TEC2014-53176-R HAVideo (2015-2017)

High Availability Video Analysis for People Behaviour Understanding

D3.2 Collaborative approaches for people behaviour understanding (December 2017)

Video Processing and Understanding Lab

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Supported by

AUTHORS LIST

Juan Carlos San Miguel Avedillo	juancarlos.sanmiguel @uam.es
Alvaro Garcia Martin	alvaro.garcia@uam.es

HISTORY

Version	Date	Editor	Description
0.1	20/03/2017	Juan C. SanMiguel	First Working Draft
0.2	24/12/2017	Álvaro García Martín	Contributions
0.3	27/03/2017	Juan C. SanMiguel	Final Draft
1.0	30/03/2017	José M. Martínez	Editorial checking
1.1	16/03/2017	Juan C. SanMiguel	Final Draft v2
2.0	23/12/2017	José M. Martínez	Editorial checking

CONTENTS:

1. Introduction

This document summarizes the work during the first and half year(s) of the project for the task T3.2 “Collaborative approaches” (WP3 “Self-configurable approaches for long-term analysis”). whose goal is to exploit interactions among multiple entities to optimize the overall performance (accuracy or resource-usage). First, we consider the processing stages where interactions are based on quality and contextual information. Second, we investigate approaches in camera networks where the quality and contextual information of each camera have to be distributed and used by other cameras in order to coordinate them.

This task T3.2 depends upon developments within WP2 (T2.1 Analysis tools for human behavior understanding, T2.2 Contextual modeling and extraction and T2.3 Quality analysis). The results of this task T3.2 will provide self-configurable approaches for long-term analysis and WP4 Evaluation framework, demonstrators and dissemination.

Here we define *collaborative* as a process in which various entities (e.g. algorithms) interact to achieve a common goal. We differentiate such collaboration from adaptation where a single entity (e.g. algorithm) adjust some of its parameters according to various indicators based on quality signals or contextual information

1.1. Document structure

The document is structured in the following chapters:

- Chapter 1: Introduction to this document
- Chapter 2: description of the contributions
- Chapter 3: Conclusions and future work

2. Contributions

This chapter compiles the contributions developed in the scope of the task T3.2

2.1. Generic developments

2.1.1. Enabling collaboration via modelling resource usage

To enable algorithm designers to identify key factors underpinning the development of collaborative and resource-aware approaches, we propose a comprehensive model of resource consumption for camera networks. This work has been published in the journal IEEE Transactions on Circuits and Systems for Video Technology [1].

We define common parameters that determine the consumption related to sensing (framesize and framerate), processing (dynamic frequency scaling and task load) and communication (output power and bandwidth). A generic abstraction model is determined based on the clock frequency and the duty cycle which considers three operational states, namely active, idle and sleep. We demonstrate the usefulness of the proposed model for the task of tracking a target and show the dependency on bandwidth and local computation resources. Moreover, cameras not operating at full hardware capacity can significantly reduce consumption with minor performance decreases. The proposed consumption model can be easily adjusted to many recent platforms, thus providing tools for further research in resource-aware camera networks.

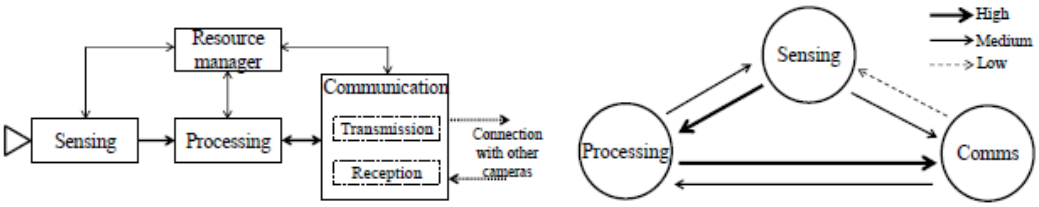


Figure 1.(left) The functional block diagram of a smart camera. The costs associated to resource usage are computed by the resource manager. (right) Typical relative influence between modules in a smart camera. Low, Medium and High indicate the impact of resource usage on the consumption of a module.

Modelling resource consumption requires to address the following items:

- Defining resources and hardware capabilities
- Defining operational states: active, idle and sleep.
- Compute the total energy consumption

We define the following operational states:

- The active state is defined when a module performs tasks (sensing photons, processing frames or transmitting data).
- The idle state occurs when a module waits to quickly become active if needed.
- The sleep state defines the operation with the lowest consumption (i.e. when most functionalities are disabled).

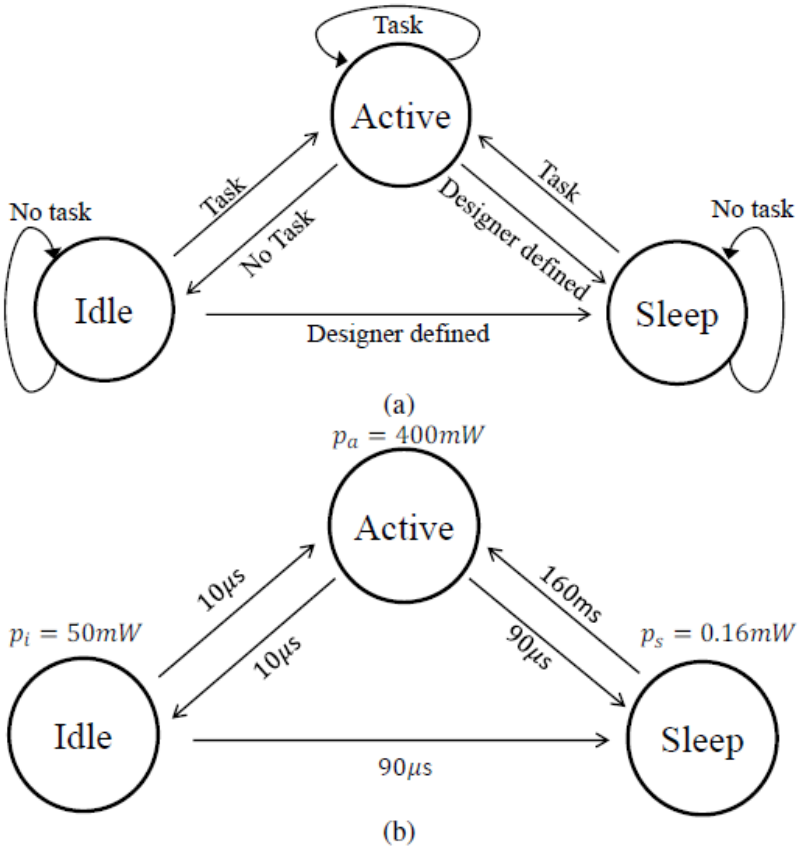


Figure 2. (a) Three-state model for module operation of smart cameras. States can be selected on demand (e.g. the processor is requested to complete a task) or via designer-defined rules (e.g. go to sleep after a time threshold). (b) Illustrative example for the transition costs of the processor SA1100 (http://research.microsoft.com/ apps/pubs/default.aspx?id=238914).Transition power is approximately pa for all cases.

The proposed consumption model is based on the power and activation times of a state-model with N = 3 states (active, sleep and idle) as depicted in Fig. 2(a). Moreover, the costs of the transition between states cannot be neglected. Fig. 2 (b) shows the transition costs for the widely used SA-1100 processor (http://research.microsoft.com/apps/pubs/default.aspx?id=238914). Each sleep → active transition employs an energy of 64mJ, representing a 15% of the active power

The following figures show examples of the achieved results with the proposed models.

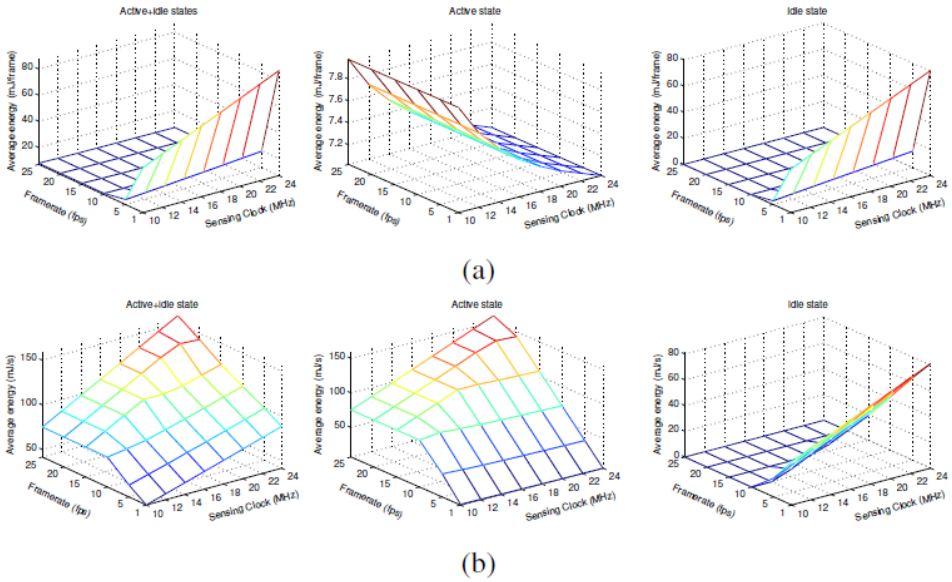
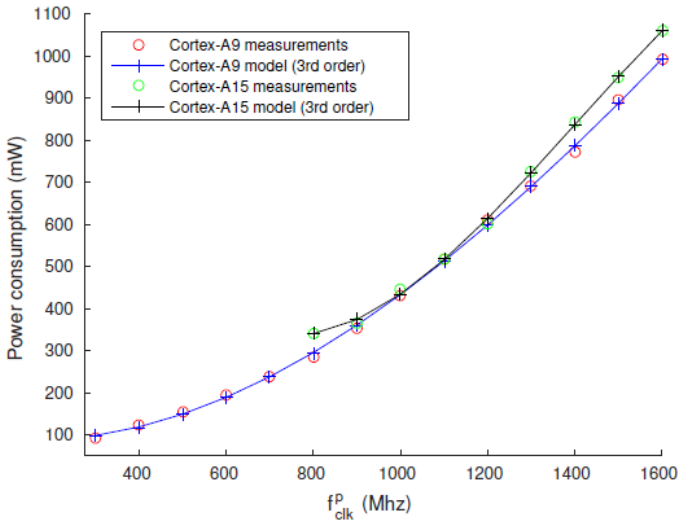


Figure 3. Energy consumption for the sensing module when changing the framerate and the operating frequency for the B3 sensor [2]. Results are for (a) each frame and (b) each second.



