C O P C A M S

Cognitive & Perceptive Cameras

Artemis-JU GA n°332913

D5.1 – Large Area Surveillance Applications Specification

WP5 –Applications & Field Tests

Editors:
Önder Altan (ASEL),
Metin Aktaş (ASEL),
Christian Fabre (CEA)

Date: 2016-01-07 at 14:13

Version: v2.0

Status: Public

CEA ref. number: LETI/DACLE/15-0903

http://copcams.eu

Document History

Date	Author	Modification
2015-03-18	Julie Foucault (CEA)	Template creation
2015-04-03	Bogdan Filipič (JSI)	Table of contents
2015-04-27	Metin Aktaş (ASEL)	First Draft
2015-06-01	Metin Aktaş (ASEL) Nizar Zorba (IQU)	System Level Simulator for communication infrastructure was added
2015-08-21	Metin Aktaş (ASEL)	"Demonstrator architecture and methodology", "Validation" and "Current state of the demonstrator" sections have been updated
2015-08-28	Önder Altan (ASEL)	“Field Test Scenario”, “Weak Observer Architecture”, “Expert Observer Architecture”, “System Functional Accuracy” and “Current State of the Demonstrator” sections have been updated.
2015-09-23	Jordi Serra (CTTC), David Pubill, Christos Verikoukis, Apostolos Georgiadis	Section 3.2 Distributed Detection of Events Based on WSN For Large Area Surveillance has been updated.
2015-09-27	Fabio Poiesi (QMUL)	Sections 3.1 Multi-target Detection and Tracking Experimental Setup, 5.1 and and 6 have been updated.
2015-10-02	Önder Altan (ASEL)	Inputs of the contributing partners have been integrated into the document v0.2. Conclusion has been added.
2015-10-20	A. Montalban; J.L. de Amaya; J. Llinares; F. Sacristán; A. Alcaide; E. Saltalamachia; J.L. Sánchez (CCTL)	Contributions to section 3.4 and 5.1
2015-11-02	Cyril Belgeron; Etienne Cappe(TCS)	Inputs have been added into the sections 5.1, 5.2 and chapter 6.
2015-11-02	Bogdan Filipic (JSI)	Editorial checks of the entire deliverable have been done.
2015-11-10	Christian Fabre (CEA)	Added CEA’s lab. experiment.
2015-11-12	Julie Foucault (CEA)	Final editorial changes

COPCAMS Partners

1	CEA	Commissariat à l'énergie atomique et aux énergies alternatives
3	TCS	THALES Communications & Security SA
4	TRT-FR	THALES Research & Technology France (repr. THALES SA)
5	INRIA	Institut national de recherche en informatique et automatique
7	CTTC	Centre Tecnològic de Telecomunicacions de Catalunya
8	CCTL	Concatel
9	IQU	Iquadrat Informatica S.L.
10	TECN	Tecnalia Research & Innovation
11	TED	Tedesys Global S.L.
12	UC	Universidad de Cantabria
13	GUT	Politechnika Gdańska
15	JSI	Institut ``Jožef Stefan''
16	DTU	Danmarks Tekniske Universitet / IMM
18	TRT-UK	THALES Research & Technology (UK) Ltd
19	QMUL	Queen Mary University of London
20	ASEL	ASELSAN Electronics Industry
21	KTOR	Kolektor Group d.o.o.
22	SOG	Sogilis
23	SQST	Squadron system

Table of Contents

List of Figures

List of Tables

1 Introduction

This document specifies the demonstration activities to show the effectiveness of COPCAMS solutions for the large area surveillance applications. The demonstration activities will be performed in two categories, i.e., a field test and laboratory experiments.

The scenarios for field test and laboratory experiments are explained in Section 2 and Section 3, respectively. In Section 3.1, the system architecture including the hardware and software configurations and the methodology for evaluating field test are explained. The validation of COPCAMS solutions on large area surveillance applications is given in Section 5 by explaining the state of the art at project start and the expected progress with COPCAMS project. Section 6 summarizes the achievements for field test and the document concludes in Section 7.

2 Demonstrator task

This section describes the field test scenario for “Large Area Surveillance Application” that is based on the requirements given in “D1.1 & D1.2 – Summary of Functional & Non-Functional Description” and use cases defined in “D1.4 – Summary of Use Cases and Field Test Definition” documents. The evaluation strategy and measurements to be collected are also described in this section.

The aim of Large Area Surveillance Application field test is to effectively monitor large areas with multiple cameras and extract meaningful information about the monitored area such as locating and classifying the moving object(s). The moving objects in the monitoring area are classified as ‘human’, ‘vehicle’ or ‘other’ based on either only the view captured by the camera at the central station or all the views available, i.e., views of the end node cameras in addition to the one at the central station.

2.1 Field Test Scenario

The field test will be performed in a test area in ASELSAN’s facility with the test setup illustrated in Figure 1. In this setup, there will be two fixed wide field of view (FOV) cameras and one narrow FOV Pan Tilt Zoom (PTZ) camera located on the surface of the building in the test area. The setup parameters of the cameras are given in Table 1. In demonstration, both for the fixed cameras and PTZ camera, Samsung SNP-3120VH camera will be used with different configurations. The specifications of Samsung SNP-3120VH camera are listed in Table 2. While, two fixed wide FOV cameras will be used in end node configuration, PTZ camera will be used in main station configuration as detailed in Section 3.1.

Based on this configuration, the following test scenario will be applied for several times with different conditions to measure the performance metrics that are defined in Section 2.2. The "target object" in the following scenario is used for "human", "vehicle" or "other[1]".

Demonstration Scenario

Throughout the demonstration, the cameras will be placed to monitor a restricted area, where the entrance is forbidden. In this case, a single "target object" will pass through the monitored area with speed in a range of 2.5-10 m/s. Then, Fixed Wide FOV Cameras on end node will detect a motion and send a meta data consists of the pixel locations of the detected "target object" and the extracted features about the "target object" to the main station via wired or wireless communication transmission. After that, on main station, the geographical coordinate of the moving object will be estimated from the pixel locations of the detected "target object" received from the end nodes. Then, PTZ camera on main station will be steered to the estimated position of the "target object". After steering, PTZ camera will start to capture the images and extracts features about the "target object". Then, multi view classification algorithm will be performed to show the class of the "target object" with a specified icon for "human", "vehicle" and "other". In meantime, the images captured by the PTZ camera will be processed with super resolution algorithm and the super resolved image will be monitored. If the detected "target object" is human or vehicle, main station commands the selected Fixed Wide FOV Cameras to stream their raw video and the received video streams from end nodes and the PTZ camera will be stored on main station for the evidence.

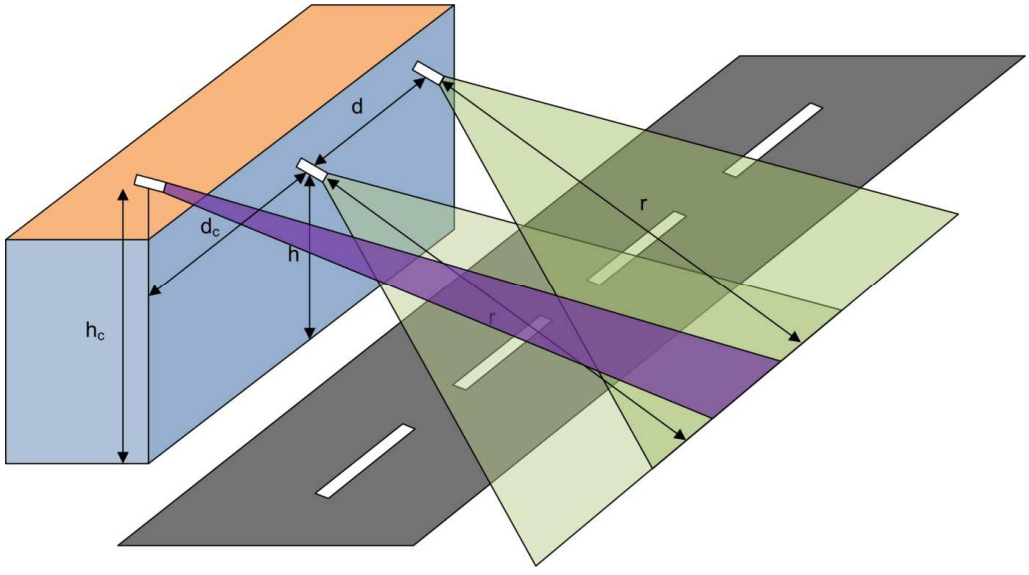

Figure 1: The illustration of test setup for Large Area Surveillance Application filed test

[1] "Other" term is used as any object that is large enough to be detected with motion estimation and different than the object used in training phase.

Table 1: The setup parameters of the cameras that are illustrated in Figure 1.

Camera Type	FOV	Relative Positions			Rotation	
		X	Y	Z	Azimuth	Elevation
Fixed	Horizontal: 54.44˚ Vertical: 42.32˚	10	0	7	0	-10
PTZ	Horizontal: 4.62˚ Vertical: 3.58˚	0	0	0	0	-10

Table 2: The specifications of Samsung SNP-3120VH camera

Camera Parameter		Value
Video	Imaging Device	1/4" Ex-view HAD PS CCD
	Total Pixels	NT : 811(H) x 508(V), PAL : 795(H) x 596(V)
	Effective Pixels	NT : 768(H) x 494(V), PAL : 752(H) x 582(V)
	Scanning System	Progressive(VPS ON) (If WDR on, Interlaced Scan)
	Frequency	NT : H : 15.734KHz / V : 59.94Hz, PAL : H : 15.625KHz / V : 50Hz
	Horizontal Resolution	Color : 600 TV lines
	Min. Illumination	Color : 0.7 Lux (F 1.65, 50 IRE, VPS OFF), 0.001 Lux (Sens up 512X) B/W : 0.07 Lux (F 1.65, 50 IRE, VPS OFF), 0.0001 Lux (Sens up 512X)
	S / N Ratio	50dB
	Video Out	CVBS : 1.0 Vp-p / 75Ω composite
Lens	Focal Length (Zoom Ratio)	3.69~44.32mm (12X)

