



Distributed tracking in self-organized silly camera network

Lobna Ben Khelifa

► To cite this version:

Lobna Ben Khelifa. Distributed tracking in self-organized silly camera network. Electronics. Université Clermont Auvergne [2017-2020], 2020. English. NNT : 2020CLFAC071 . tel-03261776

HAL Id: tel-03261776

<https://theses.hal.science/tel-03261776>

Submitted on 16 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ CLERMONT AUVERGNE
ÉCOLE DOCTORALE: SCIENCES POUR L'INGÉNIEUR

THÈSE

présentée par

Lobna Ben Khelifa

Pour obtenir le grade de:

DOCTEUR DE L'UNIVERSITÉ CLERMONT AUVERGNE

Spécialité: Électronique et Architecture de Systèmes

Titre de la thèse:

Distributed tracking in self-organized silly camera network

Thèse soutenue le 25 Septembre 2020 devant le jury composé de

Président	Mme. Evelyne Gil
Directeur de thèse	M. François Berry
Encadrant	M. Jean Charles Quinton
Rapporteurs	M. Julien Dubois M. Jorge Fernández-Berni
Examineurs	M. Richard Kleihorst M. Luca Maggiani



DOCTORAL THESIS

Distributed tracking in self-organized silly camera network

Author:

Lobna BEN KHELIFA

Supervisors:

François BERRY

Jean Charles QUINTON

*A thesis submitted in fulfillment of the requirements
for the degree of Docteur de l'Université Clermont Auvergne*

at the

**DREAM Research Group
Institut Pascal**

UNIVERSITÉ CLERMONT AUVERGNE
Ecole Doctorale des Sciences Pour l'Ingénieur
Institut Pascal

Abstract

Distributed tracking in self-organized silly camera network

by Lobna BEN KHELIFA

The Ant-Cams network is a new model of camera networks that can be used for environment monitoring and understanding. Usually, such networks are composed of smart cameras, which benefit from high resolutions, powerful processing capabilities and strategic viewpoints on the environment. Here, the network uses silly cameras, defined by much lower specifications forming the Ant-Cam model. This latter is inspired from the world of ants, where ants are able to solve complex problems by communicating despite their limited capabilities.

This model can reach efficient high-level understanding in spite of the limited information provided by each silly camera. We rather focus on the interactions between those cameras to increase the performance of the system where data exchanged between the cameras, such as timing or features characterizing the events, is as important as the visual information extracted locally.

Unlike many existing visual sensor network which require some prior knowledge of the network such as position and neighbors, the Ant-Cams do not require any knowledge about the network configuration (e.g. camera location). Once starting working and the system reaches a steady state, all the necessary information can be found through interacting with neighbors. Thus, we can find the topology of the network where links are reinforced based on observed transitions, the paths adopted by the targets and if space covering is sufficient.

Keywords: Smart Camera Network, Predictive modeling, Distributed problem solving

To my family...

Contents

Contents	7
List of Figures	9
List of Tables	13
I Introduction	17
1 Iot and IoSmartT	18
2 Ant world and IoSillyT	19
3 Research question and contribution	20
4 Thesis outline	21
II Self-organized network	23
1 Introduction	24
2 Overview	24
3 Self-organization	30
4 Network construction	30
5 Network description	37
6 Conclusion	44
III Distributed tracking in smart camera network	45
1 Overview	46
2 Pre-event Connectivity	52
3 Event Connectivity	53
4 Post-event connectivity	58
5 Conclusion	60
IV Evaluation	61
1 Smart Camera Simulation Tool	62
2 Network Evaluation	65
3 Conclusion	72
V Real Environment evaluation	75

1	LobNet platform	76
2	Implementations	82
3	Evaluation	92
4	Conclusion	99
VI	Conclusions and Perspectives	101
1	Conclusions	102
2	Perspectives: Dynamic Ant-Cam network: Towards real ants world	102
A	Existing platforms	119
B	panoramic view of IoT applications	121

List of Figures

II.1	(A) Star topology, (B) Cellular topology (C) Tree topology and (D) Mesh topology	31
II.2	Internal and external event in the Ant-Cam	34
II.3	Illustration of the local network topology.	35
II.4	Example of links created between cameras.	40
II.5	Example of a camera network scenario	41
II.6	Relationship between target vector \underline{x} and cameras.	43
II.7	Relationship between target vector \underline{x} and cameras.	43
III.1	Top-down and bottom-up	52
III.2	Network event evolution	54
III.3	Example of a target scenario with different camera views	55
III.4	Example of unpredictable sequence under Markov assumption	56
III.5	Instance of network: Connectivity graph based on events (L2,L3) and communication graph allowed by technology used for it (Wifi, LoRa..)	57
III.6	Network representing the relation between the re-identification estimation and the prediction received.	58
III.7	Camera update after each external event	59
III.8	Network architecture: each column represent a camera participating in the tracking task. Black node correspond to the processing available in the camera.	60
IV.1	Illustration of tested scenarios. Each camera is represented by a red circle with its FOV indicated by red lines.	62
IV.2	Example of cameras declaration with the simulator.	63
IV.3	Example of a camera definition function with the simulator.	63
IV.4	Example of targets declaration with the simulator.	63
IV.5	Example of a target definition function with the simulator.	64
IV.6	Example of the network environment at different instants with the simulator.	64
IV.7	Examples of the visual information detected by the 7 cameras at different instants.	66
IV.8	Example of the network construction, markov assumption is consider here($n=0$), 0 iteration.	67

IV.9	Example of the network construction, markov assumption is consider here($n=0$), 1 iteration.	67
IV.10	Example of the network construction, markov assumption is consider here($n=0$), 20 iterations	68
IV.11	probabilities of the link between camera 1 and j.	69
IV.12	Simulation of 39 nodes	69
IV.13	Evolution of the estimated time	70
IV.14	$p_{i path}$ for the same path($\sum = 1$)	71
IV.15	Probability in node C_{21}	73
IV.16	Example of evaluation scenario	74
IV.17	recurrence of different paths.	74
V.1	The Ant-Cam	76
V.2	The Ant-Cam architecture	77
V.3	Example of images taken using Ant-Cam, the resolution is 30*30 pixels	78
V.4	Different images captured with camera 1 for a target coming respectively from cameras 2, 3 and 4.	84
V.5	Projection of the target in the 2D plan.	85
V.6	Illustration of the different transformations. Spatial transformations between camera A and B. Temporal transformations between a current target detection and the reference target.	88
V.7	A series of steps is followed in each of the two cameras in order to find the transformation between each other. Starting with the Principal Components Analysis (PCA), a projection is applied to reduce the dimensions of the target space before estimating the transformation.	89
V.8	Network architecture: Each column represents a camera participating in the tracking task. Grey nodes correspond to the processing available in the camera.	91
V.9	Variation in the re-identification rate according to the number of eigenvectors considered. S.T refers to Spatial Transformation, T.T refers to Temporal Transformation. S.T.T points to Spatial and one Temporal Transformation. Total.T refers to the 3 transformations.	92
V.10	(a) corresponds to the initial features reference for camera 1, and (b) for camera 2. (c) correspond to the input features after each detection for camera 1 for 5 target detected, and (d) for camera 2. (e) correspond to the features generated via spatial transformation using (c). (f) correspond to the features generated via spatial and temporal transformation of (c). (g) correspond to the features generated via temporal transformation of (c). (h) correspond to the generated features using the whole transformation of (c) and (d). (e), (f), (g) and (h) are then used for comparison with the input (d)	93
V.11	VHDL code for the Background-Foreground Segmentation (BFS)	95
V.12	Different network states.	96
V.13	Different events in the network.	97
V.14	The vision graph building during the network's run time.	98
V.15	Evolution of the probability in the node 1 in the network.	99
VI.1	Factors of calibration	103

VI.2	Blue refers to data stored in memory for further processing. Green corresponds to the initialization of the transformations. Oranges represent the transformation performed in each camera following each detection. The pink corresponds to the matching between the generated data through the transformations, and the extracted data.	105
B.1	panoramic view of IoT applications	122

List of Tables

II.1	Comparison of conceivable network configurations.	25
II.2	Benchmark with current state of the art	29
II.3	Summary of used notation for the network model	38
II.4	Summary of used notation for the camera model	40
II.5	CRO/CRI	42
III.1	Summarize of the reconfiguration methods used for tracking, C refers to centralized processing and D to distributed processing	49
III.2	Summarize of the reconfiguration methods used for coverage, C refers to centralized processing and D to distributed processing	52
III.3	Different parameters used for re-identification estimation.	59
IV.1	Overview of the used parameters.	65
IV.2	Ranks of the Ant-Cams	71
V.1	Specifications and Electrical characteristics	79
V.2	the parametric values used for the BFS	83
V.3	Notations for used parameters	84
V.4	Pairwise identification for two datasets. SS refers to the detection of the same side detection, whereas DS is for the detection of different sides.	94
V.5	Tracking performance in the network in dataset 2.	94
V.6	the parametric values used for the BFS	95
V.7	Resource Utilization of the bfs on the Ant-Cam Platform.	95
VI.1	self calibration	104
A.1	Existing platforms for SCN	120

Glossary

AoA Angle of Arrivals. [33](#)

AODV Ad hoc On Demand Distance Vector. [33](#), [81](#), [95](#)

AOP Ant Optimization Path. [19](#)

BFS Background-Foreground Segmentation. [10](#), [13](#), [82–84](#), [95](#)

DSP Digital Signal Processor. [78](#)

FOV Field Of View. [19](#)

FPGA Field Programmable Gate Array. [21](#), [79](#), [94](#), [95](#)

GSM Global System for Mobile Communications. [32](#)

IoT Internet of Things. [18](#), [33](#), [81](#)

LQI Link quality indicator. [33](#)

LTE Long Term Evolution. [32](#)

NCF Near-Field Communication. [32](#)

PCA Principal Components Analysis. [10](#), [86](#), [89](#), [92](#), [95](#)

PIR Passive InfraRed. [18](#), [78](#), [82](#), [94](#), [95](#)

RSS Received Signal Strength. [33](#)

SCN Smart Camera Network. [24](#), [56](#), [77](#), [102](#)

SSN Smart Sensor Network. [18](#), [33](#)

TDoA Time Difference of Arrivals. [33](#)

ToA Time of Arrivals. [33](#)

UMTS Universal Mobile Telecommunications System. [32](#)

Wi-Fi Wireless Fidelity. [31](#), [32](#)

WSN Wireless Sensor Network. [31](#)

CHAPTER I

Introduction

1 Iot and IoSmartT

Thanks to the technological improvements, the concept of **Internet of Things (IoT)** is emerging progressively to include various electronic devices and new application fields. Medical care, agriculture, traffic control, crime prevention and shoplifters' identification are just few instances of the **IoT** wide use-areas. Over the past few decades, the **IoT**-based communication has been ensured through the following main architectural units: (i) a sensing unit representing the interface with the environment and providing measurement, and (ii) a communication unit playing as a network infrastructure to broadcast data between a set of different devices and a central control module. Nevertheless, this central architectural layout of the **IoT** concept is considered as limited at present time mainly because it handles only the data broadcast. As a result, the **IoT** basic structure has entailed numerous drawbacks and vulnerabilities. Mainly, these vulnerabilities are linked to the security level, the communication reliability requirements and the huge data storage capacities. Indeed, a great focus is now given to the **IoT** security concerns. Malicious hacking and spoofing attacks may happen through the connected devices broadcasting private and valuable data. Consequences of such attacks and forbidden access to data are more hazardous in case of critical safety systems such as control insulin pumps, implantable cardioverter defibrillators or several control functions of automobiles. Moreover, the early described architecture of the **IoT** systems cannot really support the huge exchanged data amounts between the early stated units. Handling a large amount of information provided by sensors requires a reliable communication technology that guarantees a suitable broadcasting range/bandwidth and an optimal energy consumption. Notably, a crucial trade-off between all these requirements is not evident. For instance, a considerable researcher work has attempted to provide self-powered cameras which are simultaneously able to ensure streaming tasks. Otherwise, the collected sensorial data need to be sent periodically to the central unit to be proceeded and then saved. The amount of data involved in the **IoT**-based application depends on features of the employed sensors. Several simple applications require just the use of scalar sensor such as **Passive InfraRed (PIR)** sensors which induce in general a small amount of data easily safeguarded. In other cases, the use of matricial sensors is mandatory. Last ones imply high data acquisition and storage requirements. Regarding to challenges and new requirements of the **IoT** applications, overcoming limitations presented by the classical centralized **IoT** layout is extremely recommended. Consequently, the actual trend is to move towards the use of smarter embedded **IoT** devices. In fact, several researchers have proposed to integrate a data processing level into the **IoT** boards. In such a way, the local processing units join sensing and communication layers to change the architectural layout to a distributed one. Thus, only semantic information is exchanged over the **IoT** network. Then, only most significant and critical results will be reported. It is worth mentioning that moving from a centralized architecture towards a distributed one offers a great autonomy degree to overall network nodes. This fact has turned the **IoT** network into a **Smart Sensor Network (SSN)**. More particularly, in the context of the visual sensors, a new generation of smart cameras is currently attracting more and more of attention due to its efficiency in tracking targets, monitoring public areas, supervising manufacturers, and identifying risks and accidents in highways. Herein, a local level proceeding of the measured data is taking place. The communication between the different parts of the network of smart cameras is triggered only to respond to the environmental changes such as movements. The set of events, captured by each local camera, are then analyzed in a higher level to provide a global perception report of the monitored scene. This could be effectively feasible as

long as the network scale remains small. However, with a huge number of cameras, the network of connected cameras will contribute to provide a collection of useless redundant information. In such a manner, the tracking efficiency depends not only on each camera capabilities, but it is tightly linked to the whole network abilities to make an abstraction of the different events and notify the central control only by the results. Correspondingly, the interactions occurring between the used cameras must be appropriately modeled to enhance the overall network global performances.

2 Ant world and IoSillyT

Nature includes many instances of phenomena that can describe and explain different concepts such as communicating systems and self-organized processes. Numerous behaviors noticed in the animal's world have inspired research efforts to solve technological problems in different fields. Such inspiring systems from nature are usually referred as reactive systems where the communication is a key component to understand and analyze facts. What may interest us here, are animals that can interact together and then accomplish cooperatively a given task. The well-organized and synchronous flying of birds is a spectacular example for self-organization processes, modeled by the theory of coupled oscillators. Thanks to this example, several distributed mechanisms were easily modeled and exploited in different areas such as neuroscience and physics for coordination and synchronization. The fact that fishes swim in well-structured shoals with the optimal motions, has also opened a vast discussion on how they can coordinate their motion to move without disturbing other's motion by the wave. Ants model, which are the most used, are attractive for their capacity to find the shortest path to a desired destination. Hence, different algorithms such as [Ant Optimization Path \(AOP\)](#) have been developed following the ants' behavior.