(a)

Model	Error for fitting P_{active} (mW^2)			
order	SA1100	Cortex-A9	Cortex-A15	Krait400
1	±7.70	±17.10	±11.21	±7.28
2	±2.96	±6.49	±5.01	±5.32
3	±2.10	±6.26	±4.82	±5.29
4	±2.09	±5.07	±4.20	±5.22
5	±2.09	±5.00	±4.19	±5.18

(b)

Figure 4. Model fitting examples for active power using available single-core measurements for SA1100 (74-204Mhz), Cortex-A9 (0-1.6GHz), Cortex-A15 (0.8-1.6GHz) and Krait400 (0.3-1.5MHz). (a) Measurements and power for active state and (b) associated fitting error.

We compare the proposed consumption model against existing models based on utilization (i.e. active time of the module) for sensing [3], processing [4] and communication [5].

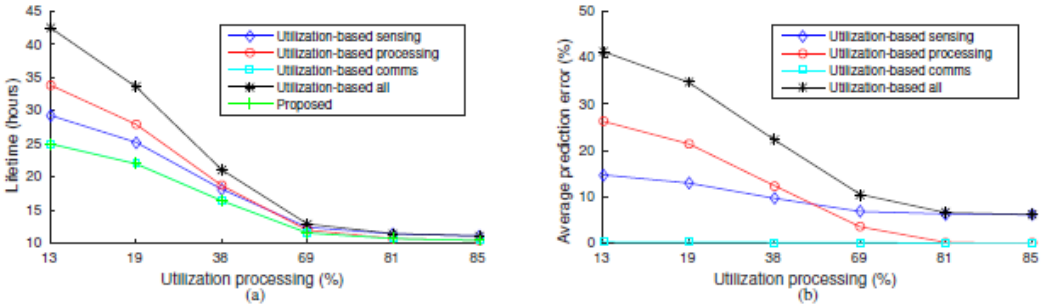


Figure 5. (a) Predicted lifetime of the camera network for the proposed and the utilization-based consumption models. Bottom (green) curve corresponds to the ideal lifetime since experimentally-validated models are used for active-sleep state modelling of sensing, processing and communication. (b) Error associated to utilization-based consumption models as compared to the proposed one.

2.2. Collaborative shadow detection

The achievement in this area corresponds to the following master thesis:

Detección de sombras en secuencias de vídeo-seguridad (Shadows detection in video-surveillance sequences), Guillermo Rodríguez Yrezabal (advisor: Juan Carlos San Miguel), Proyecto fin de Carrera (Master Thesis), Ingeniería de Telecomunicación, Univ. Autónoma de Madrid, Sept. 2016.

The main goal of this master thesis is the design and implementation of a shadow detection algorithm. Many computer vision applications such as video-surveillance require the detection and object tracking where background subtraction is commonly applied for background/foreground segmentation. However, cast shadows from moving foreground objects usually result in errors for such applications.

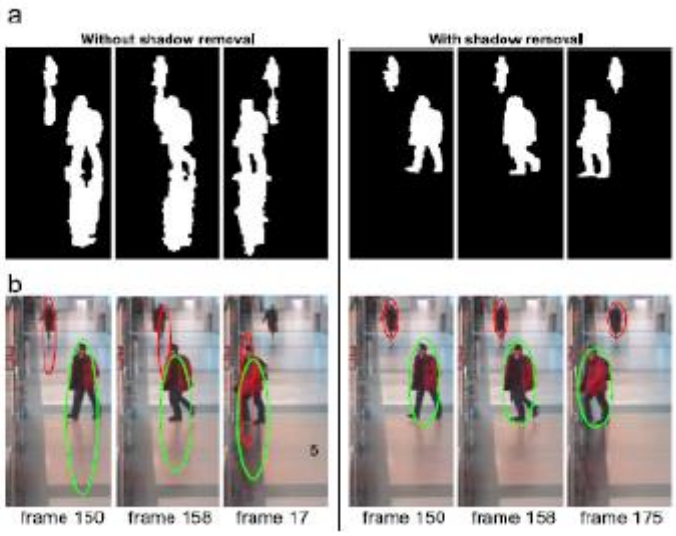


Figure 6. Effect of detected shadows in segmentation algorithms

To address these problems, this work proposes the design and implementation of a shadow detection algorithm, exploiting the colour information by means of calculating the ratios between pixels under shadow regions and background pixels for different colour spaces. For this purpose, we first studied, implemented, adapted and evaluated the main and most relevant techniques of background subtraction and shadow methods that form the basis of most detectors in the literature, highlighting the main gaps they present in detecting and removing shadows from image sequences. It is described later the proposed algorithm explaining each of the process steps such us the calculation of ratios, histograms, colour spaces channel correlation and optimization of thresholds.

The shadow detectors employed are based on thresholding operations and therefore, they require to learn proper configurations of the best thresholds to be applied. The following figure presents such scheme.

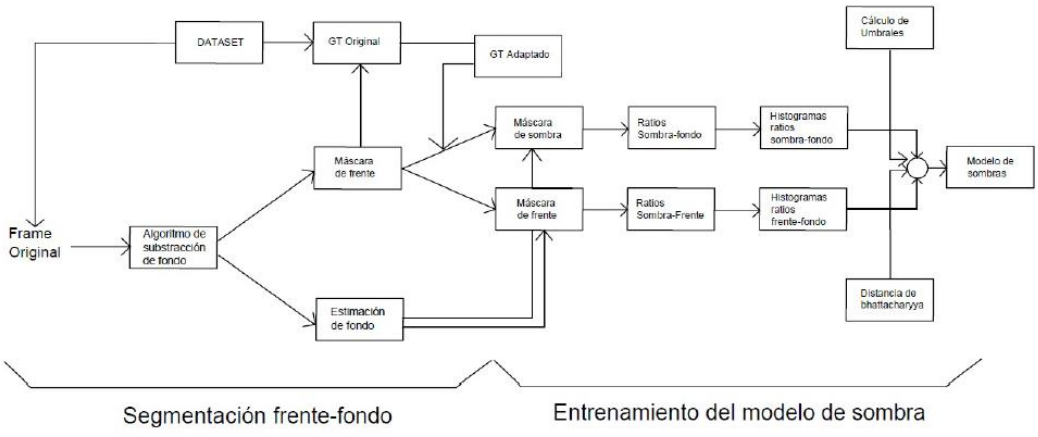


Figure 7. Proposed scheme to train the models for the collaborative shadow detectors

The methodology for collaboration is based on the *maximization of mutual agreement between independent sources* [6]. In short, two algorithms will iteratively change their decision parameters until the output converges to a similar result for both shadow detectors. This process is illustrated in the following figure.

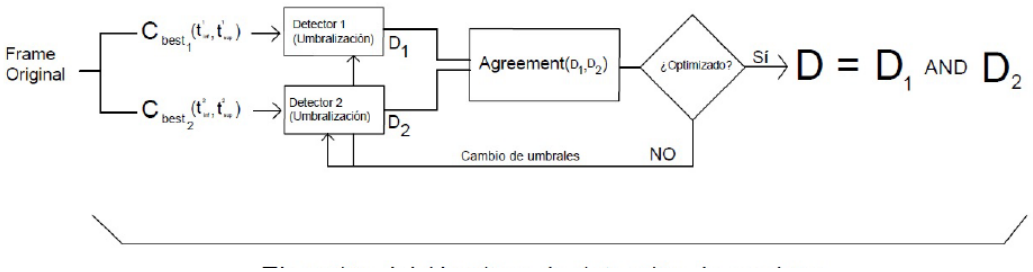


Figure 8. Proposed scheme for collaborative shadow detection

The results associated to every process of the algorithm are presented in four experiments, performing a comparative evaluation with some of the algorithms found in the literature.

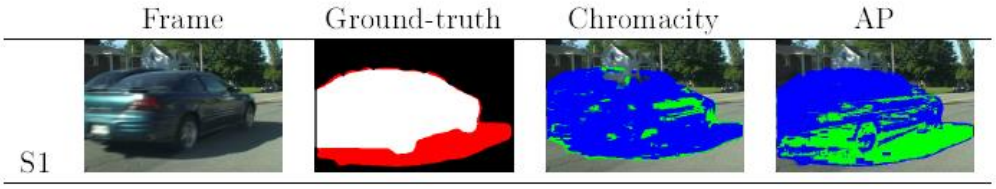


Figure 9. Example of the proposed algorithm (AP) being compared against an approach based on chromaticity.