	Max. Aperture Ratio	F1.65(Wide) / F2.01(Tele)
	Angular Field of View	H : 54.44˚(Wide) ~ 4.62˚(Tele) / V : 42.32˚(Wide) ~ 3.58˚(Tele)
	Min. Object Distance	0.2m (Wide) / 0.8m (Tele)
	Lens Type	DC Auto Iris
Pan / Tilt / Rotate	Pan Range	360˚ Endless
	Pan Speed	Preset : 650˚/sec, Manual : 0.05˚/sec ~120˚/sec (Turbo:200˚/sec)
	Tilt Range	-5˚~185˚
	Tilt Speed	Preset : 650˚/sec, Manual : 0.05˚/sec ~120˚/sec
Network	Ethernet	RJ-45 (10/100BASE-T)
	Video Compression Format	H.264, MPEG4, MJPEG
	Resolution	NT : 704x480, 640x480, 352x240, 320x240 PAL : 704x576, 640x480, 352x288, 320x240
	Max. Framerate	NT : 30fps / PAL : 25fps
	Video Quality Adjustment	H.264/MPEG4 : Compression Level, Target Bitrate Level Control MJPEG : Quality Level Control
	Bitrate Control Method	H.264/MPEG4 : CBR or VBR MJPEG : VBR
	Streaming Capability	Multiple Streaming (Up to 10 Profiles)
	IP	IPv4, IPv6
	Protocol	TCP/IP, UDP/IP, RTP(UDP), RTP(TCP), RTSP, NTP, HTTP, HTTPS, SSL, DHCP, PPPoE, FTP, SMTP, ICMP, IGMP, SNMPv1/v2c/v3(MIB-2), ARP, DNS, DDNS

	Security	HTTPS(SSL) Login Authentication Digest Login Authentication IP Address Filtering User access Log
	Streaming Method	Unicast / Multicast
	Max. User Access	10 users at Unicast Mode
	Web Viewer	Supported OS : Windows XP / VISTA / 7, MAC OS Supported Browser : Internet Explorer 6.0 or Higher, Firefox, Google Chrome, Apple Safari
	Central Management	Software NET-i viewer

2.2 Evaluation & Measurements

The described scenario in Section 2.1 will take place outdoor under ‘sufficient’ and ‘stable’ day light or artificial illumination with enough lux percentage, in addition to that no background change is allowed to happen, e.g., an object of the background such as a parked bicycle is assumed to stay in its place with no motion throughout the scenario. We assume that the system is correctly installed to cover the monitoring area and the cameras are calibrated.

Under these conditions, the following metrics will be measured for the performance evaluation.

- **Detection Rate:** For an evaluation of the first step in the single camera solution, i.e., the “other” / “anomaly” detection step, the detection rate of anomalies will be measured at a given range of bearable false alarm rates.
- **False Alarm Rate:** For an evaluation of the first step in the single camera solution, i.e., the “other” / “anomaly” detection step, the false alarm rate of anomalies will be measured at a given range of desired detection rates.
- **Classification Accuracy:** In the set of observations that are labeled as “nominal”, i.e., “not anomalous” or “other”, the accuracy for the task of “human” vs “vehicle” classification will be measured.

- **Total Transmission Bandwidth:** For an evaluation of communication overhead in distributed surveillance architecture, the total transmission bandwidth for a unit time or unit event will be measured.

3 Related Lab Experiments

3.1 Multi-target Detection and Tracking Experimental Setup

This section describes the Probability Hypothesis Density Particle Filter (PHD-PF) [14] that is a multi-target tracker developed by QMUL. PHD-PF is tested on publicly available video datasets and its performance is quantified using state-of-the-art tracking metrics.

3.1.1 Motivation

The aim of this work is the algorithmic optimization of PHD-PF in order to improve the speed performance (target: to achieve 15 fps) on embedded platforms (e.g. NVIDIA Jetson TK1). PHD-PF estimates the state of targets by propagating over time the cardinality (number) of targets. Being a Particle Filter-based approach, many operations have to be executed for each particle by making this algorithm computationally expensive while there is a version of the PHD filter that uses a closed form solution based on Gaussian Mixtures (GM-PHD) [15], which is computationally cheaper than the PHD-PF. However, GM-PHD does not offer the flexibility of choosing arbitrary target motion models and likelihood functions. This motivates our choice for the PHD-PF algorithm for multi-target tracking.

3.1.2 Scenario definition

We tested PHD-PF for monocular person tracking using. We used publicly available surveillance datasets, specifically Towncentre (http://goo.gl/aQiSdS), PETS2009-S2L1 (http://goo.gl/UNCCCl), TUD-Stadmitte (https://goo.gl/4MhNlQ) and iLids Easy (http://goo.gl/sfqYoT). These sequences contain a variable number of people as well as challenging situations of occlusions, motion variations and scale changes.

PHD-PF is developed in OpenCV running of CPU, and it is tested on a desktop computer with Intel i7 CPU 3.4GHz and 16Gb RAM. Object detection is performed using the OpenCV version of a person detector based on Histogram of Oriented Gradients (HOG) [16]. The object tracking algorithm is wholly developed by QMUL. The pipeline is tested as an end-to-end system: a video sequence is given as input and object trajectories are provided as output. Performance is evaluated using ground-truth information for object locations.

3.1.3 Evaluation strategy and Measurements

Tracking performance is quantified using Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Precision (P), Recall (R) and Identity Switches (IDS) [13]. MOTA is computed as

$$MOTA = 1 - \frac{\sum_{k=1}^{K}(c_1 FN_k + c_2 FP_k + c_3 IDS_k)}{\sum_{k=1}^{K} v_k},$$

where FN_k is the number of false negatives, FP_k is the number of false positives, IDS_k is the number of identity switches and v_k is the number of ground truth targets at time k.

MOTP is computed as

$$MOTP = \frac{\sum_{t=1}^{n_m} \sum_{k=k_{ini}^t}^{k_{end}^t} \frac{\left|E_k^t \cap G_k^t\right|}{\left|E_k^t \cup G_k^t\right|}}{\sum_{k=1}^{K} v_k},$$

where E_k^t is the estimated bounding box and G_k^t is the ground-truth bounding box of target t at k, k_{ini}^t and k_{end}^t are the initial and final time instant of target t.

Precision is computed as

$$P = \frac{|TP|}{|TP| + |FP|}$$

and Recall as

$$R = \frac{|TP|}{|TP| + |FN|}$$

where $|TP|$, $|FP|$ and $|FN|$ are the total number of True Positive, False Positive and False Negative trajectory estimations in the video sequence, respectively. An estimated trajectory is considered True Positive is the bounding box overlaps that of the ground truth at least for the 50%.

Sample results are shown in Table 3 & 4 by using 3000 and 300 particles per target, respectively. Tracking accuracy is generally higher in the case of PHD-PF using 3000 particles per target. A larger number of particles allows PHD-PF to have a better approximation of the target state space (more robust against noisy detections) and a better discrimination of targets over time (lower IDS). Fragmented tracks are also the cause of IDS because the same object will be associated to multiple identities by the PHD-PF. A large number of particles reduces the track fragmentation problem; this is why the number of IDS is lower in the case of 3000 particles. The major drawback of a large number of particles is the computation time per frame. During the experiments we measured that the

computation time per frame was between 2000 to 5000 ms per frame in the case of 3000 particles per target, whereas the computation time in the case of 300 particles per target was between 200 and 600 ms per frame. The large execution time in the case of 3000 particles per target is due the large number of targets inside Towncentre. Figure 2 shows an example of tracking result on Towncentre where a numerous targets are simultaneously tracked. In the case of TUD-Stadtmitte (Figure 3) the targets are fewer leading to a smaller computational time.

Table 3: Tracking performance of PHD-PF using OpenCV HOG person detector. 3000 particles per target are used.

Dataset	MOTA	MOTP	Precision	Recall	IDS
Towncentre	0.34	0.66	0.67	0.69	786
PETS2009-S2L1	0.12	0.63	0.61	0.39	106
TUD-Stadtmitte	0.58	0.75	0.85	0.73	27
iLIDS AB Easy	0.49	0.71	0.82	0.63	80

Table 4: PHD-PF tracking performance using OpenCV HOG person detector. 300 particles per target are used.

Dataset	MOTA	MOTP	Precision	Recall	IDS
Towncentre	0.34	0.66	0.68	0.68	894
PETS2009-S2L1	0.15	0.63	0.63	0.40	93
TUD-Stadtmitte	0.58	0.75	0.86	0.73	40
iLIDS AB easy	0.50	0.71	0.84	0.63	87

Figure 2: Example of PHD-PF tracking result on Towncentre where a large number of targets is tracked. Some of the targets are not tracked to do miss-detections. Semi-transparent box shows the estimated target state. Green box contour shows the detection.

Figure 3: Example of PHD-PF tracking result on TUD-Stadtmitte. Semi-transparent box shows the estimated target state. Green box contour shows the detection.

3.2 Distributed Detection of Events Based on WSN For Large Area Surveillance

This section describes the distributed detection system, based on WSN, proposed by CTTC as well as the specification of the scenario and the field tests strategy.

3.2.1 Motivation

Herein CTTC proposes the use of a distributed detection system for surveillance applications based on a Wireless Sensor Network (WSN). Moreover, the scenario definition and the associated field tests are described as well as the evaluation strategy and the necessary measurements. The aim is to complement the video surveillance system, proposed by other partners. Namely, the proposed detection system is based on sensors which are not based on video information but on other type of information e.g. sonar sensors. This approach permits to complement the video surveillance system in situations where the video information may be degraded, e.g. at night, in rainy or cloudy situations or in areas that cameras do not monitor properly. Moreover, the proposed WSN based system may provide useful side information for the main video surveillance system. For instance, the use of sonar sensors leads to obtain the distance of the detected object or person. And this distance can be used in the main station of the video surveillance system as side information to estimate the position of the detected object and to steer the PTZ camera properly.

3.2.2 Scenario Definition

The field test scenario is located in the CTTC facilities and the setup of the detection system is as follows. An array of multiple sonar sensors is considered to take the measurements, see Figure 4. The sonar sensors used for the experiments are the LV-MaxSonar-ez1 MB1010 by Maxbotix, whose main features are given in table 3, see [19] for further information. Herein this type of sensors are

considered to detect objects and persons in the perimeter of the monitored area. Each sonar sensor is connected to an IEEE 802.15.4 WSN node, namely the Z1 motes from Zolertia are chosen, whose main features are given in table 4, see [21] for further information. The connection between the sensor and the WSN node is done through an analog port (the Phidgets 5V port in the Zolertia Z1 mote). Moreover, the sensors provide a voltage level, related to the measured distance according to the next formula:

$$d = 2.54\frac{V_m}{8}$$

Where d is the detected distance in cm. V_m is the voltage related to the measurements that the sensor outputs between 0 V and 5 V. The scaling 8 accounts for the normalizations factors related to the volts per inch provided by the sensor and the quantization levels of the Zolertia Z1 ADC. Finally, the factor 2.54 converts from inches to cm.

