & Davis, L. S. (2007). Objects in Action: An Approach for Combining Action Understanding and Object Perception. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/CVPR.2007.383331

Gupta, Abhinav, & Davis, L. S. (2007). *Objects in Action: An Approach for Combining Action Understanding and Object Perception*. https://doi.org/10.1184/R1/6557027.v1

Haas, L. (1995). Reprints available directly from the publisher Photocopying permitted by license only. *Review of Education, Pedagogy, and Cultural Studies*, *17*(1), 1–6. https://doi.org/10.1080/1071441950170102

Han, W., Khorrami, P., Paine, T. L., Ramachandran, P., Babaeizadeh, M., Shi, H., … Huang, T. S. (2016). Seq-NMS for Video Object Detection. *ArXiv:1602.08465 [Cs]*. Retrieved from http://arxiv.org/abs/1602.08465

Hanckmann, P., Schutte, K., & Burghouts, G. J. (2012). Automated Textual Descriptions for a Wide Range of Video Events with 48 Human Actions. In A. Fusiello, V. Murino, & R. Cucchiara (Eds.), *Computer Vision – ECCV 2012. Workshops and Demonstrations* (pp. 372–380). Springer Berlin Heidelberg.

Harasse, S., Bonnaud, L., & Desvignes, M. (2006). Human model for people detection in dynamic scenes. *18th International Conference on Pattern Recognition (ICPR'06)*, *1*,

335–354. https://doi.org/10.1109/ICPR.2006.638

Haritaoglu, S., Harwood, D., & Davis, L. S. (2000). *W4: Real-time surveillance of people and their activities.* 809–830. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8).

Hartley, R. I. (1997). In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 19, No. 6*, 580–593.

Hartley, R., & Zisserman, A. (2003a). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hartley, R., & Zisserman, A. (2003b). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *ArXiv:1703.06870 [Cs]*. Retrieved from http://arxiv.org/abs/1703.06870

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. 770–778. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

Heel, J. (1990). *Direct Estimation of Structure and Motion from Multiple Frames* (No. AI-M-1190). Retrieved from MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB website: https://apps.dtic.mil/docs/citations/ADA223903

Herath, S., Harandi, M., & Porikli, F. (2017). Going deeper into action recognition: A survey. *Image and Vision Computing*, *60*, 4–21. https://doi.org/10.1016/j.imavis.2017.01.010

Herbst, E., Ren, X., & Fox, D. (2013). RGB-D flow: Dense 3-D motion estimation using color and depth. *2013 IEEE International Conference on Robotics and Automation*, 2276–2282. https://doi.org/10.1109/ICRA.2013.6630885

Herrera, F., Herrera-Viedma, E., & Martínez, L. (2000). A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, *114*(1), 43–58. https://doi.org/10.1016/S0165-0114(98)00093-1

Herrera, F., & Martinez, L. (2001). A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *31*(2), 227–234. https://doi.org/10.1109/3477.915345

Hervieu, A., Bouthemy, P., & Cadre, J.-P. L. (2008). Video event classification and detection using 2D trajectories. *Proc. of the Int. Conf. on Computer Vision Theory and Applications, Visapp'08*.

Hogervorst, M. A., Kappers, A. M. L., & Koenderink, J. J. (1997). Monocular discrimination of

rigidly and nonrigidly moving objects. *Perception & Psychophysics*, *59*(8), 1266–1279. https://doi.org/10.3758/BF03214213

Hongeng, S., Nevatia, R., & Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, *96*(2), 129–162. https://doi.org/10.1016/j.cviu.2004.02.005

Horn, B. K. P., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, *17*(1), 185–203. https://doi.org/10.1016/0004-3702(81)90024-2

Hou, Y., & Pang, G. K. H. (2011). People Counting and Human Detection in a Challenging Situation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *41*(1), 24–33. https://doi.org/10.1109/TSMCA.2010.2064299

Hsieh, K., Ananthanarayanan, G., Bodik, P., Venkataraman, S., Bahl, P., Philipose, M., … Mutlu, O. (2018). *Focus: Querying Large Video Datasets with Low Latency and Low Cost*. 19.

Hu, C., Xu, Z., Liu, Y., & Mei, L. (2015). Video structural description technology for the new generation video surveillance systems. *Frontiers of Computer Science*, *9*(6), 980–989.

Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *34*(3), 334–352. https://doi.org/10.1109/TSMCC.2004.829274

Hu, W., Xie, D., Fu, Z., Zeng, W., & Maybank, S. (2007). Semantic-Based Surveillance Video Retrieval. *IEEE Transactions on Image Processing*, *16*(4), 1168–1181. https://doi.org/10.1109/TIP.2006.891352

Hu, Y.-T., Huang, J.-B., & Schwing, A. (2017). *MaskRNN: Instance Level Video Object Segmentation*. 10.

Hua, X.-S., Lu, L., & Zhang, H.-J. (2004). Automatic music video generation based on temporal pattern analysis. *Proceedings of the 12th Annual ACM International Conference on Multimedia*, 472–475. ACM.

Huang, T. S., & Netravali, A. N. (2009). *Motion and Structure from Feature Correspondences : A Review*.

ICDP conf. (2017, October 5). 8th IET International Conference on Imaging for Crime Detection and Prevention. Retrieved April 9, 2019, from All Conference Alert website: https://www.allconferencealert.org/icdp-2017/

I-Lids Dataset for AVSS 2007. (2007). Retrieved from http://www.eecs.qmul.ac.uk/~andrea/avss2007_ss_challenge.html

Ivanov, Y., Stauffer, C., Bobick, A., & Grimson, W. E. L. (1999). Video surveillance of interactions. *Proceedings Second IEEE Workshop on Visual Surveillance (VS'99) (Cat.*

No.98-89223), 82–89. https://doi.org/10.1109/VS.1999.780272

Jaffré, G., & Joly, P. (2005). Improvement of a Temporal Video Index Produced by an Object Detector. In A. Gagalowicz & W. Philips (Eds.), *Computer Analysis of Images and Patterns* (Vol. 3691, pp. 472–479). https://doi.org/10.1007/11556121_58

Jain, R. (1991). DIALOGUE Ignorance, Myopia, and naivete in computer vision systems. *CVGIP: Image Understanding*, *53*, 112–117.

Jan, T. (2004). Neural network based threat assessment for automated visual surveillance. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, *2*, 1309–1312 vol.2. https://doi.org/10.1109/IJCNN.2004.1380133

Jiang, X., Rodner, E., & Denzler, J. (2012). Multi-person Tracking-by-Detection Based on Calibrated Multi-camera Systems. In L. Bolc, R. Tadeusiewicz, L. J. Chmielewski, & K. Wojciechowski (Eds.), *Computer Vision and Graphics* (Vol. 7594, pp. 743–751). https://doi.org/10.1007/978-3-642-33564-8_89

Jiangung Lou, Qifeng Liu, Tieniu Tan, & Weiming Hu. (2002). Semantic interpretation of object activities in a surveillance system. *Object Recognition Supported by User Interaction for Service Robots*, *3*, 777–780. https://doi.org/10.1109/ICPR.2002.1048115

Jodoin, Jean-Philippe, Bilodeau, G.-A., & Saunier, N. (2014). Urban Tracker: Multiple object tracking in urban mixed traffic. *IEEE Winter Conference on Applications of Computer Vision*, 885–892. https://doi.org/10.1109/WACV.2014.6836010

Jodoin, J.-P., Bilodeau, G.-A., & Saunier, N. (2013). Urban Tracker: Suivi multiobjets en milieu urbain. Retrieved April 9, 2019, from https://www.jpjodoin.com/urbantracker/

Kang, D., Emmons, J., Abuzaid, F., Bailis, P., & Zaharia, M. (2017). *NoScope: Optimizing Neural Network Queries over Video at Scale*. Retrieved from https://arxiv.org/abs/1703.02529v3

Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., & Wang, X. (2017). *Object Detection in Videos With Tubelet Proposal Networks*. 727–735. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Kang_Object_Detection_in_CVPR_2017_paper.html

Kaptelinin, V. (2013). Activity Theory. In *The Encyclopedia of Human-Computer Interaction* (2nd Ed.). Retrieved from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/activity-theory

Kataoka, H., Aoki, Y., Iwata, K., Satoh, Y., Yoda, I., & Onishi, M. (2013). *Big Trajectory Data Analysis for Clustering and Anomaly Detection*. 4.

Kazi Tani, M. Y., Lablack, A., Ghomari, A., & Bilasco, I. M. (2015). Events Detection Using a Video-Surveillance Ontology and a Rule-Based Approach. In L. Agapito, M. M. Bronstein, & C. Rother (Eds.), *Computer Vision - ECCV 2014 Workshops* (pp. 299–

308). Springer International Publishing.

Ke, Y., Sukthankar, R., & Hebert, M. (2007). Event Detection in Crowded Videos. *2007 IEEE 11th International Conference on Computer Vision*, 1–8. https://doi.org/10.1109/ICCV.2007.4409011

Khammar, M. (2012). Evaluation of different block matching algorithms to motion estimation. *International Journal of V LSI and Embedded Systems-IJVES, ISSN: 2249 – 6556*, 148–153.

Khan, M. U. G., Lei Zhang, & Gotoh, Y. (2011). Towards coherent natural language description of video streams. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 664–671. https://doi.org/10.1109/ICCVW.2011.6130306

Khoreva, A., Benenson, R., Ilg, E., Brox, T., & Schiele, B. (2017). *Lucid Data Dreaming for Object Tracking*. 6.

Klappstein, J., Vaudrey, T., Rabe, C., Wedel, A., & Klette, R. (2009). Moving Object Segmentation Using Optical Flow and Depth Information. In T. Wada, F. Huang, & S. Lin (Eds.), *Advances in Image and Video Technology* (pp. 611–623). Springer Berlin Heidelberg.

Ko, T. (2008). A survey on behavior analysis in video surveillance for homeland security applications. *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, 1–8. https://doi.org/10.1109/AIPR.2008.4906450

Ko, Teddy. (2011). A Survey on Behaviour Analysis in Video Surveillance Applications. *Video Surveillance*, (Prof. Weiyao Lin (Ed.)), 17.

Kojima, A., Izumi, M., Tamura, T., & Fukunaga, K. (2000). Generating natural language description of human behavior from video images. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, *4*, 728–731 vol.4. https://doi.org/10.1109/ICPR.2000.903020

Kojima, A., Tamura, T., & Fukunaga, K. (2002). Textual description of human activities by tracking head and hand motions. *Object Recognition Supported by User Interaction for Service Robots*, *2*, 1073–1077 vol.2. https://doi.org/10.1109/ICPR.2002.1048491

Kojima, Atsuhiro, Tamura, T., & Fukunaga, K. (2002). Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy of Actions. *International Journal of Computer Vision*, *50*(2), 171–184. https://doi.org/10.1023/A:1020346032608

Korpipää, P., Jani, M., Kela, J., Malm, E.-J., & others. (2003). Managing context information in mobile devices. *IEEE Pervasive Computing*, (3), 42–51.

Kovesi, P. (n.d.). MATLAB and Octave Functions for Computer Vision and Image Processing. Retrieved from https://www.peterkovesi.com/matlabfns/index.html

Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Saenko, K., & Guadarrama, S. (2013). Generating natural-language video descriptions using text-mined knowledge. *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Kumar, P., & Mittal, A. (2007). Study of robust and intelligent surveillance in visible and multi-modal framework. *Informatica*, (31), 4.

Kuno, Y., Watanabe, T., Shimosakoda, Y., & Nakagawa, S. (1996). Automated detection of human for visual surveillance system. *Proceedings of 13th International Conference on Pattern Recognition*, *3*, 865–869 vol.3. https://doi.org/10.1109/ICPR.1996.547291

Kunt, M. (1991). Comments on "Dialogue," a series of articles generated by the paper entitled "Ignorance, Myopia, and Naiveté in Computer Vision." *CVGIP: Image Understanding*, *54*(3), 428–429.

Lavee, G., Rivlin, E., & Rudzsky, M. (2009). Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *39*(5), 489–504.

Lee, Y. J., Kim, J., & Grauman, K. (2011). Key-segments for video object segmentation. *2011 International Conference on Computer Vision*, 1995–2002. https://doi.org/10.1109/ICCV.2011.6126471

Li, H., Li, Y., & Porikli, F. (2016). DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking. *IEEE Transactions on Image Processing*, *25*(4), 1834–1848. https://doi.org/10.1109/TIP.2015.2510583

Li, H., Tang, J., Wu, S., Zhang, Y., & Lin, S. (2010). Automatic Detection and Analysis of Player Action in Moving Background Sports Video Sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, *20*(3), 351–364. https://doi.org/10.1109/TCSVT.2009.2035833

Li, P., Wang, D., Wang, L., & Lu, H. (2018). Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, *76*, 323–338. https://doi.org/10.1016/j.patcog.2017.11.007

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81. Retrieved from https://www.aclweb.org/anthology/W04-1013

Lipton, A. J., Fujiyoshi, H., & Patil, R. S. (1998). Moving target classification and tracking from real-time video. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*, 8–14.

https://doi.org/10.1109/ACV.1998.732851

Lipton, Alan J. (1999). *Local Application of Optic Flow to Analyse Rigid versus Non-Rigid Motion*.

Lipton, Alan J, & others. (1999). *Local application of optic flow to analyse rigid versus non-rigid motion*. Carnegie Mellon University, The Robotics Institute.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21–37). Springer International Publishing.

Long, X., Gan, C., & de Melo, G. (2018). Video Captioning with Multi-Faceted Attention. *Transactions of the Association for Computational Linguistics*, *6*, 173–184. https://doi.org/10.1162/tacl_a_00013

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, *293 (5828)*, 133–135.

Love, N. S., & Kamath, C. (2006). *An Empirical Study of Block Matching Techniques for the Detection of Moving Objects* (No. UCRL-TR-218038). https://doi.org/10.2172/898460

Loy, C. C. (2010). *Activity understanding and unusual event detection in surveillance videos* (Thesis). Retrieved from https://qmro.qmul.ac.uk/xmlui/handle/123456789/664

Lucas, B. D., Kanade, T., & others. (1981). *An iterative image registration technique with an application to stereo vision*.

Luong, Quang-Tuan, Deriche, R., Faugeras, O., & Papadopoulo, T. (1993). *On determining the fundamental matrix : analysis of different methods and experimental results* [Report]. Retrieved from INRIA website: https://hal.inria.fr/inria-00074777/document

Luong, Quan-Tuan, & Faugeras, O. D. (1996). The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, *17*(1), 43–75. https://doi.org/10.1007/BF00127818

Ly, N. Q., Truong, A. M., & Nguyen, H. V. (2016). Specific Behavior Recognition Based on Behavior Ontology. In D. Król, L. Madeyski, & N. T. Nguyen (Eds.), *Recent Developments in Intelligent Information and Database Systems* (pp. 99–109). https://doi.org/10.1007/978-3-319-31277-4_9

Makris, D., & Ellis, T. (2005). Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(3), 397–408. https://doi.org/10.1109/TSMCB.2005.846652

Makris, Dimitrios, & Ellis, T. (2002). Path detection in video surveillance. *Image and Vision Computing*, *20*(12), 895–903. https://doi.org/10.1016/S0262-8856(02)00098-7

Maninis, K.-K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L.

(2017). Video Object Segmentation Without Temporal Information. *ArXiv:1709.06031 [Cs]*. Retrieved from http://arxiv.org/abs/1709.06031

Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., & Van Gool, L. (2016). Convolutional Oriented Boundaries. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 580–596). Springer International Publishing.

Mann, R., Jepson, A., & Siskind, J. M. (1996). Computational perception of scene dynamics. In B. Buxton & R. Cipolla (Eds.), *Computer Vision — ECCV '96* (pp. 528–539). Springer Berlin Heidelberg.

Marzat, J. (2008). *Estimation temps réel du Flot Optique*. in Rapport de Stage Ingénieur, Institut National de Recherche en Informatique et Automatique.

McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., & Wechsler, H. (2000). Tracking Groups of People. *Computer Vision and Image Understanding*, *80*(1), 42–56. https://doi.org/10.1006/cviu.2000.0870

Meinhardt-Llopis, E., Sánchez Pérez, J., & Kondermann, D. (2013). Horn-Schunck Optical Flow with a Multi-Scale Strategy. *Image Processing On Line*, *3*, 151–172. https://doi.org/10.5201/ipol.2013.20

Meyer, D., Denzler, J., & Niemann, H. (1997). Model based extraction of articulated objects in image sequences for gait analysis. *Proceedings of International Conference on Image Processing*, *3*, 78–81 vol.3. https://doi.org/10.1109/ICIP.1997.631988

Meyer, Dorthe, Pösl, J., & Niemann, H. (1998). *Gait Classification with HMMs for Trajectories of Body Parts Extracted by Mixture Densities*.