2.3. Collaborative People detection

2.3.1. Detection threshold adaptation during runtime

Applying people detectors to unseen data is challenging since pattern distributions may significantly differ from the ones of the training dataset.

In this work, we propose a framework to adapt people detectors during prediction time. Such adaptation combines multiple detectors to identify their best configurations (i.e. detection thresholds) without requiring manually labelled data. This combination is based on the maximization of mutual information by correlating the output of pairs of detectors which allows to obtain a set of hypotheses for the detection thresholds. These hypotheses are later combined by weighted voting to obtain a global decision for the detection threshold of each detector. The proposed approach does not require re-training detectors and uses standard detector outputs, therefore it can combine various types of detectors. The experimental results demonstrate that the proposed approach outperforms state-of-the-art detectors whose optimal configuration is learned from training data

Figure 10. People detection results for Faster R-CNN [13] (sequence S1-T1-C, http://www.cvg.reading.ac.uk/PETS2006). Each row corresponds to the detection thresholds tao1 = 0:25 (row 1), tao2 = 0:5 (row 2) and tao3 = 0:75 (row 3). Finding an optimal threshold for all cases is challenging due to the variability of viewpoints, people sizes and occlusions.

The results of this work have been submitted to the International Conference on Image Processing 2017.

2.4. Collaborative tracking

2.4.1. Single-target single-camera tracking 1

First an approach for single-camera settings where multiple trackers are combined based on quality measures defined in the task "T2.3 Quality analysis". This work has been published in the journal IEEE Transactions on Circuits and Systems for Video Technology.

The proposed approach is inspired by the test and select framework [7] for ensemble combination where accurate classifiers are fused if their errors are diverse. Considering trackers as classifiers, we extend this framework to video tracking by introducing spatio-temporal correlation and adaptive online performance evaluation in the following figure.

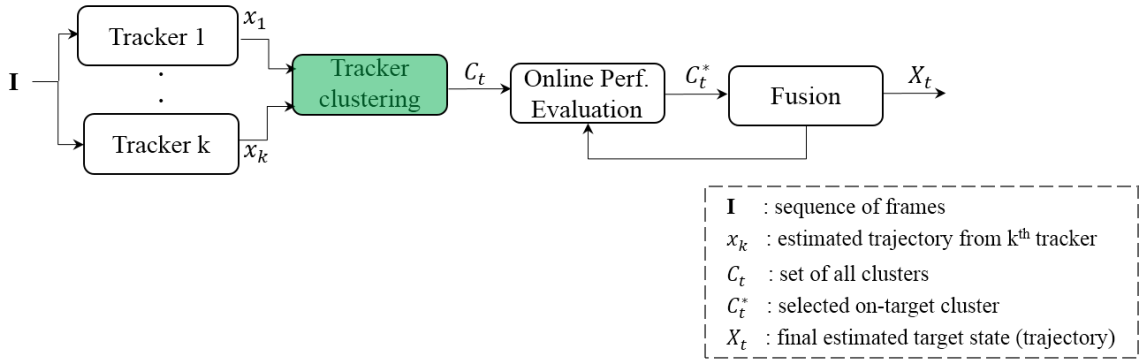


Figure 11. Block diagram of the proposed approach to fuse the output of K trackers.

In this section, we focus on the fusion part which is initially approached as the average position of the selected N trackers out of the pool of M trackers (M>N):

$$x_t^* = \frac{1}{N}\sum_{n=1}^{N} x_t^n.$$

and present the results obtained by this module.

As trackers we use: The first tracker is the Sparse features based Tracker (ST) [8], which is PF-based and uses sparse (intensity) features to generate the target appearance model. The Maximum a Posteriori criterion is employed to estimate the target state. The second tracker is the Adaptive Fragments-based Tracker (AFT) [9] that models the target appearance with various fragments. Fragment reliability is based on colour similarity between the current and previous fragment, to integrate highly-reliable fragments within a PF framework. The third tracker is the Locally Orderless Tracker (LOT) [10]
that divides the target into superpixels using the HSV color space. This PF-based tracker weights each particle by the distance between the model and the noisy observations. Final target state is estimated as the weighted average of particles. The fourth tracker is the Incremental Visual Tracker (IVT) [11] that uses an on-line update approach to account for appearance changes, and a PF to track the target over time. The fifth tracker is the Scale and Orientation Adaptive Mean Shift Tracker (SOAM) [12], that estimates the changes in scale and orientation of the target using the mean shift framework, by employing Gaussian kernels and image moments. The sixth tracker is the Fast Compressive Tracker (FCT) [13] that projects the original image to a low-dimensional space. The projected features are then used to formulate tracking as a binary classification task via a naive Bayesian classifier. The seventh tracker is the L1 Tracker (L1APG) [14], which is based on PF and uses sparse features for target modeling. A fast minimization solver is used to reduce the computational complexity associated to L1-trackers. The eighth tracker is the Least Soft-Threshold Squares Tracker (LSST) [15] is based on PF and performs liner regression via least-soft threshold squares distance between the observation and the target model.

We have implemented six different configurations of TPF: TPF3 (ST, AFT, LOT), TPF4 (ST, AFT, LOT, IVT), TPF5 (ST, AFT, LOT, IVT, FCT), TPF6 (ST, AFT, LOT, IVT, FCT, SOAM), TPF7 (ST, AFT, LOT, IVT, FCT, SOAM, L1APG) and TPF8 (ST, AFT, LOT, IVT, FCT, SOAM, L1APG, LSST).

Sequence	State of the Art					TPF	
	AvgF	SymT [17]	VTS [13]	STR [50]	KCF	TPF_3	TPF_8
P1	.25+/-.13	.35+/-.11	.24+/-.14	.75+/-.04	.21+/-.12	.72+/-.02	.50+/-.10
P2	.40+/-.14	.65+/-.05	.77+/-.01	.68+/-.01	.87+/-.01	.81+/-.01	.73+/-.07
P3	.78+/-.03	.81+/-.01	.77+/-.01	.78+/-.01	.85+/-.01	.70+/-.02	.82+/-.01
P4	.30+/-.15	.33+/-.15	.37+/-.16	.39+/-.03	.64+/-.07	.41+/-.15	.42+/-.13
P5	.64+/-.03	.72+/-.02	.59+/-.08	.68+/-.10	.85+/-.01	.76+/-.03	.57+/-.13
P6	.12+/-.07	.12+/-.07	.14+/-.10	.13+/-.08	.10+/-.07	.13+/-.12	.12+/-.07
P7	.89+/-.02	.89+/-.01	.85+/-.01	.80+/-.01	.83+/-.01	.85+/-.01	.89+/-.01
P8	.47+/-.06	.62+/-.07	.78+/-.01	.74+/-.03	.67+/-.10	.77+/-.01	.58+/-.04
P9	.84+/-.01	.84+/-.01	.92+/-.01	.82+/-.01	.74+/-.03	.77+/-.04	.82+/-.01
P10	.12+/-.10	.12+/-.09	.14+/-.01	.91+/-.11	.11+/-.08	.77+/-.01	.76+/-.04
P11	.69+/-.02	.90+/-.01	.84+/-.01	.54+/-.10	.76+/-.02	.86+/-.01	.82+/-.02
P12	.30+/-.14	.33+/-.15	.36+/-.10	.28+/-.11	.27+/-.16	.55+/-.07	.35+/-.15
P13	.53+/-.17	.55+/-.16	.56+/-.12	.69+/-.07	.44+/-.15	.71+/.09	.67+/-.10
P14	.82+/-.03	.83+/-.02	.85+/-.16	.82+/-.02	.82+/-.03	.86+/.01	.76+/-.07
P15	.82+/-.01	.83+/-.01	.78+/-.01	.69+/-.01	.79+/-.01	.81+/.01	.82+/-.01
P16	.70+/-.10	.77+/-.05	.86+/-.01	.78+/-.05	.79+/-.02	.74+/.07	.89+/-.01
P17	.43+/-.07	.41+/-.08	.24+/-.12	.16+/-.10	.84+/-.01	.26+/.12	.36+/-.10
P18	.28+/-.12	.38+/-.13	.87+/-.01	.84+/-.01	.12+/-.10	.86+/.12	.85+/-.02
P19	.87+/-.01	.88+/-.01	.80+/-.01	.76+/-.01	.89+/-.01	.89+/.01	.89+/-.01
P20	.83+/-.02	.85+/-.01	.87+/-.01	.78+/-.01	.90+/-.01	.88+/.01	.85+/-.01
P21	.78+/-.01	.79+/-.01	.85+/-.01	.75+/-.02	.74+/-.02	.77+/-.02	.74+/-.02
P22	.43+/-.05	.54+/-.03	.37+/-.10	.13+/-.06	.13+/-.06	.43+/-.05	.50+/-.07
Mean	.56+/-.07	.61+/-.06	.62+/-.06	.63+/-.04	.61+/-.05	.70+/-.05	.67+/-.06

Table 1. COMPARISON OF PROPOSED APPROACH (TPF) WITH THE STATE-OF-THE-ART IN TERMS OF THE OVERLAP SCORE (OS) (MEAN+/- STANDARD DEVIATION). KEY - AVGF: AVERAGE FUSION; SYMT: SYMBIOTIC TRACKER; VTS: VISUAL TRACKER SAMPLER; STR: STRUCK; KCF: KERNELIZED CORRELATION FILTER TRACKER

The following figure presents the number of trackers employed from the pool of M trackers. As we can see the 100% of trackers is rarely achieved and therefore, trackers failing are discarded from the final result.