The main reason of such a success in achieving a given task by animals is their collaborative behavior. It is obvious that starting from very limited individual perception capacities, animals can reach a higher level of performances by working together. However, their organizational arrangement does not require any central coordination, but allows only a "point to point" communication. This kind of interaction offers large opportunities to optimize the communication and to acquire a great awareness about the environmental changes. At this stage, an important question arises about this level of animals' perfect self-organization. How may the networked devices take an advantage of these examples to reach such level of self-organization?

In camera network word, conventional cameras benefit from high resolutions, powerful processing capabilities and strategic viewpoints on the environment. Thus, we classically try to optimize the hardware parameters, including position, orientation, [Field Of View \(FOV\)](#), zoom, focus and resolution at the camera level. But this is also true for software components (e.g., detection algorithm) as well as topological components (e.g., number of necessary cameras to cover the task space). In this work, we rather choose to work with silly cameras, defined by much lower specifications, called Ant-Cams. We put forward a novel model following the principle of smart dust, where the Ant-Cams are scattered in the environment without a priori knowledge of their positions and their [FOVs](#). Despite their limited sensing and processing capabilities, the Ant-Cams can reach efficient high-level understanding thanks to their communication abilities. Fully exploiting the cameras interactions, the system is able to learn regularities and then infer from distributed sequences of events, passed between Ant-Cams.

3 Research question and contribution

The objective of this thesis is to demonstrate the capabilities of distributed networks of intelligent cameras defined by very low specifications to perform various tasks such as self-organization and tracking. Without any prior knowledge about the environment, cameras are able to organize themselves and adapt to the changing conditions of their environment. Monitoring and re-identification tasks at the camera level is therefore determined by the camera's own perception of the environment as well as the information transmitted and retrieved by other cameras. This information exchange is essential to improve their local performance as well as that of the entire network. The main contribution to the current state-of-the-art are as follows:

Network Construction and self-organization: Starting with wholly unknown environment, each camera learns about its neighborhood during the run-time. Indeed, and following the ant analogy, each camera spreads "pheromones" on the network, thanks to the communication technologies, to establish knowledge. These artificial pheromones, generated after each target detection, contain all its characteristics (visual, temporal and spatial information). Links are then established according to the behavior of the monitored objects. Then cameras can organize themselves throughout the network and form the so-called vision graph without central coordination. At the camera level, this knowledge reduces the communication effort with other cameras in the network, while maintaining the overall tracking performance of the entire system at a high level.

Distributed tracking model: We propose an approach for target tracking in a very low resolution (30×30) visual sensors network. This last is fully distributed and aims to accomplish the tracking task without any supervision. Each camera uses the stimulation-response combination to perform specific requirements: external stimuli that are detection following environmental measures and internal stimuli that are notifications from other cameras after external stimuli to predict. External and internal stimuli can help the camera to develop a deep understanding of its environment and build its own vision domain. These online learning of associations can lead to high-performance tracking from a global system point of view, making it possible to create a spatial-visual-temporal correlation between cameras and targets. The correlation enhances the accuracy of its prediction in terms of on-site processing or communication. In addition, by analogy to PageRank used by google to rank the Web pages, we introduce the concept of CamRank: CamRank-In and CamRank-Out. Both aim to rank cameras according to their relevance in the network.

New technique for low resolution images processing: In our modeling, we focus on low computational efforts and time-saving processing without the need for high-end hardware processors. The low specifications of Ant-Cams make the implementation of computationally intensive methods of tracking impossible. In a smart camera network, particularly when dealing with fully decentralized processing, we focus on the amount of output data and the manner of deploying it in a second camera. A critical challenge in tracking tasks is to decrease the volume of transmitted data. This typically involves the elaboration of an appearance model and a position identifier. A target generating an observation measurement in the network is portrayed by a set of features that must be relevant not only by the camera itself, but by the entire network. For this, we create the associations between the observations of different cameras that we define as magic matrix at two levels. The first is a spatial mapping. It is a camera-to-camera translation between the two observations of the same target by two cameras successively detecting it. The second is a temporal one. It is a translation between observations of two different targets by the same camera

after each detection. The re-identification is then based on the matching between different observations.

Simulations and Real World Deployments: A simulation tool was developed for the purpose of testing the network model. It offers the possibility to simulate several cameras, along with their positions, orientation and field of view, as well as targets with different characteristics such as shape, size, velocity and path followed in the network. The simulator has been used in evaluating the model proposed in this work with different scenarios. Following this simulation, we deployed our methods in a platform of intelligent cameras. The choice of very low-resolution sensors decreases privacy issues, costs, computing requirements and energy consumption. A functional set of custom Ant-Cams was designed according to the project specifications: very low-resolution images limited to 900 pixels, light processing capabilities with a [Field Programmable Gate Array \(FPGA\)](#) and advanced communication ability thanks to SmartMesh IP protocol.

4 Thesis outline

The remaining chapters of this thesis are organized as follows:

- Chapter 2 details the network model. This Chapter introduces the merging of the ant world and smart cameras to enable fully distributed tracking. It highlights how the camera can be fully autonomous and acts as a autonomous agent, which leads to a self-organized network without any centralized host.
- Chapter 3 describes how our model performs distributed tracking tasks. Furthermore, it outlines various uncertainties and their impact on the overall performance of the network. ,
- Chapter 4 The proposed approach is first evaluated with a simulation environment. This chapter gives an overview of the used simulator and its features. First, we assess the self-organization of the network based on each camera strategies learnt during run-time. Thus, we evaluate the network robustness while accomplishing a distributed tracking. Finally, the CamRank-In and CamRank-Out are evaluated and their importance is highlighted.
- Chapter 5 introduces the hardware platform developed during this thesis called Lob-Net. This platform is distinguished by its low specifications: 900 pixels for the image resolution, plain processing with MAX10 [FPGA](#) but talkative thanks to the Smart Mesh IP protocol used. It depicts how Ant-Cams have been deployed. We delineate the environment setup and different scenarios scheduled. Thus, the network model presented previously is evaluated. Furthermore, this chapter discusses the different real environment problems and their impact on the overall performance of the cameras and the whole network.
- Finally, chapter 6 draws the conclusion of the work presented in this thesis and outlines the possible perspectives.

CHAPTER II

Self-organized network

1 Introduction

While establishing a wireless [Smart Camera Network \(SCN\)](#), a priori knowledge of network topology, environmental characteristics or neighborhood relations between cameras can significantly improve network surveillance performance. However, such task requires the intervention of a central host, such as a server, or a human administrator to establish a well-designed implementation plan. This plan must take into consideration the exact number of cameras, the installation cost, the cameras' calibration (zoom, field of view, direction) and configuration, and the flexibility of the arrangement according to environmental concerns. However, under certain conditions, this information is unavailable. This risk increases particularly when the network is intended for an uncontrolled environment such as control at the bottom of an ocean, spying on the battlefield, or landing in a large warehouse, or at home. In this case, the sensors are usually deployed by dropping from aircraft or missile, catapulting, but can also be placed one by one with the assistance of a human operator or by a robot.

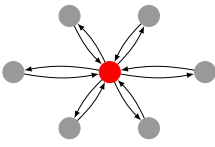
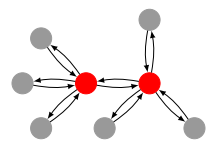
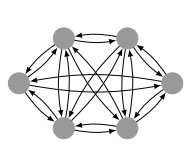
These uncontrolled parameters are not appearing only in pre-deployment phase, but even afterwards. Indeed, devices can cause change in the network topology, either by disappearing, due to malfunctioning or lack of power or problem of reachability due to noise jamming or interference. This is also can be due to task details which can lead the camera to change the hardware parameters to better perform this task. Topology is also one of the relevant metrics to be taken into account in the assessment. Indeed, it corresponds to the logical architecture of a network, defining the links between the motes of the network and a possible hierarchy between them.

While establishing a [SCN](#), the desired applications influence the choices of hardware and software parameters. New generation of [SCN](#) refuse to comply with the unique implementation and a prior fixed infrastructure support. The main challenge in this case is the self-organization: a set of independent motes must independently build a fully autonomous network without the need of human interaction or any specific knowledge about the network. Two main reasons for such a choice: (i) it is challenging to identify an optimal structure to effectively cover the environment. (ii) the latter can evolve over time (eg; light condition). For this, new devices can be either re-configurable or/and recalibrable. The former refers to all the software parameters of the device such as its topology and processing capabilities. The latter allows to change the hardware parameters like its direction, zoom and position. The [SCN](#) should also become self-healing. Indeed, Self-organization over long operating cycles must consider the failure of links, the emergence of new nodes and the shutdown of nodes due to battery depletion or malfunction.

2 Overview

In parallel to the plausible alterations in the camera architecture, another paradigm shift is occurring in the management and coordination of systems. Commencing with centralized systems control, we migrate to a distributed management system and further to widely distributed and self-organized systems. The paradigm is obviously determined by the system and its purpose.

TABLE II.1: Comparison of conceivable network configurations.

	Centralized	Decentralized	Distributed
			
Points of Failure Maintenance	• • •	• •	•
Fault Tolerance Stability	•	• •	• • •
Scalability	•	• •	• • •
Ease of development Creation	• • •	• •	•
Evolution Diversity	•	• •	• •

2.1 Centralized Network

When we refer to a one-computer system, its peripherals and, eventually, a remote system, we use the term monolithic. The term "centralized" designates a unique centralized control port for a set of systems. Nevertheless, the two terms have often been used in the same context when dealing with embedded systems and control methodologies, albeit in a much more "centralized" way. Although alternative control paradigms have been designed, centralized control may remain the most desirable approach in certain application environments. The key strengths are its simplicity and efficiency. Indeed, the implementation is straightforward. In fact, a well-defined control process responsible for the upkeep of all subsystems should be defined. All information concerning the latter such as addresses, links, tasks is part of the setup and implementation of this process. The latter entails a reconfiguration, probably manual, each time a network parameter such as topology is modified. However, this update remains fast since there is only one machine to update. Another advantage is the easy detachment of a node from the system. All that is necessary is to remove the connection between the both. This is important especially when a node stops working for any reason. This has no direct impact on the network. On the other hand, there are disadvantages to this system: transparency, scalability and degradation. Transparency refers to more flexible resources and increased scalability based on the number of systems under control, with the possibility of dynamic configuration changes at runtime. Degradation is more worrisome, wherein in the eventuality of a fault at the central node, the overall system undergoes a sudden failure.

2.2 Decentralized Network

Multi-level architectures are a straightforward extension of the subdivision of processes into processing units and a data level. The various tiers are directly related to the application's logical structure. Indeed, the operations are logically and physically segregated on several devices, each device being adjusted to a specific set of operations.

2.3 Distributed Network

According to Tanenbaum, a suitable definition of distributed system that outlines all the relevant attributes of distributed systems would be: *A distributed system is a collection of independent computers that appear to its users as a single coherent system.*

From a high-level point of view, the nodes that make up a peer-to-peer system are all equal. These nodes represent the functions to be performed at the network level to accomplish the requested task. As a result, much of the interaction between the nodes is symmetrical: each node with its process will act both as an actuator and a servitor. Given this symmetrical behavior, a problem arises regarding the organization of these nodes in the network with this peer-to-peer architecture. In these networks, the process represents the nodes and the links symbolize the communication between these nodes. In such a situation, a process may not communicate directly with another random process, but is expected to provide prompts as requested by the tasks.

Ressources: All available resources must be conveniently accessed from each node. These include processes to localize and assess resources such as data or processing units. Meanwhile, interoperability challenges should be tackled through abstraction layers and open interfaces. Linking nodes and resources also facilitates collaboration and the exchange of useful information. In a generic situation, almost everything can be swapped between nodes. In this context, we refer to distributed entry as distributed perception in the case of Multiview systems, for example, or distributed delivery as the result of local processing transferred over the network. Sharing resources is done in a cooperative way based on well-fixed algorithm.

Transparency: Hiding the fact that its processes and resources are physically distributed over several nodes is a major purpose of a distributed system. A system is considered transparent if it can represent itself as if it were one computing system. The transparency concept encompasses various dimensions of a distributed system, like location. System transparency masks differences in both the representation of data and how resources can be gained by nodes. This degree of transparency reflects the quality of a distributed system. In addition, transparency mechanisms permit the integration of new nodes and assets that may not have been previously identified at the developing time. Transparency of the distribution is typically deemed desirable to facilitate implementation. Nevertheless, this is obviously not always a good idea. A trade-off between the degree of transparency and the performance of the system must be carefully considered.

Scalability: The scalability of the system concerns its size and manageability. Size can be the number of nodes and the ability to easily scale and add nodes and resources without degrading performance, or the geographic size that affects not only communication but also responses to needs that require collaboration. Control of the system, its links and resources are essential to achieve this scalability. Indeed, nodes do not have complete information on the global system and do not necessarily have a global clock, they make decisions based only on local information. For that, the failure of a machine does not ruin the algorithm.

Distributed systems provide a control paradigm that focuses on systems with distributed activities. They enable efficient use of resources with a high degree of fault tolerance by using tools such as asynchronous communication, distribution, replication and caching. These mechanisms tackle the issues of centralized control, i.e. transparency and scalability, but can in fact drive other concerns such as coherence issues. Indeed, a high number of nodes or insufficient resources can cause synchronization and scalability problems. Other paradigms must be used to control the system.

Distributed computing systems are commonly adopted for high-performance solutions, typically from the parallel computing domain. Nevertheless, just because distributed systems can be built does not make it a good strategy. Indeed, it is difficult to design and debug algorithms for each node considering its neighborhood and its calibration, for example. The complexity increases with the growing number of nodes. For this purpose, we distinguish two categories of networks: first, those that are self-organized and second, those that are not.

2.3.1 Pooling

In multi-camera tracking, a fundamental challenge is how to coordinate the camera's tasks. This coordination involves vast amounts of resources, such as memory and processing power of individual cameras. To do this, the primary tasks of detecting and tracking objects by a single camera must be enhanced by a coordination mechanism. These strategies vary according to the required assumptions for the camera network, data distribution and processing, and the resources required. We discuss the related work by considering the following aspects:

Time Synchronization: The information content of an image can be unuseable without appropriate reference to when the image was acquired. Since many processing tasks that involve more than one camera depend on snapshots of highly synchronized cameras, this permits to derive an accurate relationship between the cameras according to the occurrence of objects.