Michael, B. J. (1992). *Robust incremental optical flow*. PhD diss., PhD thesis, Yale university.

Milan, A. (2012). Discrete-Continuous Energy Minimization for Multi-Target Tracking. Retrieved April 9, 2019, from http://research.milanton.de/dctracking/index.html

Milan, A. (2014). Continuous Energy Minimization for Multi-Target Tracking. Retrieved from http://research.milanton.de/contracking/

Milan, A., Rezatofighi, S. H., Dick, A., Reid, I., & Schindler, K. (2017). Online Multi-Target Tracking Using Recurrent Neural Networks. *Thirty-First AAAI Conference on Artificial Intelligence*. Presented at the Thirty-First AAAI Conference on Artificial Intelligence. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14184

Milan, A., Roth, S., & Schindler, K. (2014). Continuous Energy Minimization for Multitarget Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(1), 58–72. https://doi.org/10.1109/TPAMI.2013.103

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2), 90–

126. https://doi.org/10.1016/j.cviu.2006.08.002

Moore, D. J., Essa, I. A., & Hayes, M. H. (1999). Exploiting human actions and object context for recognition tasks. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, *1*, 80–86 vol.1. https://doi.org/10.1109/ICCV.1999.791201

Morris, B. T., & Trivedi, M. M. (2008). A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(8), 1114–1127. https://doi.org/10.1109/TCSVT.2008.927109

Motion-Based Multiple Object Tracking - MATLAB & Simulink. (n.d.). Retrieved April 9, 2019, from https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html

Muller-Schneiders, S., Jager, T., Loos, H. S., & Niem, W. (2005). Performance evaluation of a real time video surveillance system. *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 137–143. IEEE.

Multi-camera pedestrians video – CVLAB. (n.d.). Retrieved March 26, 2019, from https://cvlab.epfl.ch/data/data-pom-index-php/

Multiple Object Tracking Tutorial - MATLAB & Simulink. (n.d.). Retrieved April 9, 2019, from https://www.mathworks.com/help/driving/examples/multiple-object-tracking-tutorial.html

Naeem, U., & Bigham, J. (2007). A comparison of two hidden markov approaches to task identification in the home environment. *2007 2nd International Conference on Pervasive Computing and Applications*, 383–388. IEEE.

Nam, H., & Han, B. (2016). *Learning Multi-Domain Convolutional Neural Networks for Visual Tracking*. 4293–4302. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Nam_Learning_Multi-Domain_Convolutional_CVPR_2016_paper.html

Nevatia, R., Zhao, T., & Hongeng, S. (2003). Hierarchical Language-based Representation of Events in Video Streams. *2003 Conference on Computer Vision and Pattern Recognition Workshop*, *4*, 39–39. https://doi.org/10.1109/CVPRW.2003.10038

Neves, J., Narducci, F., Barra, S., & Proença, H. (2016). Biometric recognition in surveillance scenarios: a survey. *Artificial Intelligence Review*, *46*(4), 515–541. https://doi.org/10.1007/s10462-016-9474-x

Nghiem, A. T., Bremond, F., Thonnat, M., & Valentin, V. (2007). ETISEO, performance evaluation for video surveillance systems. *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, 476–481. https://doi.org/10.1109/AVSS.2007.4425357

Nishida, F., & Takamatsu, S. (1982). Japanese-English Translation Through Internal

Expressions. *Proceedings of the 9th Conference on Computational Linguistics - Volume 1*, 271–276. https://doi.org/10.3115/991813.991856

Nishida, F., Takamatsu, S., Tani, T., & Doi, T. (1988). Feedback of Correcting Information in Postediting to a Machine Translation System. *Proceedings of the 12th Conference on Computational Linguistics - Volume 2*, 476–481. https://doi.org/10.3115/991719.991737

Ochs, P., & Brox, T. (2011). Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. *2011 International Conference on Computer Vision*, 1583–1590. https://doi.org/10.1109/ICCV.2011.6126418

Ochs, P., & Brox, T. (2012). Higher order motion models and spectral clustering. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 614–621. https://doi.org/10.1109/CVPR.2012.6247728

Ochs, P., Malik, J., & Brox, T. (2014). Segmentation of Moving Objects by Long Term Video Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(6), 1187–1200. https://doi.org/10.1109/TPAMI.2013.242

Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C., Lee, J. T., … Desai, M. (2011). A large-scale benchmark dataset for event recognition in surveillance video. *CVPR 2011*, 3153–3160. https://doi.org/10.1109/CVPR.2011.5995586

Oliver, N. M., Rosario, B., & Pentland, A. P. (2000). A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 831–843. https://doi.org/10.1109/34.868684

Oltramari, A., & Lebiere, C. (2012). *Using Ontologies in a Cognitive-Grounded System: Automatic Action Recognition in Video Surveillance*. 8.

OWL Web Ontology Language Overview. (2004). Retrieved from https://www.w3.org/TR/owl-features/

Pan, Y., Mei, T., Yao, T., Li, H., & Rui, Y. (2016). *Jointly Modeling Embedding and Translation to Bridge Video and Language*. 4594–4602. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/Pan_Jointly_Modeling_Embedding_CVPR_2016_paper.html

Pan, Y., Yao, T., Li, H., & Mei, T. (2017). *Video Captioning With Transferred Semantic Attributes*. 6504–6512. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Pan_Video_Captioning_With_CVPR_2017_paper.html

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. S. (2007). Human Computing and Machine Understanding of Human Behavior: A Survey. In T. S. Huang, A. Nijholt, M. Pantic, & A. Pentland (Eds.), *Artifical Intelligence for Human Computing* (pp. 47–71). Springer Berlin Heidelberg.

Papazoglou, A., & Ferrari, V. (2013). *Fast Object Segmentation in Unconstrained Video*. 1777–1784. Retrieved from https://www.cv-foundation.org/openaccess/content_iccv_2013/html/Papazoglou_Fast_Object_Segmentation_2013_ICCV_paper.html

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Park, S., & Aggarwal, J. K. (2004). Event semantics in two-person interactions. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, *4*, 227-230 Vol.4. https://doi.org/10.1109/ICPR.2004.1333745

Park, S., & Trivedi, M. M. (2007). Homography-based Analysis of People and Vehicle Activities in Crowded Scenes. *2007 IEEE Workshop on Applications of Computer Vision (WACV '07)*, 51–51. https://doi.org/10.1109/WACV.2007.30

Park, Sangho, & Aggarwal, J. K. (2003). Recognition of Two-person Interactions Using a Hierarchical Bayesian Network. *First ACM SIGMM International Workshop on Video Surveillance*, 65–76. https://doi.org/10.1145/982452.982461

Park, Sangho, & Aggarwal, J. K. (2006). Simultaneous tracking of multiple body parts of interacting persons. *Computer Vision and Image Understanding*, *102*(1), 1–21. https://doi.org/10.1016/j.cviu.2005.07.011

Park, Sangho, & Trivedi, M. M. (2007). Multi-person interaction and activity analysis: a synergistic track- and body-level analysis framework. *Machine Vision and Applications*, *18*(3–4), 151–166. https://doi.org/10.1007/s00138-006-0055-x

Patel, B., Kshirsagar, R. V., & Nitnaware, V. (2013). *Review and comparative study of motion estimation techniques to reduce complexity in video compression*. *2*(8), 11.

Pathak, A. R., Pandey, M., Rautaray, S., & Pawar, K. (2018). Assessment of Object Detection Using Deep Convolutional Neural Networks. In S. Bhalla, V. Bhateja, A. A. Chandavale, A. S. Hiwale, & S. C. Satapathy (Eds.), *Intelligent Computing and Information and Communication* (pp. 457–466). Springer Singapore.

Pattern recognition network - MATLAB patternnet. (n.d.). Retrieved April 9, 2019, from https://www.mathworks.com/help/deeplearning/ref/patternnet.html

Pavlidis, T. (1992). Why progress in machine vision is so slow. *Pattern Recognition Letters*, *13*(4), 221–225.

Pellegrini, S., Ess, A., Schindler, K., & Gool, L. van. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. *2009 IEEE 12th International Conference on Computer Vision*, 261–268. https://doi.org/10.1109/ICCV.2009.5459260

Pellegrini, Stefano, Ess, A., & Van Gool, L. (2010). Improving Data Association by Joint

Modeling of Pedestrian Trajectories and Groupings. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (pp. 452–465). Springer Berlin Heidelberg.

Pesquet-Popescu, B., Cagnazzo, M., & Dufaux, F. (2013). *Motion Estimation Techniques*.

Peursum, P., West, G., & Venkatesh, S. (2005). Combining image regions and human activity for indirect object recognition in indoor wide-angle views. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, *1*, 82-89 Vol. 1. https://doi.org/10.1109/ICCV.2005.57

Pham, H.-H., Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S. A. (2019). Learning to Recognize 3D Human Action from A New Skeleton-based Representation Using Deep Convolutional Neural Networks. *IET Computer Vision*, *13*(3), 319–328. https://doi.org/10.1049/iet-cvi.2018.5014

Piccardi, M. (2004). Background subtraction techniques: a review. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, *4*, 3099–3104 vol.4. https://doi.org/10.1109/ICSMC.2004.1400815

Piciarelli, C., Micheloni, C., & Foresti, G. L. (2008). Trajectory-Based Anomalous Event Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1544–1554. https://doi.org/10.1109/TCSVT.2008.2005599

Pirsiavash, H. (n.d.). Hamed Pirsiavash Website. Retrieved April 9, 2019, from https://www.csee.umbc.edu/~hpirsiav/

Pirsiavash, H., Ramanan, D., & Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. *CVPR 2011*, 1201–1208. https://doi.org/10.1109/CVPR.2011.5995604

Possegger, H., Mauthner, T., Roth, P. M., & Bischof, H. (2014). Occlusion Geodesics for Online Multi-object Tracking. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1306–1313. https://doi.org/10.1109/CVPR.2014.170

Prest, A., Leistner, C., Civera, J., Schmid, C., & Ferrari, V. (2012). Learning object class detectors from weakly annotated video. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3282–3289. https://doi.org/10.1109/CVPR.2012.6248065

Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, *12*(8), 1550147716665520. https://doi.org/10.1177/1550147716665520

Rashidi, P., & Cook, D. J. (2009). Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, *39*(5), 949–959.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. 779–788. Retrieved from https://www.cv-

foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

Redmon, J., & Farhadi, A. (2017). *YOLO9000: Better, Faster, Stronger*. 7263–7271. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.html

Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, *24*(6), 843–854. https://doi.org/10.1109/TAC.1979.1102177

Remagnino, P., Foresti, G. L., & Ellis, T. (2005). *Ambient intelligence: a novel paradigm*. Springer.

Remagnino, P., Tan, T., & Baker, K. (1998a). Agent orientated annotation in model based visual surveillance. *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, 857–862. IEEE.

Remagnino, P., Tan, T., & Baker, K. (1998b). Multi-agent visual surveillance of dynamic scenes. *Image and Vision Computing*, *16*(8), 529–532.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv:1506.01497 [Cs]*. Retrieved from http://arxiv.org/abs/1506.01497

Resource Description Framework (RDF): Concepts and Abstract Syntax. (2004). Retrieved from https://www.w3.org/TR/rdf-concepts/

Revathi, A. R., & Kumar, D. (2012). *A SURVEY OF ACTIVITY RECOGNITION AND UNDERSTANDING THE BEHAVIOUR IN VIDEO SURVEILLANCE*. 14.

Rodríguez, N. D., Cuéllar, M. P., Lilius, J., & Calvo-Flores, M. D. (2014). A Survey on Ontologies for Human Behavior Recognition. *ACM Comput. Surv.*, *46*(4), 43:1–43:33. https://doi.org/10.1145/2523819

Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., & Schiele, B. (2014). Coherent Multi-sentence Video Description with Variable Level of Detail. In Xiaoyi Jiang, J. Hornegger, & R. Koch (Eds.), *Pattern Recognition* (pp. 184–195). Springer International Publishing.

Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., … Schiele, B. (2017). Movie description. *International Journal of Computer Vision*, *123*(1), 94–120.

Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., & Schiele, B. (2013). Translating video content to natural language descriptions. *Proceedings of the IEEE International Conference on Computer Vision*, 433–440.

Ros, M., Cuéllar, M. P., Delgado, M., & Vila, A. (2013). Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal

windows. *Information Sciences*, *220*, 86–101.

Roshan Zamir, A., Dehghan, A., & Shah, M. (2012). GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. Retrieved from Computer Vision – ECCV 2012 website: http://crcv.ucf.edu/projects/GMCP-Tracker/

Ryoo, M. S., & Aggarwal, J. K. (2007a). Hierarchical Recognition of Human Activities Interacting with Objects. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/CVPR.2007.383487

Ryoo, M. S., & Aggarwal, J. K. (2007b). Hierarchical Recognition of Human Activities Interacting with Objects. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/CVPR.2007.383487

Ryoo, M. S., Chen, C.-C., Aggarwal, J. K., & Roy-Chowdhury, A. (2010). An Overview of Contest on Semantic Description of Human Activities (SDHA) 2010. In D. Ünay, Z. Çataltepe, & S. Aksoy (Eds.), *Recognizing Patterns in Signals, Speech, Images and Videos* (pp. 270–285). Retrieved from http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html

SanMiguel, J. C., Martinez, J. M., & Garcia, Á. (2009). An Ontology for Event Detection and its Application in Surveillance Video. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 220–225. https://doi.org/10.1109/AVSS.2009.28

Sargano, A., Angelov, P., & Habib, Z. (2017). A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition. *Applied Sciences*, *7*(1), 110. https://doi.org/10.3390/app7010110

Sasikanth, & Kroon, D.-J. (2010). 2D - 2D Projective Homography (3x3) Estimation - File Exchange - MATLAB Central. Retrieved April 9, 2019, from https://www.mathworks.com/matlabcentral/fileexchange/28760

Sikora, T. (1997). The MPEG-4 video standard verification model. *IEEE Transactions on Circuits and Systems for Video Technology*, *7*(1), 19–31. https://doi.org/10.1109/76.554415

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

Singla, G., Cook, D. J., & Schmitter-Edgecombe, M. (2010). Recognizing independent and joint activities among multiple residents in smart environments. *Journal of Ambient Intelligence and Humanized Computing*, *1*(1), 57–63. https://doi.org/10.1007/s12652-009-0007-1

Snoek, C. G. M., & Worring, M. (2003). Time interval maximum entropy based event indexing in soccer video. *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, *3*, III–481.

https://doi.org/10.1109/ICME.2003.1221353

Son, K. D. (2019). *Contribute to son-oh-yeah/Moving-Target-Tracking-with-OpenCV development by creating an account on GitHub* [Java]. Retrieved from https://github.com/son-oh-yeah/Moving-Target-Tracking-with-OpenCV (Original work published 2015)

Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., & Shen, H. T. (2017). From Deterministic to Generative: Multi-Modal Stochastic RNNs for Video Captioning. *ArXiv:1708.02478 [Cs]*. Retrieved from http://arxiv.org/abs/1708.02478

SPEVI: Audiovisual people dataset, Courtesy of EPSRC funded MOTINAS project (EP/D033772/1). (n.d.). Retrieved from ftp://motinas.elec.qmul.ac.uk/pub/av_people

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, *2*, 246-252 Vol. 2. https://doi.org/10.1109/CVPR.1999.784637

Stoll, M., Volz, S., & Bruhn, A. (2013). Adaptive Integration of Feature Matches into Variational Optical Flow Methods. In K. M. Lee, Y. Matsushita, J. M. Rehg, & Z. Hu (Eds.), *Computer Vision – ACCV 2012* (Vol. 7726, pp. 1–14). https://doi.org/10.1007/978-3-642-37431-9_1

Szczodrak, M., Kotus, J., Kopaczewski, K., Lopatka, K., Czyzewski, A., & Krawczyk, H. (2011). Behavior Analysis and Dynamic Crowd Management in Video Surveillance System. *2011 22nd International Workshop on Database and Expert Systems Applications*, 371–375. https://doi.org/10.1109/DEXA.2011.16

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). *Going Deeper With Convolutions*. 1–9. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

Szeliski, R. (2011). Feature detection and matching. In R. Szeliski (Ed.), *Computer Vision: Algorithms and Applications* (pp. 181–234). https://doi.org/10.1007/978-1-84882-935-0_4

Taha, A., H. Zayed, H., E. Khalifa, M., & M. El-Horbaty, E.-S. (2014). Exploring Behavior Analysis in Video Surveillance Applications. *International Journal of Computer Applications*, *93*(14), 22–32. https://doi.org/10.5120/16283-6045

Taj, M., & Cavallaro, A. (2010). Recognizing Interactions in Video. In H. T. Sencar, S. Velastin, N. Nikolaidis, & S. Lian (Eds.), *Intelligent Multimedia Analysis for Security Applications* (pp. 29–57). https://doi.org/10.1007/978-3-642-11756-5_2

Tekalp, M. A. (1995). *Digital Video Processing*. Prentice Hall PTR.

Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., & Mooney, R. (2014). Integrating language and vision to generate natural language descriptions of videos in the wild. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 1218–1227.

Thonnat, M., & Rota, N. (1999). Image understanding for visual surveillance applications. *Proc. 3rd Int. Workshop on Cooperative Distributed Vision, Nov. 1999*. Retrieved from https://ci.nii.ac.jp/naid/10020720477/

Torr, P. H. S., & Zisserman, A. (2000). Feature Based Methods for Structure and Motion Estimation. In B. Triggs, A. Zisserman, & R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice* (pp. 278–294). Springer Berlin Heidelberg.

Tracking Interacting Objects – CVLAB. (n.d.). Retrieved March 26, 2019, from https://cvlab.epfl.ch/research/research-surv/trackinteractobj/

Tsai, R. Y., & Huang, T. S. (1984). Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*(1), 13–27. https://doi.org/10.1109/TPAMI.1984.4767471

Tu, K., Meng, M., Lee, M. W., Choe, T. E., & Zhu, S.-C. (2014). Joint video and text parsing for understanding events and answering queries. *IEEE MultiMedia*, *21*(2), 42–70.

Varadarajan, J., & Odobez, J. (2009). Topic models for scene analysis and abnormality detection. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 1338–1345. https://doi.org/10.1109/ICCVW.2009.5457456

Velastin, S. A. (2009). CCTV Video Analytics: Recent Advances and Limitations. In H. Badioze Zaman, P. Robinson, M. Petrou, P. Olivier, H. Schröder, & T. K. Shih (Eds.), *Visual Informatics: Bridging Research and Practice* (pp. 22–34). Springer Berlin Heidelberg.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., & Saenko, K. (2015). Sequence to Sequence -- Video to Text. *2015 IEEE International Conference on Computer Vision (ICCV)*, 4534–4542. https://doi.org/10.1109/ICCV.2015.515

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., & Saenko, K. (2014). Translating Videos to Natural Language Using Deep Recurrent Neural Networks. *ArXiv:1412.4729 [Cs]*. Retrieved from http://arxiv.org/abs/1412.4729

Verma, K. K., Kumar, P., & Tomar, A. (2015). Analysis of moving object detection and tracking in video surveillance system. *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1758–1762.

Vezzani, R., & Cucchiara, R. (2010). Video Surveillance Online Repository (ViSOR): An Integrated Framework. *Multimedia Tools Appl.*, *50*(2), 359–380. https://doi.org/10.1007/s11042-009-0402-9

Video Surveillance Online Repository. (2017). Retrieved April 7, 2019, from

http://www.openvisor.org/config_schema.asp

Villalonga, C., Razzaq, M. A., Khan, W. A., Pomares, H., Rojas, I., Lee, S., & Legran, O. B. (2016). Ontology-based high-level context inference for human behavior identification. *Sensors (Switserland)*, *16*(10), 1–26. https://doi.org/10.3390/s16101617

Vishwakarma, S., & Agrawal, A. (2013). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, *29*(10), 983–1009. https://doi.org/10.1007/s00371-012-0752-6

Vorobjov, D., Zakharava, I., Bohush, R., & Ablameyko, S. (2018). An Effective Object Detection Algorithm for High Resolution Video by Using Convolutional Neural Network. In T. Huang, J. Lv, C. Sun, & A. V. Tuzikov (Eds.), *Advances in Neural Networks – ISNN 2018* (Vol. 10878, pp. 503–510). https://doi.org/10.1007/978-3-319-92537-0_58

Vygotsky, L. S., & Cole, M. (1978). *Mind in Society*. Harvard University Press.

Wactlar, H., Christel, M., Kanade, T., Faloutsos, C., Lafferty, J., Hauptmann, A., & Yang, Y. (2001). *Informedia-ii: Auto-summarization and visualization over multiple video documents and libraries*.

Wang, B., Liang, J., Qian, Y., & Dang, C. (2015). A Normalized Numerical Scaling Method for the Unbalanced Multi-Granular Linguistic Sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. https://doi.org/10.1142/s0218488515500099

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (2013). Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision*, *103*(1), 60–79. https://doi.org/10.1007/s11263-012-0594-8

Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2014). Learning Actionlet Ensemble for 3D Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(5), 914–927. https://doi.org/10.1109/TPAMI.2013.198

Wang, K., Lin, L., Zuo, W., Gu, S., & Zhang, L. (2016). *Dictionary Pair Classifier Driven Convolutional Neural Networks for Object Detection*. 2138–2146. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/Wang_Dictionary_Pair_Classifier_CVPR_2016_paper.html

Wang, Liang, Hu, W., & Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, *36*(3), 585–601. https://doi.org/10.1016/S0031-3203(02)00100-0

Wang, Lijun, Ouyang, W., Wang, X., & Lu, H. (2015). *Visual Tracking With Fully Convolutional Networks*. 3119–3127. Retrieved from http://openaccess.thecvf.com/content_iccv_2015/html/Wang_Visual_Tracking_With_ICCV_2015_paper.html

Wang, T., & Collomosse, J. (2012). Probabilistic Motion Diffusion of Labeling Priors for Coherent Video Segmentation. *IEEE Transactions on Multimedia*, *14*(2), 389–400. https://doi.org/10.1109/TMM.2011.2177078

Wang, X., Türetken, E., Fleuret, F., & Fua, P. (2016). Tracking Interacting Objects Using Intertwined Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(11), 2312–2326. https://doi.org/10.1109/TPAMI.2015.2513406

Wang, Xiaogang, Tieu, K., & Grimson, E. (2006). Learning Semantic Scene Models by Trajectory Analysis. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision – ECCV 2006* (pp. 110–123). Springer Berlin Heidelberg.

Wedel, A., & Cremers, D. (2011). *Stereo Scene Flow for 3D Motion Analysis*. Springer Science & Business Media.

Wedel, A., Meißner, A., Rabe, C., Franke, U., & Cremers, D. (2009). Detection and Segmentation of Independently Moving Objects from Dense Scene Flow. In D. Cremers, Y. Boykov, A. Blake, & F. R. Schmidt (Eds.), *Energy Minimization Methods in Computer Vision and Pattern Recognition* (pp. 14–27). Springer Berlin Heidelberg.

Wilson, G., & Shpall, S. (2016). Action. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016). Retrieved from https://plato.stanford.edu/archives/win2016/entries/action/

Wixson, L., & Selinger, A. (1998). Classifying moving objects as rigid or non-rigid. *Proc. of DARPA Image Understanding Workshop*, 341–358.

Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., & Rehg, J. M. (2007). A Scalable Approach to Activity Recognition based on Object Use. *2007 IEEE 11th International Conference on Computer Vision*, 1–8. https://doi.org/10.1109/ICCV.2007.4408865

Wu, X., Li, G., Cao, Q., Ji, Q., & Lin, L. (2018). Interpretable Video Captioning via Trajectory Structured Localization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6829–6837. https://doi.org/10.1109/CVPR.2018.00714

Wu, Z., Yao, T., Fu, Y., & Jiang, Y.-G. (2017). Deep Learning for Video Classification and Captioning. *ArXiv:1609.06782 [Cs]*, 3–29. https://doi.org/10.1145/3122865.3122867

Xiang, T., & Gong, S. (2008). Video Behavior Profiling for Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(5), 893–908. https://doi.org/10.1109/TPAMI.2007.70731

Xiang, Y., Alahi, A., & Savarese, S. (2015a). Learning to Track: Online Multi-object Tracking by Decision Making. *2015 IEEE International Conference on Computer Vision (ICCV)*, 4705–4713. https://doi.org/10.1109/ICCV.2015.534

Xiang, Y., Alahi, A., & Savarese, S. (2015b, December). Learning to Track: Online Multi-object Tracking by Decision Making. Retrieved April 9, 2019, from 2015 IEEE International Conference on Computer Vision (ICCV) website:

http://cvgl.stanford.edu/projects/MDP_tracking/

Xu, J., Mei, T., Yao, T., & Rui, Y. (2016). *MSR-VTT: A Large Video Description Dataset for Bridging Video and Language*. 5288–5296. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/Xu_MSR-VTT_A_Large_CVPR_2016_paper.html

Xu, L., & Song, J. (2016). A Video Structural Event Description Model for Traffic Surveillance System. *Proceedings of the 2016 4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016)*. Presented at the 2016 4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016), Guilin, China. https://doi.org/10.2991/amitp-16.2016.69

Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., … Huang, T. (2018). YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (Vol. 11209, pp. 603–619). https://doi.org/10.1007/978-3-030-01228-1_36

Xu, R., Xiong, C., Chen, W., & Corso, J. J. (2015). *Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework*. 7.

Xu, Z., Hu, C., & Mei, L. (2016). Video structured description technology based intelligence analysis of surveillance videos for public security applications. *Multimedia Tools and Applications*, *75*(19), 12155–12172.

Xu, Z., Zhi, F., Liang, C., Lin, M., & Luo, X. (2014). Semantic annotation of traffic video resources. *2014 IEEE 13th International Conference on Cognitive Informatics and Cognitive Computing*, 323–328. IEEE.

Xue, M., Zheng, S., & Zhang, C. (2012). Ontology-based surveillance video archive and retrieval system. *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, 84–89. https://doi.org/10.1109/ICACI.2012.6463126

Yan, J., & Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. *European Conference on Computer Vision*, 94–106. Springer.

Yang, M.-T., Shih, Y.-C., & Wang, S.-C. (2004). People tracking by integrating multiple features. *Proceedings of the 17th International, 4*, 929-932 Vol.4. https://doi.org/10.1109/ICPR.2004.1333925

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., & Courville, A. (2015). *Describing Videos by Exploiting Temporal Structure*. 4507–4515. Retrieved from http://openaccess.thecvf.com/content_iccv_2015/html/Yao_Describing_Videos_by_ICCV_2015_paper.html

Yet Another Computer Vision Index To Datasets (YACVID). (2018). Retrieved March 26, 2019,

from http://riemenschneider.hayko.at/vision/dataset/index.php

Youssef, Wael. F. (2015). *Deformable vs Non-deformable Dataset*. Retrieved from http://www.irit.fr/recherches/SAMOVA/CORPORA/DND/DNDO.zip

Yu, H., Wang, J., Huang, Z., Yang, Y., & Xu, W. (2016). *Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks*. 4584–4593. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/Yu_Video_Paragraph_Captioning_CVPR_2016_paper.html

Zaman, Halimah. B., Robinson, P., Smeaton, A., Shih, T. K., Velastin, S., Jaafar, A., & Ali, N. M. (2017). *Advances in Visual Informatics*. Retrieved from https://books.google.com/books/about/Advances_in_Visual_Informatics.html?id=w2A-DwAAQBAJ

Zang, D., Doerschner, K., & Schrater, P. R. (2009). Rapid inference of object rigidity and reflectance using optic flow. In Xiaoyi Jiang & N. Petkov (Eds.), *Computer Analysis of Images and Patterns* (pp. 881–888). Springer Berlin Heidelberg.

Zen, G., Lepri, B., Ricci, E., & Lanz, O. (2010). Space Speaks: Towards Socially and Personality Aware Visual Surveillance. *Proceedings of the 1st ACM International Workshop on Multimodal Pervasive Video Analysis*, 37–42. https://doi.org/10.1145/1878039.1878048

Zhai, M., Chen, L., Mori, G., & Roshtkhari, M. J. (2019). Deep Learning of Appearance Models for Online Object Tracking. In L. Leal-Taixé & S. Roth (Eds.), *Computer Vision – ECCV 2018 Workshops* (pp. 681–686). Springer International Publishing.

Zhang, Dengsheng, & Lu, G. (2001). Segmentation of moving objects in image sequence: A review. *Circuits, Systems and Signal Processing*, *20*(2), 143–183. https://doi.org/10.1007/BF01201137

Zhang, Dong, Javed, O., & Shah, M. (2013). *Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions*. 628–635. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2013/html/Zhang_Video_Object_Segmentation_2013_CVPR_paper.html

Zhang, Z., Huang, K., Tan, T., & Wang, L. (2007). Trajectory Series Analysis based Event Rule Induction for Visual Surveillance. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/CVPR.2007.383076

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., & Xiong, C. (2018). End-to-End Dense Video Captioning with Masked Transformer. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8739–8748. https://doi.org/10.1109/CVPR.2018.00911

Zisserman, A., Capel, D., Fitzgibbon, A., Kovesi, P., Werner, T., & Wexler, Y. (2012). MATLAB Functions for Multiple View Geometry. Retrieved April 9, 2019, from http://www.robots.ox.ac.uk/~vgg/hzbook/code/

VIII. Appendices

VIII.1. Appendix 1: related works in the domain of video surveillance ontologies

In addition to what was mentioned in the chapter II, other interesting ontologies in the video surveillance domain can be found in the literature, we mention:

- A work of detecting events and objects is presented by (Kazi Tani et al., 2015) which propose to use, together, an ontology with a rule detection system. The system uses probabilities to estimate certain events as well as its initial and final times.
- Video event analysis ontology is presented in (SanMiguel et al., 2009). Their ontology is based on two levels of knowledge: the application domain (high level semantic concepts as objects, context, and events) and the analysis system (algorithms, reactions to events, etc.). The ontology and the case study are specialized for the Underground video surveillance domain.
- In the Mind's Eye project (Oltramari & Lebiere, 2012) they used machine learning algorithms for features extraction from a camera input step, and to know patterns and detect suspicious behaviours they match them using a cognitive model operating over those visual features. Then the output is filtered back in the form of new knowledge patterns into the cognitive model and as feature utility into the perceptual algorithms.
  In (Ly et al., 2016) a behaviour ontology is proposed. The ontology is evaluated in the PETS 2006 and PETS 2007 datasets. This approach use prior knowledge, for detecting behaviour without training data of entire process. They sets that a specific behaviour can be acted in various ways but they still share a general plot.

Another interesting ontologies can be found in (Kazi Tani et al., 2015), (Xue, Zheng, & Zhang, 2012), (Carles Fernández et al., 2008), (Akdemir, Turaga, & Chellappa, 2008), Etiseo project (Nghiem, Bremond, Thonnat, & Valentin, 2007), (Bai, Lao, Jones, & Smeaton, 2007), (Francois, Nevatia, Hobbs, Bolles, & Smith, 2005).

VIII.2. Appendix 2: Motion estimation techniques

We should note a duality between motion detection and estimation where the motion detection can be used to find the active region of motion in frames. The inactive region is assumed with zero motion vectors. Then the motion estimation can be applied to active regions only. The early detection of regions with zero motion vector leads to significant reduces in computation.

Also another duality should be noticed, between motion estimation and segmentation operations. In order to correctly estimate the motion, regions of homogeneous ones need to be known. Contrariwise, for better segmentation of these regions, it is essential to apply, previously, motion estimation. This problem can be tackled by joint motion estimation and segmentation techniques (Pesquet-Popescu, Cagnazzo, & Dufaux, 2013).

Another perspective to look at those two approaches (motion detection and motion estimation) is to see them as same approach. To estimate a motion vector, that means detect it and estimate it resulting into foreground segmentation of moving objects, that's why approximately all the motion estimation methods can be and are used for motion detection. But concerning the detection of the motion, can it estimate the motion! Yes, instead of assigning values to each pixel in the visual input, which can be a complex task, detecting motion focus on extracting of moving features (here the feature is a moving objects in the scene), then estimate its position in the next frame, revealing the vector motion of the object. This approach lay under the semantic level (object-based level), where can be called object-based detection or Object-matching.

Many good surveys mentioned this with details (Vishwakarma & Agrawal, 2013), (Candamo et al., 2010), (W. Hu et al., 2004), (Oliver et al., 2000).

The most used methods for motion estimation are:

1. **Background subtraction**: Background subtraction is one of most popular motion detection methods, especially when working with a static camera. It detects moving regions in an image by calculating the difference, pixel-wise, between the reference background image and the current image. Pixels having the result of difference near null are assumed as background pixels, and the rest of pixels belong to a moving object. These methods assume that background changes are much weaker compared to object changes. Background image is simple to use, but they suffer from shadows and reflections of moving objects which may be highlighted in the difference image. In addition, it is extreme sensitivity to changes in luminosity or dynamic scenes. Therefore, to reduce the influence of some of these changes, it is highly recommended having good model for background, (Gandhamal & Talbar, 2015), (Haritaoglu, Harwood, & Davis, 2000), (McKenna, Jabri, Duric, Rosenfeld, & Wechsler, 2000), (Stauffer & Grimson, 1999).
2. **Temporal differencing**: it performs pixel-wise differences between consecutive frames to extract moving regions. After that, to determine changes, a threshold-based function is used. The extracted moving sections are, then by applying a connected component

analysis, clustered into motion regions. This approach is adaptive to dynamic environments, but sometimes fails to extract all the relevant pixels. Other problem is that non-moving parts of objects are non-detected. Besides, that these kinds of methods are very sensitive to noise and luminosity. Furthermore, temporal changes between consecutive images may be detected in areas that are close to object boundaries and in uncovered background. In addition, deposited or removed objects cannot be correctly detected using successive images (Verma, Kumar, & Tomar, 2015), (Aishy, 2001), (A. J. Lipton, Fujiyoshi, & Patil, 1998).