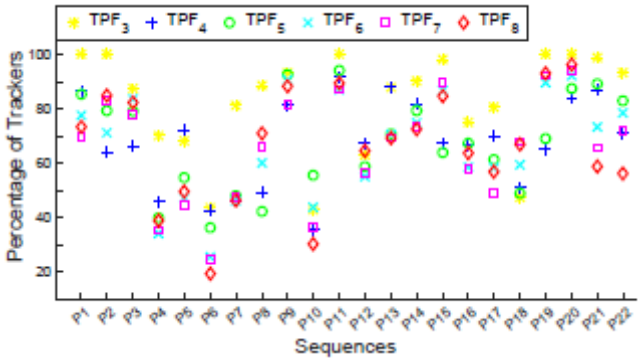


Figure 12. Percentage of trackers used in the proposed approach

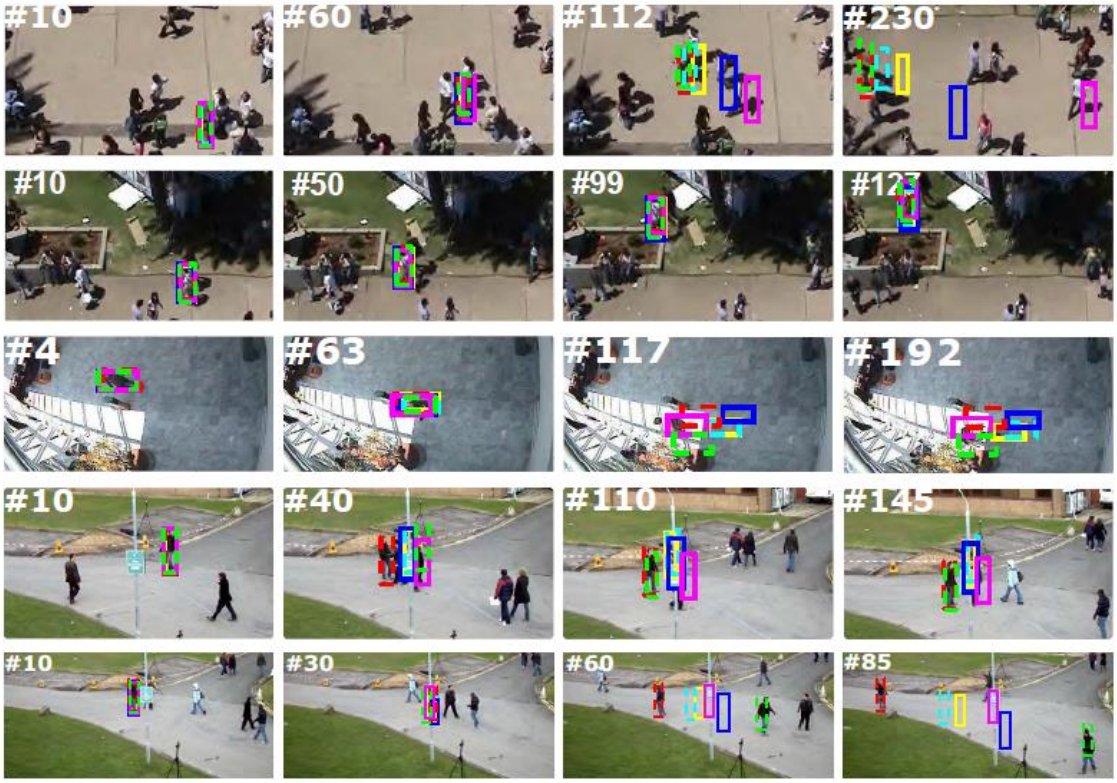


Figure 13. Sample tracking results. From top to bottom row: Students-P1, Students-P3, CAVIAR-P4, PETS-P10 and PETS-P12. TPF (green dotted); STRUCK (red dotted); VTS (blue solid); SymT (blue dotted); AvgF (yellow solid); KCF (magenta solid).

2.4.2. Single-target single-camera tracking 2

The achievement in this area corresponds to the following master thesis:

Moreno De Pablos, E. "Seguimiento de objetos basado en múltiples algoritms", Trabajo Fin de Grado, Degree ITST, Universidad Autonoma de Madrid, July 2016

The main objective of this work consists on studying metrics to compare algorithms for tracking objects in video sequences using a set of search or tracking algorithms. The results of this work is mainly focused on the quality analysis and hence it has been reported in the deliverable D2.3 corresponding to the task "T2.3 Quality Analysis" A summary is provided here with respect to the collaborative tracking.
The proposed framework for tracker combination is described in the following figure. It can be seen that depends on some inter-tracker and intra-tracker measures (described in D2.3).

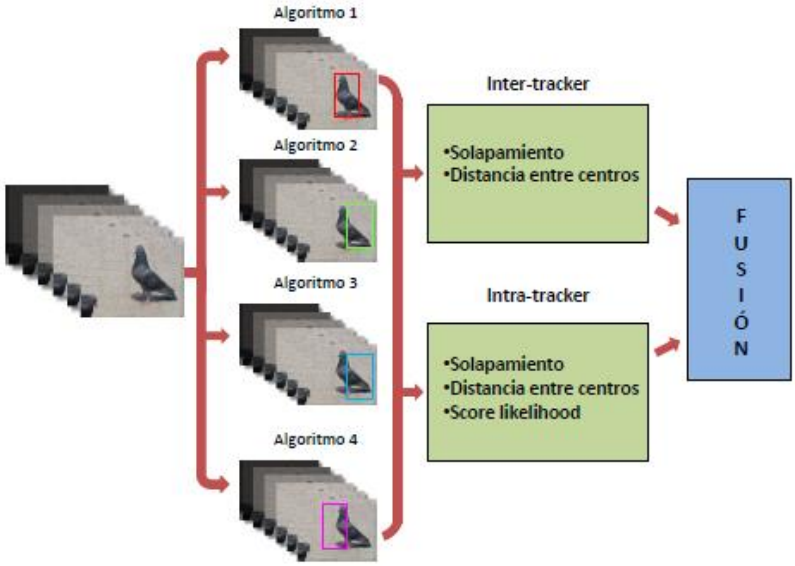


Figure 14. Block diagram of the proposed approach to combine four trackers.

To show the applicability, a simple fusion algorithm has been implemented taking those trackers with high similarity (e.g. values over 0.7). Two fusion versions are considered: equal and similarity-weighted combinations.

For the experiments, a set of 10 sequences has been extracted from the Tracker Benchmark 1.0 (https://sites.google.com/site/trackerbenchmark/benchmarks/v10). As trackers, this study used the following:

- **MS** (PSU R.Collins, CSE. Mean-shift Tracking. 2006)
- **CBWH** (David Zhang y Chengke Wu Jifeng Ning, Lei Zhang. Robust mean shift tracking with corrected background-weighted histogram. IET CVI 2010)
- **PFC** (Fabian Kaelin, An Adaptive Color-Based Particle Filter. ECCV 2010)
- **ACA** (Michael Felsberg y Joost van de Weijer Martin Danelljan, Fahad Shahbaz Khan. Adaptive color attributes for real-time visual tracking. CVPR 2014)

Sample results are show in the following figures/tables.

Comparación algoritmos	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS-CBWH	.01±.002	.20±.005	.15±.004	.99±.001	.09±.007	.47±.109	.50±.041	.44±.062	.97±.011	.28±.039
MS-PFC	.34±.003	.34±.015	.45±.012	.85±.064	.41±.006	.60±.083	.36±.024	.67±.069	.92±.047	.39±.018
MS-ACA	.11±.001	.12±.003	.90±.043	.95±.035	.11±.001	.20±.024	.82±.075	.31±.019	.90±.058	.26±.012
CBWH-PFC	.35±.006	.23±.009	.53±.029	.00±.000	.36±.009	.66±.061	.61±.038	.82±.079	.95±.019	.36±.030
CBWH-ACA	.12±.002	.18±.007	.92±.025	.88±.035	.12±.005	.38±.060	.94±.015	.65±.128	.90±.036	.42±.039
PFC-ACA	.19±.004	.40±.001	.86±.073	.97±.012	.33±.002	.47±.039	.88±.059	.32±.001	.89±.050	.64±.034

(a)

Comparación algoritmos	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS-CBWH	.03±.008	.30±.009	.33±.016	.04±.004	.17±.026	.10±.033	.58±.079	.50±.059	.04±.038	.39±.052
MS-PFC	.90±.006	.79±.003	.69±.005	.20±.128	.79±.0060	.54±.150	.68±.014	.75±.088	.09±.062	.69±.009
MS-ACA	.22±.001	.21±.012	.12±.070	.01±.006	.21±.0050	.37±.044	.13±.069	.46±.040	.07±.040	.53±.034
CBWH-PFC	.86±.038	.78±.005	.83±.026	.00±.000	.76±.010	.27±.127	.65±.118	.23±.128	.08±.063	.58±.057
CBWH-ACA	.24±.006	.28±.013	.15±.104	.03±.025	.23±.020	.10±.031	.14±.108	.44±.163	.12±.079	.61±.060
PFC-ACA	.89±.009	.79±.004	.15±.088	.02±.018	.69±.002	.75±.021	.15±.094	.69±.003	.13±.084	.51±.209