Calibration: In camera networks, the majority of image processing algorithms request information on the location of camera nodes as well as camera orientation. This can be achieved through a camera calibration procedure, which recovers details of the camera's intrinsic and extrinsic properties. The assessment of calibration settings generally entails knowledge of a series of characteristic point matches between camera images.

Architectures: Often, cameras are associated with other kinds of sensors in a heterogeneous network, so that cameras are only activated when an event is detected by other sensors used in the network. A further possibility is the use of cameras with different hardware architectures.

FOV: The issue of overlapping and non-overlapping fields of view is not a mere calibration problem, but rather a processing concern. Indeed, the proposed techniques do not automatically work with cameras with overlapping FOVs as well as with cameras with non-overlapping FOVs.

Processing: The nature of the target application influences the way cameras process data, which has a direct impact on the expected end-to-end quality and complexity in terms of energy consumption, processing time and IT resources required. For this purpose, the processing can be completely distributed, partially distributed or centralized. Regardless of the choice, cameras must take into consideration the results of others in order to avoid sending the same data in the case of central processing or to benefit from the results of the other cameras to continue its task in the distributed system.

Prior knowledge: An a priori knowledge of the position of each camera or of neighborhood relations may be required to distribute activity in the network. It can be preset for all cameras both manually or in a previous learning phase. This knowledge can be bypassed as it can be difficult to set it up in particular contexts, such as the presence of a large number of cameras. nevertheless, [ELYR14] have shown that the network can remain effective by implementing new approaches of network construction.

Category: All the parameters presented above can either be set in pre-arranged configuration without adjustment: static network; or can fluctuate over time according to the external or internal aspects: dynamic network. Indeed, we are referring to the dynamism of the network in terms of both the linkages between the cameras or in the dynamism of the camera itself. The first one evokes the policies of communication and data interchange among cameras that can vary: reconfiguration (related to the software part). The latter is about the straightforward adjustment of the calibration level: recalibration. This concerns factors such as positions, zooms and direction, which can be adapted either in response to the environment (as is the case with cameras positioned on mobile robots), or to provide better visibility to satisfy the needs of the processing.

Paper	[ELYR14]	[EDPA]	[OS00]	[QT08]	[MKLK10]	[LB11]	[SSRCF08]	[MEB04b]	[Det07]	[LXG09]	[EDPA]	Here
Distributed	+	-	-	+	+	-	+	-	-	-	-	+
No prior knowledge	+	-	+	+	-	+	-	-	+	-	-	+
No-calibration	+	-	+	+	-	+	-	+	-	+	-	+
No-synchronization	+	-	+	-	+	-	+	+	+	-	-	+
Non-overlapping	+	+	-	-	-	+	-	+	+	+	+	+
Static	+	+	+	-	+	+	-	+	+	+	+	+
No Best view	-	-	-	+	+	+	+	-	-	-	-	+
Low specification	-	+	-	-	-	-	-	-	-	-	+	+

+ means that it was considered and - means it was not considered

TABLE II.2: Benchmark with current state of the art

Table II.2 covers the above-mentioned properties of several platforms within the distributed camera network. These platforms operate with several levels of specifications. The further details of these selections will be discussed in the subsequent section, as well as cameras behavior.

3 Self-organization

Nature is always used as an example for processes in different fields such as technology and economics. It features many phenomena that can describe and explain different concepts such as connecting, communicating and self-organizing. This latter is revealed particularly in the animals' kingdom where behavior attracts interest of scientists who try to model the animals' behaviors and apply it in different fields. In such systems, communication has become a key point to understand and analyze behavior. What may interest us here are animals which can interact together and then accomplishing tasks together. Birds are one of the examples; their synchronous flashing is a spectacular example for self-organization, gathering in trees and then flashing in unison using some distributed mechanisms which were sources of study for a long time before being modeled by applying the coupled-oscillators theory. This latter is then utilized in different areas such as neuroscience and physics for coordination and synchronization. Fish swim in well-structured shoals with optimal motions, opening a discussion on how they can coordinate their motion to move without disturbing others motion. Ants are attractive for their capacity to find the shortest route to their destination. Hence, various algorithms such as ant colony optimization algorithm have been developed following the ants' behavior. It is a class of optimization algorithms inspired from ant colony behavior, and representing multi-agent systems as artificial ants for various systems such as internet or vehicle routing.

The major key of such a success is the collaboration between them. However, their organizational structure does not require any central coordination, allowing only "point to point" interaction, giving them the opportunity to withstanding environmental changes and influences. This high level of self-organization, showing its robustness and efficiency, addresses the question of how networked devices can be designed to reach such a level of self-organization. From a high-level point of view, the nodes that make up a peer-to-peer system are all identical. These nodes represent a set of functions to be performed at the network level to accomplish the requested task. As a result, much of the interaction between the nodes is symmetrical: each node with its process will act both as an actuator and a servitor. Given this symmetrical behavior, a problem arises regarding the organization of these nodes in the network with this peer-to-peer architecture. In these networks, the process represents the nodes and the links symbolize the communication between these nodes. In such a situation, a process may not communicate directly with another random process but is expected to provide prompts as required by the tasks.

4 Network construction

Starting from completely unknown environment, logical topology can be find out by tracking [Jav08, WL13, CMCP08] or association between pairs of camera [KHN10]. This

can be done in case of non overlapping cameras [LLP15, Jav08] or completely overlapping views [BJKD12a, SR11]. During runtime, event-based approaches [MEB04a] are used for topology inference. Indeed, spatio-temporal correlation between cameras [NRCC07, LLP15], as well as statistical dependence [DG05] can figure out useful links. This can require calibration to be performed in advance [RBSF] or not [MPC06]. The construction is not only about the pre-deployment phase, but should be kept updated during runtime, prediction-feedback relations improve data association in a unsupervised way [MBKQ⁺16] based on positively-correlated observations [MBKQ⁺16]. Links can be created locally with neighbors [MBKQ⁺16] or extracted to further ones [KDIM17].

These links can be established according to several datatypes. First, it is the basic information retrieved thanks to the wireless communication technology employed. Alternatively, this can be performed according to visual information, using a background construction or images. This does not concern us since it is impossible to do it with the qualities of the images involved. Finally, this can be achieved with the processing output of each camera as a result of the events.

4.1 Technological based method

In [Wireless Sensor Network \(WSN\)](#), a range of technologies are available to establish the spatial configuration of a network. On the hardware side, (GPS) is the most widely adopted solution. However, in addition to high material costs and high energy consumption, it fails to perform efficiently in indoor environments. For this purpose, researchers have been oriented towards Location Based Services (LBS); It is the process of determining the location of unknown sensors. This process retrieves the relative positions of the sensors with regard to the others. The term connecting a mote¹ usually evokes wireless communications and technologies such as [Wireless Fidelity \(Wi-Fi\)](#), Bluetooth or cellular. These protocols have different characteristics (rate, range, energy consumption, cost, etc.) allowing them to meet different needs. Before building the network, it is important to define several parameters that will allow a fully understand how the protocols work and to provide some answers: In what environment will the mote evolve (Urban, underground, rural, indoor or outdoor space)? How much information, data to be communicated per day? How often will this data be delivered? Is it a moving or fixed mote? The mote must be traced in real time? What type of power source supplies the mote (mains, battery or battery)? The mote needs to be geolocated? If so, what is the tolerated accuracy?

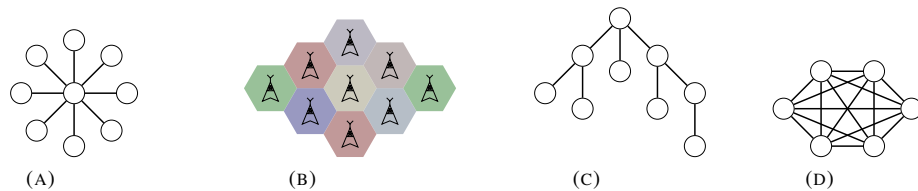


FIGURE II.1: (A) Star topology, (B) Cellular topology (C) Tree topology and (D) Mesh topology

Even in a wholly unknown environment, the technology utilized provides a preliminary overview of the network topology.

¹sensor node

Star model: The simplest model for connecting the motes together is the star model as illustrated in Fig. II.1a. Each part is connected to a central mote while remaining independent of the other mote to keep a rather low complexity and allow a fast inference during the learning process. This topology makes it easy to insert or remove nodes without impacting the rest of the network. In the meantime, all network intelligence is clustered on a single node, making it easier to manage the network. However, if the concentrator has a technical problem, then the whole network is down. This type of topology is used extensively in indoor environments (especially with Wi-Fi) or in mobile technologies (clothing, bracelets, etc. connected) where the smartphone acts as a gateway. This topology is proposed by various technologies such as Bluetooth, Near-Field Communication (NFC) and Wi-Fi.

Mesh model: The mesh models require all motes to be connected to all others. This grants denser description and therefore permit more extensive interactions between motes. The main disadvantages are that, as a result, the complexity of learning links and inference is exponential depending on the number of motes, as well as high energy consumption and the risk of collision of the exchanged packets. Figure II.1d shows a mesh topology fully connected. This topology is offered by various technologies such as Zigbee, Z-Wave, CPL and SmartMesh IP.

Broadcast model: In this type of topology, a mote transmits a message without specifying a particular receiver. This means that the message is analyzed by all the objects that have received the message correctly. This operation is suitable when several devices are to be reached without distinction, as is the case with the LoRaWAN and Sigfox protocols, for example. One of major problem of this model is the high-power consumption.

Cellular model: A cellular topology is based on the division of a territory into areas called cells. The radius of a cell can vary from a few hundred meters (urban environment) to several kilometers (rural environment). At the center of the cell, an antenna ensures the radio link between the objects and the Internet. The principle is summarized in the following figure II.1b where each cell has a different color to indicate that the antenna uses a different radio frequency band than the neighboring cells. This type of topology is the basis of mobile networks (e. g. 2G/Global System for Mobile Communications (GSM), 3G/Universal Mobile Telecommunications System (UMTS) and 4G/Long Term Evolution (LTE)).

The topologies presented above refer to the physical topology of a network which is determined by the capabilities of the network access devices, the required level of control or fault tolerance and the cost associated with the wiring or telecommunications circuits. These are the layout of the wiring, node locations and links between the nodes. The physical locations are then determined by means of the so-called anchor² or beacon reference sensors. The latter is generally deposited manually or equipped with a GPS module. For these relative positions, communication technologies employed by these sensors are generally used. The localization of each mote is a two-step process: a physical distance measurement between motes followed by an estimation of the location based on the measured distances.

These technological based Method can be summarized in four main categories:

²Anchor nodes are nodes whose location is known

- Statistical Approximation methods: consist in implementing standard approximation methods such as Semidefinite Programming, Least Square, Maximum Likelihood, Multi-Dimensional Scaling... to estimate location.
- Geometric methods: consist on estimating location by exploiting geometric parameters such as triangular information iteration or angulation.
- Path planning: consist on using an anchor node moving in a specified path in the network to find out the whole network localization.
- Mobility model: consists on relying on mobility pattern of some motes like random walk or random direction...

These technological based algorithms estimate location information based on the range-based measurement parameters such as [Received Signal Strength \(RSS\)](#), [Link quality indicator \(LQI\)](#), [Time of Arrivals \(ToA\)](#), [Time Difference of Arrivals \(TDoA\)](#) or [Angle of Arrivals \(AoA\)](#).

On the other hand, another concept of topology has been identified: logical topology. It is defined as the way in which signals act on the network, or the way in which data passes from one device to another across the network without regard to the physical interconnection of the devices. Logical topology is used associated to the routing protocols, allowing the network to not get stuck in its physical topology, and remains important to define the network performance. For instance, starting from the Mesh model, a logical topology can make it possible to prioritize a Mesh topology to define levels in order to manage the network more efficiently. In this case, we obtain the "cluster tree", Fig. II.1c. Tree models define a hierarchy of motes where each has only one parent and therefore no connection is allowed between the different branches. This structure also allows for an effective inference. However, trees require a study of the environment to be covered. This topology is widely used in home automation, where some objects cannot connect to the gateway because of distance or noise and surrounding obstacles.

Protocols such as flooding and [Ad hoc On Demand Distance Vector \(AODV\)](#) are usually used to optimize the network performance and define its logical topology. In the network context, communications depend on various parameters related the application.

4.2 Event-based network

[SSN](#) systems can be generally classified into two categories according to how they gather data: event driven or time driven. The former one refers to the category of periodic reports on environmental or habitat phenomena observed. The other is intended to capture as much data as feasible. after an relevant event has triggered sensor nodes. Obviously, the baseline scenario has a major impact on all the significant choices for the [SSN](#) protocols used.

Event-driven models give a valuable framework for the analysis of [IoT](#) systems. This section presents the event-driven analysis procedures to determine the main design criteria for [IoT](#) systems. Event analysis permits us to extract the properties of the network event population over time.

4.2.1 What is an event?

At the level of a camera, an event is a change of state resulting from an external stimulus. As shown in Figure II.2, two sorts are distinguished. External events that are detection following environmental measures and internal events that are notifications from other cameras. These notifications are generated by the cameras following external events. Either to prevent other cameras from a possible future detection, or to acknowledge the reception of an expected target.

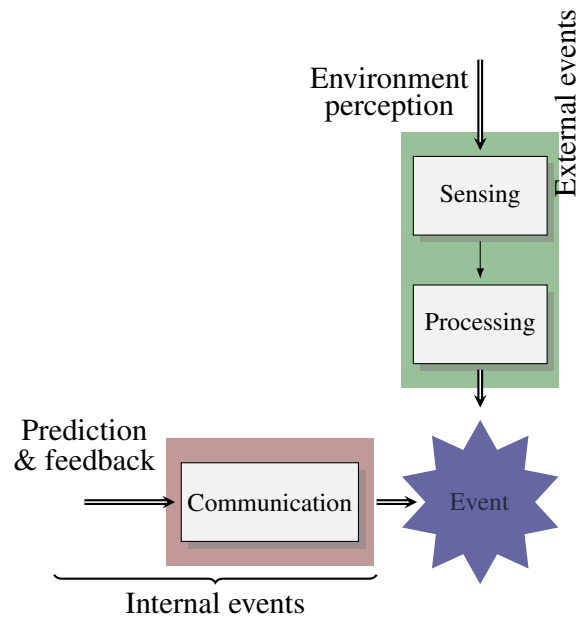


FIGURE II.2: Internal and external event in the Ant-Cam

Our model is an event-based one, each Ant-Cam starts a task depending on the event generated. Hence, we suppose that a camera can autonomously detect targets appearing in its FOV and extract a suitable description. Furthermore, in the absence of any neighbourhood information in the beginning, the camera starts by broadcasting the information in the network thanks to the communication technology. However, even if our model does not require to know the topology a priori, the broadcasting method to exchange information is inefficient in terms of communication cost. Thus, this method is used initially until building up the vision graph. This latter will be based on shared activities between two cameras detecting respectively the same target. Over time, the camera will be able to identify relevant neighbours who may share their internal events. The communication overhead will be significantly scaled down. The events are classified in two parts: internal events and external event. Internal event are the notifications of the others cameras, however the external events correspond to the detections of a target appearing in its FOV. From the perspective of an individual camera, it has 3 main tasks: The first consists of detecting each target appearing in its FOV by selecting the most suitable software resources. This task gives it the ability to perform the second task, which is re-identification based on the information shared by the other cameras. The third task concerns the exchange of information between the cameras. This can either be as a prediction: sending corresponding and appropriate information to the other cameras; or feedback to confirm re-identification.