3. **Optical flow:** It is defined as the apparent motion of the brightness pattern. In other terms, it captures the spatial and temporal pixel intensities variation in image sequences. It is similar of velocity measurement, and reflects the image variations during a time interval due to motion. Optical flow is used without any prior knowledge of the frame content, and is suitable for large variety of motions. Also, can be used to detect a moving object in the occurrence of camera motion. Optical flow can be calculated with multiples methods (Differential techniques (Variational techniques), Region-Based Matching (Correlation-Based Methods), Energy-Based Methods (frequency-based methods), Phase-Based Techniques), described in Appendix 1.The optical flow analysis approach gives very dense and approximately accurate results, in each pixel. It is considered to be one of the most detailed and rich motion representations of an image from a video signal. In general, this method can be applied if the intervals between consecutive images are very short, and if no significant change occurs. On the other hand, flow computation methods are computationally complex and time consuming methods. Besides, shadows and reflections of moving objects can be highlighted in the result image and non-moving parts of objects are non-detected. (Barron et al., 1994), (D. Meyer, Denzler, & Niemann, 1997), (Dorthe Meyer, Pösl, & Niemann, 1998).

4. **The Block Matching Algorithm**: Block matching is a special case of the region-based approach. In this approach, motion estimation algorithms are based on the matching of blocks between two frames, with the objective to minimize a dissimilarity measure. For that, current frame is partitioned into blocks for purpose to find out the corresponding motion vector for each block according to its relative displacement from the previous frame. The same displacement vector is assigned to all pixels within a block.
   Block matching is widely used in video coding for transmission or compression purposes. Another important reason for this wide use is the low computational cost it involves. According to (Love & Kamath, 2006), Block matching techniques consist of three main components: block determination, search methods, and matching criteria. Block location, the size, and the scale can be determined by a simple or a hierarchical approach. There are several block-based motion estimation methods (search methods), to mention: Full Search Algorithm (FSA) or Exhaustive Search Algorithm (ESA), two dimensional Logarithmic Search (LOGS) , three-Step Search (3SS), four Step Search (4SS) , Adaptive Rood Pattern Search (ARPS), Diamond Search (DS), Modified Orthogonal Search Algorithm (MOSA), Cross Search Algorithm (CSA), Binary Search (BS), Hierarchical Search Algorithm (HSA), etc.

   Concerning Matching criteria, also known as error or matching functions, we can mention:

- The sum of the absolute values of the differences in the two blocks (SAD).
- The mean squared error (MSE).
- The sum of squared errors (SSE).
- The sum of absolute transformed differences (SATD).
- The mean of the absolute values of the differences in the two blocks (MAD).
- The mean of the square of the differences in the two blocks (MSD).
- The sum of the non-matching pixels in the two blocks, where a match is defined as the difference absolute value is less than a threshold (MPC).

Some Block matching approach can give a very dense motion field (each pixel), even more, some others can give sub-pixel accuracy, but the computational cost, in either case, is extremely high. Using the Full Search Algorithm (FSA) can give very good results but it is an extremely time consuming method. Besides, shadows and reflections of moving objects can be highlighted in the result image and non-moving parts of objects are non-detected. To know more about Block matching Algorithm the reader is referred to (Khammar, 2012), (Love & Kamath, 2006), (Patel, Kshirsagar, & Nitnaware, 2013).

5. **The feature-based approach:** also known as feature correspondence or matching. This approach is based, as first stage, on extracting a set of relatively sparse and discriminatory features in the corresponding images, such as edges, corners or other distinguished points. Inter-frame correspondence (matching) is then established between these features, as second stage, to remove matches that do not correspond to the actual motion and give a set of motion vectors. The resulting motion field is estimated at those feature points only. The uncertainty in determining invariant, accurate and reliable features due to various image distortions at the feature detection stage. At the feature matching stage, the well-known correspondence problem of ambiguous potential matches occurs. Main works were carried out especially on feature detectors and recently good results were achieved; for that reason the most recent SFM approaches prefers using the feature-based approach…
   Motion estimation based on correspondence of interest points (feature points) works well for inter-frame non-small time intervals. Sparse and non-regular correspondence points are also detected. Highly textured objects which are common in real scene cannot adequately be handled by primitives like edges, lines and corners…
   Besides, shadows and reflections of moving objects can be highlighted in the difference image.
   For more information for the feature-based approach, the reader is referred to (Heel, 1990), (Farin & With, 2005), (Szeliski, 2011), (Szeliski, 2011), (Farin Dirk, de With Peter H.N., 2005), (Torr & Zisserman, 2000).

Next, Table VIII-1 compares major approaches and classifies them into: Poor, Moderate, Good, and Very Good.

NB: The assessment of those approaches is aimed for the approach in general. Bad assessment means that the approach in general suffers from this kind of problem, but does not deny the existence of some methods able to overcome the mentioned problem.

For more information the reader is referred to (Aggarwal & Nandhakumar, 1988), (Pesquet-Popescu et al., 2013), (Dufaux & Moscheni, 1995), (Patel et al., 2013), (Patel M. B., Kshirsagar R. V., Nitnaware V., 2007).

Table VIII-1: Comparison between the background subtraction, the temporal differencing, the optical flow approaches, block matching and feature correspondence.

Approach / Problem	Background Subtraction	Temporal Differencing	Optical Flow	Block Matching	Feature Correspondence
Used for Motion Detection	Very Good	Very Good	Very Good	Good	Good
Used for Motion Estimation	Moderate	Moderate	Very Good	Very Good	Very Good
Without prior knowledge about the content of frames	Poor	Good	Very Good	Very Good	Very Good
Minimum hypothesis, assumptions and constraints	Poor	Moderate	Good	Very Good	Very Good
Detection with camera motion	Poor	Poor	Very Good	Good	Good
Non Sensitivity to changes in dynamic scenes	Poor	Very Good	Good	Moderate	Moderate
Non detection of shadows for moving objects	Poor	Poor	Poor	Poor	Moderate
Non detection of reflections for moving objects	Poor	Poor	Poor	Poor	Moderate
Non Sensitivity to illumination changes	Poor	Poor	Poor	Moderate	Good
Non Sensitivity to noise	Poor	Poor	Poor	Poor	Poor
Detection of removed or deposited objects	Poor	Poor	Good	Poor	Poor
Detection with Occlusion and Transparency	Poor	Poor	Poor	Poor	Poor
Detection of Turning object	Poor	Poor	Good	Poor	Poor
Detection of slow object motions	Moderate	Poor	Good	Moderate	Moderate
Detection of non-moving Parts	Very Good	Poor	Poor	Poor	Poor
Object Boundaries	Good	Poor	Good	Poor	Moderate
Non-missing parts or Holes in detected objects	Moderate	Poor	Very Good	Moderate	Moderate
Detection of each pixel displacements	Poor	Poor	Very Good	Good	Moderate
Accurate displacements (sub-pixel)	Moderate	Poor	Very Good	Good	Moderate
Allow Tracking each objects pixels throw frames	Poor	Poor	Very Good	Good	Moderate
Allow Tracking and Prediction of moving objects	Poor	Poor	Good	Moderate	Good
object shape changing during motion	Poor	Poor	Good	Poor	Poor
Used to motion 3D estimation	Poor	Poor	Good	Poor	Very Good
Unrequired filtering	Poor	Poor	Moderate	Moderate	Moderate
Resulting in Motion Vectors	Poor	Poor	Very Good	Very Good	Moderate
Density	Poor	Poor	Very Good	Good	Moderate
Simplicity of the method	Moderate	Very Good	Poor	Moderate	Moderate
Non-Expansivity in time consumption	Good	Very Good	Poor	Moderate	Moderate

VIII.3. Appendix 3: Argumentation of the maximization equation

In this study, as shown in the sub-section III.3.5, the problem is not object detection or face localization, but it is a classification. Then, for the probabilistic approach, in the maximization equation, $\mathrm{Arg}\max_{N,N_2} P[(X \geq N_2) \cap (Y < N_2)]$, X and Y are not completely independent, in addition, Y=N-X. For that, it seems enough to maximize only the X term without the Y term ($\arg\max_{N,N_2} P[(X \geq N_2)$). But if we use, for the maximization equation ($\arg\max_{N,N_2} P[(X \geq N_2)$), the same graphical representation that was introduced as a numerical resolution for the maximization problem ($\arg\max_{N,N_2} P[(X \geq N_2) \cap (Y < N_2)]$), we will have the graphical representation showing Figure VIII-1.

Figure VIII-1 shows Graphical representation of probability of $\arg\max_{\mathrm{N,N_2}} \mathrm{P}[(\mathrm{X} \geq \mathrm{N_2})$ for 1<= N <=40 and 1<= N2 <=N. With the probability of success p = 82.58 and the probability of failure q= 17.42 (see Table III-1) of the fundamental matrix as transformation, the Symmetric epipolar distance as distance measures, the normalization level 2, the Mapping error threshold γ_{F}^2 =2.2, the Motion rigidity threshold δ_{F}^2 = 80.16).

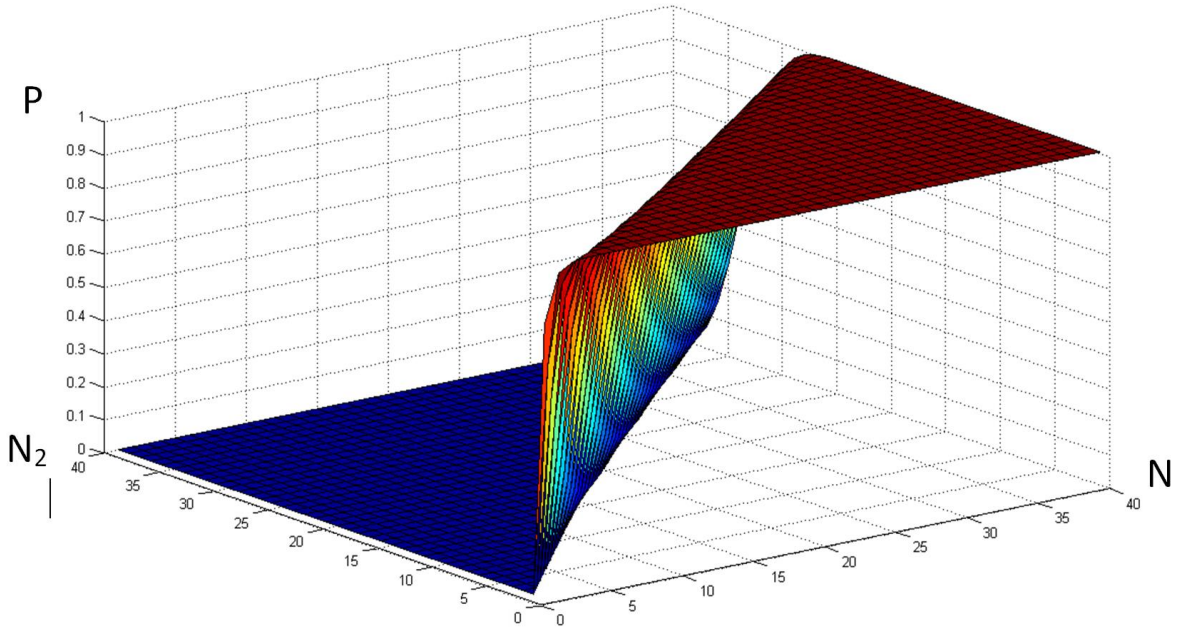


Figure VIII-1: Graphical representation of probability of $\boldsymbol{arg\ max_{N,N_2}\ P[(X \geq N_2)}$ *for 1<= N <=40 and 1<=* N_2 *<=N, having p = 82.58, q= 17.42, f, Symmetric epipolar distance, normalization level 2,* $\boldsymbol{\gamma_F^2}$ *=2.2, and* $\boldsymbol{\delta_F^2}$ *= 80.16.*

Table VIII-2: Numerical representation of probability of $\boldsymbol{arg\ max_{N,N_2}}\ \boldsymbol{P}[(\boldsymbol{X \geq N_2})$ *with p = 82.58 and q= 17.42, and N, N_2 = 1:15.*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.8242	0.9691	0.9946	0.999	0.9998	1	1	1	1	1	1	1	1	1	1
2	0	0.6793	0.9181	0.9811	0.9959	0.9991	0.9998	1	1	1	1	1	1	1	1
3	0	0	0.5599	0.8552	0.959	0.9894	0.9974	0.9994	0.9999	1	1	1	1	1	1
4	0	0	0	0.4615	0.786	0.9286	0.9787	0.9941	0.9985	0.9996	0.9999	1	1	1	1
5	0	0	0	0	0.3803	0.7146	0.891	0.9633	0.9887	0.9968	0.9991	0.9998	0.9999	1	1
6	0	0	0	0	0	0.3135	0.6441	0.8476	0.9429	0.9807	0.9939	0.9982	0.9995	0.9999	1
7	0	0	0	0	0	0	0.2584	0.5763	0.7999	0.9178	0.9696	0.9897	0.9967	0.999	0.9997
8	0	0	0	0	0	0	0	0.2129	0.5124	0.7493	0.8882	0.9553	0.9836	0.9944	0.9982
9	0	0	0	0	0	0	0	0	0.1755	0.4532	0.6973	0.8546	0.9376	0.9755	0.9911
10	0	0	0	0	0	0	0	0	0	0.1447	0.399	0.6448	0.8177	0.9165	0.9652
11	0	0	0	0	0	0	0	0	0	0	0.1192	0.3498	0.593	0.7782	0.8922
12	0	0	0	0	0	0	0	0	0	0	0	0.0983	0.3056	0.5424	0.7368
13	0	0	0	0	0	0	0	0	0	0	0	0	0.081	0.2661	0.4939
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0668	0.231
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.055

It appears obviously that, in the graph, a big plateau of probability = 1 exists, and that could not be correct. For example for N = 40 and N_2 = 1, 2 or3…22, the probability = 1 and that it totally wrong.

If we compare those results with the numerical and graphical results of the maximization expressions: $\arg\max_{N,N_2} P[(X \geq N_2) \cap (Y < N_2)]$, we find that the graph of the maximization expression (eq. III-15) is descendent on the right side when N is growing and N_2 is remain respectively small. And this it seems more logical. For example: if N = 40 and N_2 = 3, the probability = 0.019.

Finally, the maximization expression must contain both terms X and Y, where we have to validate all the correct classification X, and reject all the false classification Y.