On its turn, the WSN processes the measurements provided by the sonar sensors in a distributed manner. Namely, each node takes a buffer of measurements, which is provided by the sensor plugged to it, and applies a detection algorithm. Several alternatives will be considered to implement these algorithms. On the one hand, a detection method based on a heuristic threshold will be implemented. Where the threshold will be set to a value higher than the noise variance and will be determined experimentally. The other detection algorithm that will be considered is a Generalized Likelihood Ratio Test (GLRT), which is a conventional method widely applied in statistical detection theory, see e.g. [18], and consists on deciding a positive detection based on the next test statistic:

$$\sum_{n=0}^{N-1} x^2(n) > \gamma$$

Where, N is the number of samples in the processing buffer, $x(n)$ is a sample provided by the sensor and γ is a threshold whose value depends on the noise variance and the desired probability of false alarm.

Afterwards, the decision of each node is sent via a wireless link to a fusion centre. Note that thanks to the distributed approach each node has to send only one bit after each processing buffer of length N, instead of sending the N samples to the fusion centre. Namely, a bit with value 1 will be sent in case that an object is detected. At this point, it is important to mention that the software of the WSN nodes (needed to carry out the tasks described above) runs on Contiki OS, an open source operative system that is widely adopted for the WSN and the Internet of Things (IoT) worlds [17].

Regarding the fusion centre, it consists of an IEEE 802.15.4 Zolertia Z1 node, which acts as a WSN sink, and is connected via USB to a Raspberry Pi 2, which implements the fusion algorithm and

the communication with external systems. Namely, the data fusion method receives the detection decisions taken by each node and combines them to improve the detection performance compared to the one of a single node. The fusion method that we will consider is the standard "k out of n" fusion rule, see [20]. This method contains as particular examples logical rules such as the AND or the OR functions. More specifically, the "k out of n" fusion rule decides that an object is detected if

$$u_1 + u_2 + \cdots + u_n \geq 2k - n$$

Where u_i is the decision of the $i-th$ WSN node, and its possible values are 1 if a detection is decided or -1 otherwise. Moreover, n is the number of WSN nodes, and k is a design parameter. The interpretation of the "k out of n" fusion rule is that if k or more WSN nodes decide that an object has been detected, then the global decision is to decide that an object is present within the monitored area.

It is assumed that each WSN node can reach the fusion centre in one hop. Moreover, for the field test purposes all the WSN nodes are assumed to be plugged to the electricity grid. It is worth to mention that the aim of the system is to detect an intruder within the monitored area. The system can detect both people and objects, though the classification process is beyond its scope and it is assumed to be carried out in the main video surveillance system.

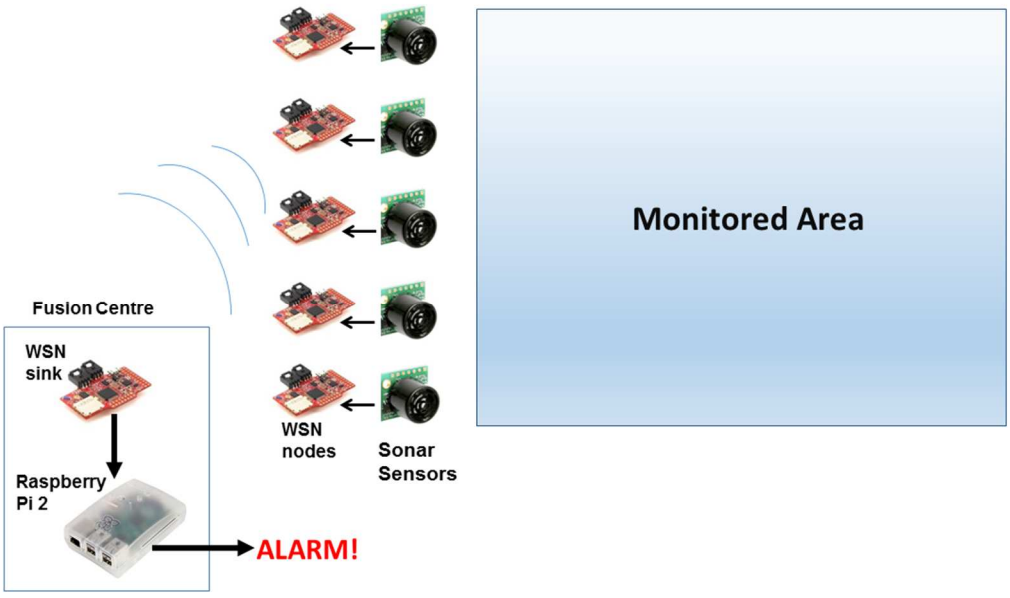

Figure 4: Setup of the detection system based on sonar sensors.

Table 5: Main feaures of the sonar sensor LV-MaxSonar-ez1 MB1010.

Electrical properties	2.5V DC to 5.5V DC voltage supply with 2mA typical current draw. Max current consumption 3 mA.Output impedance 14.7 kΩ.
Readings rate	Up to 50 ms.
Detection range	From 15.24 cm up to 6.45 m.
Sensor type	Distance (sonar)
Sensor output type	Ratiometric
Measurement Distance Resolution	25.4 mm
Operating temperature	From -40ºC to 65ºC.

Table 6: Main features of the Zolertia Z1 WSN node.

Micro Controller Unit (MCU)	16 bit ultralow power MCU based on the 2nd generation of the MSP430 by Texas Instruments.
Memory	92 KB flash, 8 KB RAM
Wireless communication	2.4 GHz IEEE 802.15.4
RF transceiver	Widely adopted CC2420 by Texas Instruments plus an embedded or external antenna.
Operative System (OS)	Contiki
Analog I/O's interfaces	2 x 3V Phidgets, 2 x 5V Phidgets

3.2.3 Evaluation Strategy and Measurements

In order to assess the proposed detection system the next assumptions are supposed to hold:

- There is not background change in the monitored area. That is, the objects present in the monitored area are static to avoid unnecessary false alarms.
- The WSN nodes are plugged to the electricity grid.
- The WSN nodes reach the sink in one hop.

The aim of the evaluation procedure is to assess the detection performance metrics as a function of the system parameters and event parameters that have a direct influence on them. Namely, the performance metrics are:

- The detection rate.
- The false alarm rate.

Moreover, the event parameters that have a direct influence on the metrics are:

- The amplitude of the event.
- The duration or length of the event (in samples provided by the sensor).

The system parameters are:

- The threshold of the detector method applied at each WSN node.
- The number of WSN nodes.
- The length (in samples) of the processing buffer at each node. This is the window where the detection algorithm is applied at each WSN node.
- The total number of processing buffers.

A twofold strategy is proposed to carry out the experiments. In the first approach, a hybrid experimental and simulated procedure is proposed to assess thoroughly the performance metrics. More specifically, the WSN will monitor the area in a real experiment and the events will be added artificially in each mote at a given number of processing buffers. Moreover, the number of processing buffers will be high enough to obtain reliable statistical values of the performance metrics. In this way, the amplitude and the length of the event can be controlled and we can obtain plots of the performance metrics as a function of them.

The second approach, to carry out the experiments, will consider real events and the aim is to see whether the system obtains a performance which is within the bounds predicted by the first hybrid approach.

3.3 Communication Infrastructure Simulation

This section describes the System-Level Simulator (SLS) developed by IQU and its implementations related to large area surveillance application.

3.3.1 Motivation

Modern mobile applications have boosted the amount of video content exchanged among user equipment terminals (UEs), which participate in Wi-Fi based Device-to-Device (D2D) networks. Cooperative techniques and Network Coding (NC) are widely used for enhancing the performance of D2D communication and alleviate the wireless channel access issues. Bidirectional video transmission, with its stringent bandwidth and Quality of Service (QoS) requirements, can greatly benefit from such advanced techniques to improve user experience without increasing network congestion.

Digital video is a key driver of the explosion in mobile data traffic of Long Term Evolution (LTE) networks, due to the increased expansion of demanding multimedia applications, such as video streaming, online gaming, social media networking and Web TV, among others. Mobile carriers face complex technical challenges, as the QoS requirements of delay sensitive applications, such as video traffic, have to be met without inflating the capital (CAPEX) and operational (OPEX) expenditures of cellular networks. Concurrently, the user experience should be maintained in high levels, unaffected by the escalating network load [22].

Nowadays, the LTE network performance is evaluated not only in terms of QoS, but also in terms of Quality of Experience (QoE), which is an upgraded indicator of the users' satisfaction with the offered service[23]. Especially for the case of video-based mobile applications, the QoE can be assessed by various video quality metrics, such as the Mean Opinion Score (MOS) [3].

In the last few years, various QoE evaluation models have been proposed, aiming to improve the user experience in video transmission scenarios over cellular networks. In [4], a video quality model and QoE optimization scheme have been presented, which aim to reduce the video distortion. Another framework for QoE inference is the MintMOS framework [5], which compares parameters of video streams in real time to QoE parameters already obtained by subjective quality assessment, in order to present realistic MOS values.

With the aim of improving the QoE for the mobile users, the offloading of mobile traffic to D2D connections seems to be a viable solution to the cellular network congestion problem. The direct connectivity among UEs is based on Wi-Fi links that reside in the unlicensed spectrum (*outband D2D*) [6]. The content sharing among UEs can be initiated in light of two main factors: i) the desire for data exchange with D2D bidirectional flows, as induced by numerous multimedia applications, and ii) the participation in cooperative communications, when the UEs serve as relays that support other UEs' communication.

Despite its capability to enhance user experience, outband D2D communication is affected by inherent issues of Wi-Fi connectivity. The contention for channel access among multiple UEs has a severe impact on the performance of D2D links. Additionally, bad channel conditions increase the number of packet retransmissions. To handle these problems, several cooperative MAC protocols have been already proposed. A considerable number of them utilize the NC technique, aiming to further improve the network performance. With NC, in the D2D context, the cooperating UEs can encode and transmit multiple overheard packets.

As advocated in [7], NC can be applied in cooperative MAC layer schemes, allowing the nodes to retransmit overheard packets of different flows. For relay-aided bidirectional communication under saturated conditions, the NCCARQ-MAC protocol [8] has been proposed. Nonetheless, with NCCARQ-MAC, the relays cooperate only if NC conditions are fulfilled. Furthermore, corrupted packets can be used for the retrieval of original packets exchanged between two nodes, as proposed in the NCAC-MAC scheme [9], in order to improve the throughput and delay performance. However,

this process requires strictly synchronized cooperative transmissions. Aiming to further exploit NC opportunities in D2D communications, the ACNC-MAC protocol [10] allows neighbouring UEs, which overhear packets and cooperation requests, to act as relays in bidirectional D2D transmissions.