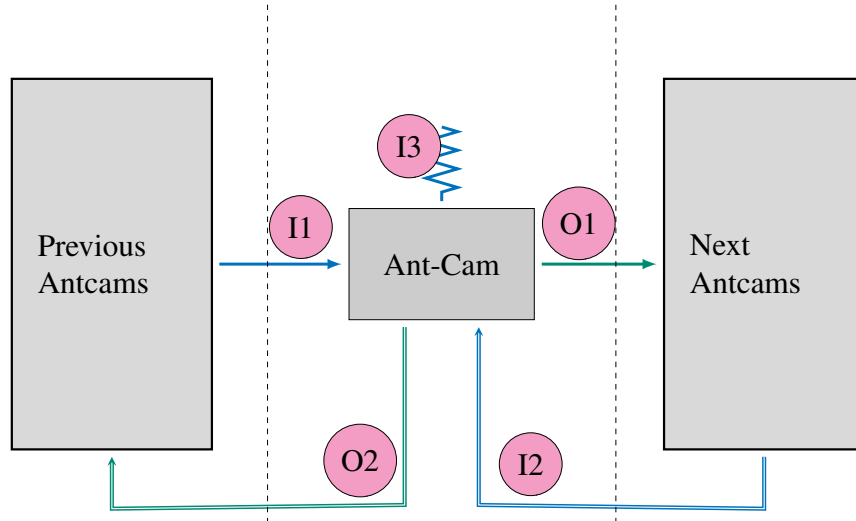


FIGURE II.3: Illustration of the local network topology.

The Fig.II.3 describes the inputs and outputs of each Ant-Cam in the network. Each camera may receive three different inputs generated either from its own activity or others' one.

4.2.1.a Inputs

A External Event

Observation: It is the only external event (I3 on Fig. II.3) generated when a target is detected by the Ant-Cam. In case when it is predated by a prediction, this observation is used to confirm or not the prediction. At this stage, the camera is required to select its own activity based on the local observations, the available resources and the prediction received. Furthermore, it should estimate the certainty of this activity to better control the impact of its activity on the overall performance of the network.

An event is composed by :

$$Event = \{Feature, time_gen, time_release\} \quad (II.1)$$

The features give the semantics of the event, it will be described in the next section. The source corresponds to the camera releasing the prediction in the network. The temporal properties of an event is estimated important to understand the event. The event is created at *time_gen* while released at *time_release*. These two parameters are used to estimate the life time of each event occurring in the network and defined by:

$$event_life_time = time_release - time_gen \quad (II.2)$$

Important for event analysis, the *event_life_time* can be a classifier for event and also gives information such as the velocity of the target detected.

B Internal Event

Prediction: It is an internal event (I1 on Fig. II.3) in the network resulting from activity in previous nodes (in terms of object trajectories), it is sent to advertise the camera and predict a future event to be observed. The camera then is ready to observe the target, after the delay estimated by the previous cameras. The previous nodes should label the target coherently to make the re-identification possible. This prediction contains different information about the target such as its visual features and previous trajectory. This will be detailed in the next section.

An event is composed by :

$$Event = \{Feature, life_time, time_sent, time_received, source\} \quad (II.3)$$

Acknowledgment: Internal event (I2 on Fig. II.3) from the following nodes which confirms a predicted event was received. When the prediction sent to the others cameras is confirmed by an observation, a feedback is sent to the camera to confirm. This acknowledgment contains a reward metric, highlighting the certitude of the re-identification. Depending on this parameter, the link strength between the 2 cameras will be reduced or on contrary will rise up. This link depicts the amount of shared target detected respectively by these 2 cameras.

An event is composed by :

$$Event = \{path, source, event_confidence\} \quad (II.4)$$

For the acknowledgment, the *event_confidence* corresponds to the reidentification accuracy. The source represents the camera generating the event, while the *path* corresponds to the target's path.

4.2.1.b Outputs

A Internal Event The outputs are generated by the camera itself after every event, and sent to the others cameras to inform about the fulfillment.

Prediction: Internal event (O1 on Fig. II.3) sent to neighbors predicting that a specific event should be received at a given time. It can be sent to more than 1 camera and wait for their answers. This prediction is generated after detecting a target and extracting a complete set of characteristics. From the perspective of the receiving camera, it is the input (I1 on Fig. II.3).

$$Event_prediction = \{path_time, source, event_confidence\} \quad (II.5)$$

Acknowledgment: Internal event (O2 on Fig. II.3) sent back to previous node which confirms a predicted event was indeed observed. When a prediction is received followed by an observation, the camera may decide whether it is the same or not. In the former case, a feedback is sent to the previous camera conveying information to help improve the prediction of future similar events.

$$Event_ack = \{event_confidence\} \quad (II.6)$$

B External Event In our case, and contrary to animals kingdom here, we do not consider any external output. We suppose that the camera does not have any action in the environment

5 Network description

5.1 Network model

Let's consider a set of cameras C . The network encompassing these cameras is defined as:

$$NM = \{C, G^{LI}, G^{LO}, G^P\}^t \quad (II.7)$$

where G^P represents the place graph, and G^{LO} and G^{LI} represent the output and input link graphs. The input graph of a camera gathers the links it has with the cameras from which it receives the predictions, and thus the targets. The output graph is for the cameras to which it provides the predictions. Both G^{LI} and G^{LO} represent the vision graph and are defined as :

$$G^L = \{C, L\}, L = \{p_{ij}\} \quad (II.8)$$

where C is a set of cameras with links L . $p_{ij} \in L$ represents a weighted connection between C_i and C_j . This connection expresses the likelihood and rates of the object's re-appearance in C_j after C_i . Thus, each camera C_i creates its neighboring camera set $Nb(C_i)$. Thus, for each camera C_i , the graphs are created independently and defined by:

$$Nb(C_i) = C_{j \in N, |p_{ij}| > 0} \quad (II.9)$$

$$G_i^L = \{Nb(C_i), L_i\} \quad (II.10)$$

where L_i is the link set of the camera i , and G_i^L correspond to the graph link of the camera i .

At each instant t , the network is defined by state s^t , which is itself defined by states s_i of each camera C_i , so the state vector representing the network is defined as:

$$\underline{s}^t = \{s_1, \dots, s_N\}^t \quad (II.11)$$

where N is the number of the cameras in the network. Each camera may observe the targets moving in the network, either all of them or a part of them. The vector of this observation is then defined as:

$$\underline{z}^{(k,t)} = \{z_1^k, \dots, z_N^k\}^t \quad (\text{II.12})$$

At each instant t , the network predicts its future state:

$$\hat{\underline{s}}^{t+1} = F^t(\underline{z}^t, G^t(\underline{s}^t, \underline{s}^{t-1}, \dots, \underline{s}^0)) \quad (\text{II.13})$$

where $G^t(\cdot)$ represents the non-markovian chain and $F^t(\cdot)$ represents the state transition function. The state vector $\hat{\underline{s}}^{t+1}$ at the sample time $t + 1$ then reverts to a dependency of the preceding states vector $\underline{s}^t, \dots, \underline{s}^0$, and the measurements conducted \underline{z}^{kt} .

TABLE II.3: Summary of used notation for the network model

Index	corresponding
NM	Network Model
\underline{z}^t	Network observation at instant t
\underline{s}^t	Network state at instant t
$\underline{z}^{(k,t)}$	observation of camera k at instant t
z_M^k	observation of target M from camera k
s_N	state of camera N
G^L	Link graph
G^{LO}	Output Link graph
G^{LI}	Input graph
G^P	Place Link graph

5.2 Camera model

Dealing with fully distributed sensors require defining well the camera as an agent independent of the whole network model. Thus, the camera reacts autonomously in response to the environment solicitation. The camera is able to develop its perceptual understanding and performance [Dep09]. The primary concern is improve its predictability for the next states by including all the information received (either prediction, positive or negative feedback) in its determinism mechanisms. These latter underlay the camera knowledge to improve the self-learning and self-regulation capability:

$$CM = \{K_n, G^L, G^P CR^i, CR^o\}^t \quad (\text{II.14})$$

where K_n corresponds to the processing resource available in the camera, defining its expertise. CR^i is the CamRank-In and CR^o is the CamRank-Out. G^L is the Link graph of camera, while G_k^P corresponds to the place graph of camera. The G^L is then defined as:

$$G^L = \{G^{LO}, G^{LI}\}^t \quad (\text{II.15})$$

where G^{LO} is the Output Link graph of the camera. It corresponds to the camera's link to the neighbors detecting the targets just before. The G_k^{LI} is the input graph of the camera, it corresponds to the camera's link with the neighbors detecting the targets immediately after, all at an instant t .

5.2.1 Expertise

It characterizes the cameras knowledge and capacities of evaluating the current situation. The declarative knowledge of each camera, which is identical for all the cameras, leads the camera to guilelessly respond to any external stimulus. It is provided here with the available treatments. The internal event is then primordial for the regulation and control of learning. The prediction-acknowledges messages allow the camera to create its own procedures and strategies to automatically perform the reidentification tasks better. The camera is then able to know how to monitor any event occurring in the network. This strategic knowledge achieves a better level of robustness. Typically, the strategies can be created based on the environment perception. Targets can be classified based on the type. This is possible when considering pedestrians, cars, dogs and so on, or even when considering all pedestrians but in different perspectives: side, back or front appearances. The path can also be a critical key while choosing the processing. This will be detailed in the next section. The history of the network helps to be aware of distracting stimulus and take decision more efficiently.

5.2.2 Links

Starting from a disordered system, the camera should be able to learn about its environment only from local interactions with its neighborhood. Here, we can consider two notions of neighborhood. First, it is the neighborhood discovered by the technology used. In other words, the cameras may receive notification using the communication capacity for the broadcasting mode. The second one is more related to the cameras which share activities with this camera. For instance, a target moving from camera C_j to camera C_i creates a link between these two cameras. Here, we focus more on the second type, and to be more precise, we distinguish input and output links. For the target moving from C_j to C_i , this link will be considered as G^{LI} for camera C_i and G^{LO} for camera C_j .

Furthermore, we choose not to work with the Markovian model. Thus, links will not be presented by matrix $N \times N$ relating each two cameras, where N is the number of cameras in the network. The links will depend on the path followed by the target before arriving. Therefore, these links will be presented by tensor $N \times N \times \dots \times N$, considering the number of cameras which constructs the chosen path.

The resulting network is wholly decentralized, where each camera C_i manages its own connections. The links are important for each camera to understand and construct the environment. Meanwhile, the links differ from one camera to another depending on the event shared between the two cameras. These links are not equally important. The strength of each link depends on the activities shared between both cameras.

In figure II.4, (a) represents the initial state where no connection is established, (b) represents the first connection established via the broadcast communication mode allowing to discover the neighbors, (c) represents the links created during the run time based on the occurrence of the events in the network. In both II.4(a) and II.4(b), red links represent

TABLE II.4: Summary of used notation for the camera model

Index	corresponding
CM	Camera Model
G_k^L	Link graph of camera K
G_k^{LO}	Output Link graph of camera K
G_k^{LI}	Input graph of camera K
G_k^P	Place Link graph of camera K

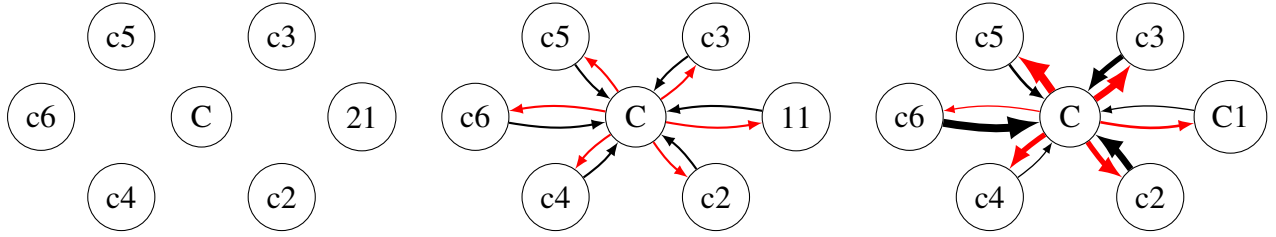


FIGURE II.4: Example of links created between cameras.

Red links represent the output links and constitute the output link graph while the black one represent the input links and constitute the input link graph. (a) represents the network at instant $t=0$ where no connection is defined. (b) represents the network at instant $t=1$ where the camera establish connections thanks to the communication allowed, links here equally important. (c) represents the network during runtime where links strength change depending on the events occurring.

the output links of camera C and build the output vision graph G_C^{LO} , while the black links represent the input links and build the input link graph G_C^{LI} .

5.2.3 The CamRank

Links created between cameras can be an indication about the importance of the camera in the network. Building graphs contributes to a holistic overview of the network and the inter relatedness of the cameras. Unfortunately, this might not be satisfactory for network analysis. Indeed, if the investigation of the required number of cameras, their arrangement and linkages is not carried out beforehand, this is worthwhile to be considered after a period of work. In fact, it is necessary to evaluate the dispersion of cameras in the environment, and refine it if appropriate.

By analogy to PageRank proposed by google to evaluate the importance of the webpages, we introduce two CamRank: CamRank-In and CamRank-Out : the first related to the target moving in to the FOV of the camera and coming from other cameras, while the latter is related to the targets leaving the FOVs and going to others one. The CamRank value of a camera corresponds to the relative frequency the target pass through that camera, assuming that the target goes on infinitely.