VIII.4. Appendix 4: Examples of experiments on deformable vs non-deformable classification

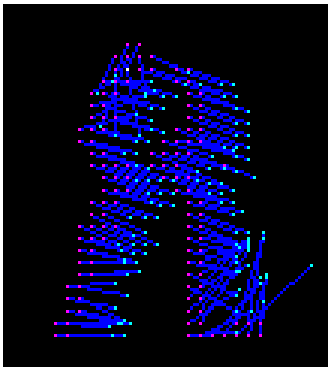
Here we can see samples of experiments done on the classification of deformable vs non-deformable objects:

- Scene of "Stairwell" :
  - Frame 31:

a

b

c



d

*Figure VIII-2: Scene of "Stairwell": a- frame 31, b- frame 28, c- motion's vectors, and d- motions' vectors zoomed (128*2 corresponding points)*

Table VIII-3: Percentage of correctly mapped points ($\gamma'^{n}_{F} = 1$ and $\gamma'^{n}_{H} = 1$)											
Normalisation / Transformation			**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
H	**mean distance**	p^{n}_{H}	**85.15**	**75**	**69**	**43.75**	**37.5**	**30.46**	**23.43**	**19.53**	**12.5**	**10.15**
		(δ'^{n}_{H}):	**78.06**	**43.61**	**25**	**15**	**9.88**	**6.74**	**4.8**	**3.4**	**2.65**	**1.94**
F		p^{n}_{F}	**99.21**	**96.87**	**91.40**	**82.81**	**75**	**67.96**	**63.28**	**50**	**42.18**	**40.62**
		(δ'^{n}_{F}):	**92.49**	**79.13**	**67.22**	**57.87**	**49.82**	**42.16**	**36.62**	**31.47**	**27.3**	**23.21**

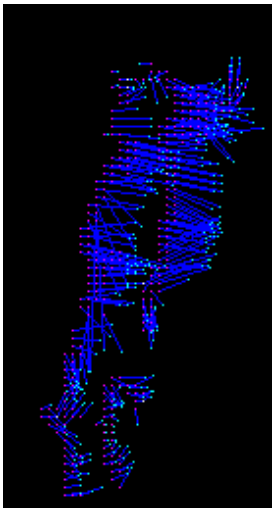
Table VIII-4: Percentage of correctly mapped points (Normalization = 2, γ^{2}_{F} =2.2, δ^{2}_{F} = 80.16)			
F	**Symmetric epipolar distance**	p^{n}_{F}	**96.8750**
		(δ'^{2}_{F}):	**80.16**

- Frame 44:

a

b

c



d

*Figure VIII-3: Scene of "Stairwell": a- frame 44, b- frame 40, c- motion's vectors, and d- motions' vectors zoomed (293*2 corresponding points)*

Table VIII-5: Percentage of correctly mapped points ($\gamma'^{n}_{F} = 1$ and $\gamma'^{n}_{H} = 1$)										
Normalisation / Transformation			1	2	3	4	5	6	7	8	9
H	mean distance	p^{n}_{H}	58.020	20.47	6.484	3.754	2.047	1.023	1.023	1.023	1.0239
		(δ'^{n}_{H}):	78.06	43.61	25	15	9.88	6.74	4.8	3.4	2.65
F		p^{n}_{F}	80.54	51.53	35.83	27.64	23.20	19.11	16.72	15.01	11.945
		(δ'^{n}_{F}):	92.49	79.13	67.22	57.87	49.82	42.16	36.62	31.47	27.3

Table VIII-6: Percentage of correctly mapped points (Normalization = 2, γ'^{2}_{F} =2.2, δ^{2}_{F} = 80.16)			
F	Symmetric epipolar distance	p^{n}_{F}	53.5836
		(δ'^{2}_{F}):	80.16

In this scene of moving person on the “Stairwell”, we can see a series of non-deformable movements, when only the upper body is shown and moving in a quite translating way like the motion between frame 28 and frame 31 (motion frame 31); But after that, legs appears at frame 36, and motions became clearly deformable like the motion frame 44. All of those motions classifications can be clearly inferred when comparing percentages of correctly mapped points to the corresponding threshold (see tables above). Plus if we put the percentages of correctly mapped points of those frames scene in a graph, it appears clearly how and when the motion became deformable:

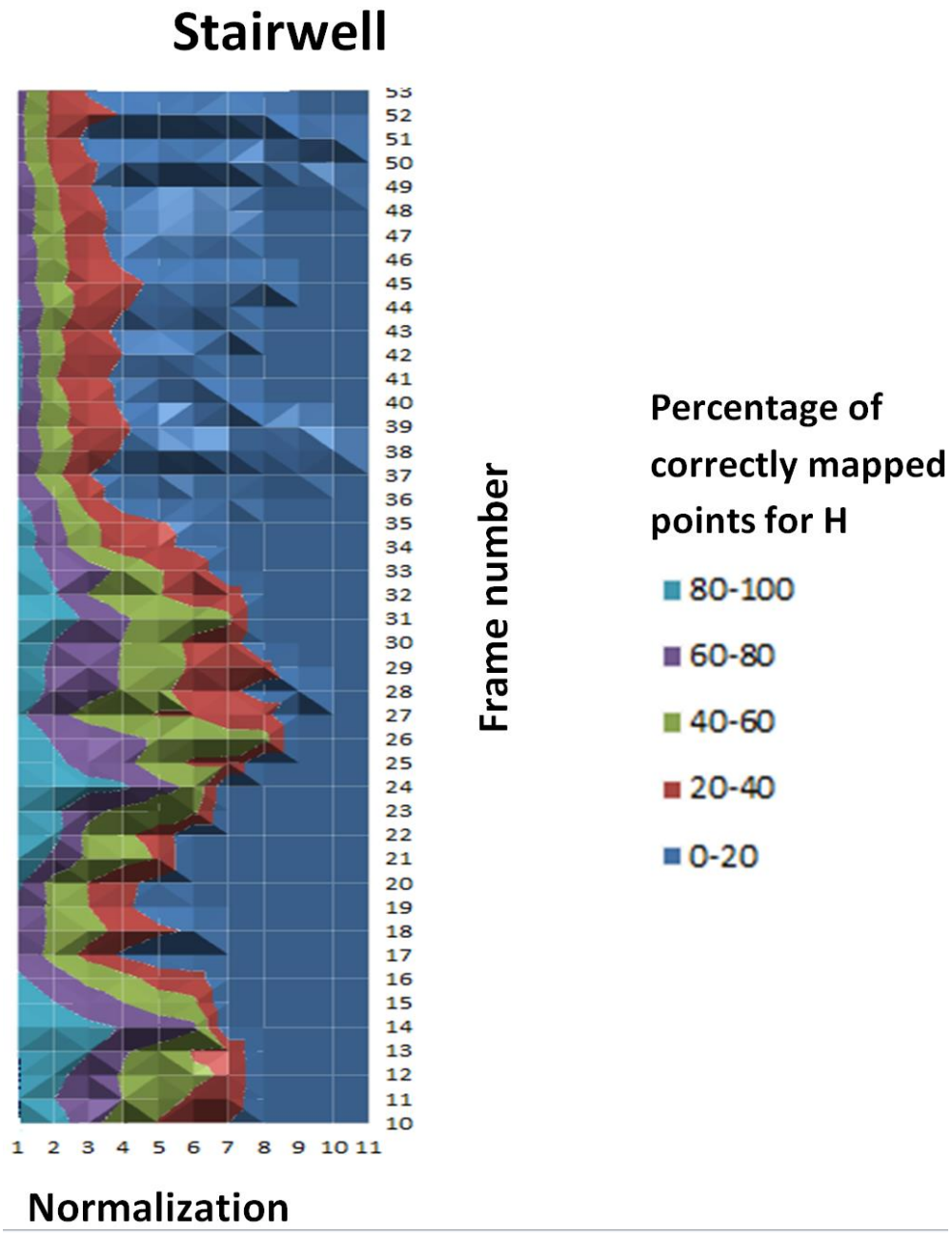
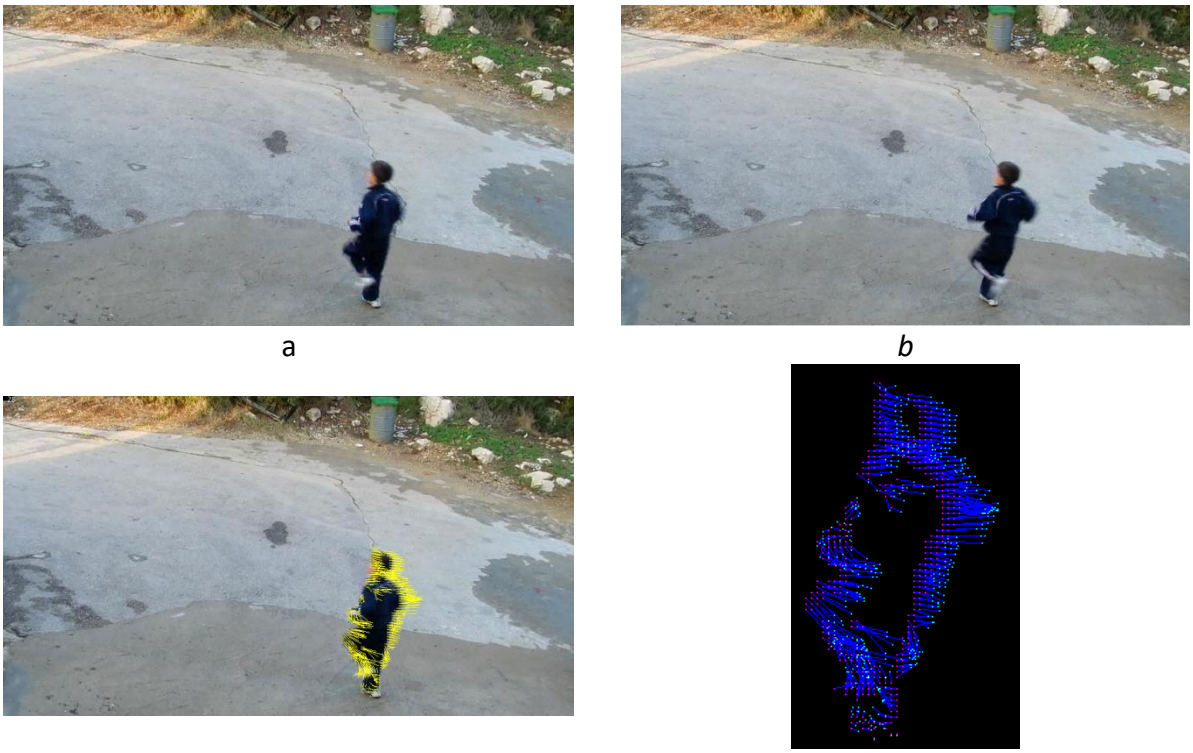


Figure VIII-4: ***Scene “Stairwell”: Percentage of correctly mapped points for H***

- Scene of "Running" :
  - Frame 28:



a b c d

*Figure VIII-5: Scene of " Running ": a- frame 28, b- frame 27, c- motion's vectors, and d- motions' vectors zoomed (351*2 corresponding points)*

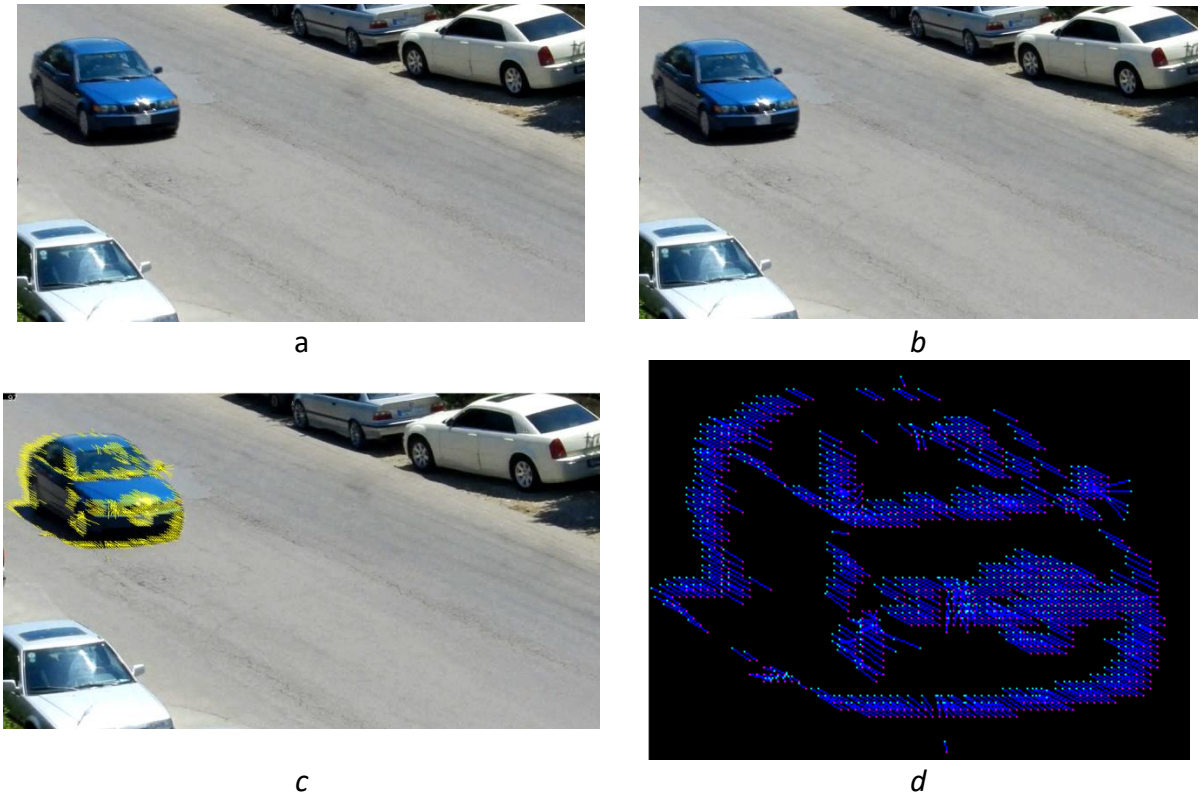
Table VIII-7: Percentage of correctly mapped points ($\gamma'^{n}_{F} = 1$ and $\gamma'^{n}_{H} = 1$)									
Normalisation / Transformation			1	2	3	4	5	6	7	8
H	mean distance	p^{n}_{H}	56.849	32.191	21.461	11.415	7.9909	5.9361	4.3379	3.1963
		(δ'^{n}_{H}):	78.06	43.61	25	15	9.88	6.74	4.8	3.4
F		p^{n}_{F}	80.593	55.9361	47.4886	37.8995	31.9635	27.1689	23.5160	20.7763
		(δ'^{n}_{F}):	92.49	79.13	67.22	57.87	49.82	42.16	36.62	31.47

Table VIII-8: **Percentage of correctly mapped points (Normalization = 2, γ^{2}_{F} =2.2, δ^{2}_{F} = 80.16)**			
F	Symmetric epipolar distance	p^{n}_{F}	57.5342
		(δ'^{2}_{F}):	80.16

In this scene of a running person, for this motion of frame 28, we can see clearly that the percentages of correctly mapped points are smaller than its corresponding thresholds whatever the normalization for F or H, and then this motion is deformable. Almost all motions in this series are deformable motions.

NB: For us, the decision will be taken according to the Percentage of correctly mapped points when Normalization = 2, γ^{2}_{F}=2.2 and δ^{2}_{F} = 80.16.

- Scene of "Highway 2":
  - Frame 97:



a b c d

*Figure VIII-6: Scene of " Highway 2 ": a- frame 97, b- frame 96, c- motion's vectors, and d- motions' vectors zoomed (690*2 corresponding points)*

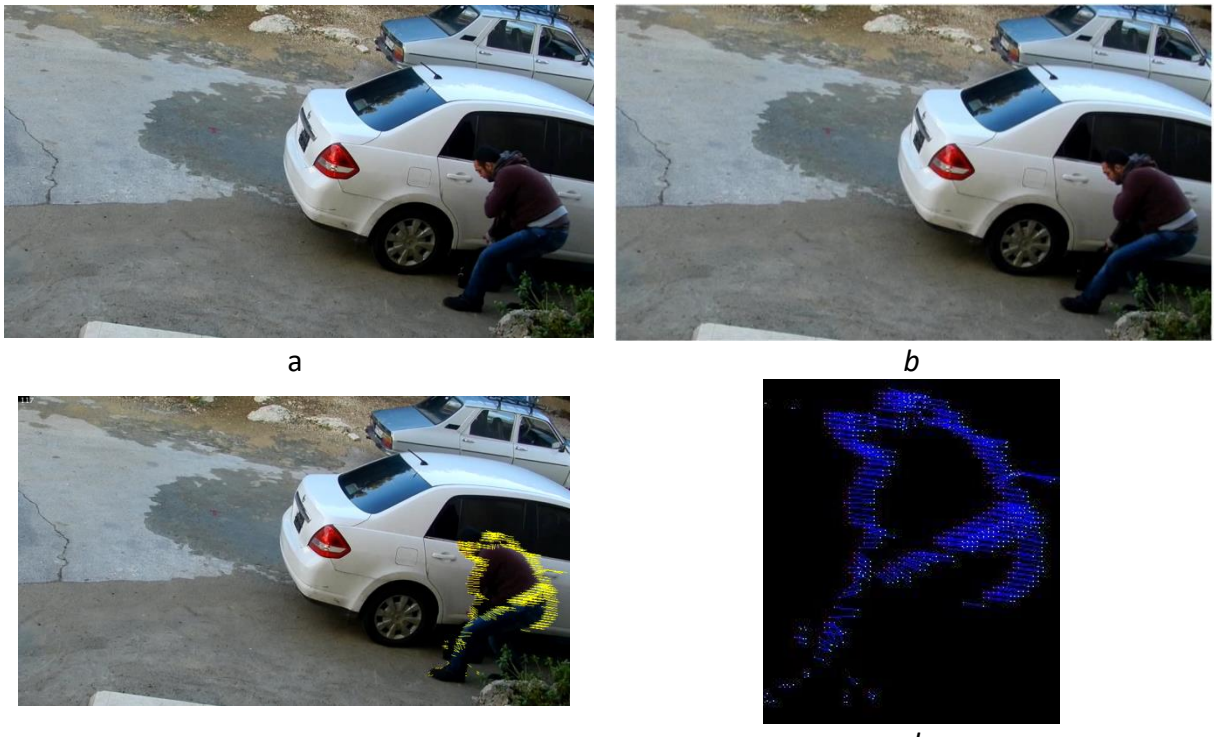
Table VIII-9: ***Percentage of correctly mapped points*** ($\gamma'^{n}_{F} = 1$ *and* $\gamma'^{n}_{H} = 1$)										
Normalisation / Transformation			1	2	3	4	5	6	7	8	9
H	mean distance	p^{n}_{H}	91.390	88.344	85.827	58.807	32.053	23.708	19.337	16.158	12.715
		(δ'^{n}_{H}):	78.06	43.61	25	15	9.88	6.74	4.8	3.4	2.65
F		p^{n}_{F}	98.41	94.17	93.11	92.05	90.72	90.19	89.00	87.54	86.75
		(δ'^{n}_{F}):	92.49	79.13	67.22	57.87	49.82	42.16	36.62	31.47	27.3

Table VIII-10: Percentage of correctly mapped points (Normalization = 2, γ^{2}_{F} ***=2.2,*** δ^{2}_{F} ***= 80.16)***			
F	Symmetric epipolar distance	p^{n}_{F}	94.4371
		(δ'^{2}_{F}):	80.16

In this scene of a car on the highway, for this motion of frame 97, we can see clearly that the percentages of correctly mapped points are bigger than its corresponding thresholds whatever the normalization for F or H, and then this motion is non-deformable. Almost all motions in this series are non-deformable motions.