(b)

Table 2. Average results for the inter-tracker and intra-tracker distance using the spatial overlap. Higher values indicate that the trackers have similar results

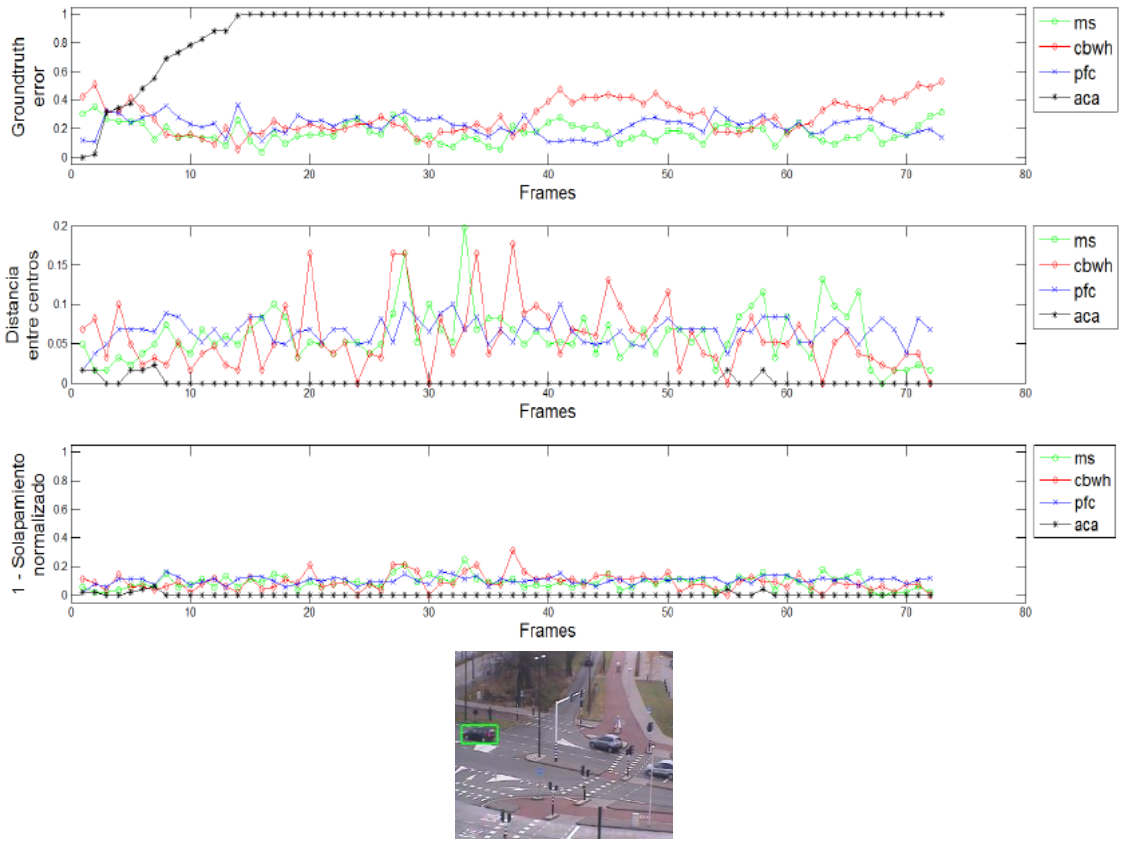


Figure 15. Similarity results for the sequence **I3_car_basic_2**. Top: ground-truth error, spatial overlap and normalized overlap score. Bottom: sample frame.

The final results of the fusion approaches are show in the following Table

Algoritmo	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Skiing
MS	.23±.004	.49±.013	.84±.076	.22±.001	.26±.014	.17±.005	.78±.048	.62±.086	.96±.020	.43±.030
CBWH	.20± .012	. 61±.035	.99±.032	.24±.006	.35±.020	.28±.013	.82±.042	.75±.105	.95±.013	.73±.095
PFC	.19± .008	. 50±.019	.28±.007	.66±.059	.35±.020	.22±.004	.77±.106	.54±.110	.91±.048	.16±.003
ACA	.08± .002	. 19±.031	.97±.017	.00±.000	.16±.005	.91±.048	.86±.081	.47 ±.024	.90±.060	.16±.004
Fusión Equitativa	.11± .001	. 35±.013	.91±.058	.16±.002	.21±.009	.13±.005	.79±.053	.49±.069	.90±.064	.23±.013
Fusión ponderada	.09±.001	. 47±.021	.99±.004	.08±.001	.27±.0152	.87±.083	.98±.003	.41±.057	.99±.004	.56±.175

Table 3. Average accuracy results for proposed fusion approaches based on the inter-tracker and intra-tracker distance computed previously.

2.4.3. Single-target multi-camera tracking

Third, a multi-camera single-target tracking approach that takes advantage of the consumption models developed within this task T3.2 (see section 2.1.1). This work has been published in the journal IEEE Transactions on Circuits and Systems for Video Technology.

In this work, we make use of the following consumption models for:

- Sensing: capturing frames
- Processing: computer vision task
- Communication: exchange of metadata among smartcameras

After decomposing these items for the sensing, processing and communication parts of a smart camera, we apply the proposed consumption models to a coalition-based approach for multi-camera target tracking [4].

The following figure provides an overview of the multi-camera tracker based on coalitions

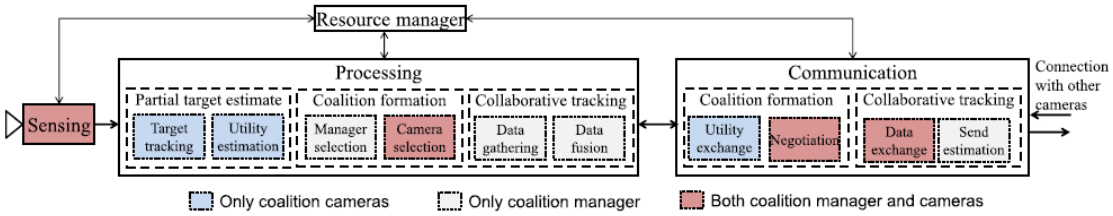


Figure 16. Operations performed by cameras for coalition-based target tracking [4]. Colors indicate subsets of operations for the existing coalition roles. Please refer to [4] for a detailed description of each block.

The following figure describes the setup employed for experiments (PETS 2009 dataset).

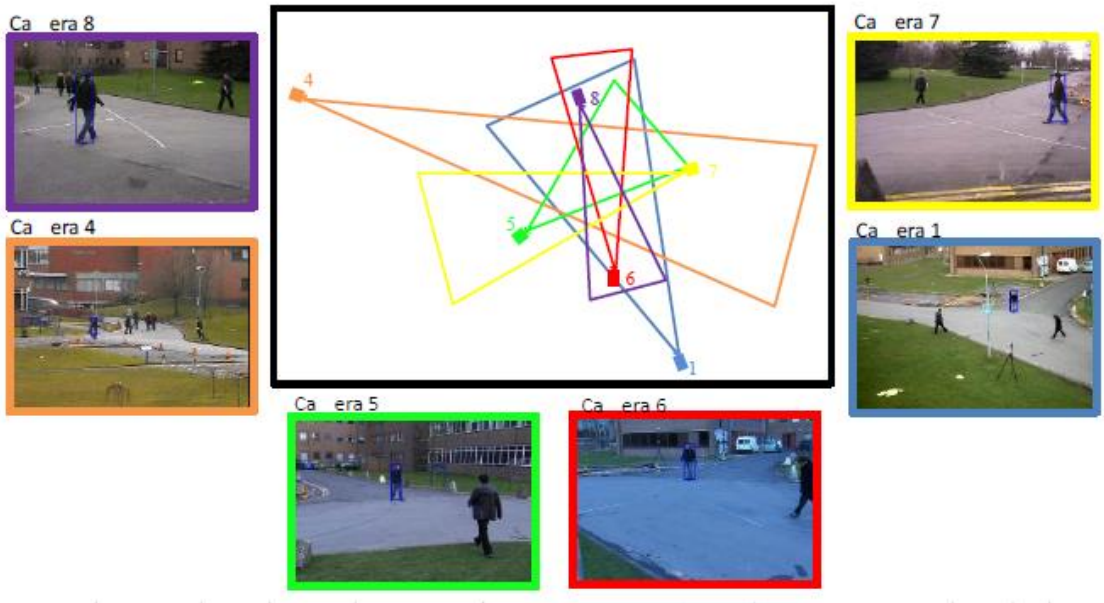


Figure 17. Fields-of-view on the ground-plane for the camera network of S2 L1 sequence (PETS2009) and target initialization on each view (blue box).

Some results are presented in the following figures. The following figure presents the evolution of the consumption and the associated tracking error when increasing the sensing framerate or processing clock.