The rank of the camera depends on its activities in the network. The more targets a camera receives, the higher its rank-in is, following the concept of the pagerank [Fra11] used by research engine to evaluate a page. An Ant-Cam's rank will be high if the neighbors relating to it have a high rank. This ranking provides information about the importance of that camera in the network.

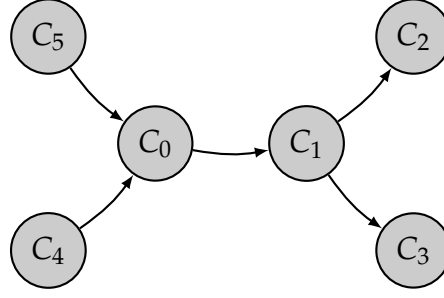


FIGURE II.5: Example of a camera network scenario

5.2.3.a CamRank-In The CamRank-In expresses the importance of the camera in term of incoming event. It is represented by the target moving from the other cameras to the one concerned. Each camera has then a notion of its own self-importance. It is evaluated following:

$$CR^I(C_k) = (1 - \alpha) + \alpha * \sum_{\substack{i=0 \\ i \neq k}}^n \frac{CR^I(C_i)}{L^I(C_i)}. \quad (\text{II.16})$$

- $CR^I(C_k)$ is the CR^I for the Ant-Cam C_k
- $CR^I(C_i)$ is the CR^I of Ant-Cams C_i which send internal event I1 to the Ant-Cam C_k .
- $L^I(C_i)$ is the number of C_i 's neighbors which link to C_k or outgoing links.
- α is a learning factor that can be set at 0.75.

Here, we take the example of node C_1 , following the graph presented in Fig II.5, C_0 is the only camera which send internal stimulus to C_1 . This latter has 2 neighbors with outgoing links, amongst them, it spreads targets out. Here C_2 and C_3 , which gives:

$$CR^I(C_1) = (1 - \alpha) + \alpha * \frac{CR^I(C_0)}{2} \quad (\text{II.17})$$

5.2.3.b CamRank-Out The CamRank-Out expresses the importance of the camera in term of outcoming event. It is represented by the target moving from the concerned camera to other cameras. It evaluated following:

$$CR^O(C_k) = (1 - \alpha) + \alpha * \sum_{\substack{i=0 \\ i \neq k}}^n \frac{CR^O(C_i)}{L^O(C_i)} \quad (\text{II.18})$$

- $CR^O(C_k)$ is the CR^O for the Ant-Cam C_k
- $CR^O(C_i)$ is the CR^O of the Ant-Cams C_i which send the event O1 to the Ant-Cam C_k
- $L^O(C_i)$ is the number of Ant-Cams which send the event O2 to C_i
- α is a learning factor that can be set at 0.75

Here, we take the example of node C_4 , following the graph presented in Fig 2, C_4 receives internal stimulus from 2 cameras C_6 and C_5 and has only one outgoing link to C_1 , which give:

$$CR^O(C_4) = (1 - \alpha) + \alpha * (CR^O(C_5) + CR^O(C_6)). \quad (\text{II.19})$$

TABLE II.5: CRO/CRI

Ant-Cam	HIGH CRI	LOW CRI
HIGH CRO	mean that the camera has received a lot of targets and released most of them, hence becoming an important intermediate Ant-Cam	mean that the camera has released a lot of targets. Here, those targets was not necessary spreading from other cameras, this camera is a start point of the network.
LOW CRO	point out that the camera has received a lot of targets without releasing them to other cameras, so it has located the destination of the target	indicate that the the camera does not receive or release a lot of targets, and it is an intermediate Ant-Cam

5.3 Target model

While re-identification is based on the prediction received, the observation and the history of the network, the camera may not be able to make the decision about re-identification. Thus, the camera can handover the re-identification responsibility with the neighbors who are most likely going to observe the target later. The camera, depending on the situation, will deal with two different approaches: top-down and bottom-up approach. The later does not take any interest on the final objectives. Once a target is detected, the camera will harvest all the available characteristics, merge and process them to extract all information may it be or not useful for achieving its goal. It is the case when no prediction anticipate the arrival of that target. Conversely, top-down approaches start with objectives, mainly received from previous camera as a prediction, and select useful information and the best strategies that fit the situation, to finally go down to the processing level and look for sensor data adapted to the goal to be achieved. This approach and their directions for use will be detailed in the next section.

In fact, an object description is in reality a parameterization process which aim to depict the best the situation with a data structure made up of a set of primitives and of relations among them. Indeed, a simple listing of visual primitives can not afford a sufficient description that can be used later to re-identify. Thus, to decide about the current situation, we choose not to limit the re-identification to an evaluation of the correspondences of the visual appearances of the targets. Thus, the decision depends on 2 main factors: (i) the target itself and its characteristics based on its observations by the previous cameras, (ii) the network behaviour and its history.

$$\underline{x} = \begin{bmatrix} z_1^{(t)} & z_2^{(t)} & \cdots & z_N^{(t)} \\ z_1^{(t-1)} & z_2^{(t-1)} & \cdots & z_N^{(t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{(0)} & z_2^{(0)} & \cdots & z_N^{(0)} \end{bmatrix}_{K \times N}$$

FIGURE II.6: Relationship between target vector \underline{x} and cameras.

$$\underline{x} = \begin{bmatrix} 0 & z_2^{(t)} & \cdots & 0 \\ 0 & 0 & \cdots & z_N^{(t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{(0)} & 0 & \cdots & 0 \end{bmatrix}_{K \times N}$$

FIGURE II.7: Relationship between target vector \underline{x} and cameras.

Targets could be defined by the observations of all the cameras N at each instant t as in figure II.6. We choose to limit the observation to one camera at an instant t II.7.

5.3.1 Target appearances

Appearance is definitely the most revealing factor when it comes to monitoring targets through time, both within and across the cameras. Unfortunately, two confusing issues complicate appearance matching for tracking and re-identifying targets in the network. First, the observations are ambiguous to the extent that different targets can be wrongly confused with each other. Inversely, a variation in lighting, position, angle or other elements may result in differences in appearance for a given target, and thus may not be re-identified as the same in two different cameras.

Thus, in re-identification tasks in camera network, human appearance can be modeled by visual appearances such as color [KHN10, Jav08, WL13] or texture [DN12]. Tracking in multi-camera has been reformulated as matching features such as color and texture between two observations in two cameras. When target is appearing in more than one camera [DN12], learning data association could be on unsupervised way, usually based on the colors also [ZPIE17, MWFF17]. More adapted metrics have been introduced such as time occurency or spatial correlation [ARG07a].

To accomplish this, we implement a set of parameters $\chi \in X$ delineating each target:

1. Visual features: e.g., target color, velocity, category
2. Temporal information: detection time, path time, event life time
3. Spatial information: path through the network

5.3.1.a Visual information They are all what may characterize the appearance of a target such as color, velocity, distance category and so on. As we are dealing with low

quality images, we develop a new model for tracking and This parameters χ_v is detailed in chapter V.

5.3.1.b Temporal information Each Ant-Cam can be seen as a neuron of a spiking neural network, where spikes are event-like signals travelling between neurons at specific times. The relative timing of the spikes in different neurons can be used for learning, as in the spike-timing-dependent-plasticity model [GKvHW96]. Moreover, the number of spikes from the same type, called firing rate, can give information about how the network is dynamic. By analogy, we introduce the temporal parameter χ_t , containing the information about the detection time. It corresponds to the time needed to go from one camera to another.

5.3.1.c Spatial information The path followed by the target before arriving to the concerned camera may be a key to identify it and predict its destination. For instance, considering smart road traffic surveillance, targets follow specific rules, and thus taking into account their path will increase the performance of the network, which can be used later either to improve the roads or traffic light configuration.

Most of the existing SCNs opt for the Markov assumption for prediction. Fig. II.5 illustrates the limits of assuming that the future state of the system only depends on the present one. If the most frequent paths through the network are for instance $C_5 > C_0 > C_1 > C_2$ and $C_4 > C_0 > C_1 > C_3$, the predicted camera after C_1 (C_2 or C_3) cannot simply be deduced from the current camera or even when considering the previous camera (C_0 in both cases). Considering the sequences of previous cameras will however disambiguate the trajectories. Furthermore, the system may form different graphs depending on the events characteristics, for example if qualifying pedestrians or cars. In addition, using the path can be interesting to keep the system working even in case of the dysfunction of a camera. These information received in each prediction and extracted after each observation are used to make sure that two observed targets correspond to the same genuine target or event.

6 Conclusion

The main objective is to introduce a networking model capable of giving the camera full autonomy and acting as a self-interested agent in the network. This is especially significant when dealing with a large scale network in which it is impractical to individually configure the cameras. Thus, the camera reacts autonomously in response to the environment solicitation with different levels of granularities. The camera is able to develop its perceptual aliasing and performance according to the targets detected in the network. The target, constrained to have very limited visual information, is defined by its transit time between cameras, and its path followed in the network. The aim is therefore to render this information complementary in order to better meet the network's needs in terms of re-identification.

CHAPTER III

Distributed tracking in smart camera network

1 Overview

Most of the cameras used in SCN context has a sufficiently high resolution, and require the input images to be noisy-free. Thus, they propose algorithms dedicated to the use case proposed. However, their performance degrades drastically when we decrease the resolution specially when we have tiny images contaminated by noise. The main goal of this work is to learn and exploit the regularities in the correlated activity of cameras. The system should be able to build precise predictions based on two components: a model (of the spatio-temporal behavior of expected/previous targets) and observations (related to the current target). To identify and track a target through a network, the observations inform about the path followed by the target. The system should pick out as much information as possible to better predict the next state (hence not being limited to the Markov assumption). The model in turn depends on the network structure and the assumptions about the trajectories or targets. Thereby, it should be built from observations, indicating which cameras should observe the target at future times, allowing to provide a multi-camera behavior analysis [MMM⁺14]. The survey [VBC13] tackles different issues and aspects of re-identification challenges.

Our model includes two principal parts: The first one is the cognitive knowledge which permits prediction decision using the the detected information and that received from other cameras. The second one is the regulation of that cognition using the feedback received from the other cameras after each event. This cognition control allows the network to be self-monitoring and then to have its own self-regulation process.

Accordingly, the camera is not just a member of the network learning the parameters that may influence its performance, but it extends its capacity to perform things based on the interaction with the other cameras. Consequently, this model enables the cameras to do some tasks more automatically and then to go further with the control of knowledge. This knowledge can be consolidated by the other cameras presented in the network. [DCRc] proposes a Network Consistent Re-identification framework which improve the camera pairwise re-identification performance between camera pairs, this performance has been evaluated in [MMF⁺16]. Keeping exchanging signatures with the neighborhood until finding the valid matches and then improve next reidentifications is proposed in [RiMP]. Moreover, targets can hardly be viewed in a similar pose by two cameras, simple comparison between the two views can not lead to accurate results. [MM14] proposes to find the optimal correspondence between images patches using a local matching technique.

Before taking the decision, the camera should analyze all the data of the environment in order to provide the best depiction of the current situation. However, the interpretation of an event and the derived information depends on the camera and its resources, the level of detail received by previous camera and the history of the network.

Cameras are interacting among themselves, in pairs or in groups, to accomplish specific tasks such as monitoring, enhancing environmental coverage or optimizing resources. The purpose of networks is to empower the camera to be autonomous and act as a self-interested agent in the network. Furthermore, the cameras are not restricted to being identical with the common algorithm, configuration and calibration. Availability of heterogeneous specifications can greatly increase performance. Hence, the camera opts for the most appropriate configuration at runtime to better tailor itself to any given situation, all without requiring any central coordination.

A rich set of mathematical tools for decision making can be applied to model the interactions between cameras [MBKQ⁺16]. The associated approaches can be labeled as socio-economic [ELYR14], game-theoretic [SSRCF08], stochastic [YSN09] or optimization oriented (e.g., particle swarm in [MADR12]). Added to that, they can rely on other forms of meta-heuristics such as the genetic algorithms [Dep09]. The survey in [PEK⁺16] offered a deeper understanding of the existing models used for the SCNs. Some models would increase flexibility even more by exploiting self-reconfigurable [JCK⁺14] and self-calibrating cameras to maximize their performances. The resulting network might adjust camera parameters such as position [RCH11], orientation, pan-tilt-zoom [MADR12], or select where processing should occur. Indeed, a reduced (dynamical) set of cameras with dedicated processing might be sufficient and could better achieve tracking [ELYR14], coverage [MADR12], path planning [RCH11] or target detection under various visibility conditions and satisfy given quality requirements [RGM⁺16]. Yet, it would offer heterogeneity in the network and more efficiency while the cameras could learn how to be different [LEC⁺13], [LEC⁺]. Most of SCNs deal with a known environment [EDPA], [MKLK10] and [LXG09]. Thus, all the necessary information such as topology, position and orientation are known. In such a case, a learning phase preceded the real time test presenting miscellaneous cases. However, as it complicated to have all the necessary information upstream, some works deal with unknown environment requires taking other point into account, such as communication and network modeling [BKMQB16] and [MBKQ⁺16]. In addition, it is considered impossible to cover up all the possible case upstream. Following this idea, it is estimated useless to have a previous learning before the SCN starts performing. The re-identification tasks are learned in real time. The survey [VBC13] tackles different issues and aspects of re-identification challenges. The survey in [PEK⁺16] offered a deeper understanding of the existing models used for the SCNs. In case of unknown environment, the communication is performed due to a spatio-temporel correlation [NRCC07]. The events scheduling is not only useful to re-identification but also to reconstruct the network.

Three scenarios can be distinguished according to the degree of cooperation among the cameras:

- **Data collection:** The role of a smart camera may be restricted to only local processing of the environmental measurement performed. Indeed, the target detection or tracking can be carried out at the camera level, which merely forwards its results to neighboring cameras or to a centralized data processing and collection unit. In this scenario, cameras can be deemed to be collaborative but independent, in the meaning that they can rely on the processing performance of other cameras to pursue a target for example, but such results do not influence the camera's software or hardware configuration.
- **Re-Configuration:** Self-reconfiguration behavior, in its turn, is more about learning its software parameters, and handle the calibration. The camera are modelled, figuratively, by ants [BKMQB16], auctioneer [ELYR14], gene [IBMCO9] and even gamer [MZA⁺13]. While all of the representation aims to mitigate the problem of dependency, each one is challenging with different aspect.
- **Re-Calibration:** The camera calibration operation is the modelization of the process of forming the images. It aims to find the relationship between the spatial coordinates of each point of the space with the associated point in the image taken

by the camera. The calibration uses two kind of parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system and fixed by the position, orientation and zoom. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates and fixed by the iris and focus. In order to better perform tasks, cameras may change those parameters depending on different factors: (i) Variable environmental conditions prod the camera to change to self-calibrate? such as illumination condition, or the obstacles which can be static or dynamic, (ii) the others cameras presented in the networks, whether because it receives a sub-task from another while a task decomposition from another due to its limited performance or to continue a task started in another camera such as tracking in the best condition. (iii) the performance which should be evaluated by the camera before starting a task, such as the accuracy, the timeliness and the energy needed to perform that, in case of overcharging, task can be split up in many sub-task and associated to other cameras in the network. This concept that helps complex systems to adapt themselves autonomically to their environment. The idea is to let the system itself find the appropriate parameters. The goal of such complex task is to improve the environment coverage [MADR12], to move up by the image quality [PMF10] and to optimize the resources consumption [KGZH10].