Taking into account the increasing popularity of video services and the benefits of outband D2D communication, we will apply ACNC-MAC in a bidirectional video transmission scenario. We also propose a new performance valuation framework, which employs the Iquadrat (IQU) System-Level Simulator (SLS) for a realistic simulation of the aforementioned scenario. Our SLS faithfully emulates all aspects of the cooperative transmission and QoE prediction techniques, assessing the effects on perceived video quality.

3.3.2 Scenario Definition

In the D2D network, depicted in Fig. 1, two UEs (u_1 and u_2) are involved in bidirectional video communication (video sharing/conferencing). Poor wireless channel conditions might lead to packet losses in the D2D links, thus retransmissions might be required. These are performed by N neighboring idle UEs that can overhear and retransmit packets, acting as relays $\{r_1,..., r_i,\ldots, r_N\}$.

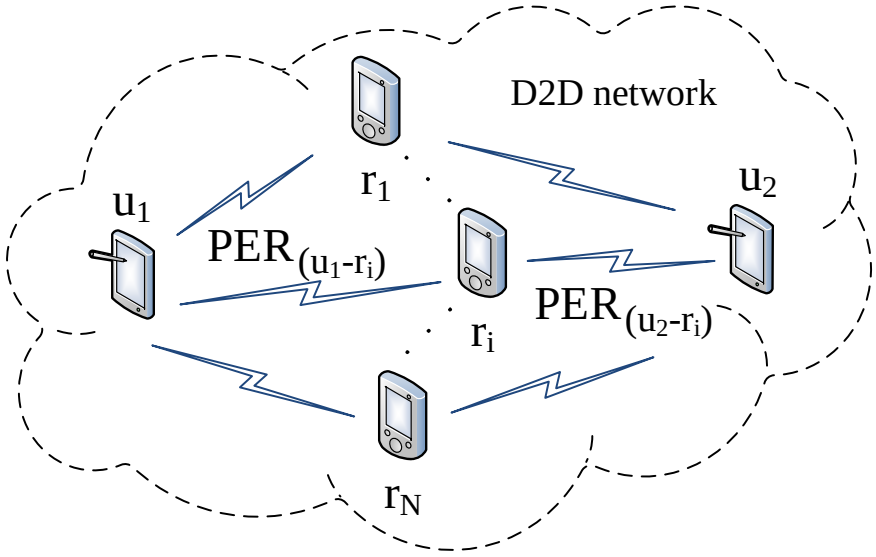

Figure 5: Outband D2D network topology

The exchanged video consists of either metadata or MPEG4 video frames, i.e., I-frames (or intra frames) P-frames (predictive frames) and B-frames (bidirectional frames). Packets to be transmitted are stored in the buffers of the two UEs. It must be noted that metadata transmission greatly decreases the required bandwidth, but on the other hand it is expected to be much more sensitive to losses. The MPEG4 decoder typically operates even after multiple packet losses (albeit with significant noise and visible artifacts) while we assume that even a single packet loss can't be tolerated in metadata transmission mode.

The UEs' transmissions are managed either with the 802.11 Distributed Coordination Function (DCF) MAC mechanism [11], which is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) method, or with the ACNC-MAC protocol [10]. According to the DCF rules, collisions can be resolved via retransmissions that employ an exponential backoff window. In the initial backoff stage, the value of contention window has the minimum value. After a collision occurs, the contention window is doubled, until the maximum value is reached. ACNC-MAC was chosen due to its particularly good performance in servicing bidirectional traffic, as in case of video conferencing applications.

Let us now provide a short description of the ACNC—MAC operation. Considering that u_1 and u_2 establish a bidirectional flow and wish to transmit packets p and p', respectively. According to ACNC-MAC protocol, after failing to decode packet p, u_2 sends a Request-For-Cooperation (RFC) packet, piggy-backing its own packet p' destined for u_1. Upon receiving the RFC, relays that have overheard and stored p in their buffer, will compete for channel access to assist in the packet retransmission.

Part of the strength of ACNC-MAC lies in the relay backoff value selection process, to avoid contention among relays. For each relay the choice of the contention window is based on the number of packets it has overheard. The relay that gains access to the channel will transmit an "Eager-To-Cooperate (ETC)" packet. This specifies the number of packets to be transmitted and also it indicates the expected number of ACK packets, so that the cooperation phase ends.

Relays with two packets stored in their buffer (one from each flow) are assigned a lower backoff value to make sure they gain access to the channel. If none of the relays has received both packets, but some of them have overheard only one packet, then a relay with one packet will be prioritized. In case that all relays have failed to decode any packet, an ETC packet, transmitted by the relay that gains channel access, terminates the cooperation round.

In a nutshell, three possible cases are identified:

- A relay has correctly received both p and p' packets and is able to perform NC. The XORed packet p $\oplus$ p' is piggy-backed to the ETC packet (Fig. 2).
- Only one of p and p' has been correctly decoded by the relay. This packet is again piggy-backed to the ETC.
- All relays fail to decode any packet, thus only an ETC packet ends the cooperation.

To make a fair comparison between DCF used as baseline and ACNC-MAC, we assume that the joint packet loss probability at the relay and packet destination in ACNC-MAC is equal to the packet error rate (PER). In Fig. 1, assuming relay r_i wins the contention phase and transmits its packet(s), we obtain:

$$(1 - PER_{u1-u2}) = (1 - PER_{u1-ri})((1 - PER_{u2-ri}).$$

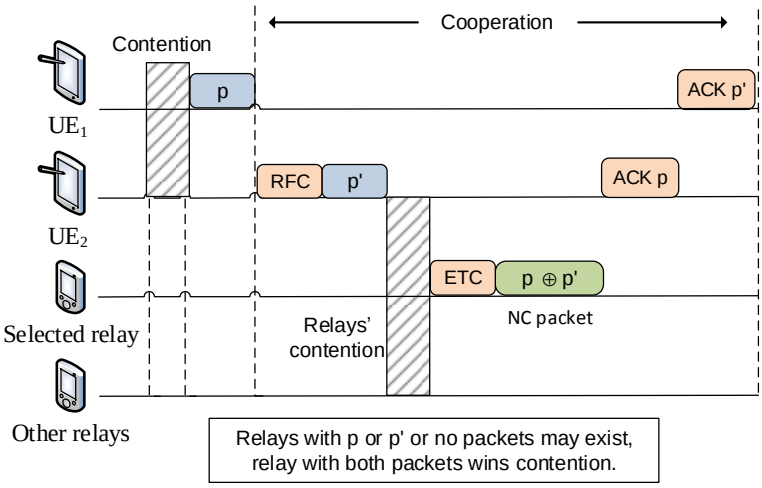

Figure 6: ACNC-MAC operation example

3.3.3 Evaluation Strategy and Measurements

The IQU SLS calculates a wide range of traffic statistics, both node-level and network-level, over the run-time of its operation. These statistics are stored in a set of trace files at the end of each simulation cycle, for further analysis with statistical software or for generating plots and diagrams. These include common QoS metrics such as packet error rate, packet delay and saturation throughput for given channel conditions. These metrics can be employed to evaluate the video metadata transmission, e.g. calculating the maximum number of supported video streams and the achievable throughput to guarantee certain QoS parameters.

Additionally, a new performance evaluation framework for the ACNC-MAC protocol is proposed, based on the IQU System-Level simulator (SLS) platform along with a Quality of Experience (QoE) prediction model. Our goal is to measure the effect of ACNC-MAC on enhancing user experience, quantified with the Mean Opinion Score (MOS) metric.

3.3.3.1 IQU System Level Simulator

The System Level Simulator (SLS) is a simulation platform for wireless networks. Its focus is in simulating Layer-2 protocols, but it also implements physical layer functionalities, i.e., simulating the underlying wireless channel and the propagation of wireless signals. The SLS is a flexible software tool, which allows rapid prototyping and validation of algorithms and scenarios. One of the key strengths of the SLS is the availability of a Graphical User Interface (GUI) which increases the efficiency of the simulation process. It also visualizes the operation of the network in real time. The SLS has been developed in the C++ programming language, using Microsoft .NET Framework.

The ACNC-MAC protocol was implemented as a new module which was added to the SLS, to evaluate its performance in a close-to-realistic environment. Protocol implementations at the SLS are in

the form of Finite State Machines (FSMs). The SLS node model was also accordingly modified to support multicasting of XORed packets, which is required by ACNC-MAC protocol.

In Figure 3, the main SLS screen is depicted, with two UEs exchanging video traffic and two relays assisting in the transmission. A color code is employed, representing the operating state of the mobile stations (e.g., green is for transmitting and blue for receiving stations). During run-time, the GUI interoperates with the SLS engine which implements the networking protocols and the components which are responsible for generating traffic, calculating traffic statistics, and writing trace files. The simulation is controlled by a set of buttons (Play, Pause, Step, Stop). The SLS allows the pause of the simulation at any moment to inspect the variables and operation state, and then continue the simulation or advance the time step-by-step. This is helpful for validating the correct operation of networking protocols, facilitating protocol development.

It must be noted that time representation at the SLS is measured in time-step intervals (or slots), of 10 μs. The main simulation loop advances time by one time interval per simulation step. The SLS modules all have a common time representation and in each simulation step all modules interact with each other to implement the network operations.