NB: In our experiments, the decision will be taken according to the Percentage of correctly mapped points when Normalization = 2, γ^{2}_{F}=2.2 and δ^{2}_{F} = 80.16. It's clear that we can infer the non-deformability of motion from the Homography too.

- Scene of "Bomb 2" :
  - Frame 117:



a b c d

*Figure VIII-7: Scene of " Bomb 2 ": a- frame 117, b- frame 114, c- motion's vectors, and d- motions' vectors zoomed (317*2 corresponding points)*

Table VIII-11: Percentage of correctly mapped points ($\gamma'^{n}_{F} = 1$ and $\gamma'^{n}_{H} = 1$)											
Normalisation / Transformation			1	2	3	4	5	6	7	8	9	10
H	mean distance	p^{n}_{H}	92.42	71.29	48.58	33.12	23.97	18.92	12.93	11.35	9.77	7.57
		(δ'^{n}_{H}):	78.06	43.61	25	15	9.88	6.74	4.8	3.4	2.65	1.94
F		p^{n}_{F}	99.053	94.6	84.22	79.4	75.39	71.92	65.9306	60.56	54.57	48.5
		(δ'^{n}_{F}):	92.49	79.13	67.22	57.87	49.82	42.16	36.62	31.47	27.3	23.21

Table VIII-12: Percentage of correctly mapped points (Normalization = 2, γ^{2}_{F} =2.2, δ^{2}_{F} = 80.16)			
F	Symmetric epipolar distance	p^{n}_{F}	95.2681
		(δ'^{2}_{F}):	80.16

In this scene of a person planting a bomb, for this motion of frame 117, the person is moving in a non-deformable way, we can see clearly that the percentages of correctly mapped points are bigger than its corresponding thresholds whatever the normalization for F or H, and then this motion is non-deformable. Almost all motions in this series are deformable motions.

NB: In our experiments, the decision will be taken according to the Percentage of correctly mapped points when Normalization = 2, γ^{2}_{F}=2.2 and δ^{2}_{F} = 80.16. It's clear that we can infer the non-deformability of motion from the Homography too.

VIII.5. Appendix 5: State of the art: Video Analysis

VIII.5.1. Object detection

Mainly visual surveillance analysis approaches start, after detecting regions corresponding to moving objects in the frames and images, by locating objects of interest in the scene. Following that, object classification and tracking, also behaviour analysis and recognition are significantly dependent on it.

As mentioned, in visual surveillance, motion is the key feature, and the temporal information is widely exploited by detection approaches. The process of object detection usually starts with environment/background modelling (Neves et al., 2016), (Revathi & Kumar, 2012), (W. Hu et al., 2004), (W. Hu et al., 2004), and motion segmentation.

Several basic conventional approaches for object detection can be used: Background subtraction, temporal differencing, optical flow and feature-based approach; the reader is referred to chapter III and Appendix II for more explanation, references and comparison for each of these approaches.

Detecting objects in still images in the late years showed very good results. Different object types were detected using deep CNNs (Szegedy et al., 2015),(Girshick, Donahue, Darrell, & Malik, 2014), (Girshick, 2015),(He, Zhang, Ren, & Sun, 2016), (Redmon, Divvala, Girshick, & Farhadi, 2016), (K. Wang, Lin, Zuo, Gu, & Zhang, 2016). He et al. (He et al., 2016), and ResNet152 classifier CNN (He et al., 2016), proposed a novel Residual Neural Network (ResNet) which train very deep networks with over one hundred layers, resulting in a very good performance classifying 1000 object classes on a public image dataset. You Only Look Once YOLO (Redmon et al., 2016), YOLOv2 (Redmon & Farhadi, 2017) and Single Shot MultiBox Detector SSD (Liu et al., 2016), Deconvolutional Single Shot Detector (DSSD) (Fu, Liu, Ranga, Tyagi, & Berg, 2017) generated multiple boxes from the image, then simultaneously aim to predict these bounding boxes for each object on image and correspondent class labels for them, and apply classification according to probabilistic scores. NoScope (D. Kang, Emmons, Abuzaid, Bailis, & Zaharia, 2017) has improved the filtering of frames, then they perform heavy CNNs. A Faster R- CNN is proposed by (Ren, He, Girshick, & Sun, 2015), to share full-image convolutional features with the detection network, they uses the fully convolutional network called Region Proposal Network (RPN) which, mainly, predicts object boundaries and scores at each position. The RPN is trained to produce high-quality region proposals, to be used by Fast R-CNN for detection. Then, counting on the “attention” mechanisms, they merge RPN and Fast R-CNN into a single network by sharing their convolutional features.

For the object detection in videos, also many works were introduced. Han et al. (Han et al., 2016) proposed a NMS method to sequences still-image detections, then apply the sequence-level NMS on the results. Weaker class scores are then boosted by the detection on the same sequence. Kang et al. (K. Kang et al., 2017) proposed a T-CNN Tubelets with Convolutional Neural, the proposed network generates many of tubelet proposals simultaneously for object detection from videos. Galteri et al. (Galteri, Seidenari, Bertini, & Bimbo, 2017) in the goal of improving window ranking, they feed back their algorithm with the object detection results on previous frame, using a closed loop framework.

Another interesting works can be seen for object detection in video (Vorobjov, Zakharava, Bohush, & Ablameyko, 2018), (Pathak, Pandey, Rautaray, & Pawar, 2018), and for detection and classification in video, Focus (Hsieh et al., 2018).

VIII.5.2. Moving object segmentation

Segmentation of video objects consists of separating the foreground objects from the background in a video (Lee, Kim, & Grauman, 2011), (Ochs & Brox, 2012), (T. Wang & Collomosse, 2012). It is important for a wide range of advanced video applications, including video content encoding (Sikora, 1997), (Dengsheng Zhang & Lu, 2001), video summary, annotation and video search, providing a spatial support for learning models of object class (Prest, Leistner, Civera, Schmid, & Ferrari, 2012), video surveillance, estimation of object movements, tracking, description of multimedia content (Tekalp, 1995), (Ebrahimi, 1997) intelligent signal processing, recognition of objects and activities, recognition of action (Gorelick, Blank, Shechtman, Irani, & Basri, 2007).

An object segmentation algorithm classifies the pixels of a video image into a number of classes that are homogeneous with respect to a few features.

The segmentation of an object is, therefore, an active research domain that has produced a wide variety of segmentation methods. Some methods focus on, depth information (3D movements) like (Klappstein, Vaudrey, Rabe, Wedel, & Klette, 2009), (Wedel, Meißner, Rabe, Franke, & Cremers, 2009) and (Herbst, Ren, & Fox, 2013), others group the points tracked during image pairs (Brox & Malik, 2010), (Ochs & Brox, 2011) or triplets (Ochs & Brox, 2012), long-term trajectories (Ochs, Malik, & Brox, 2014). Some methods seek the infrastructure of different segments and shapes of the object in order to separate the segments of that object from the bottom (Papazoglou & Ferrari, 2013), (Dong Zhang, Javed, & Shah, 2013), (Lee et al., 2011), and background subtraction (Piccardi, 2004).

Recent approaches have used many methods of deep learning for object segmentation in videos:

Some were based on a recurrent neural network (RNN)(Y.-T. Hu, Huang, & Schwing, 2017), on CNNs (Maninis, Pont-Tuset, Arbeláez, & Van Gool, 2016), on optical flow (Khoreva, Benenson, Ilg, Brox, & Schiele, 2017), on LSTM (N. Xu et al., 2018), Fully Convolutional Networks (FCNs) (OSVOSS) (Maninis et al., 2017), on R-CNN (Girshick et al., 2014), Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017), on modified Faster-RCNN (Ren et al., 2015), and on Encoder-Decoder Seg-Net (Badrinarayanan, Kendall, & Cipolla, 2015).

VIII.5.3. Object classification

Object classification is the process of identifying what type of object is present in the environment among different available ones; for instance, to tell whether the moving objects are humans, vehicles, animals, inert objects or others. Object classification could distinguish remarkable motion from those caused by specular reflections, moving clouds, or other dynamic occurrences. Some problems appear when the background may contain element features similar to the foreground objects, e.g., when a many persons are moving

together in a crowd (Cavallaro, Steiger, & Ebrahimi, 2005). Three main categories of approaches for classifying moving objects (W. Hu et al., 2004), (Loy, 2010), (Cilla et al., 2014): Shape-based method (Collins et al., 2000), (A. J. Lipton et al., 1998), (Kuno, Watanabe, Shimosakoda, & Nakagawa, 1996), Motion-based method (R. Cutler & Davis, 2000), (Alan J. Lipton, 1999), and Feature-based method (Yang, Shih, & Wang, 2004), (Harasse, Bonnaud, & Desvignes, 2006).

More recent works, using deep neural networks, for object classification, especially in images, have taken a big success. Some approach detect objects using conventional techniques and then the detected objects can be classified using object classification CNN architectures such as ResNet (He et al., 2016), AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) and VGG (Simonyan & Zisserman, 2014). Other well-known one-stage techniques are YOLO, YOLOv2 and Faster RCNN, where they detect jointly the objects and classify them.

VIII.5.4. Video action analysis

A variety of state of the art which is in the field of the video action analysis, most are limited by restrictions, we mention:

A. Scene Type: For example, traffic scene (Gerber et al., 2002), outdoor scene (Nevatia et al., 2003).
B. The type of the object: human (Liang Wang et al., 2003), manipulation of a single human hand (Mann, Jepson, & Siskind, 1996), human and car (Ivanov et al., 1999).
C. The type of action: they are limited by some gestures for example: move forward, turn right, turn left, stop, walk, run, up, down, approach, punch, kick, push (A. Kojima et al., 2002), (Aggarwal, 2004).
D. The scenario: type of limited interaction between humans (Thonnat & Rota, 1999), football (Snoek & Worring, 2003), sport (H. Li et al., 2010).
E. The type of experimentation: (Ryoo & Aggarwal, 2007b), (J. Wu et al., 2007), (Abhinav Gupta & Davis, 2007).

VIII.6. Appendix 6: Features extraction

Five types of features were extracted, spatial, temporal, inter-objects, inter-frames and trajectory features; here we present detailed explanations on the first three types as follow:

A. **Object spatial Features:** after the segmentation, in **each of the scene frames**, the following features, shown in Table VIII-13, are extracted from the object 1 (obj1) state and object 2 (obj2) state, see Figure VIII-8.

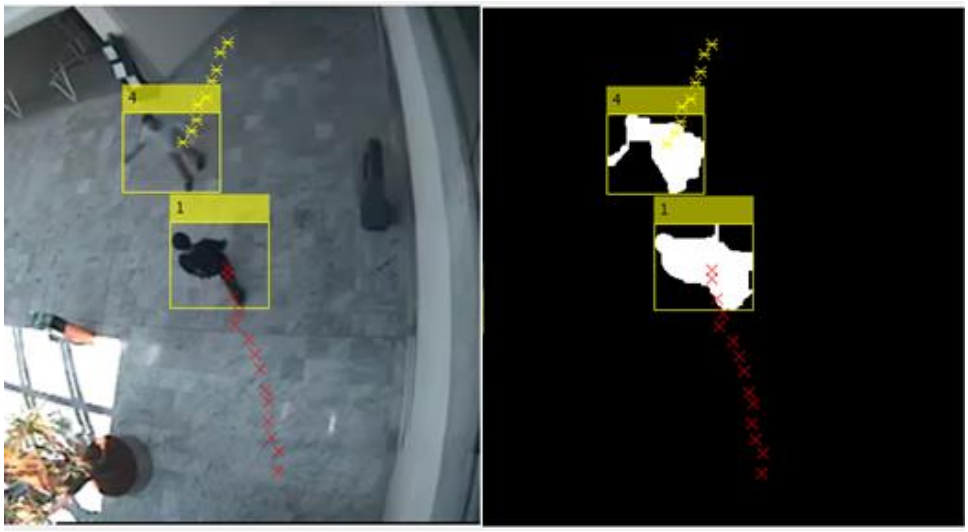


Figure VIII-8: Two objects states in a frame.

Table VIII-13: Spatial features of objects	
Obj_width(obj)	The object width normalized by the frame width (bbox_width*100/frame_width).
Obj_height(obj)	The object height normalized by the frame height (bbox_height*100/frame_height).
xC_pos(obj)	The x position of the object centroid C(obj) with respect of the screen (x*100/ frame_width).
yC_pos(obj)	The y position of the object centroid C(obj) with respect of the screen (y*100/frame_height).
S_bbox(obj)	The bbox surface normalized by the frame surface (bbox_width * bbox_height *100/ frame_width * frame_height).
S1_obj (obj)	The object surface normalized by the frame surface (nb_pixels(obj)*100/ frame_width * frame_height).
S2_obj (obj)	The object surface normalized by the bbox surface (nb_pixels(obj)*100/ bbox_width * bbox_height).
Bmask(obj)	The bbox binary mask differentiating the object pixels from background (see Figure VIII-9).
Intensity(obj)	The bbox grey matrix presenting the object pixels.
RGB(obj)	The bbox RGB colour matrix presenting the object pixels (see Figure VIII-9).
Int_mean(obj)	The mean of the grey matrix of the object pixels.
RGB_mean(obj)	The mean of the RGB colour 3D matrix of the object pixels.
Int_std(obj)	The standard deviation (std) of the grey matrix of the object pixels.
RGB_std(obj)	The standard deviation (std) of the RGB colour 3D matrix of the object pixels.
P_obj (obj)	The object perimeter normalized by the object surface (p(obj)*100/ nb_pixels(obj)).
Hu(obj)	The Hu invariant moments vector (7 moments)[Invariant Moments] of the object.
Hu_comb(obj)	The 7 Hu invariant moments of the object combined in one value.

Reliability(obj)	The reliability of the segmented object in the frame, it depends on which is the object predicted (occluded), directly detected after appearance and before disappearance.
Type_obj(obj)	The object type if deformable or not, according to the algorithm of the chapter III.

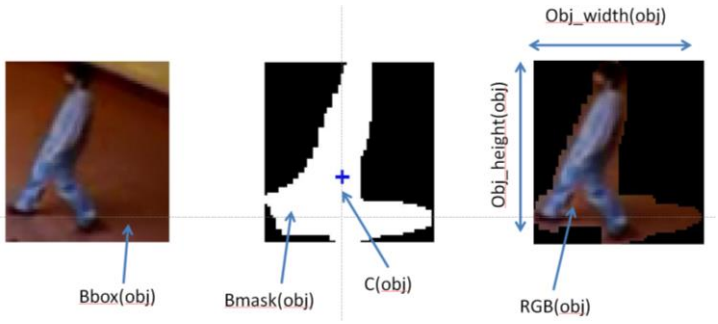


Figure VIII-9: Figure showing bbox, mask and pixels features extracted from an object

B. **<u>Object temporal Features:</u>** after extracting the spatial features, the following features, shown in Table VIII-14, are extracted. Those features designate the variations occurring between the past frame (f-n) and current one (f) for object 1 (obj1) and object 2 (obj2) each one separately. Here n is set to mainly to one, and for some features as speed and direction, is set to five.