In this Section, a brief overview of the models used is presented. This overview is not meant to be exhaustive, but to highlight the main ideas applied in SCNs and how the agent of a network can be modeled. Each camera acts as an independent agent in the network. its purpose may be a simple information report, collaboration with other cameras to perform monitoring tasks or a more in-depth analysis for ontology.

1.1 Tracking

Having a network of cameras in an uncontrolled environment in order to track multiple people is a very challenging task due to the non-rigid nature of the human body, where the appearance of the person changes with body movements, with a wide degree of variation in their pose and orientation and with quickly changing lighting conditions in uncontrolled environments. Tracking a single individual is not difficult as tracking multiple people moving around in the scene, which usually contains static occludes (for example: furniture for an indoor environment, and lamp posts, trees, etc. for an outdoor environment). A person may sometimes be occluded by another person(s) or object(s) in the scene in a camera view. When tracking a target person from a particular view is not good enough due to either type of occlusions, the camera will collect more information about them by inquiring whether other candidate cameras have a better observation. The other cameras in the network respond with their local information in the form of no or lesser occlusion. The camera analyzes the local information from the other cameras in order to have a global view of target people. Finally, the camera chooses the cameras with the best views about the target person, and it asks these cameras for assistance in tracking of target until the occlusion in its view is less severe or over.

Tracking people in low-resolution constraints was studied in [END⁺14, NDE⁺14]. The users' locations and mobility statistics were obtained from a robust people tracker based on recursive maximum likelihood principles in a lab setup of 5 low-resolution visual cameras. The multi-camera tracking system of [BFTF11] first utilized the concept of probabilistic occupancy mapping to find the persons' positions. Then the known positions of

each person would be linked using the k-shortest path algorithm. In [YGA12] the authors first tracked people in each camera separately and then integrated these results using a Bayesian approach and relying on the principles of epipolar geometry. [BJKD12b] first detected people utilizing the detector of [DT05] and [FGMR10] in each camera view and would afterwards track the detection with a particle filter within each view. The same people in different views were associated by triangulation, using a greedy matching approach. Their technique relied on color features. The technique in [GJNC⁺14] could work in grayscale sequences as it was based on optimizing the likelihood of foreground/background segmentation images given a hypothesized position in a 3D space. The actual data fusion involved a Kalman filter. The method was able to track multiple persons in real-time, but because of the Kalman filter, it would lose people when they suddenly change direction.

TABLE III.1: Summarize of the reconfiguration methods used for tracking, C refers to centralized processing and D to distributed processing

Algorithm used	P	calibration	Processing
Game-theoretic	[LB11]	Camera selection	C
Failure containment	[KGZH10]	Camera selection	C
POMDP	[NHW ⁺ 14]	PTZ	C
Greedy best-first search	[QT11]	Camera selection & PTZ	C
Production rules	[SQ11]	Camera selection & PTZ	C
Task assignment	[HFK ⁺ 09]	Position	D
Negotiation	[MCC ⁺ 10]	Position	D
Socio-economic approach	[ELYR14]	Camera Selection	D
Optimization	[SJAR11]	Position & direction	D
Game-theoretic	[SSRCF08, DSM ⁺ 12]	Camera selection & PTZ	D

A real time visual surveillance system for detecting and tracking multiple people and monitoring their activities in an outdoor environment has been proposed in [HHD00]. The authors employed a combination of shape analysis and tracking to locate persons and their body parts such as the head, the hands, the feet and the torso, in order to build models of persons' appearance, so that they could be tracked through interactions such as occlusion. In another camera system [SM], the authors fused the output of a number of detection and tracking algorithms to achieve robust tracking of people in an indoor environment. Depth information was utilized as well for person tracking. In [LSA11], the authors presented a novel framework which integrated an a-priori person detector with an on-line learnt person detector and a Multi-Hypothesis Tracker (MHT), so as to estimate the motion state of multiple people in 3D using three vertically mounted Kinect sensors. The framework integrated two detectors and a tracker that involved a track interpretation feedback to control learning. Their approach did not rely on learning a background model or a ground plane assumption. Santos and Morimoto [SM11] put forward a framework to track a group of people using sparse uncalibrated cameras. They integrated all available information of all cameras before any detection decision based on the homography constraint that did not rely on the single view segmentation of the subjects or previous tracking information. [LLZ⁺15] suggested a three-stage cascade structure framework using RGB-D videos. The first stage transformed the RGB-D data to point an ensemble of image from plan-view perspective. Next, an unsupervised detector was used in the second stage to retrieve positions, where these positions were further refined by a classifier

utilizing two new features: a histogram of height difference and joint histogram of color and height. The 3D trajectories were generated in the last stage.

Deep learning is one of the promising techniques for accurate people tracking in outdoor environments. In [FXWG10], the authors trained a Convolutional Neural Network (CNN) to estimate the location and scale of a person given their previous location. Their CNN learnt spatial and temporal features, where multiple pathways were introduced to better fuse local and global information. The authors in [XLCH16] applied a deep-learning-based pedestrian classifier which outperformed handcrafted features and traditional classifiers like the SVM. They also introduced a probabilistic tracking method combining the deep learning classifier with a probabilistic motion model, which tracked people by greedily maximizing the posterior probability. [YLXG09] employed a CNN for multiple human tracking. Their CNN incorporated and combined multiple cues based on color models, shape matching, and bags of local features. In [LLP15], the authors used a three-layer CNN model to distinguish the target object from its surrounding background. Their CNN relied on a tracking-by-detection strategy where the CNN was updated in an online manner. A family of deep neural network classifiers using on-line AdaBoost for person tracking was proposed in [ZXZZ14].

1.2 Ontology

Cameras tasks are used to have high level information to understand the environment. An indoor or outdoor environment equipped with a network of camera devices and actuators is referred to an “Intelligent environment”. Understanding activity patterns of people in an intelligent ecosystem can be used to optimize the monitoring and control tasks, as well as productivity and the comfort of supervisor. The sensory signal outputs from a monitoring system can be used to recognize several activity patterns such as “arriving to work late”, “leaving the office early”, “working non-stop”, and so on. By learning and detecting long-term activity patterns, the environment becomes aware of each person’s preferences in order to increase work productivity and decrease stress. For example, a person who works continuously for longer hours than usual without a break, the environment can recommend him to have a coffee break. In another situation, when the environment notices a change in a person’s behavior by arriving and leaving the environment late, the environment can notify them how such a change in their habit can make them less socially interactive. Based on observations and learned models, the environment compares how the observations deviate from previous activity patterns, in order to suggest healthier habits.

Humans perform activities based on habits, so inferring patterns that describe the past and present activities is important in order to define future activities as well. Accordingly, an environment can proactively activate and deactivate some devices based on learnt patterns (e.g. switching off the computer automatically when a person leaves their office, or the light when a person leaves their home). Apart from automating actions or devices, patterns can also be used to understand a person’s activity behavior and act in accordance with it (e.g. issuing meeting reminders). Besides, making the environment more efficient in terms of saving energy (e.g. switching off the lights when a person has gone to lunch or a meeting) or increasing safety (e.g. locking door when a person is not present) helps to improve work productivity and encourages people to manage stress.

Chen et al. [CA11, CABAA11, CUWA11] studied the problem of discovering the social interactions in indoor environments using a network of high-resolution cameras and

RFID. The head poses and the locations of people were tracked using Chamfer matching. Afterwards, a classifier was used to estimate the head orientation based on the location, relative distance and head orientation of people. Added to that, a probabilistic model was used to infer the use of space by individuals and their interactive behavioral patterns. Moreover, probabilistic graphical models, such as the HMM, the dynamic Bayesian network, and the Conditional Random Fields (CRFs), have been used to model the activity transition sequence for activity recognition purposes. In [OGH04, OH05, OHG02], the authors compared the Layered HMMs (LHMMs) and the dynamic Bayesian networks for identifying indoor or outdoor activities from multi-modal sensors such as video, audio and user's interaction with the computer. Hence, the dynamic Bayesian networks are only included at higher levels of LHMMs. [WNS06] proposed a multi-level HMM framework for multi-person activity recognition (meeting, paperwork, discussion, etc) with simultaneous tracking of users in the room using audio and video cues. In [EDPA16], the authors presented an approach to detect the habitual absence patterns of users in the indoor environment, so as to offer a better use of the indoor space with others.

On the other hand, many unsupervised approaches have been proposed to handle the problem where activity labels are not available. In [CAA11], a system consisting of a visual processing and a learning module were proposed to discover accurate patterns that represented the user's frequent behaviors in an indoor or outdoor environment by associating the semantic locations of the user to activities. [HMJ⁺09] put forward the idea that global structural information of human activities could be encoded using a subset of their local event sequences. They regarded discovering structure patterns of activities as a feature selection process. [SPYZ11] studied the daily activities of people from videos, by automatically learning event grammar under the information projection and minimum description length principles in a coherent probabilistic framework, without manual supervision about what events would happen and when they would happen.

1.3 Aggregation

Data aggregation methods rely on optimizing the overall coverage of the scene being monitored, e. g. by maximizing the deployment of sensors that monitor relevant areas or by minimizing non-observed areas of the environment. The problem was initially conceived in the field of computational geometry as the "art gallery problem" [Fra]. However, the latter can not be directly implemented in real distributed camera networks, since it ignores several issues such as camera directivity or range. Since then, several studies evoking different geometric and topological models have been analysed. Those models are analyzed in the survey [MC].

1.4 application use-case

Defining various architectures and specifications were always endorsed by the steering wheel to integrate connectivity into everything that can affect human life. This can directly influence human life by providing a smarter home, an intelligent transportation system or health care services. But also indirectly, by affecting the environment more intelligently through the energy security and control system, and more wisely industries by managing manufacturing and buildings. The figure B.1 illustrates a panoramic overview of the IoT application fields.

TABLE III.2: Summarize of the reconfiguration methods used for coverage, C refers to centralized processing and D to distributed processing

Algorithm used	P	Recalibration	Processing
Particle swarm optimization	[KC13, MADR12]	PTZ	C
Expectation-maximization	[PMF10]	PTZ	C
Simulated annealing	[MD04, MD08]	Position & PTZ	C
Genetic algorithm	[IBMC09]	Position & PTZ	C
Greedy min-set cover	[ARG07b]	Position & orientation	C
Coverage path planning	[QKWS ⁺ 10]	Position	C
Integer linear programming and greedy	[AA06]	Position & orientation	C& D
Greedy search algorithm	[SWJT10, Wah10]	Position	D
Max-sum task assignment	[DRX ⁺ 12]	Position	D
Lawn mower search pattern	[KYR14]	Position	D

Before taking the decision, the camera should analyze all the data of the environment in order to provide the best depiction of the current situation. However, the interpretation of an event and the derived information depends on the camera and its resources, the level of detail received by previous camera and the history of the network. Contrariwise, cameras are not depend from each other. For example, If no prediction is received, camera can proceed wholly its tasks considering that it is the starting point of the object, or may leave the decision to the next cameras.

2 Pre-event Connectivity

Before receiving the target, the camera may receive its prediction in case if it has already detected by some other cameras and not just entering in the network. Thus, the camera has an idea about what it is waiting for. The prediction received contains different information describing the object, and describing its state. At this level, two approaches are adopted depending on the situation.

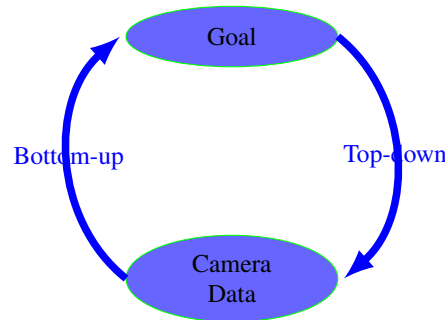


FIGURE III.1: Top-down and bottom-up

2.1 Bottom-up approach

It is used in case no prediction precede the appearance of a target. In this case, the camera will extract all the information corresponding to that target. Here, the camera will consider itself as the starting point of the target, and advertise the network about that.

2.2 Top-down approach

The top-down approach is used when a prediction is received before the target. In this case, the camera has an idea about what it is waiting for, and will apply the corresponding algorithm to estimate the similarity between them. Therefore, the camera takes its own previous detection in consideration in order to alleviate any bias effect due to the prediction received.

Alternating the two approaches gives the Ant-Cam more flexibility and independence between cameras accomplishing specific tasks. This flexibility is insured using a predefined set of rules. The latter are therefore the same for all the Ant-Cams. These rules are focused on the images interpretation. Indeed, image quality makes it difficult to have a robust algorithm which can face all cases, and it can easily disrupt the re-identification. Thus, we choose to classify the most important cases to allow the camera using its resources in the best way and then maximizing its chances to get the good answer. Depending on the case, the task will be chosen by the camera after analyzing the situation.

3 Event Connectivity

This step can be seen as the governor step and divided in two parts: concluding the pre-event phase, if it exists, by determining if the target is the same or not, and trigger the post-event phase by predicting the next state of the object and spreading it in the network. Two information categories are considered here. The first is the prediction characteristics broadcasted in the network after each detection. However, the second one, is the deeper information extracted from the target such as visual metrics. These latter are stored in the camera until the others need them.

However, in most of case, decision can be difficult to take for different reasons such as noise or low quality features extracted. In this case, cameras may find that it is more rational to not take the decision and ask the next ones to do it. Machine who does know what it doesn't know is the new concept of learning, pairwise decision can then help to get better performance in the network.

In this step, the target is detected by the camera. Two main situations can be present: For the first one, there is no prediction preceding this step. In this case, we do not consider the re-identification steps. The camera is considered the starting point of the target and all the necessary information will be extracted and sent to the next cameras. The second case is when the detection joins a prediction. Here, two tasks are presented: (i) The camera has to find out whether the received prediction matches with the detection, and (ii) the camera has to predict the future position of the target.