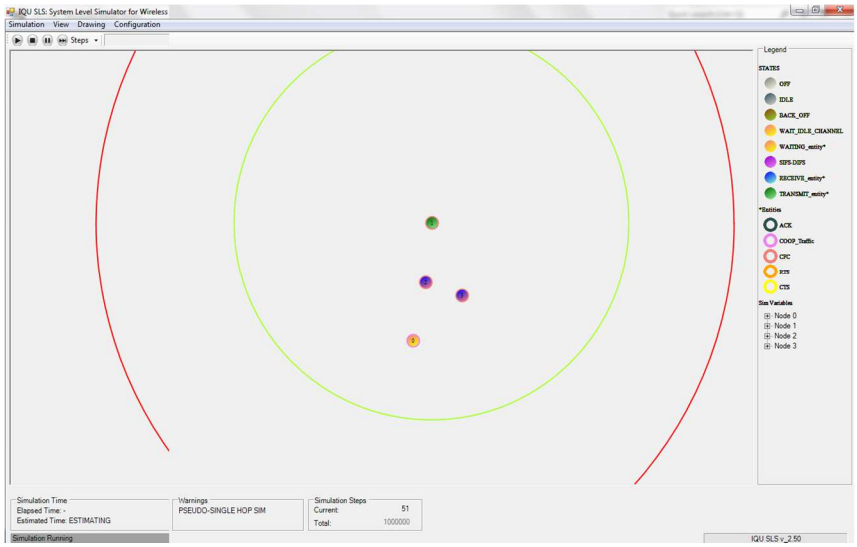

Figure 7: SLS Main screen

3.3.3.2 Measurement Metrics

Traditional QoS metrics, such as delay and packet loss ratio, are already supported by the SLS and as previously mentioned can be employed to evaluate the metadata video transmission. For example, we can calculate the maximum number of supported video streams for given channel conditions and calculate the maximum throughput to guarantee a given packet delay or a given packet error rate probability. However, QoS parameters are not accurate predictors of user experience when video traffic (i.e., MPEG 4 frames) are transmitted. Thus, to assess the efficiency of MAC protocols in perceived video quality without resorting to costly field tests, several QoE prediction models have been proposed

in the literature. In our performance evaluation framework, we employ the QoE prediction model described in [12], which takes into account the effect of packet losses in different frame types. The authors employ a database of videos for a range of different packet error rates, and employ Video Quality Model (VQM) to assess their quality. This process has a very good correlation with the MOS score, which evaluates the perceptual video quality as experienced by experts. However, VQM assessment it is still too resource intensive to be employed in a real-time network simulation. The output of the author's analysis is a linear model that predicts the MOS score from I_{loss}, B_{loss} and P_{loss}, namely, the frame loss ratios of I-frames, B-frames and P-frames, respectively:

$$\mathrm{MOS} = 4.9 - 1.08 \cdot \mathrm{I_{loss}} - 3.28 \cdot \mathrm{B_{loss}} - 3.23 \cdot \mathrm{P_{loss}}$$

In the abovementioned model, the authors assume that a single packet loss at an I-frame is recoverable (as long as it is not at the frame header) and P-frame losses have a bigger impact on video quality than B-frame losses.

To implement the aforementioned QoE prediction model at the SLS, we added support for the MPEG4 Group of Pictures (GOP) pattern, which specifies the order of frame types. The GOP starts with an I-frame followed by two B-frames (denoted as an "IBB" pattern) and then by a pattern of multiple "PBB" patterns, as depicted in Fig. 4. The number of frames in a GOP is referred to as the GOP length. The three different frame types in the GOP support a different compression ratio:

- I-frames are frames that can be independently decoded, and have the lowest compression ratio. An average 7:1 compression ratio is assumed.
- B-frames have the highest compression ratio, by referencing past and future frames. An average 50:1 compression ratio is assumed.
- P-frames stand in between I-frames and B-frames, with an average compression ratio of 20:1.

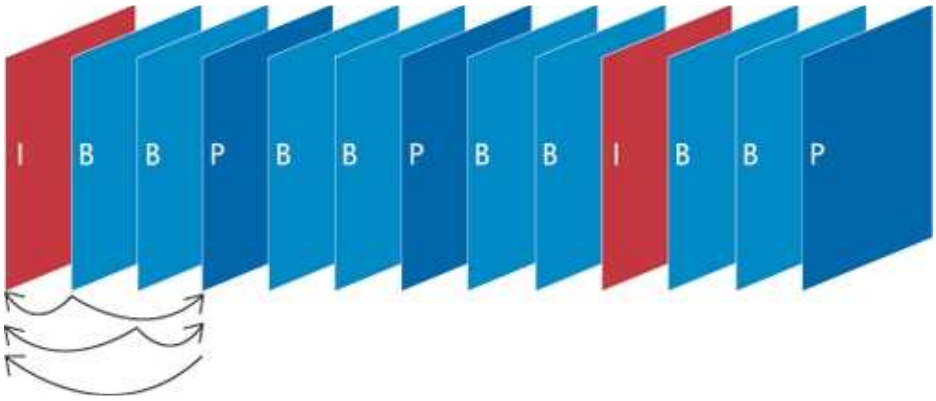

Figure 8: MPEG4 video GOP structure

To simulate the abovementioned GOP structure at the SLS we mark transmitted packets accordingly. The GOP structure starts with an "IBB" pattern, with the I-frame taking up 14 packets and each B-frame 2 packets. Then the "PBB" pattern is repeated 8 times, with P-frames taking-up 5 packets and each B-frame again 2 packets. It can be seen that the relative sizes of the frame types reflect their average compression ratio.

3.3.4 Results

We have programmed the QoE prediction for video transmission in the IQU SLS and we will provide the results soon.

3.4 Cognitive and Perceptive Cameras Systems for Smart Facility Management Domain

3.4.1 Motivation

Facilities management is gaining increasing recognition as a significant contributor to the overall effectiveness of many organisations. Smart Facility and Building Management (SF&BM) generally involves a number of disciplines and services. The most general description to identify the market segment is understanding Smart F&BM as integrated management process that considers people, process and place in organisational context, being focused in the design and improvement of intelligent buildings (IB) and the coordination and optimization of several domains: facilities, life security, physical security and information technology. In this context, companies are becoming more interested in exploring opportunities to consolidate multiple services from single suppliers as a way of improving value. There is a significant consolidation opportunity for service providers able to deliver an integrated solution.

With buildings responsible for about half of all energy consumption and greenhouse gas emissions, establishing, managing, optimizing, and maintaining sustainability objectives is becoming a core driver. Also, forward-looking companies and public entities are adopting a new approach, where not only a coordinated work and integrated I&FM service is being required, but also providing new smart services that can take corrective actions automatically.

Smart Buildings – Automatic Corrective Actions

.One of the key trends is to provide solutions that can take remedial actions automatically, providing a coordinated response in the “foundational systems” such as security, electrical distribution or HVAC (heating, ventilation, and air conditioning).

Smart Video Applications

While the video surveillance system is a mainstay of building security, it may serve many purposes. The analysis of digital images addresses aspects of physical security but may go way beyond that to provide data and information for building life safety, energy management and overall building performance.

However, though during last years a wide range of new applications within computer vision have been enabled, the network bandwidth, server processing and cost have been inhibitors for these opportunities up until now. Additionally, the traditional vision of a vertically structured market prevented the adaptation to a growing demand of dynamism and flexibility in the context of Smart Facility Management. The market is demanding not only more efficient, flexible and autonomous surveillance systems, but the integration of video systems to provide more data and information for energy management and enhanced building performance.

In this context, the Cognitive & Perceptive Video Systems (CPVS) enabled by COPCAMS would represent a significant step towards wider adoption of embedded vision systems within the smart facilities & smart building management domain. This new approach would provide advanced features in an emerging market, where and improved performance and reduced energy consumption will facilitate the use of embedded cameras not only as simple sensors, but as a distributed cognitive system, going beyond smart surveillance.

The motivation in this use case is to test and iteratively improve the approach (together with the use case in T5.3), in order to identify a minimum viable service (MVS) that can be provided to different clients as a comprehensive solution within the Smart Facilities and Smart Buildings Management domain.

The motivation for this use case is based on the potential use of a CPVS to provide different functions/profiles depending of different situations. That is, to explore the potential of COPCAMS approach, -with embedded and powerful vision systems- to sense the surrounding environment, and react to changes. In this case, the field test aims to explore the possibility of a COPCAMS system that is initially working in “asset recognition mode” to change to “surveillance mode” to detect intrusions in the specific zone. This can be used in different environments, as public or industrial facilities to control different assets (trolleys in airports, containers in maritime cargo terminals, special vehicles in public/private facilities e.g.)

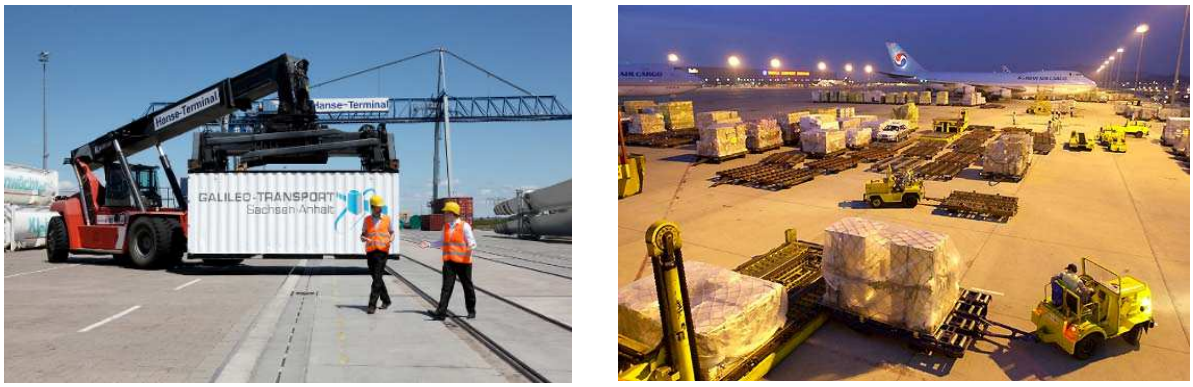

Figure 9: Cargo Terminals Sample

3.4.2 Scenario Definition

The field test scenario is initially simulated in CCTL facilities, and the setup of the system is as follows: A COPCAMS platform (initially a PC+GPGPU, but extensible to STHORM platform) with a single camera will be placed to monitor a working area, where specific assets must be identified. These assets will be identified thanks to a specific image pattern, and will remain stopped during a timeframe of 2-4 seconds, simulating a routine control.