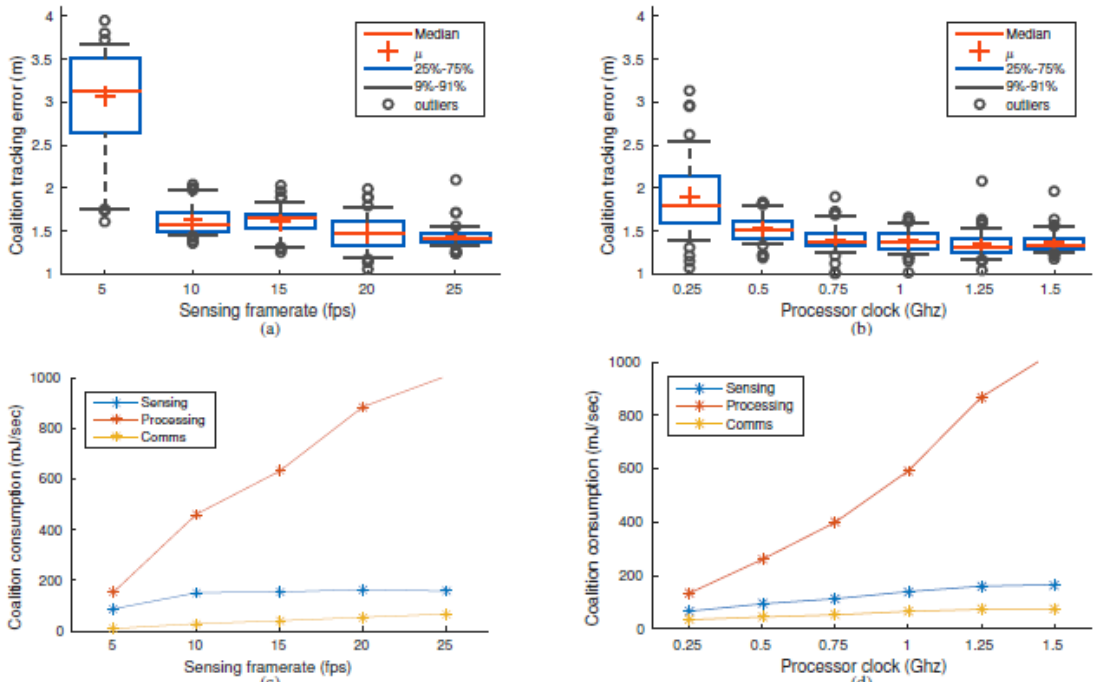


Figure 18. Coalition tracking error and associated energy consumption for dynamic sensing and processing capabilities (left and right columns, respectively). (a)-(c) Correspond to various sensing framerates (fp = 1.5Ghz) whereas (b)-(d) correspond to various processing clocks (framerate = 25fps).

Figure 4 presents the associated error and camera consumption when the data is captured at different framerates. We can observe some differences in the average tracking error due to

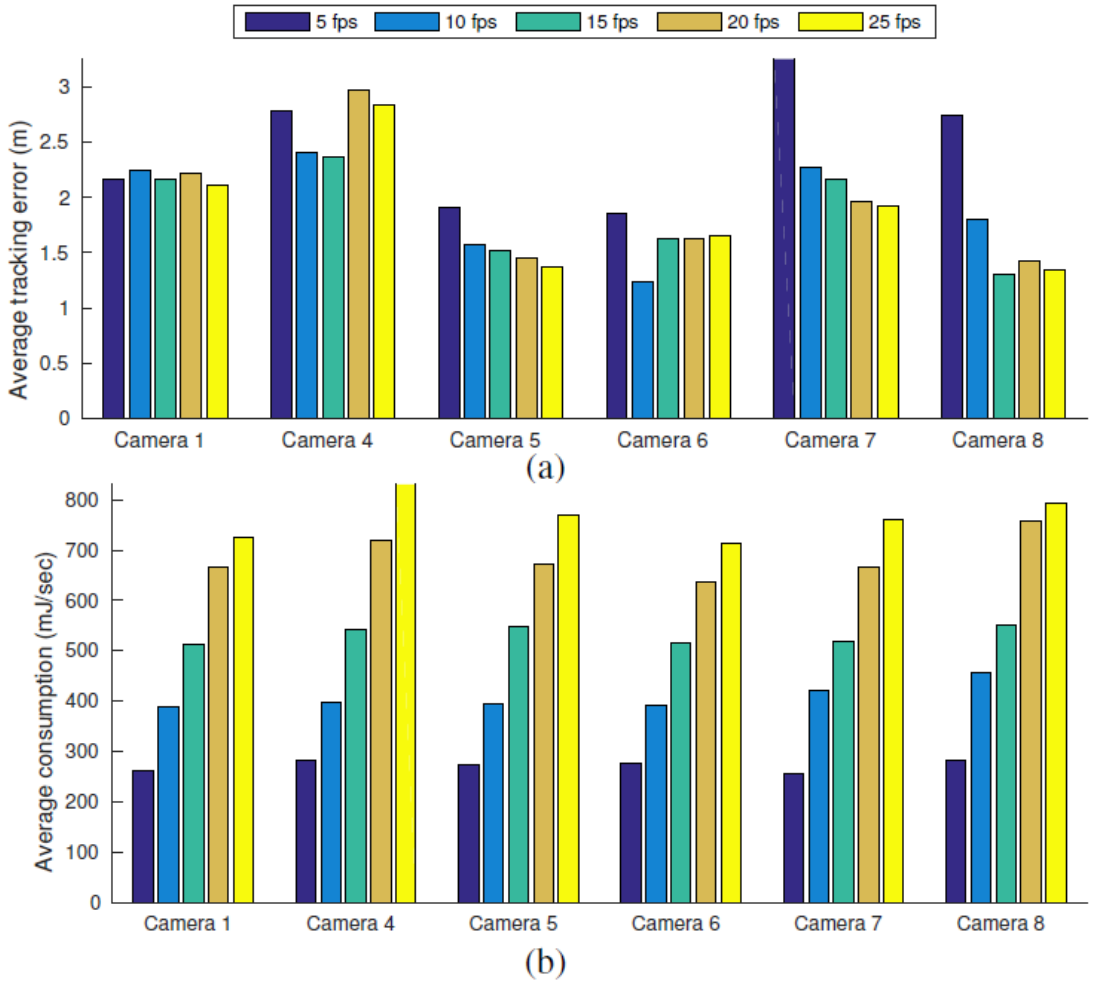


Figure 19. Camera tracking error on the ground-plane and energy consumption for various framerates. The processing clock is fp = 1.5GHz.

The following figure shows visual tracking examples for different framerates. While tracking accuracy increases in the selected camera views from 1fps to 5fps (first to second columns), there is no significant accuracy improvement from 5fps to 7fps (second to third columns), although camera consumption increases since more frames are captured and processed.

Figure 20. Visual tracking results for frame 21 on each camera view under different sensing framerates (fps). Green and blue boxes correspond to the estimated and ground-truth target locations for each camera. (From top to bottom rows: 1fps; 5fps and 7fps)

2.4.4. Multi-target multi-camera tracking

We have developed an approach for multi-camera multi-target tracking in smart camera networks based on target detection quality (see "T2.3 Quality analysis") and the simulator developed in the WP1. This work has been published in the journal *IEEE Computer*.

This case study considers distributed fusion where cameras exchange information to perform tasks without a leader organizing this collaborative processing [16]. Hence, no local fusion centers exist and often local communication is used (i.e. camera neighbours) which offers scalability for sensor fusion.

Consensus-based approaches are widely used to distributedly achieve the average over a quantity among the nodes of the network. For each consensus iteration, cameras share the data and then, compute the mean of the received data by local neighbours. As the number of iterations increases, cameras obtain the same value in a distributed fashion. This concept can be applied to track targets on the ground-plane by using a Kalman-Consensus Filter (KCF) [17]. Each camera runs a KCF whose output is broadcast to all neighbor cameras which take the average of the received target state. By repeating this process over time, all network nodes obtain the same information so the state of the target being tracked (e.g. its position) is known by all cameras. To improve capabilities of KCF and avoid problems when the target is not observed by a particular camera (i.e. it would share an empty result), the Information-Consensus filter (ICF) [18] is proposed to efficiently share data across the network by also exchanging measurement information of targets.

This case study considers a simulated camera network with eight wireless cameras is defined over an 500m-by-500m area where four targets move around the monitored area during 200s (see Fig. 6a). Cameras get measurements at a frequency of 4Hz (i.e. sampling time of 0:25s) and have a communication range of 250m. We compare the accuracy and energy consumption of KCF and ICF approaches under ideal and real network conditions over 10 independent runs. The following figure presents an schematic of the setup.