The figure [III.2](#) illustrates the presence of internal and external events in a network of four cameras. A detection at instant t can be preceded by a prediction at instant $t-1$. The

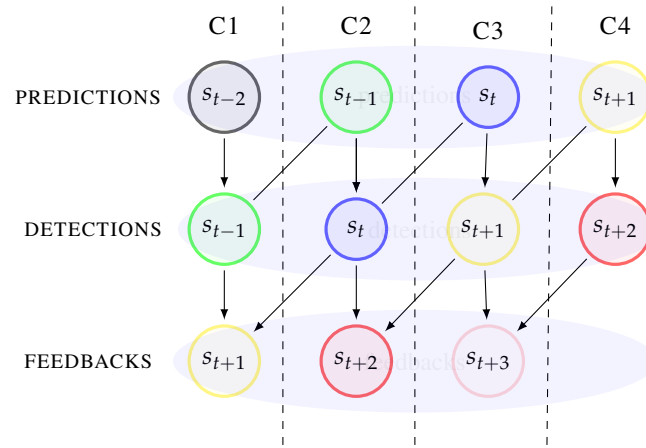


FIGURE III.2: Network event evolution

processing of this detection then generates a feedback to the source camera at time $t+1$, and receives in turn a feedback at time $t+2$.

At each detection, the camera extracts all the information related to the target (visual, temporal and spatial). If there is a prediction that preceded this detection, the re-identification is performed according to the correspondence between each computed parameter and the one received, which is called confidence.

3.1 Target confidences

3.1.1 visual confidence

When considering perception systems, visual stimuli trigger the sensory organs continuously. Thus, perceptual and analytic phase starts. It is an abstraction of important subset of the scene and order them in a certain way. The last step, is the interpretation of these primitives. The latter depend not only on the scene's observation, but also on the whole environment. While humans are naturally equipped with such a system, the goal of the research is to replicate this in the artificial systems. Indeed, for smart cameras, most of time we steer the re-identification to a simple comparison to the primitives extracted by 2 cameras. Here, we want to go one step further and learn the cameras how to match completely different primitives.

Thus, visual feature vectors extracted after each observation z_i , characterize the object in a certain way which can be exploited by the cameras. They prototype the current object and contain the visual mark of the target such as the color, the velocity and the category. As the target may be seen from different side we take into account those parameters separately. The main goal here is to find a matching between visual features even when they are different. In other words, cameras have to learn the difference between their observations. Figure III.3 presents examples of situations when a target is seen differently in each camera. While C1 observes the target as a circle, C2 gets a rectangle and C3 a piece of both.

The visual confidence ϕ_v is estimated. It represents the similarity between the observation $z_i^t : \chi_v(o)_i$ and the prediction received from the other camera j representing $z_j^{t-dt} : \chi_v(o)_j'$. This confidence is defined by:

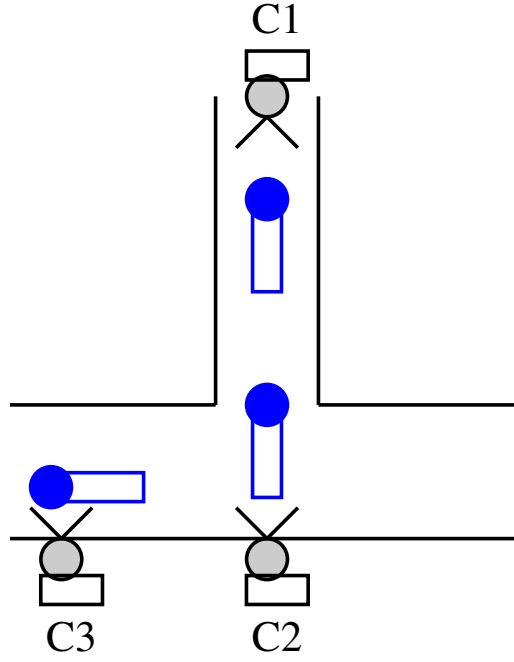


FIGURE III.3: Example of a target scenario with different camera views

$$\phi_v = f_v(\chi_v(o)'_j, \chi_v(o)_i) \quad (\text{III.1})$$

With values of the similarity function f_v ranging from 0 to 1 (perfect match). The function is here kept abstract as one such function may be introduced for each target type. The exact computations may also depend on the processing used in each camera.

3.1.2 Temporal confidence

Temporal confidence $\hat{\chi}_t(o)$ is estimated by the Ant-Cam to predict how long the target needs to catch the next Ant-Cam and sent via the prediction, whereas $\chi_t(o)$ is the time really needed for the target to move from one Ant-Cam to another extracted from the detection. Here, the external and internal inputs events (I1 and I3 on Fig II.3) is similarly valuable, as it measures the precision of the temporal prediction. Hence, we introduce function f_t and the associated result ϕ_t :

$$\phi_t = f_t(\hat{\chi}_t(o), \chi_t(o)) \quad (\text{III.2})$$

where f_t is a decreasing function of the difference between the estimated time ($\chi_t(o)$) and observation time ($\chi_t(o)'$). If the difference is too high, the observed event should be considered as something unrelated to the prediction. The value produced (in $[0, 1]$) allows estimating the temporal validity of the prediction in a graded fashion.

This factor is evaluated using a Gaussian probability distribution $\mathcal{N}_{ij}(\chi_t(o)_{ij}, \sigma_{ij}^2)$ centred at the delay time expectation $\chi_t(o)_{ij}$ between cameras C_i and C_j . For instance, if a target x_1 is detected from node C_i and then from node C_j , the delay time $\chi_t(o)'_{ij}$ can be measured. If this target is periodically detected between the nodes, an expected delay time $\chi_t(o)_{ij}$ can be also estimated. Accordingly, in order to evaluate whether nodes are temporally correlated or not, the measured $\chi_t(o)'_{ij}$ is compared to the $\chi_t(o)_{ij}$ through the

time delay distribution. The closer $\chi_t(o)_{ij}$ to the model average, the higher correlation probability results. By iteratively updating the $\chi_t(o)_{ij}$ mean and giving the likelihood probability of correlation, the nodes strengthen their coordination in case of a correlated event. Since the probability distributions \mathcal{N}_{ij} are computed on-line, the rewards are continuously updated to meet environmental variations and to detect abnormal events that might occur.

3.1.3 Spatial confidence

Most of the existing SCNs commit to the Markov assumption for prediction. Figure III.4 illustrates the limits of assuming that the future state of the system only depends on the present one. If the most frequent paths through the network are for instance $path_i = C5, C0, C1, C2$ and $path_i = C4, C0, C1, C3$, the predicted camera after C_1 (C_2 or C_3) cannot simply be deduced from the current camera or even when considering the previous camera (C_0 in both cases). Considering the sequences of previous cameras will on the contrary disambiguate the trajectories. In addition, the system may form different graphs depending on the events characteristics, for example if qualifying pedestrians or cars. In addition, using the path can be interesting to keep the system working even in case of dysfunction of a camera.

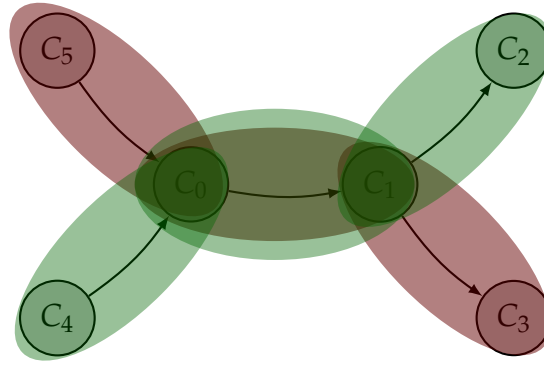


FIGURE III.4: Example of unpredictable sequence under Markov assumption

We put forward a function f_s and associated value ϕ_s depending on the previous cameras that detected the target:

$$\phi_s = f_s(\chi_s(o)', \chi_s(o)) \quad (\text{III.3})$$

3.2 SVT parameters

The spatial, visual and temporal (SVT) parameters presented above contribute to the construction of each target model and then the decision of re-identification. Here, two objectives are set: precision and reliability. The first is about the detection, it depends only on the camera's capacity of processing. The latter is more about the certainty about this re-identification and estimate with the confidence parameters explained above. This reliability characterizes the interaction between the cameras. The links between them cannot be defined just with the number of shared tasks, but with the certainty of those tasks.

$$\phi = g(\phi_s, \phi_t, \phi_v) \quad (\text{III.4})$$

The ϕ parameter represent the probability that the observed target correspond to the prediction received from previous cameras.

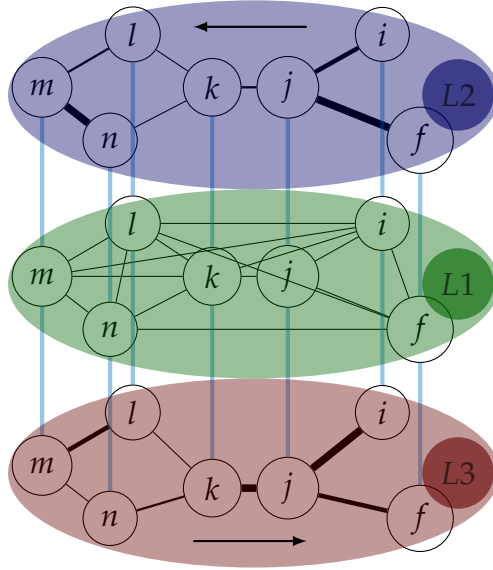


FIGURE III.5: Instance of network: Connectivity graph based on events (L2,L3) and communication graph allowed by technology used for it (Wifi, LoRa..)

As it is illustrated in figure III.2, the re-identification task depends on the previous state. Hence, state s_t depends on the external stimulus generated by a detection d and on the previous states s_{t-1} and s_{t-2} . The involvement of these factors contributes to estimate how much the detected object corresponds to the received prediction. The re-identification of target o by camera C_i is estimated by $p(o_i^t|Z^t)$ and depends on different parameters in the network, as presented in figure III.6. Table III.3 lists the different parameters to be taken into account in this estimation.

This estimation is evaluated by:

$$p(o_i^t|Z^t) = \frac{p(z^t|o_i^t, Z^{t-dt})p(o_i^t|Z^{t-dt})}{p(z^t|Z^{t-dt})} \quad (\text{III.5})$$

where:

- $p(z^t|o_i^t, Z^{t-dt})$ is the observation likelihood. It is the probability of getting the camera input if we know that the object is present in front of it at instant t . We assume that the current and past observations are independent of each other and conditioned on the location. In other words, the appearance of the object of the perspective of the camera should be independent from the path followed by the object. Consequently, it becomes equivalent to $p(z^t|o_i^t)$ and evaluated relying on the similarity defined in the feature domain ϕ_v defined by Eq. IV.1.
- $p(o_i^t|Z^{t-dt})$ is the prior belief about the location of the object in the network, conditioned by all previous observations on all cameras. This term can be evaluated from $p(o_j^{t-dt}|Z^{t-dt})$, where dt is the time of the previous event in the camera network. For this, we need to consider the set of observed/memorized paths in the camera network, with associated delays. We evaluate this term using the transition matrix [BKMQB16] with the adjusted probabilities having the temporal similarity ϕ_t and the spatial information ϕ_s .

- In addition, $p(z^t|Z^{t-dt})$ is a normalizing term. The latter is independent of the camera at all times. Subsequently, we obtain:

$$p(o_i^t|Z^t) \propto p(z^t|o_i^t)p(o_i^t|Z^{t-dt}) \quad (\text{III.6})$$

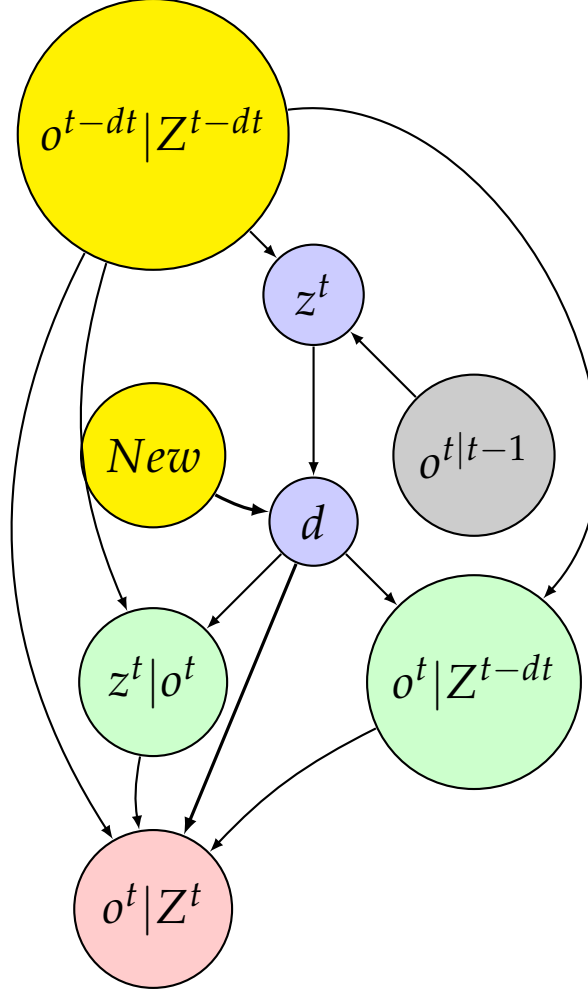


FIGURE III.6: Network representing the relation between the re-identification estimation and the prediction received.

As will be explained in chapter 5, the estimated visual prediction is computed on the basis of the current and previous observations. This estimation is considered not only for re-identification, but also for adjusting the link between two cameras.

4 Post-event connectivity

In this step, we reevaluate the transitions between the cameras depending on the feedback received, or the absence of that feedback. Thus, these would help the camera to learn more about the network and thus identify the cameras with whom it should broadcast the information in the future. Thus, we reduce the communication cost for the next event. In fact, this may lead to a second representation of the network. While the first is about the connection allowed by the communication technology for the camera situated in its range(layer L1 in figure III.5), the second is more specific about the network behavior. It, the second, highlights only the transition due to common interest to a specific events.

TABLE III.3: Different parameters used for re-identification estimation.