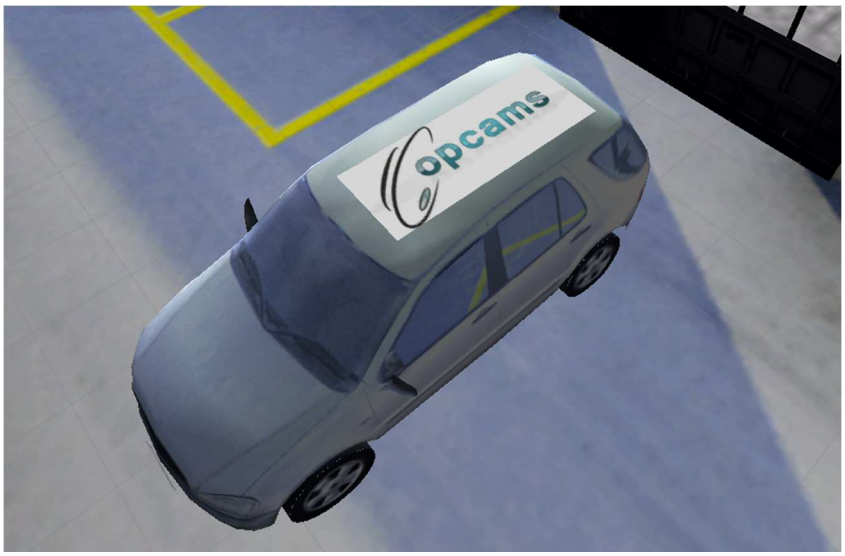

Figure 10: Sample Vehicle with Specific Pattern

On a particular moment (triggered by the end of working time or by a specific simulated alarm that will be captured by the system), the system will be required to change to surveillance mode, and detect human intrusions in that specific zone. The results will be registered, in order to be sent to a main station, that could feed a business intelligence unit, a dashboard or a decision making system.

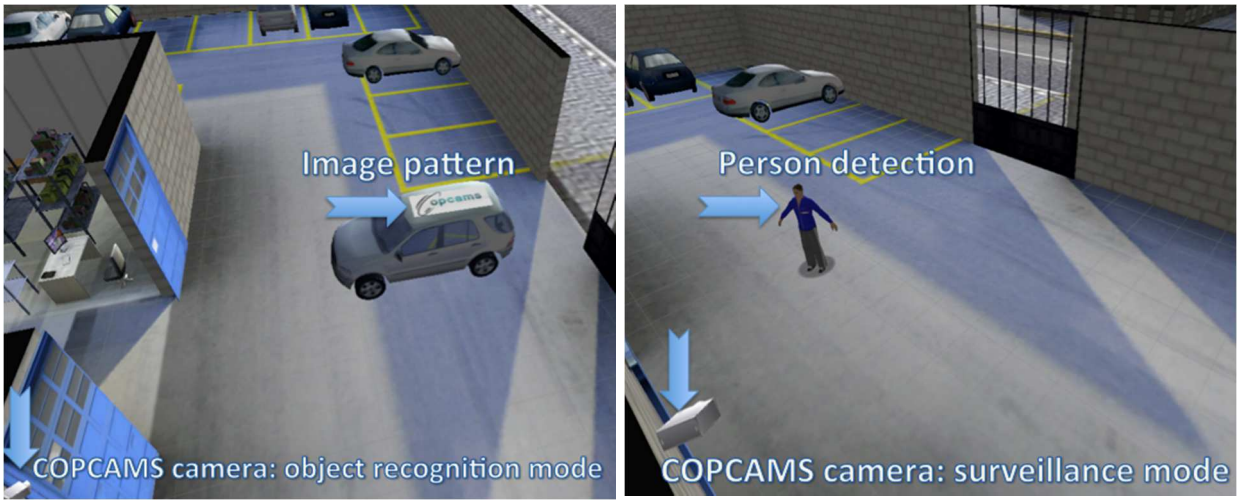

Figure 11: Sample Vehicle with Specific Pattern

3.4.3 Evaluation Strategy and Measurements

The described scenario will take place in a controlled area, with stable illumination. We assume that the system is correctly installed to cover the monitoring area and the cameras are calibrated.

Under these conditions, the following metrics will be measured for the performance evaluation.

- **Detection Rate:** For both modes, the detection rate will be measured.
- **False Alarm Rate:** For both modes, the false alarm rate will be measured.

The evaluation strategy will follow an iterative process, where the algorithms, and the overall COPCAMS performance will be analyzed in different target platforms.

3.5 Face Detection System

This section describe the *Face Detection System* developed by CEA using their HOE2 methodology.

3.5.1 Motivation

CPVS development teams have to cope with usual constraints of industrial organizations developing embedded systems, including: (1) End-to-End Engineering: the full development cycle goes from requirement formalization to the final integration and assessment of the application on its platform. (2) Incremental & collaborative development: To organize efficiently the work of large teams, it is critical to regularly distribute and integrate work, and to measure progress towards the objectives.

The motivation of this lab. experiment is to measure the benefits of a well-organized development method and assess its benefits, especially for reuse *e.g.* when a given applications has to be specialized when for several platforms.

3.5.2 Scenario Definition

The CEA scenario will be to develop a Face Detection System (FDS) using the ⟨HOE⟩^{2} method [24-27]. The FDS will be made of an *application* running on a *platform*. The development of each of those two systems will be initiated separately from their own *use cases*, presented Table 7 and Table 8 respectively. The platform's use cases will be implemented on two different hardware platforms: One made of a Raspberry and an Arduino, the other made of an i.MX6 board.

Table 7: Use Cases of the Face Detection System's Application

ID	Causality	Name
Description		
1	Primary	Detect presence
The actor wants to know when somebody enter the monitored zone.		
2	Primary	Track faces
The actor wants to track faces of people entering the monitored zone.		
3	Secondary	Toggle camera control
The actor wants to switch between manual and automatic tracking modes.		
4	Secondary	Query camera control mode
The actor wants to know the current tracking mode.		
5	Secondary	Orientate camera
The actor set the camera's orientation.		
6	Secondary	Query camera orientation
The actor wants to know the camera's current orientation.		

Table 8: Use Cases of the Face Detection System's Platform

ID	Causality	Name
Description		
1	Primary	Install firmware
The actor installs the new firmware on the platform.		
2	Primary	Execute the application
The actor executes the application.		
3	Secondary	Query date and time
The actor wants to know the current date and time on the platform.		
4	Secondary	Set date and time
The actor sets the date and time on the platform.		
5	Secondary	Query firmware version
The actor wants to know the current version of the firmware.		

3.5.3 Evaluation Strategy and Measurements

The goal of this experiment is to assess the gains in (1) code reuse, (2) tool reuse and (3) dev elopement time while developing an application for several platforms. We will compare the amount of hand written code against generated code and the reuse of code generation tools across the two platforms. We will also measure the time spent on modeling and developing the application and the two platforms.

4 Demonstrator Architecture and Methodology

The field test scenario described in Section 2.1 will be configured as a distributed heterogeneous sensor architecture as illustrated in Figure 12. In this configuration, each fixed wide FOV camera will be used as a “weak observer” whose task is to continuously monitor the scene of interest with low computational resources, seeking salient event/target traces. The PTZ camera will be used as an “expert observer” with much higher computational resources. When an interesting event/target is detected by a weak observer, the “weak observer” passes the corresponding information (pixel locations and extracted features of the detected event/target) to the expert observer. Then the “expert observer” looks into the detected event/target and provides the final decision. This approach is to be realized by ASELSAN’s proposed smart surveillance architecture.

In this architecture, the use of a high quality PTZ camera enables the system to gather richer sets of information, i.e., spatial and temporal features, which cannot be obtained by a fixed FOV camera only system. For instance, extracting histogram of gradient features (HOG) using low resolution images typically results in noisy features due to noisy gradients, and in turn degrades overall system performance. This issue can be rectified by zooming capability of a high quality PTZ camera. Furthermore, the use of a PTZ camera facilitates the manual visual inspection of a detected event/target, which is significantly critical in eliminating highly undesired false alarms/ misdetections, as well as providing information on corner cases that can be corrected by improving the system accordingly.

The proposed cluster based smart surveillance system can provide wide area coverage with the advantages described above at a low overall cost. This is due to the use of multiple cheap, wide and fixed FOV cameras providing full coverage of a wide area at all times; and the use of a single PTZ camera providing higher quality coverage of a smaller portion that is of interest if/when necessary. The fixed FOV cameras transmit metadata or encoded video frames only when salient event/target detection occurs, reducing data transmission and power consumption.

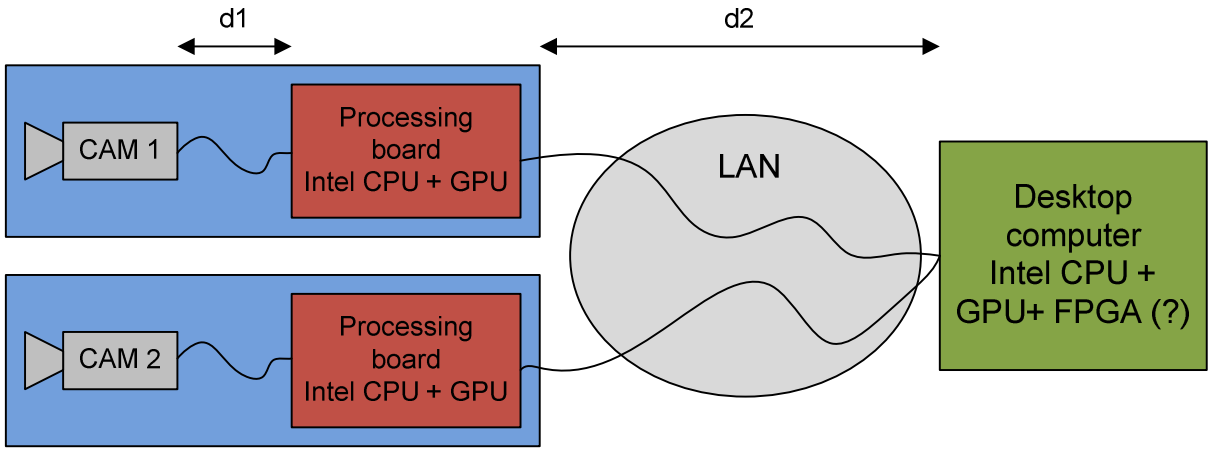

Figure 12: Distributed heterogeneous sensor architecture

The detailed hardware and software architectures for “weak observer” and “expert observer” as well as the methodology for using these architectures in the distributed surveillance application are given in the following sections.

4.1 Weak Observer Architecture

Each weak observer composed of fixed wide FOV camera and a local processing unit with low computational resources. The main task of weak observer is to continuously monitor the scene of interest and seeking salient event/target traces. The local processing unit on weak observer composed of

1. Intel Core i7-4860EQ CPU
2. Intel Iris Pro 5200 Embedded (same die) GPU
3. AMD E6760 Embedded (PCIe) GPU
4. 4GB DRAM
5. 256 GB SSD HD

In the configuration of weak observers, the video frames will be captured via Ethernet port and the captured frames will be feed to the processing unit.

The software components that will be run on the processing unit are listed below.

A Motion Detection: The motion in the scene of interest is detected and registered at each pixel location in the imaging plane, if it is significant, i.e., above a certain threshold.