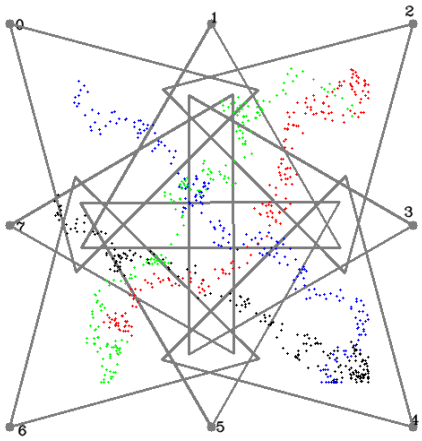
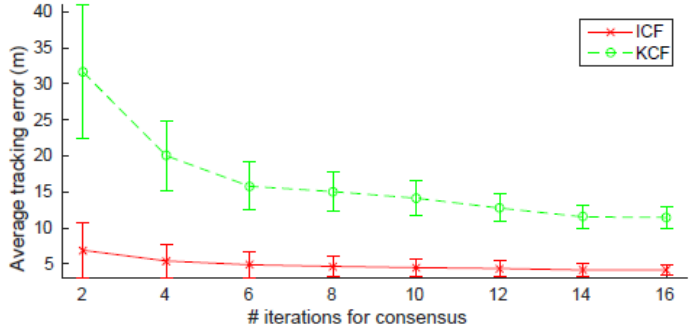
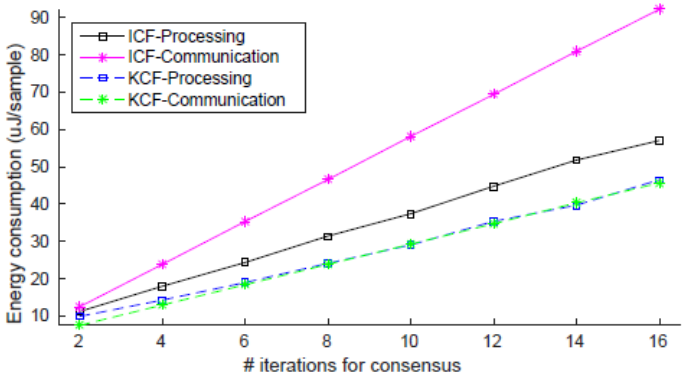


Figure 21. Simulated network composed of eight wireless cameras. Four sample trajectories of targets are in green, red, blue and black colours (only 100 samples are shown for each one).

The following figure presents the results for ideal network conditions. As expected, the tracking error decreases with increasing number of iterations since the estimation error of each camera is diffused over the other cameras. ICF outperforms KCF by sharing prior information about absence of measurements when the targets are outside the FoV of cameras. We compare the consumption of these two approaches. The same figure shows that ICF requires more than twice the energy of KCF for any iteration of the consensus. The increased ICF accuracy requires extra resources for processing and communication.



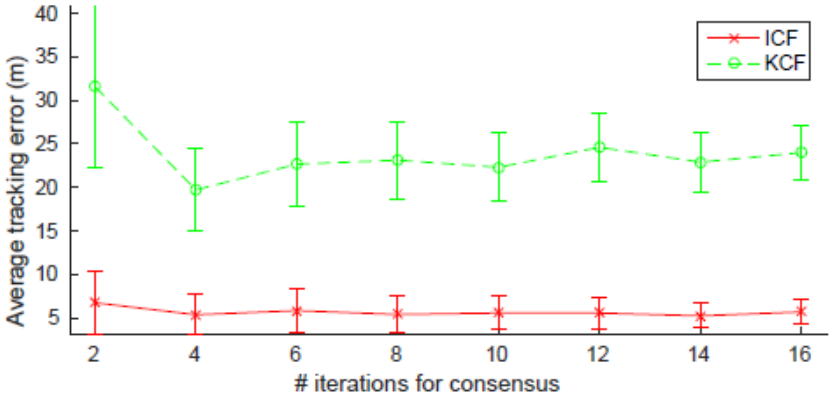
(a)



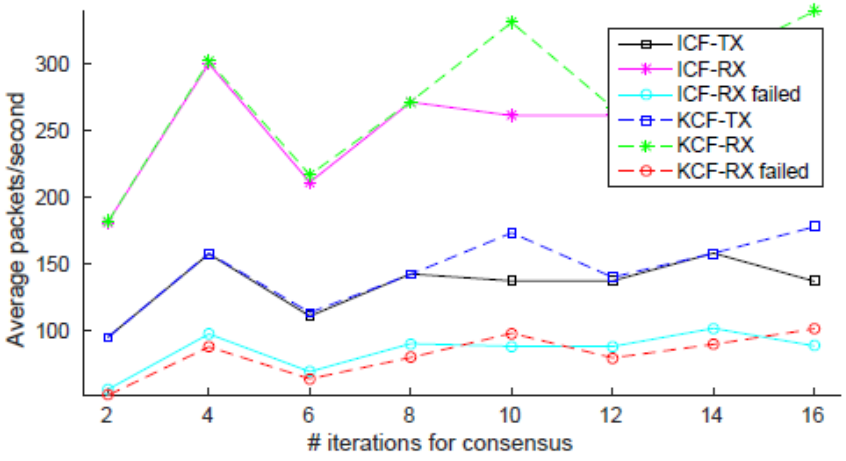
(b)

Figure 22. Comparative results for distributed-based target tracking using the consensus-based approaches (ICF and KCF) and assuming ideal network conditions.

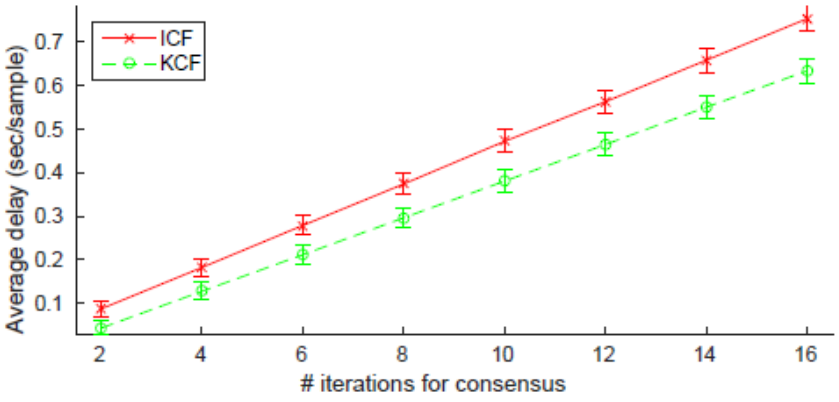
The following figure presents the same tracking results for real network condition. Real conditions should account many factors such as the transceiver (radio) models; the communication protocol (e.g. MAC); interference and attenuation of the wireless communication channel; and the latencies of the camera modules. Another key issue is the synchronization of the cameras in the network which can be internal and external.



(a) Average tracking error for all cameras.



(b) Average number of transmitted/received packets for all cameras.



(c) Average delay to process each sample by all cameras.

Figure 23. Comparative results for distributed-based target tracking using the consensus-based approaches (ICF and KCF) and assuming real network conditions.

Unlike ideal network conditions, the tracking error for both ICF and KCF does not decrease as the consensus iterations are increased. This dramatic change in the accuracy trend is due to the accumulated delay of the iterations in the consensus which is reported in the bottom graph.

2.4.5. Collaborative target tracking using Multiple Camera Networks

Target tracking is an important research line given its wide relevance in several fields such as video surveillance, medical research, business intelligence, unmanned vehicles, etc. Regarding the increasing popularity of SCNs, much of the literature focuses primarily on the integration of the tracking task in distributed networks. In this context, the estimation of the target position exploits the information gathered by all the network nodes, among which a consensus is established to solve a joint estimation. This collaboration is achieved thanks to consensus-based algorithms, which converge by averaging the information exchanged iteratively between the neighboring nodes. Locally, on every node, several approaches for estimating the target state from a set of observations are based on the Kalman filter. The algorithms merging these two strategies (average consensus and Kalman filtering) are the Information Consensus Filter (ICF) and the Multi-Target Information Consensus (MTIC). Those were the focus area of study in this project given their suitability in SCNs.

The Information Consensus Filter (ICF) was developed to solve one of the two main issues in vision-based solutions for distributed multi-target tracking, known as naivety (i.e. sensors may not be sensing the same targets). This is achieved by taking into account the quality of neighbors' information. Despite that naivety is addressed, ICF assumes that the measurement to track association is given. This is known as data association (i.e. which measurement of the available set corresponds to each target) and it is the remaining issue in Multi-Target Tracking (MTT).

The Multi-Target Information Consensus (MTIC) extends ICF to address the data association problem. To that end, the Joint-Probabilistic Data Association Filter (JPDAF) is utilized. JPDAF is a centralized method of associating existing targets with new available observations in each time step. Briefly, each target is mapped to a new measurement which is computed as a weighted probability average of the set of candidate measurements. MTIC integrates JPDAF and ICF to handle distributed multi-target tracking. However, MTIC fails in several conditions when working with real video. This work analyzed its shortcomings and proposed an improvement of JPDAF filter for data association.

Regarding the proper evaluation of such distributed algorithm, this project used the WiSE-MNet++ open source simulator. The Wireless Simulation Environment for Multimedia Sensor Networks (WiSE-MNet++) is a modular simulation framework that allows the testing and modeling of distributed algorithms for Wireless Multimedia Sensor Networks (WMSNs), i.e., networks composed by sensors dealing with complex vectorial data (such as video and audio). The design of WiSE-MNet++ is flexible and a camera model and its functionalities are provided to enable the simulation of resource-aware distributed computer-vision algorithms at a high level of abstraction.

The goal of this work was to take a real scenario into account. Thus, the first aim of this project was to improve the current implementation in WiSE-MNet++ to support the new requirements related to the use of real-life data (and thus issues found in real camera networks), such as target detection and removing.
Such development started with the integration of a detection layer. Three different detection methods were included and can be selected by the user for each simulation. The detection is performed for each camera view and then mapped to a common coordinate system (ground plane) regarding multi-camera calibration techniques. The following figure depicts the entire scenario used for simulations. Each camera has different observations of the existing targets, which will not be usually exactly the same for each view.

Figure 24. Case scenario (PETS09 dataset) with seven cameras and their position in the ground plane

As for the distributed tracking task, every camera needs to be using same data, this is to say that each measurement needs to be projected to a common world before MTIC is performed, see the following figure.