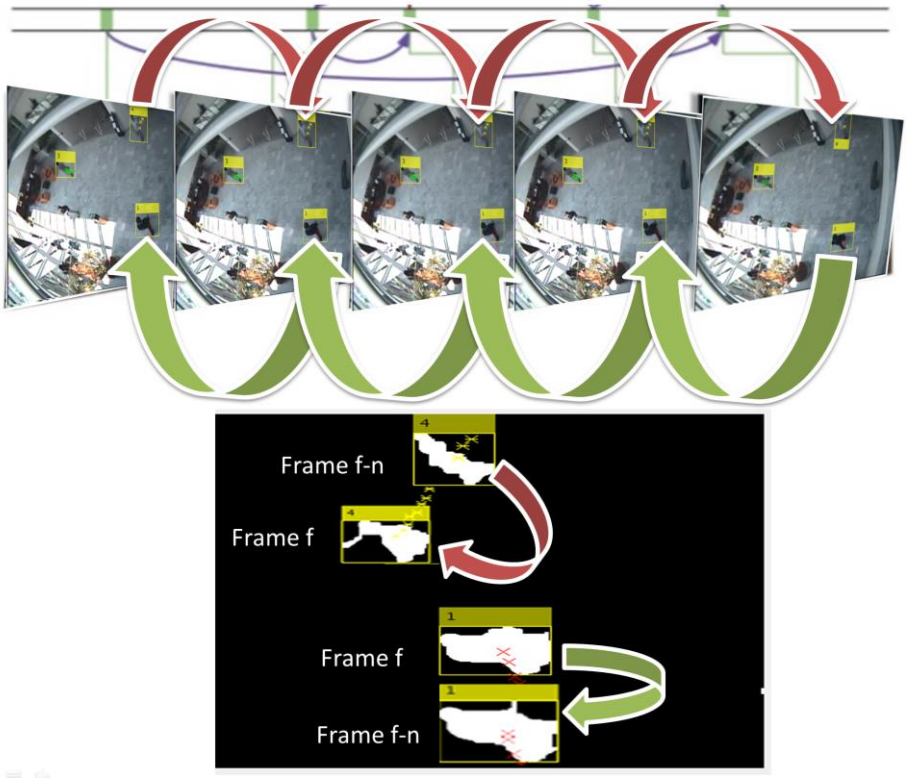


Figure VIII-10: Temporal features of objects

Table VIII-14: Temporal features of objects	
S_bbox_change(obj)	The bbox surface change normalized by the frame surface ($S\_bbox^{f}(obj)$- $S\_bbox^{f-1}(obj)$).
S1_obj_change (obj)	The object surface change normalized by the frame surface ($S1\_obj^{f}(obj)$- $S1\_obj^{f-1}(obj)$).
S2_obj_change (obj)	The object surface change normalized by the object surface (($S1\_obj^{f}(obj)$- $S1\_obj^{f-1}(obj)$)*100/ $S1\_obj^{f-1}(obj)$).
P_obj_change (obj)	The object perimeter change normalized by the object surface ($P\_obj^{f}(obj)$- $P\_obj^{f-1}(obj)$).
Dist1(obj)	The distance between the object centroids (current frame f and f-1) normalized by the frame surface (d($C^{f}(obj)$, $C^{f-1}(obj)$)*100/ frame_width*frame_height).
Dist2(obj)	The distance between the object centroids (current frame f and f-5) normalized by the frame surface, (d($C^{f}(obj)$, $C^{f-5}(obj)$)*100/ frame_width*frame_height).
Speed(obj)	The speed between the object centroids (current frame f and f-5) normalized by the frame surface, as 1 sec =25 frames, (Dist2(obj)/(5/25)).
Angle1(obj)	The angle formed by 3 object centroids (f-1, current frame f and f+1).
Angle2(obj)	The angle formed by 3 object centroids (f-5, current frame f and f+5).
Hu_changes(obj)	The Hu invariant moments vector changes (7 moments) of the object (sum(abs($Hu^{f}(obj)$- $Hu^{f-1}(obj)$))).
Huα_changes(obj)	The Hu invariant moments α changes of the object ($Hu\alpha^{f}(obj)$- $Hu\alpha^{f-1}(obj)$), where α ϵ [1,…,7].
Form_change1(obj)	The change of the object forms normalized by the object surface. This feature calculates the difference of surfaces between binary masks, after co-centring the centroids, by counting the number of pixels which do not coincide, divided by the nb_pixels(obj), (see Figure VIII-12).
Form_change2(obj)	The upper part only of the change of the object forms normalized by the object surface. We took the upper part only because it is more reliable than the lower part, because of the shadow. This feature calculates upper part of the difference of surfaces between binary masks, after co-centring the centroids, by counting the number of pixels above the centroid which do not coincide, divided by the nb_pixels(obj), (see Figure VIII-12).
Obj_width_change(obj)	The normalized change of the object width (($Obj\_width^{f}(obj)$- $Obj\_width^{f-1}(obj)$) / $Obj\_width^{f-1}(obj)$ *100/ frame_width).
Obj_height_change(obj)	The normalized change of the object height (($Obj\_height^{f}(obj)$- $Obj\_height^{f-1}(obj)$)*100/frame_height).
Int_mean_change(obj)	The normalized change of the intensity mean (($Int\_mean^{f}(obj)$- $Int\_mean^{f-1}(obj)$)*100/ $Int\_mean^{f-1}(obj)$).
Int_std_change(obj)	The normalized change of the intensity standard deviation (($Int\_std^{f}(obj)$- $Int\_std^{f-1}(obj)$)*100/ $Int\_std^{f-1}(obj)$).
RGB_mean_change(obj)	The normalized change of the RGB mean (($RGB\_mean^{f}(obj)$- $RGB\_mean^{f-1}(obj)$)*100/ $RGB\_mean^{f-1}(obj)$).
RGB_std_change(obj)	The normalized change of the RGB standard deviation (($RGB\_std^{f}(obj)$- $RGB\_std^{f-1}(obj)$)*100/ $RGB\_std^{f-1}(obj)$).

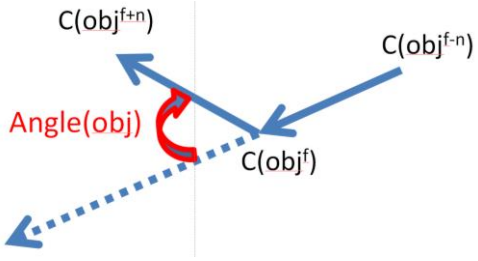


Figure VIII-11: Figure showing the angle feature for object movement

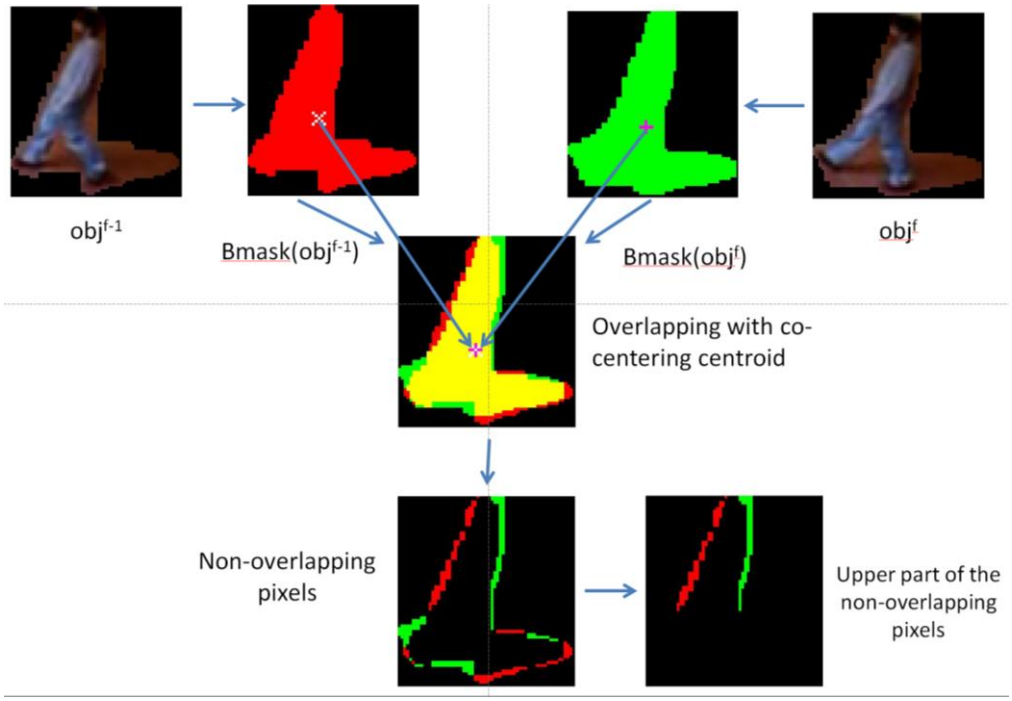


Figure VIII-12: Figure showing bbox, mask and pixels features extracted from an object

C. **Inter-Objects Features:** after extracting the spatial features, the following features, shown in Table VIII-15, are extracted. Those features designate the difference between features of the object 1 (obj1) and those of the object 2 (obj2).

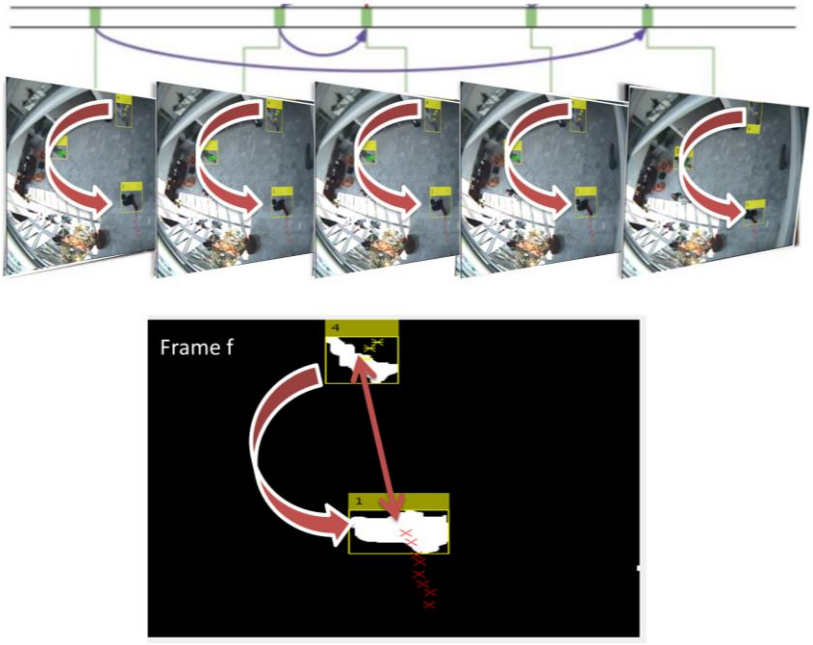


Figure VIII-13: Inter-Objects features.

Table VIII-15: Inter-object features	
S_bboxes_objs_diff	The bbox surface differences between the two objects, normalized by the frame surface ((S_bbox(obj1)- S_bbox(obj2)) / S_bbox(obj1))*100/ frame_width * frame_height).
S1_objs_diff	The surface differences between the two objects, normalized by the frame surface (S1_obj(obj1)- S1_obj(obj2) / S_bbox(obj1))*100/ frame_width * frame_height)..
S2_objs _diff	The surface differences between the two objects, normalized by the objects surface mean ((S1_ob(obj1)- S1_obj (obj2))/ S1_obj(obj1) *100/).
P_objs_diff	The perimeter differences between the two objects, normalized by the object surface (P_obj(obj1)-P_obj(obj2))*100/ (S1_ob(obj1)+S1_obj (obj2)/2).
Dist_objs	The relative distance between the two object centroids normalized by the frame surface (d(C(obj1), C(obj2))*100/ frame_width*frame_height).
Angle1_objs	The angle formed by the two objects vectors ($\overrightarrow{C(obj1)^{f-1}C(obj1)^{f}}$ and $\overrightarrow{C(obj2)^{f-1}C(obj2)^{f}}$).
Angle2_objs	The angle formed by the two objects vectors ($\overrightarrow{C(obj1)^{f-5}C(obj1)^{f}}$ and $\overrightarrow{C(obj2)^{f-5}C(obj2)^{f}}$).
Angle3_objs	The angle formed by the two objects vectors ($\overrightarrow{C(obj1)^{f-1}C(obj2)^{f-1}}$ and $\overrightarrow{C(obj1)^{f}C(obj2)^{f}}$).
Angle4_objs	The angle formed by the two objects vectors ($\overrightarrow{C(obj1)^{f-5}C(obj2)^{f-5}}$ and $\overrightarrow{C(obj1)^{f}C(obj2)^{f}}$).
Hu_objs_diff	The Hu invariant moments vector difference (7 moments) of the two object (sum(abs(Hu(obj1)- Hu(obj2))).
Huα_ objs_diff	The Hu invariant moments α difference of the two objects (Huα(obj1)- Huα(obj2)), where α ϵ [1,…,7].
Form_objs_diff1	The difference between the two objects forms normalized by the objects surface mean, in a way similar to Form_change1(obj).
Form_objs_diff2	The upper part only of the difference between the two objects forms normalized by the objects surface mean, in a way similar to Form_change2(obj).
Objs_width_diff	The normalized difference between objects widths ((Obj_width(obj1) - Obj_width(obj2))/Obj_width(obj1)) *100/frame_width.
Objs_height_diff	The normalized difference between objects heights ((Obj_height(obj1) - Obj_height(obj2))/ Obj_height(obj1)) *100/frame_height.
Int_mean_objs_diff	The normalized difference of the objects intensity means ((Int_mean(obj1)- Int_mean(obj2))*100/ Int_mean(obj1)).
Int_std_objs_diff	The normalized difference of the objects intensity standard deviations ((Int_std(obj1)- Int_std(obj2))*100/ Int_std(obj1)).
RGB_mean_objs_diff	The normalized difference of the objects RGB means ((RGB_mean(obj1)- RGB_mean(obj2))*100/ RGB_mean(obj1)).
RGB_std_objs_diff	The normalized difference of the objects RGB standard deviations ((RGB_std(obj1)- RGB_std(obj2))*100/ RGB_std(obj1)).

D. **<u>Inter-Frames Features:</u>** We take a window of M frames. In this window, and **for each frame**, we extract:

- The spatial features of object one and object two
- The temporal features of object one and object two
- The inter-objects features for object one and object two

For almost each of the features, we extract the derivative and second derivative, if it can be applied. Then for **each of the features (f)**, **its derivative (df)** and **its second derivative (ddf)**, we find seven global **inter-frames features** as follow in Table VIII-16.

Table VIII-16: Inter-frames features	
Min/Max	The minimum value of the f/df/ddf between the window frames, normalized by the maximum value between the window frames.
First/Max	The value of the f/df/ddf in the first frame of the window frames, normalized by the maximum value between the window frames.
Last/Max	The value of the f/df/ddf in the last frame of the window frames, normalized by the maximum value between the window frames.
Middle/Max	The value of the f/df/ddf in the middle frame of the window frames, normalized by the maximum value between the window frames.
Average/Max	The average value of the f/df/ddf vector of all the window frames, normalized by the maximum value between the window frames.
Median/Max	The median value of the f/df/ddf vector of all the window frames, normalized by the maximum value between the window frames.
STD/Max	The standard variation value of the f/df/ddf vector of all the window frames, normalized by the maximum value between the window frames.