B Block Processing: The motion detected in step A is processed and grouped into rectangular and overlapping blocks of pixels in the imaging plane. Then, blocks with significant amount motion are detected and registered.

C Feature Extraction: A certain (pre-defined) set of visual features are extracted from each detected block.

D Decision: The extracted features for each detected block are processed and a decision is made: i) The activity in the corresponding block is immediately labeled as human/car/other; or ii) There is not enough evidence and the extracted features along with the pixel coordinates are propagated to the central station for a detailed analysis.

1. Motion Estimation: In this algorithmic block, the pixel locations where some predefined motions are detected and the bounding box locations will be produced as output.
2. Object Selection: The object corresponding to the biggest/most accurate/… will be selected among all the detected objects in order to report single object to the main station.
3. Classification: Features for the detected objects will be produced.

When any moving object is detected, the weak observer will send the pixel locations of the upper left and lower right corners of the bounding box and the extracted features for the detected object to the main station through Ethernet.

4.2 Expert Observer Architecture

The expert observer composed of narrow FOV PTZ camera and a local processing unit with high computational resources. The main task of expert observer is to merge the information received from multiple weak observers and decide final decisions about the detected objects. The processing unit on expert observer composed of

1. Intel Core i7-4470 CPU
2. NVIDIA GTX780 or GTX 980 or Quadro K6000 or Quadro M6000 GPU(s)
3. 16 GB DRAM
4. 256 GB SSD HD

In the configuration of expert observer, the video frames will be captured via Ethernet port and the captured frames will be feed to the processing unit.

The software components that will be run on the processing unit are listed below.

A. Object Localization: The 3D coordinate of the detected object will be estimated from the pixel locations of the detected object received from at least two weak observers.
B. PTZ Steering: The PTZ camera will be steered to the detected object coordinate by converting the 3D coordinate to the azimuth and elevation rotation angles of PTZ camera.

C. Multi-view Classification: Features for the detected objects will be produced from the video frames captured from PTZ camera. Then, these features and the features received from weak observers will be merged and classification result will be produced.
D. Superresolution: From multiple frames captured from the PTZ camera, the super resolved video frames will be monitored.

1. Object Localization: The 3D coordinate of the detected object will be estimated from the pixel locations of the detected object received from at least two weak observers.
2. PTZ Steering: The PTZ camera will be steered to the detected object coordinate by converting the 3D coordinate to the azimuth and elevation rotation angles of PTZ camera.
3. Multi-view Classification: Features for the detected objects will be produced from the video frames captured from PTZ camera. Then, these features and the features received from weak observers will be merged and classification result will be produced.
4. Superresolution: From multiple frames captured from the PTZ camera, the super resolved video frames will be monitored.

4.3 Methodology

The performance of the distributed heterogeneous sensor architecture to be tested in demonstration will be measured based on the following methodologies. The performance metric measurements will be handled in two categories, i.e., system functional accuracy and system resources.

4.3.1 System Functional Accuracy

4.3.1.1 Classification

To measure the classification accuracy, video frames will be captured from the monitored area and stored to be used in training and testing the classification algorithm. Based on the demonstration scenario, there will be only one moving object in the scene at any time. These objects can be a walking person or moving car with a speed less than 40 km/h or moving vehicle other than a car such as motorcycle, pickup truck, etc. Throughout the training phase of the classification algorithm, the video frames can contain only a walking person or a moving car. Therefore, any moving object other than a “walking person” or “moving car” will be defined as “other” in the classification. To measure the accuracy of the classification algorithm, we first manually label each frame as “human”, “vehicle”, “other” and “normal” and then the following evaluation metrics will be calculated by comparing the results of the classification algorithm with the labeled video frames.

- **Detection Rate:** The empirical probability of that a truly “other” object is labeled as “other”.

- **False Alarm Rate:** The empirical probability of that a truly “normal” object is labeled as “other”.
- **Classification Accuracy:** This is calculated for only “normal” objects. The empirical probability of that a truly “human activity” is labeled as “human activity”. This is the classification accuracy for “human activity”; and it is defined similarly for the vehicles.

- **Detection Rate:** Give the textual and mathematical definition
- **False Alarm Rate:** Give the textual and mathematical definition
- **Classification Accuracy:** Give the textual and mathematical definition

4.3.1.2 Superresolution

In order to measure the superresolution accuracy, the user experienced quality measurement will be used. Both the original and superresolved images will be shown to the users and the users will be asked to give a quality number between 0 and 5 (0 is the worst and 5 is the best quality) based on the following three criteria [22],

1. Fidelity Preserving: Does the superresolved image has the same general appearance as the original image (0: Completely different, 5: The same)
2. Detail Enhancing: Does the superresolved image has sharp features where they are expected (0: Worst result, 5: Best result)
3. Smoothness: Does the superresolved image has continuity where it is expected and avoid unnatural high-frequency artifacts (0: Wors result, 5: Best result)

Each of the above criteria will be measured over several images by different users and will be averaged to determine the quality metric.

4.3.2 System Resources

4.3.2.1 Total Transmission Bandwidth

In order to measure the bandwidth resources, we will use a network analyzer tool such as Wireshark and count the number of bits received at the main station within a unit time or triggering event. For the same scenario, the measurement will be performed for both the distributed heterogeneous sensor architecture and centralized architecture. In the former, weak observers will send only the pixel locations of the detected object and a feature vector corresponding to the detected object. On the other hand, in the centralized architecture, video frames will be sent to the main station continuously. The transmission bandwidth comparison between centralized and distributed architectures will be used as a quality metric measurement for distributed architecture.

5 Validation

5.1 State of The Art at Project Start

As of 2015, the dominant approach for large area surveillance applications is the IP camera based centralized architecture. IP camera based video surveillance has been gaining popularity over the traditional analog systems since early 2000s. A centralized architecture uses a master database located on a central control server. All configuration information, related to the cameras and NVRs/DVRs that constitute the installation, as well as all content is transmitted to the master database; for subsequent access and analysis [23]. The main drawbacks in this architecture can be listed as [23].**Erreur ! Source du renvoi introuvable.**

- Continuous communication of users with the central office requires expensive infrastructure of high-end switches and also uses up precious bandwidth
- In case of WAN failure, remote users are left stranded with no access to the live and recorded video from cameras which may actually be on the local network
- All users of the system rely on the central database for login and license permission checks. If this database fails, the entire system fails.
- As and when more cameras and users at remote locations are added to the network, bandwidth becomes congested.
- The network and servers need to cope with increased levels of traffic – database changes, user authentications, storage and transmission of recordings.
- Surveillance cameras do not respect demanding requirements placed on privacy issues specially for a system in public places

Recently, decentralized IP cameras have been introduced to the security and surveillance market. These decentralized cameras have on board processing power and storage to perform low complexity processing tasks such as image enhancement and motion/activity detection. However, the amount processing power available on board is limited and the resulting architecture is not truly distributed as independent cameras do not collaborate to perform a common security/surveillance task. Typically, the processing done on the end node decentralized IP cameras are not utilized at the center or at another end node camera.

This situation also applies to other domains, like smart facility and building management, where additionally, the traditional vision of a vertically structured market prevented the adaptation to a growing demand of dynamism and flexibility in the context of Smart Facility Management. As explained above, the market is demanding not only more efficient, flexible and autonomous surveillance systems, but also the integration of video systems to provide more data and information

for energy management and enhanced building performance. The Cognitive & Perceptive Video Systems (CPVS) enabled by COPCAMS will represent in this context a significant step towards wider adoption of embedded vision systems within the smart facilities & smart building management domain; providing advanced features in an emerging market.

QMUL began the implementation of a full pipeline for multi-target detection and tracking based on PHD-PF from a MATLAB prototype developed in house. Target detection is carried out using Histogram of Oriented Gradient approach [16]. The MATLAB implementation of PHD-PF has the drawback of having large latency, especially if aimed at running on an embedded platform, such as NVIDIA Jetson TK1. Therefore QMUL developed a C++ version of PHD-PF that will allow its parallelization at a lower programming level than MATLAB.

5.2 Targets at Project End

COPCAMS project will attempt to resolve the drawbacks of centralized surveillance architecture by proposing and designing heterogeneous distributed surveillance architecture. Unlike the centralized architecture, in a distributed architecture the data is spread across the system, generally close to where it is produced or needed.

In COPCAMS architecture, the cameras distributed over the monitored area have local processing units and analyze the video frames to decide whether there is an activity on the scene or not. The activity can be defined based on the video surveillance task, such as detecting a specified object. If no activity is detected on the scene, it is not need to send any information to the main station. Hence, we can eliminate the unnecessary data transmission.

Our main target is to improve the current (state-of-the-art) centralized surveillance architecture in the following points:

- **Bandwidth:** In the distributed surveillance architecture, the cameras in the weak observer will not send live video frames to the main station. Instead, they will analyze the video frames in the local processing unit to detect an activity that can be defined based on the mission. Then, if any defined activity is detected, they further analyzing the activity and extract the features that describe the activity. After performing the processing, the pixel locations and features of the detected object will be sent to the main station. In this structure, network bandwidth is not used for continuous communication with remote locations. Data is streamed to the central station only in event of an operational incident. Hence, overall communication cost is expected to decrease.
- **Distributed Computing:** As opposed to the centralized surveillance architecture, since the weak observers have local processing unit there is no need to analyze the whole video

frames on the central station. Some of the tasks or some part of the whole task can be performed on the node outside of the central station. This approach decreases the computational complexity requirement in central station. Hence, the central station in distributed architecture can be cost effective or can perform much complex tasks as compared to the centralized architecture.

- **Scalability:** With distributed architecture, additional cameras and users can be added to a sub-location to increase the coverage area with minimal increase to network traffic and computational capacity of the central station.
- **Security and Privacy:** we will improve support for privacy protection by implement algorithms of anonymization and encryption scheme, for videosurveillance vision systems.

6 Current State of the Demonstrator

The current status about the Large Area Surveillance Application Demonstration can be summarized as follows.