Next step was the integration of the set of local measurements as the input of the MTIC algorithm. The original parameters of the algorithm were designed for a synthetic simulation example, however, their value needed to be set for a real scenario. Here, we realised that the distribution used to model false measurements or clutter (Poisson

distribution) does not truly represent real cases. Also, there were some specific challenges not handled by the original MTIC proposal, such as the birth and death of targets. Two protocols (remove lost/gone targets and create new ones during runtime) were implemented in the WiSE-MNet++ distributed environment in order to solve both situations.

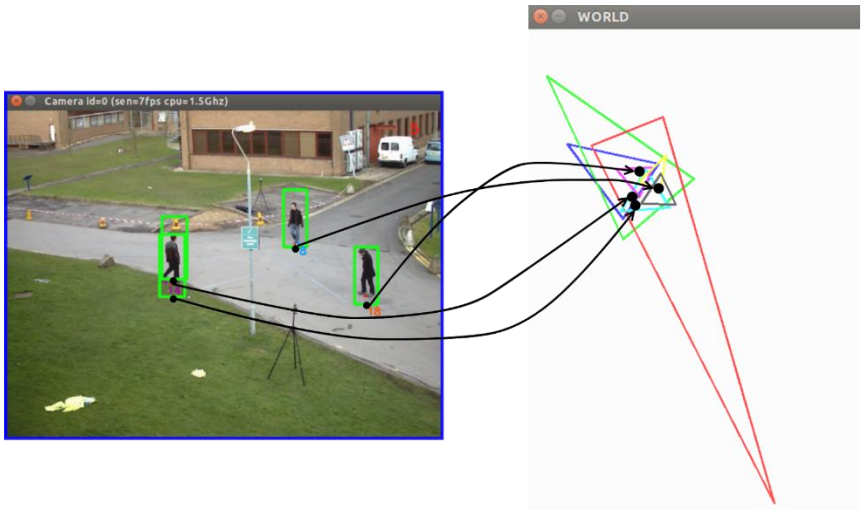


Figure 25. Example of projection from 2D camera detection to 3D world target position

Once the system was designed and fully working with real-life data, its performance was evaluated. It was then clear the data association issue. MTIC, in particular JPDAF, has poor performance when there are multiple targets in the scene and its positions come close. In this case, their tracks are either mixed or lost, see Figure 3. In order to solve this track coalesce issue, the first approach was to enrich the measurements vector by exploiting multiple features, from position coordinates to color information. This approach was still poor due to two specific aspects of JPDAF.

First, noting that JPDAF was designed for radar applications, where more than one measurements are usually obtained from each target, it is understandable the computation of a weighted mean measurement, also for smoothing reasons. However, in video tracking applications such smoothing may cause coalescence. Second, JPDAF assumes that the target states are independent, this is to say that a measurement can be assigned to one target no matter whether the same measurement was already assigned to another target.

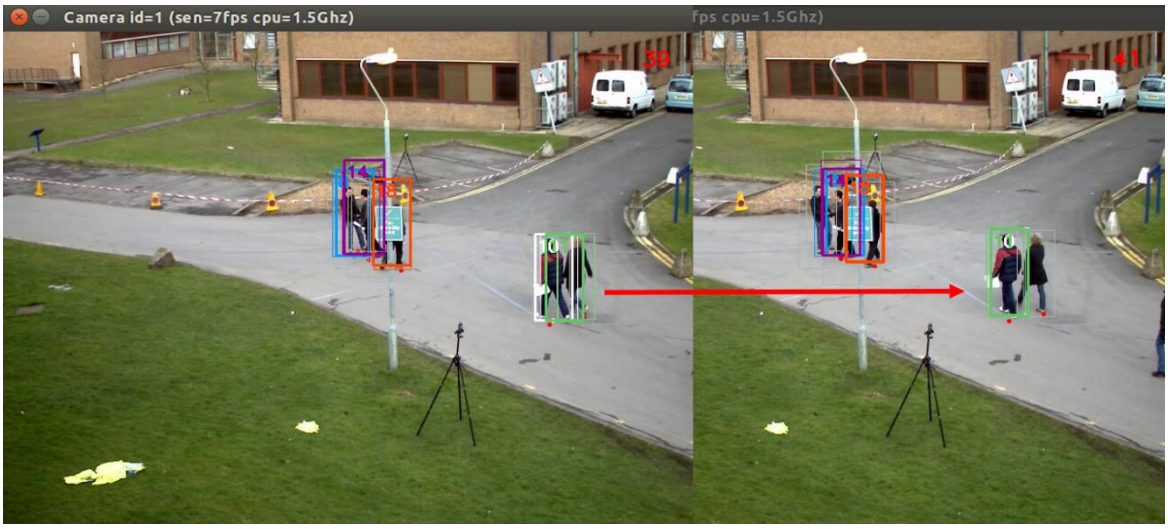


Figure 26. Example of track coalescence of JPDAF, tracks of two different targets are joined.

In the view of the above, two different computation approaches were implemented: NNJPDAF and RNNJPDAF. NNJPDAF, instead of computing an average, assigns the most likely measurement, thus avoiding the smoothing. RNNJPDAF is the recursive approach, which accounts for the dependence between target states (in short, a measurement that is likely assigned to a target can not be assigned to any other). Figure 4 shows the same situation of Figure 3 but using the RNNJPDAF approach.

Figure 27. Example of solved track coalescence using RNNJPDAF.

3. Conclusions and future work

3.1. Achievements

As summary, the achievements of task 3.2 are:

- A model describing the usage of resources of smart cameras which enables the development of collaborative approaches based on resource-aware policies.

- Collaborative approaches to adapt parameters during runtime via maximizing the agreement of independent sources. This strategy has been applied to
  - shadow detection
  - people detection.

- Several approaches for collaborative video tracking and detection based on quality signals for detections and tracking results. The following setups have been explored to combine multiple algorithms for:
  - single-target and single-view
  - single-target and multi-view
  - multi-target and multi-view

3.2. Future work

As future work, we will focus on the following:

- Collaborative people detection. Improvement of the previous prototype
- Combination of multiple features for single-target and single-view
- Combination of multiple algorithms and features for multi-target and single-view

4. References

[1] Juan C. SanMiguel, Andrea Cavallaro, "Energy Consumption Models for Smart-Camera Networks", IEEE Transactions on Circuits and Systems for Video Technology, (online September 2016), IEEE, ISSN 1051-8215 (Digital Object Identifier 10.1109/TCSVT.2016.2593598).

[2] R. Likamwa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy Characterization and Optimization of Image Sensing Toward Continuous Mobile Vision," in International conference on Mobile systems, applications, and services (MOBISYS), 2013, pp. 69–81.

[3] A. Redondi, D. Buranapanichkit, M. Cesana, M. Tagliasacchi, and Y. Andreopoulos, "Energy Consumption of Visual Sensor Networks: Impact of Spatio-Temporal Coverage Based on Single-Hop Topologies," IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 12, pp. 2117–2131, Dec. 2014

[4] J. C. SanMiguel and A. Cavallaro, "Cost-aware coalitions for collaborative tracking in resource-constrained camera networks," IEEE Sensors Journal, vol. 15, no. 5, pp. 2657 – 2668, May 2015

[5] Z. He, D. Wu, Zhihai He, and Dapeng Wu, "Resource allocation and performance analysis of wireless video sensors," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 5, pp. 590–599, May 2006.

[6] J. C. SanMiguel and S. Suja, Skin detection by dual maximization of detectors agreement for video monitoring, Pattern Recognition Letters, vol. 34, no. 16, pp. 2102 2109, 2013.

[7] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in Proceedings of Neural Information Processing Systems, 2009, pp. 2035–2043

[8] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," IEEE Trans. Image Process., vol. 22, no. 1, pp. 314–325, Jan 2013.

[9] E. Erdem, S. Dubuisson, and I. Bloch, "Fragments based tracking with adaptive cue integration," Comput. Vis. and Image Understanding, vol. 116, no. 7, pp. 827 – 841, July 2012.

[10] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," Int. Jour. of Comput. Vis., vol. 111, no. 2, pp. 213–228, 2014.

[11] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," Int. Jour. of Comput. Vis., vol. 77, no. 1-3, pp. 125–141, 2008.

[12] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Scale and orientation adaptive mean shift tracking," IET Comput. Vis., vol. 6, no. 1, pp. 52–61, Jan 2012.

[13] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 10, pp. 2002–2015, Oct 2014.

[14] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in IEEE Conf. on Comput. Vis. and Pattern Recogn. (CVPR), June 2012, pp. 1830–1837.

[15] D. Wang, H. Lu, and M. Yang, "Least soft-threshold squares tracking," in IEEE Conf. on Comput. Vis. and Pattern Recogn. (CVPR), June 2013, pp. 2371–2378.

[16] M. Taj and A. Cavallaro, "Distributed and Decentralized Camera Tracking," IEEE Signal Processing Magazine, vol. 28, no. 3, pp. 46–58, 2011.

[17] R. Olfati-Saber, J. A. Fax, and Murray R.M., "Consensus and Cooperation in Networked Multi-Agent Syst." IEEE Proc., vol. 95, no. 1, pp. 215–233, 2007.

[18] A. T. Kamal, J. Farrell, A. K. Roy-Chowdhury et al., "Information weighted consensus filters and their application in distributed camera networks," IEEE Transactions on Automatic Control, vol. 58, no. 12, pp. 3112–3125, 2013.