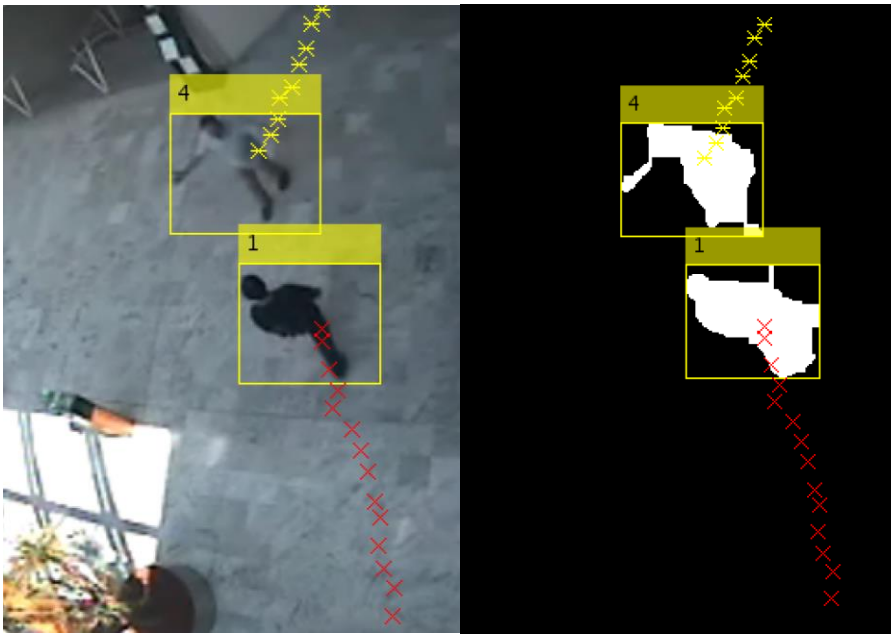
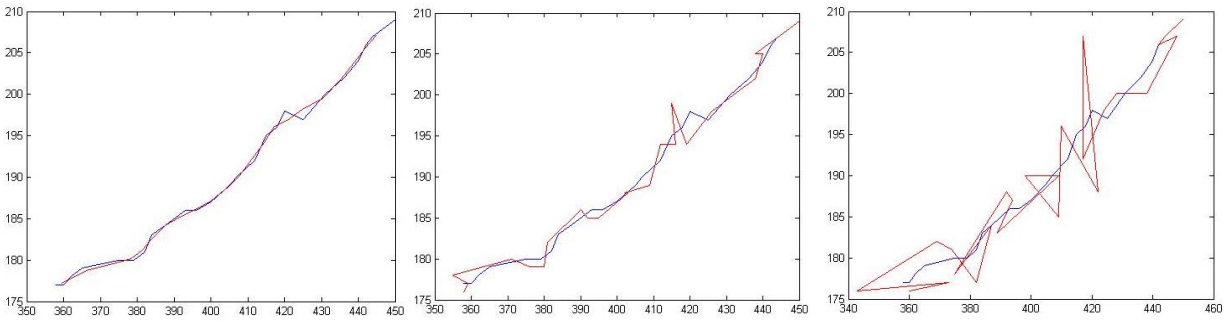


Figure VIII-14: Objects Trajectories

E. **<u>Trajectory Features:</u>** The last set of features is related to trajectory. For that, we compute the trajectory of object one and object two centroids through window frames, and the trajectory of the middle points between object one and object two centroids through window frames. For each of those three trajectories, we apply three smoothing filters:

- **Average filter:** This filter generates new trajectory points, where each point is the average between the precedent and the subsequent one. The goal here is to smooth rough trajectories obtain at the previous step. See an example in Figure VIII-15-a.
- **First order prediction filter**: This filter generates new trajectory points p', where each point is predicted from the end point of the last speed vector of earlier movements ($\overrightarrow{p^{-2}\ p^{-1}}$) and where its start point is the precedent one p^{-1}. See an example Figure VIII-15-b.
- **Second order prediction filter**: This filter generates new trajectory points p'', where each point is predicted from the position of the previous point on the trajectory, from the last speed vector, and is corrected with the last acceleration vector. See an example Figure VIII-15-c.



a b c

Figure VIII-15: Original trajectories are in blue. The new trajectories after filtering are in red. a- Average filter, b- First order prediction filter, and c- Second order prediction filter

VIII.7. Appendix 7: Graph representation of "Object characteristics template"

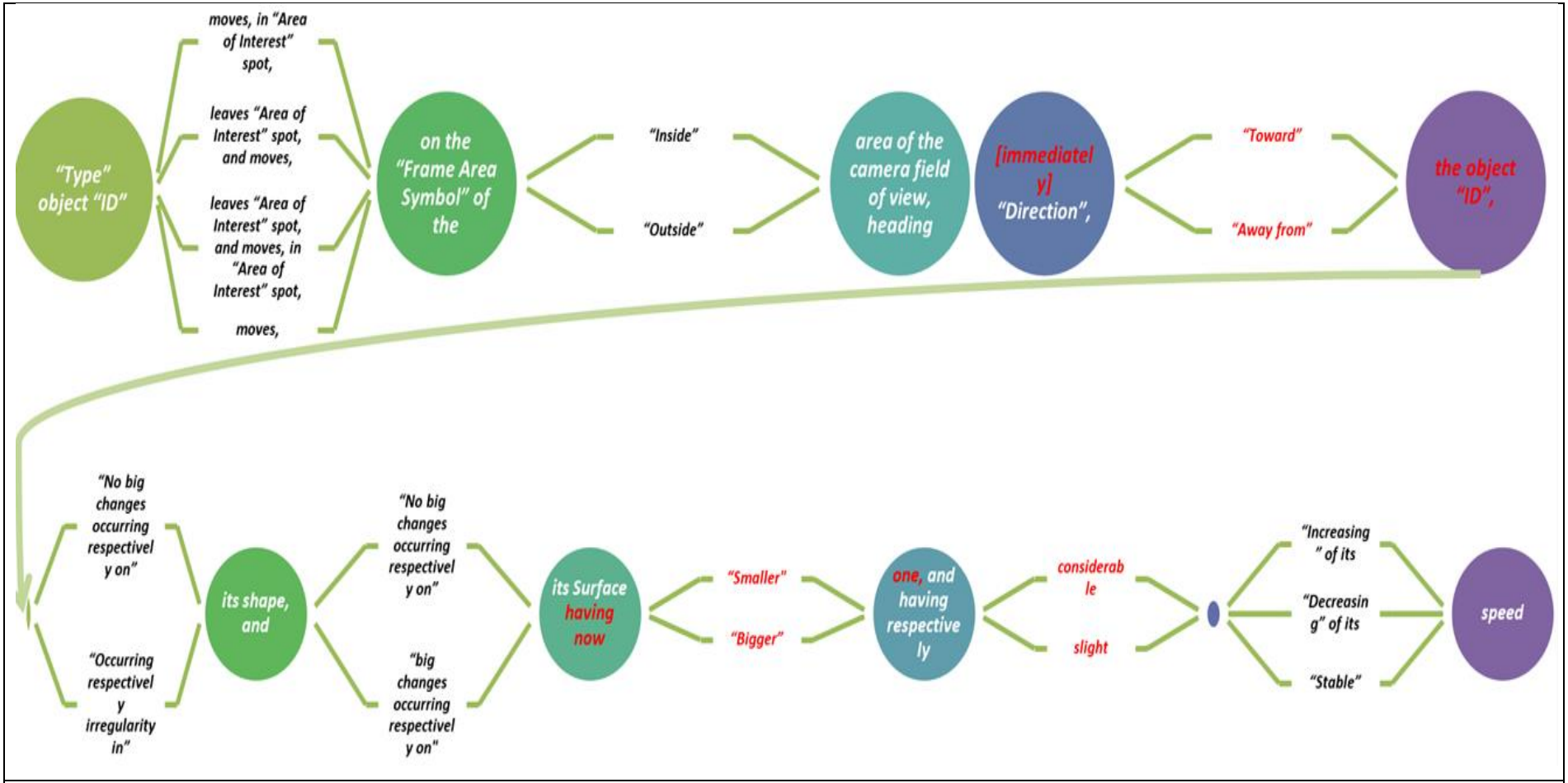


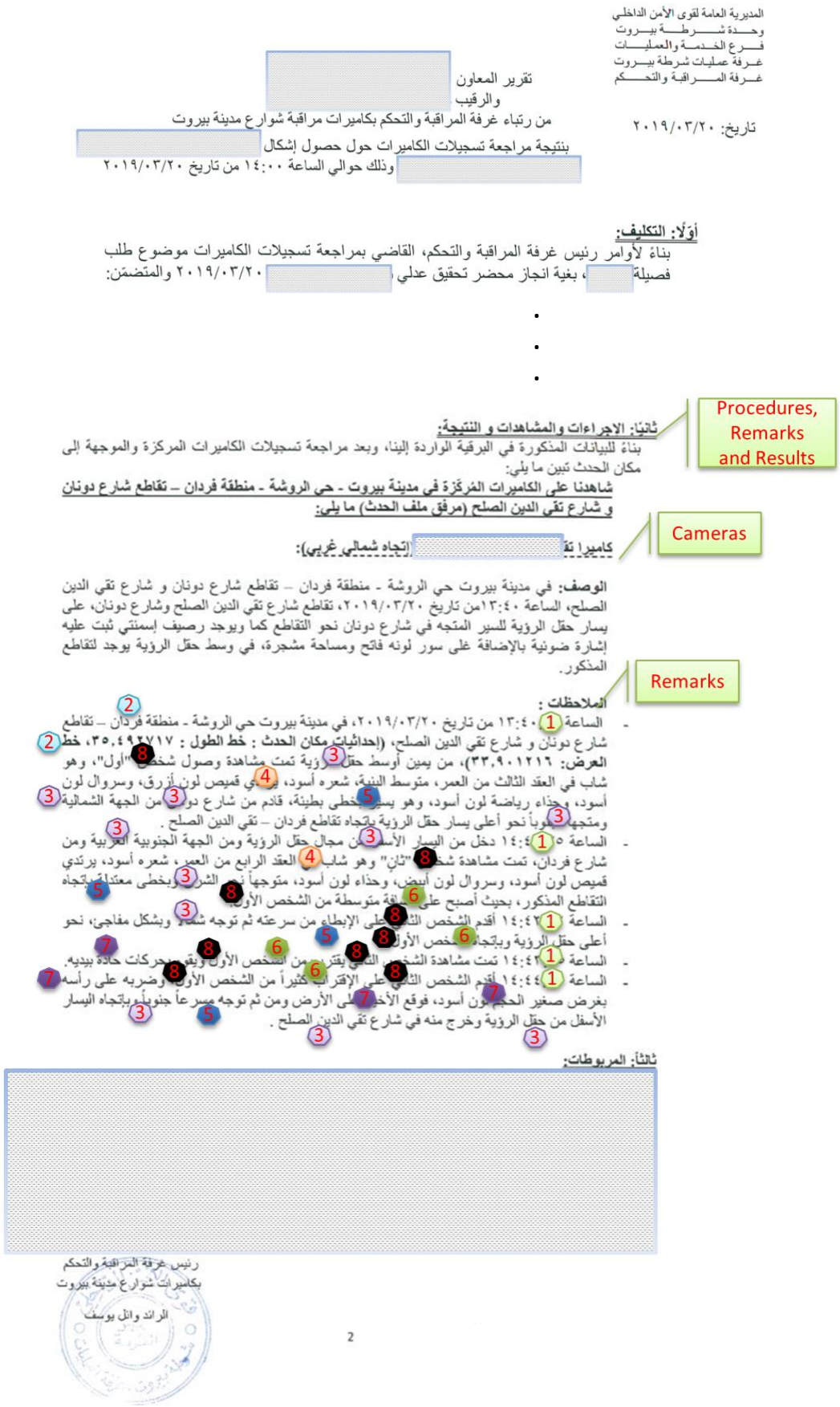
Figure VIII-16: Graph representation of "Object characteristics template". Clauses in red indicate that the clause may not be present in the output sentence (to be decided according to rules).

VIII.8. Appendix 8: Sample of a CCTV report

In the Figure VIII-17, a sample of police reports issued from Beirut CCTV control room is shown, describing an incident of attack from a person on another. Where, in this incident report, tells that a time X date Y in Beirut location Z GPS (X,Y), on the right middle of the camera field of view, a person 1 enters the scene, having middle size (and clothes descriptions), having slow speed, heading south near the intersection XX. Also, similar description for another person 2 entering the camera field of view, where at a specific time, he slow down its speed, heading immediately north toward the person 1. At specific time he approaches to person 1, performing aggressive moves with his hands. Finally, at specific time, merge with person 1, and hit him on his head with black small object (physical interaction), person 1 fall down, person 2, heading south, on the down left of the camera field of view, increasing its speed, and exits the scene from street YY.

In this report, the four W's and the How are all answered. We highlighted in Figure VIII-17 some key words, as follow: Time and date (1), location and GPS (2), position in the frame, position to an area of interest and direction (3), physical description (size, closes colours,…) of person 1 & 2 (4), speed (5), distance, and position and direction compared to the other person (6), interaction (7), and object 1 and object 2 (8).

It is clearly noticed the similarity of these key words with our description.


المديرية العامة لقوى الأمن الداخلي
وحدة شرطة بيروت
فرع الخدمة والعمليات
غرفة عمليات شرطة بيروت
غرفة المراقبة والتحكم

تقرير المعاون
والرقيب
من رتباء غرفة المراقبة والتحكم بكاميرات مراقبة شوارع مدينة بيروت
بنتيجة مراجعة تسجيلات الكاميرات حول حصول إشكال
وذلك حوالي الساعة ١٤:٠٠ من تاريخ ٢٠١٩/٠٣/٢٠

تاريخ: ٢٠١٩/٠٣/٢٠

أوّلاً: التكليف:
بناءً لأوامر رئيس غرفة المراقبة والتحكم، القاضي بمراجعة تسجيلات الكاميرات موضوع طلب فصيلة ، بغية انجاز محضر تحقيق عدلي ٢٠١٩/٠٣/٢٠ والمتضمّن:

Procedures, Remarks and Results

ثانيًا: الاجراءات والمشاهدات و النتيجة:
بناءً للبيانات المذكورة في البرقية الواردة إلينا، وبعد مراجعة تسجيلات الكاميرات المركزة والموجهة إلى مكان الحدث تبين ما يلي:
شاهدنا على الكاميرات المُركّزة في مدينة بيروت - حي الروشة - منطقة فردان – تقاطع شارع دونان و شارع تقي الدين الصلح (مرفق ملف الحدث) ما يلي:

Cameras

كاميرا تق (إتجاه شمالي غربي):

الوصف: في مدينة بيروت حي الروشة - منطقة فردان – تقاطع شارع دونان و شارع تقي الدين الصلح، الساعة ١٣:٤٠من تاريخ ٢٠١٩/٠٣/٢٠، تقاطع شارع تقي الدين الصلح وشارع دونان، على يسار حقل الرؤية للسير المتجه في شارع دونان نحو التقاطع كما ويوجد رصيف إسمنتي ثبت عليه إشارة ضوئية بالإضافة غلى سور لونه فاتح ومساحة مشجرة، في وسط حقل الرؤية يوجد لتقاطع المذكور.

Remarks

الملاحظات :
- الساعة ١٣:٤٠ من تاريخ ٢٠١٩/٠٣/٢٠، في مدينة بيروت حي الروشة - منطقة فردان – تقاطع شارع دونان و شارع تقي الدين الصلح، **(إحداثيات مكان الحدث : خط الطول : ٣٥،٤٩٢٧١٧، خط العرض: ٣٣،٩٠١٢١٦)**، من يمين أوسط حقل الرؤية تمت مشاهدة وصول شخص "أول"، وهو شاب في العقد الثالث من العمر، متوسط البنية، شعره أسود، يرتدي قميص لون أزرق، وسروال لون أسود، وحذاء رياضة لون أسود، وهو يسير بخطى بطيئة، قادم من شارع دونان من الجهة الشمالية ومتجهاً نحو أعلى يسار حقل الرؤية باتجاه تقاطع فردان – تقي الدين الصلح .
- الساعة ١٤:٤٠ دخل من اليسار الأسفل من مجال حقل الرؤية ومن الجهة الجنوبية الغربية ومن شارع فردان، تمت مشاهدة شخص "ثاني" وهو شاب في العقد الرابع من العمر، شعره أسود، يرتدي قميص لون أسود، وسروال لون أبيض، وحذاء لون أسود، متوجهاً نحو الشرق وبخطى معتدلة باتجاه التقاطع المذكور، بحيث أصبح على مسافة متوسطة من الشخص الأول.
- الساعة ١٤:٤٢ أقدم الشخص الثاني على الإبطاء من سرعته ثم توجه شمالاً وبشكل مفاجئ، نحو أعلى حقل الرؤية وباتجاه الشخص الأول.
- الساعة ١٤:٤٣ تمت مشاهدة الشخص الثاني يقترب من الشخص الأول ويقوم بحركات حادة بيديه.
- الساعة ١٤:٤٤ أقدم الشخص الثاني على الإقتراب كثيراً من الشخص الأول وضربه على رأسه بغرض صغير الحجم لون أسود، فوقع الأخير على الأرض ومن ثم توجه مسرعاً جنوباً وباتجاه اليسار الأسفل من حقل الرؤية وخرج منه في شارع تقي الدين الصلح .

ثالثاً: المربوطات:

رئيس غرفة المراقبة والتحكم
بكاميرات شوارع مدينة بيروت
الرائد وائل يوسف

2

Figure VIII-17: A sample of a police CCTV report, with highlights on key words.

Publications

1- Youssef, Wael. F., Haidar, S., & Joly, P. (2016). Classifying deformable and non-deformable video objects. 7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016), 9–6. https://doi.org/10.1049/ic.2016.0077

2- Youssef, Wael F., Haidar, S., & Joly, P. (2018). Generic Video Surveillance Description Ontology. 1st International Conference on Big Data and Cyber-Security Intelligence (BDCSIntell 2018), 6. Retrieved from http://ceur-ws.org/Vol-2343