- **Camera Installation:** Two Samsung SNP-3120VH IP cameras were installed on the outside of the building in ASELSAN's facility. One of these cameras will be used as fixed wide FOV camera in weak observer and the other one will be used as narrow FOV PTZ camera in expert observer. In this configuration, the target object location will be estimated from the pixel locations of a single camera with the assumption that target object is on the floor. Then, we will try to install the third camera before the final demo and use two fixed cameras to estimate target object without any assumption about the target location.
- **Network:** Two Samsung SNP-3120VH IP cameras are connected to the ASELSAN's local network and video capture and steering functionalities were successfully tested.
- **Processing Units:** There will be two different types of processing units to be used in weak observer(s) and central station. The processing unit in weak observer will be configured with low computing power (Intel Core i7-4860EQ, Intel HD 5200 GPU same die GPU and possibly an embedded grade GPU) and the one in expert observer will be configured with high computing power (Intel Core i7-4770 with multiple high end GPUs)
- **Algorithms:** Single camera classification algorithm design in MATLAB has been completed and we are working on algorithm porting to OpenCV. Activity detection and anonymization algorithm are currently ported on COPCAMS platforms (iMX6) to ensure privacy protection.

- **Data Collection:** We have just started to collect the video frames for classification algorithm training. We capture a video from the scene of interest. Since this video is going to be used for training and testing purposes, it should contain all kind of activities regarding “human”, “vehicle” and “other” activities. We split this video into two segments. In the first segment, it contains only the normal, i.e., non-other, activities such that the classification algorithms are trained by using this segment. In the second segment, there are also the “other” activities in addition to the normal ones. By using this segment, the classification algorithms are tested for cross-validation and performance evaluation.

QMUL is currently working towards the parallelization of PHD-PF and the computational performance achieved so far is reported in Figure 11. The parallelization is done for GPGPU on NVIDIA Jetson TK1. This performance improvement is obtained via parallelization of a clustering step (Expectation-Maximization – E-M) within PHD-PF. The clustering step is used by the PHD-PF to estimate the final state of the targets. The E-M was originally developed with OpenCV, whereas the parallelized version is developed in CUDA. Figure 12 shows the power consumption in the case of CPU version of the code and GPU version of the code. The power consumption is slightly higher for the GPGPU version but still comparable to that of the CPU version. QMUL is continuing with the optimization of the algorithm via the use of OpenMP and GPGPU functions provided in OpenCV, and the goal is to achieve tracking performance of 15 fps.

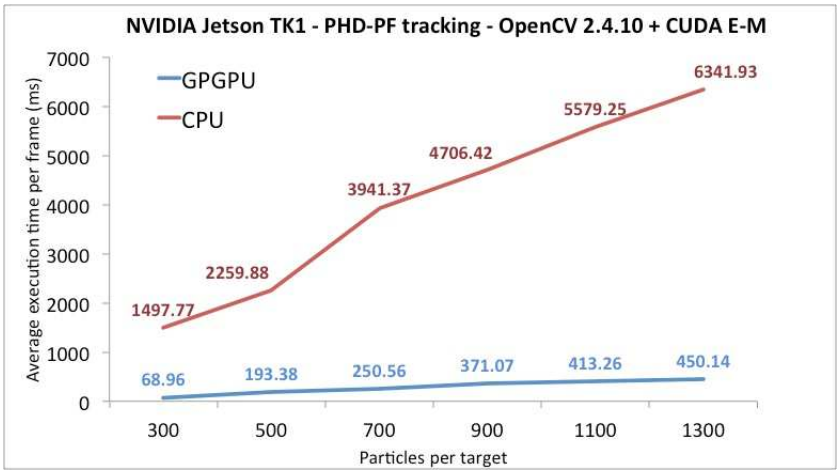

Figure 13 : GPU vs. CPU performance of NVIDIA Jetson TK1 in the case of PHD-PF tracking algorithm. The horizontal axis represents the variation of the particles per target. The vertical axis represents the average execution time in milliseconds (ms).

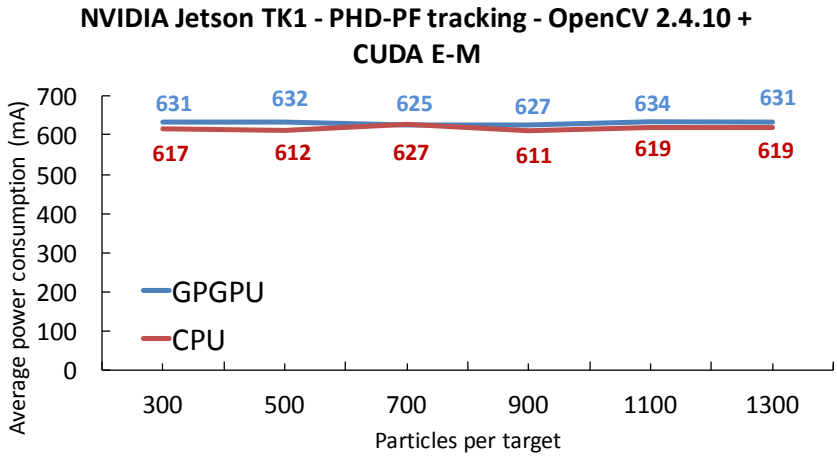

Figure 14: GPU vs. CPU power consumption on NVIDIA Jetson TK1 in the case of PHD-PF tracking algorithm. The horizontal axis represents the variation of the particles per target. The vertical axis represents the average power consumption (mA).

7 Conclusion

The specifications of the demonstration activities for the large area surveillance applications are described in two categories, i.e., a field test and laboratory experiments. The system architecture including the hardware and software configurations and the methodology for evaluating field test are explained. The state of the art at project start and the expected progress with COPCAMS project are explained. The metrics and measurement strategy of COPCAMS solutions on large area surveillance applications are described. The results explored by evaluating the field test will be reported on "Large Area Surveillance Applications Report" at the end of the COPCAMS project.

References

[1] E. Liotou, E. Papadomichelakis, N. Passas, and L. Merakos, "Quality of experience-centric management in LTE-A mobile networks: The Device-to-Device communication paradigm," Sixth International Workshop on Quality of Multimedia Experience (QoMEX), pp. 135-140, September 2014.

[2] P. Callet, S. Möller and A. Perkis, "Qualinet White Paper on Definitions of Quality of Experience", European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), March 2013.

[3] K. Zheng, X. Zhang, Q. Zheng, W. Xiang, and L. Hanzo, "Quality-of-experience assessment and its application to video services in LTE networks," Wireless Communications, IEEE , vol.22, no.1, pp. 70,78, February 2015.

[4] A. Khan, S. Lingfen, and E. Ifeachor, "QoE Prediction Model and its Application in Video Quality Adaptation Over UMTS Networks," IEEE Transactions on Multimedia, vol.14, no.2, pp. 431-442, April 2012.

[5] M. Venkataraman and M. Chatterjee, "Inferring video QoE in real time,"IEEE Network, vol.25, no.1, pp. 4-13, January-February 2011.

[6] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," Communications Surveys & Tutorials, IEEE, vol. 16, no. 4, pp. 1801-1819, April 2014.

[7] S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding", IEEE/ACM Transactions on Networking, vol. 16, no. 3, pp. 497-510, June 2008.

[8] A. Antonopoulos, C. Verikoukis, C. Skianis, and O. B. Akan, "Energy Efficient Network Coding-based MAC for Cooperative ARQ Wireless Networks", Ad Hoc Networks, vol. 11, no. 1, pp. 190-200, January 2013.

[9] X. Wang, J. Li, and M. Guizani, "NCAC-MAC: Network Coding Aware Cooperative Medium Access Control for Wireless Networks," IEEE Wireless Communications and Networking Conference (WCNC), pp. 1636-1641, April 2012.

[10] E. Datsika, A. Antonopoulos, N. Zorba, and C. Verikoukis, "Adaptive Cooperative Network Coding Based MAC Protocol for Device-to-Device Communication", IEEE International Conference on Communications (ICC), June 2015.

[11] "IEEE Standard for Information technology-Telecommunications and information exchange between systems, local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), pp. 1-2793, March 2012.

[12] D. Hernando, J.E.L. de Vergara, D. Madrigal, and F. Mata, "Evaluating quality of experience in IPTV services using MPEG frame loss rate, "International Conference on Smart Communications in Network Technologies (SaCoNeT), pp. 1-5, June 2013.

[13] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler, "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking," arXiv:1504.01942, Apr. 2015

[14] E. Maggio, M. Taj, A. Cavallaro, "Efficient multi-target visual tracking using Random Finite Sets," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 18, Issue 8, Aug. 2008, pp. 1016-1027

[15] B.-N. Vo, and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," IEEE Trans. on Signal Processing, vol. 54, no. 11, pp. 4091–4104, Nov. 2006

[16] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” Proc. of Computer Vision and Pattern Recognition, San Diego, CA, USA, Jun. 2005.

[17] (http://www.contiki-os.org/)

[18] S.M. Kay, “Fundamentals of Statistical Signal Processing, Volume II: Detection Theory”, Prentice-Hall Ed., 1998.

[19] (http://www.maxbotix.com/Ultrasonic_Sensors/Rangefinders.htm.)

[20] Z. Chair and P.K. Varshney, “Optimal data fusion in multiple sensor detection systems”, IEEE Trans. On Aerospace and Electronic systems, vol. AES22, no.1, pp. 98-101, Jan. 1986.

[21] (http://issuu.com/zolertia/docs/z1_brochure?e=1376732/2711667)

[22] Liu, J. Wang, S. Zhu, M.Gleicher and Y. Gong, “Visual-Quality Optimizing Super Resolution”, Computer Graphics Forum, Vol. 0 (1981), Num 0, pp. 1-14, 2008.

[23] (www.mistralsolutions.com/networked-ip-video-surveillance-architecture-distributed-centralized/)

[24] Nicolas Hili, Christian Fabre, Sophie Dupuy-Chessa, and Stéphane Malfoy. Efficient Embedded System Development: A Workbench for an Integrated Methodology. In Proc. Of the 6th Embedded Real-Time Software and Systems Congress (ERTS2 2012), Toulouse, France.

[25] Nicolas Hili, Christian Fabre, Sophie Dupuy-Chessa, and Dominique Rieu. A Model-Driven Approach for Embedded System Prototyping and Design. In Proc. of The IEEE International Symposium on Rapid System Prototyping (RSP 2014), New Delhi, India.

[26] Nicolas Hili, Chrisitan Fabre, Ivan Llopard, Sophie Dupuy-Chessa, and Dominique Rieu. Model-Based Platform Composition for Embedded System Design. In Proc. of IEEE 8th International Symposium on Embedded Multicore Many-core Systems-on-Chip (MCSoC-14), University of Aizu, Japan.

[27] Ivan Llopard, Albert Cohen, Christian Fabre, and Nicolas Hili. A Parallel Action Language for Embedded Applications and Its Compilation Flow. In Proceedings of the 17th International Workshop on Software and Compilers for Embedded Systems, SCOPES ’14, pages 118–127, New York, NY, USA, 2014. ACM.