Index	corresponding
$o^{t-1} Z^{t-dt}$	prediction received for t-dt from previous camera detection the target re-identification estimation
$o^t Z^t$	
$o^t t-dt$	The target exists
z^t	Observation at t
d	detection
<i>New</i>	The information are new
$z^t o^t$	observation likelihood
$o^t Z^{t-dt}$	prior belief about the location of the target in the network

The second one is in reality a representation of the real world behavior (L2 and L3 in figure III.5). However, if we want to keep our system open to the environment changes, we should keep the 2 representations. A change of habits, such as open a new issue may change the behaviors and thus the system should be ready to take this change into account. Thus, the connection weight p_{ij} between two cameras i and j is updated following:

$$p'_{ij} = h(p_{ij}, \phi) \quad (\text{III.7})$$

4.1 Graph update

The validation of a target re-identification in a camera i triggers an update of connection weights defining the vision graph G_i^L . This adjustment process depend on the internal events (feedback and prediction) and external event (observation).

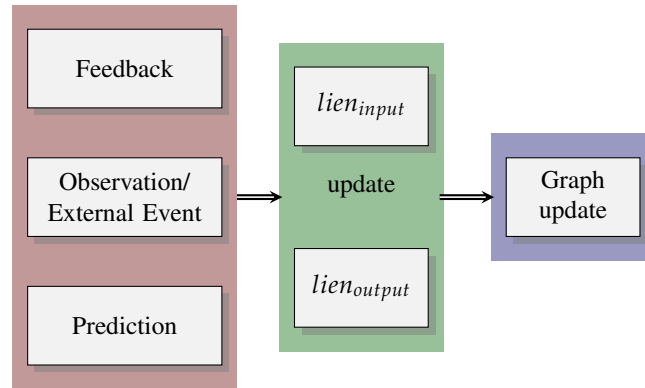


FIGURE III.7: Camera update after each external event

The input link graph is updated based on the previous camera observation/prediction, while output link graph is updated based on the next camera observation/feedback.

4.2 Database association

The network is represented in the Fig.III.8, the black nodes represent the processing available in the camera. The different layers represent the cameras which detect the target in the environment. Thus the input is each target moving in the network, and the output of each layer is its own representation of the target in the network. Depending on the results

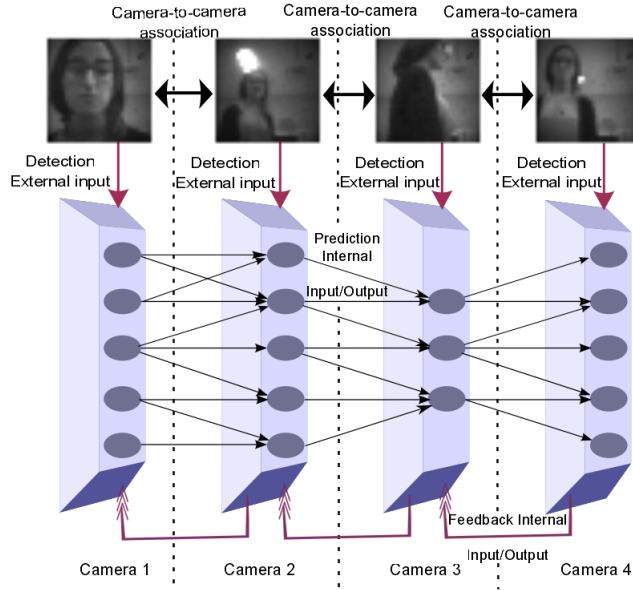


FIGURE III.8: Network architecture: each column represent a camera participating in the tracking task. Black node correspond to the processing available in the camera.

offered by the previous camera, the camera will analyze the situation and choose which processing to apply to better fit the situation. As mentioned before, the result of a same processing may differ depending on the situation. Following the same idea, a target moving around the cameras may have different appearances. The goal here is to learn these transformations between the cameras to optimize re-identification results.

5 Conclusion

The intended network is fully distributed and aim to accomplish the tracking task without any supervision. The choice of very low-resolution sensors decreases privacy issues, costs, computing requirements and energy consumption. Each camera uses the stimulation-response combination to perform specific tasks: external stimuli that are detection following environmental measures and internal stimuli that are notifications from other cameras after external stimuli to predict or feedback. External and internal stimuli can help the camera to develop an in-depth understanding of its environment and build its own vision graphic. These online learning of associations can lead to high-performance tracking from a global system point of view, making it possible to create a spatial-visual-temporal correlation between cameras and targets. The correlation enhances the accuracy of its prediction and feedback in terms of onsite processing or communication.

CHAPTER IV

Evaluation

1 Smart Camera Simulation Tool

We choose simulations for replicability and controllability (e.g. between learning sequences, and between slightly different scenarios). A simulation tool has been developed to implement the required network models. It allows virtual cameras instances with scalable topologies and flexible configurations. Such a tool illustrates how the network coordinates autonomously the interactions between cameras within the network as a function of a particular event models. A network of intelligent virtual cameras is set up to evaluate the proposed techniques with a number of facilities: simplified communication, abstraction of computer vision processing, a variety of target movement models, ease of generating various test scenarios.

Unlike real camera networks, debugging is simpler and problems and anomalies are rapidly detected. Furthermore, the use of discrete time steps make it possible to extract the targets' position and cameras' states.

1.1 Simulation environment

An environment is generated to simulate scenarios resembling an indoor environment, which can be a residential building, an office or even factory corridors. The grey squares simulate obstacles that can be walls or barriers. The cameras are placed in different locations with no direct links between them.

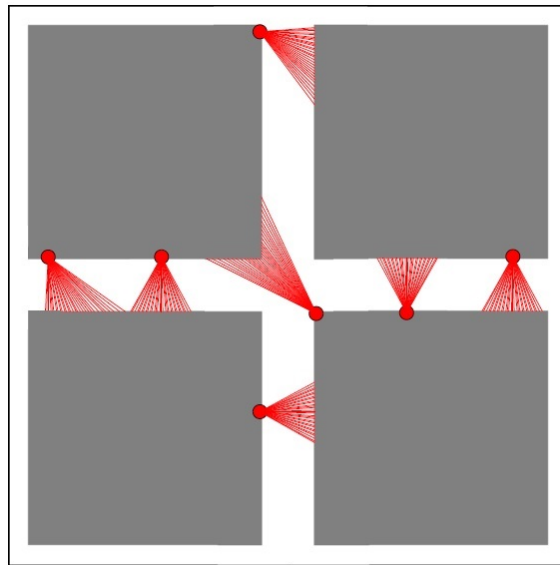


FIGURE IV.1: Illustration of tested scenarios. Each camera is represented by a red circle with its FOV indicated by red lines.

1.1.1 Cameras

Cameras are represented by red circles and their FOVs are indicated by red lines. Location, FOVs, viewing angles and directions are set up based on the scenario intended. The resolution is fixed at 30*30 pixels [IV.3](#). This image turns to target color as soon as an object is within the FOV of the concerned camera, and the visibility of the target is estimated at each time step.

```
# List of cameras in environment
rw = 7
cams = list(
  Antcam$new(c(2,2),c(-0.2,-0.2),fov=40),
  Antcam$new(c(-9,-2),c(0,1)),
  Antcam$new(c(env$xr[2]/2,2),c(0,-1)),
  Antcam$new(c(-16,-2),c(0.5,1)),
  Antcam$new(c(16,-2),c(0,1)),
  Antcam$new(c(-2,9),c(1,0)),
  Antcam$new(c(-2,-18),c(0.7,0.3))
)
```

FIGURE IV.2: Example of cameras declaration with the simulator.

In this simulation, we utilized 7 cameras set up as shown in the figure IV.2. Figure IV.5 shows a screen shot of initial tested scenario with 7 cameras.

```
#-----
#' Constructor
#' @param xy      camera location
#' @param dxy     camera orientation
#' @param fov     field of view (in degrees)
#-----
initialize = function(xy,dxy,fov=60,res=30) {
  # Copy parameters as fields
  self$xy = xy
  self$dxy = dxy
  self$fov = fov
  self$res = res
  # Generate a new ID
  self$id = paste0(Antcam$ID_PREFIX,Antcam$ID)
  Antcam$ID = Antcam$ID+1 # increase ID
},
```

FIGURE IV.3: Example of a camera definition function with the simulator.

1.1.2 Objets

Targets are identified and distinguished in the network at any time by a globally unique ID. Each target is predefined by: color, size, shape, starting point, end point, direction, speed and path. During the run-time, each target move through predefined waypoints. It is a straight line with a predefined speed.

```
# List of objects in environment
objs = list(

  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(0+66,2+66,3+66)),s,'blue'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(4+66,6+66,7+66)),s,'red'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(2+66,4+66,5+66)),s,'black'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(6+66,8+66,9+66)),s,'green'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(10+66,12+66,13+66)),s,'yellow'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(15+66,17+66,18+66)),s,'blue'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(26+66,28+66,29+66)),s,'red'),
  TgtBall$new(data.table(x=c(-25,0,0),y=c(0,0,25),t=c(31+66,66+66,34+66)),s,'yellow'),

  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(1,3,4)),s,'green'),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(7,9,10)),s,'red'),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(9,11,12)),s,env$colors$pink),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(13,15,16)),s,'blue'),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(14,16,17)),s,env$colors$purple),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(17,18,19)),s,'green'),
  TgtCube$new(data.table(x=c(-25,0,25),y=c(0,0,0),t=c(23,24,25)),s,'blue').
```

FIGURE IV.4: Example of targets declaration with the simulator.

1.2 Scenarios

Here, we have chosen to evaluate the model with a real-world scenario describing the traffic rules in a building. Thus, we want to highlight how our model act in case of changes

```

#' @param xyt      reference points
#' @param rad      ball radius
#' @param col      ball color
#-----
initialize = function(xyt,rad=1,col='blue') {
  # Copy parameters as fields
  self$xyt = xyt
  self$rad = rad
  self$col = col
  # Generate a new ID
  self$id = paste0(TgtBall$ID_PREFIX,TgtBall$ID)
  TgtBall$ID = TgtBall$ID+1 # increase ID
},

```

FIGURE IV.5: Example of a target definition function with the simulator.

in rules like closing doors or opening new ones. At the end, we set back the rules and we evaluate the system. The system is defined by a set of cameras able to generate events based on the visual environment change in their own visibility range and communicate with the others.

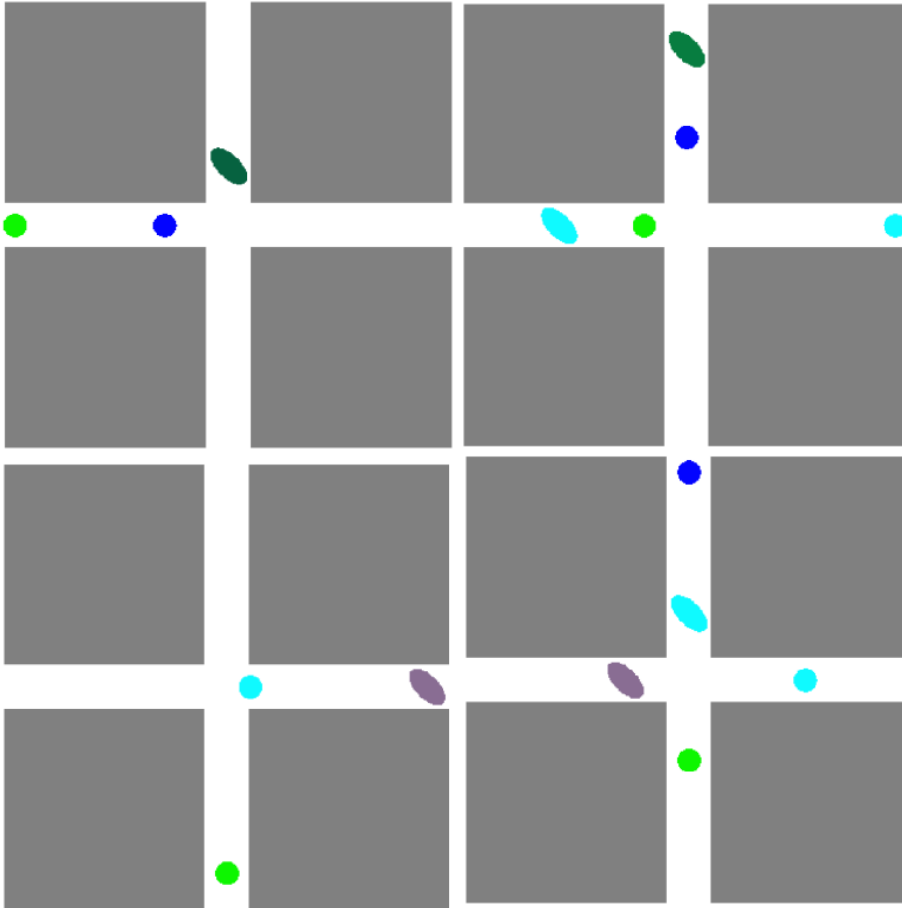


FIGURE IV.6: Example of the network environment at different instants with the simulator.

1.3 Startup parameters

All parameters should be set up before running the scenario. An overview of all possible parameters is presented:

In order to highlight the different situations and the influence of the variations of each metrics used in our models, we choose to mention multiple cases. Consequently, we

Element	Parameters
Environment	height, length, x-position, y-position, color
Background	height, length, x-position, y-position, color
Camera	$Camera_{ID}$
Camera	Camera location with the simulation environment
Camera	viewing angle (the width of the FOV)
Camera	resolution of the camera
Camera	direction of the camera
Target	$Target_{ID}$
Target	starting position of the target.
Target	end position of the target.
Target	speed of the target
Target	direction of the target
Target	color of the target
Target	shape of the target
Event	time step
Vision graph	graph with equiprobable links

TABLE IV.1: Overview of the used parameters.

can follow the system changes in various conditions. The created targets differ in color, size, shape and velocity, so we take these parameters in consideration while extracting the visual features. The time step is fixed to 0.01s for the whole system. The spatial feature is extracted from the history of each camera shared with the others after each detection and following the target. The initially unbiased state transition policy imposes to initialize all the transition probabilities to $1/\text{number of cameras}$. These probabilities evolve according to the trajectories generated by our simulator. An example of events are shown in figure IV.7.

2 Network Evaluation

2.1 Evaluation of self organization and network construction

One of the hypothesis in the network model is the unknown environment. Thus, here we evaluate the network construction and the self-organization of the cameras. The communication graph is also evaluated based on the information exchanged between the cameras. In the first iteration, the cameras broadcast the information to the whole network, while after 30 iterations, each camera share the information only with the cameras who are in its set of neighbors.

The graph represents the links created between the cameras based on the shared events. The graph highlights how the cameras learn about the environment when they receive internal and external events. Although they start randomly, they can reach a stable state reflecting the perceived regularities in the environment. Moreover, we choose to change the environment distribution in order evaluate how our model can face any change in a real environment such as a condemned door or an internal change. Our system converges to the new situation slowly.

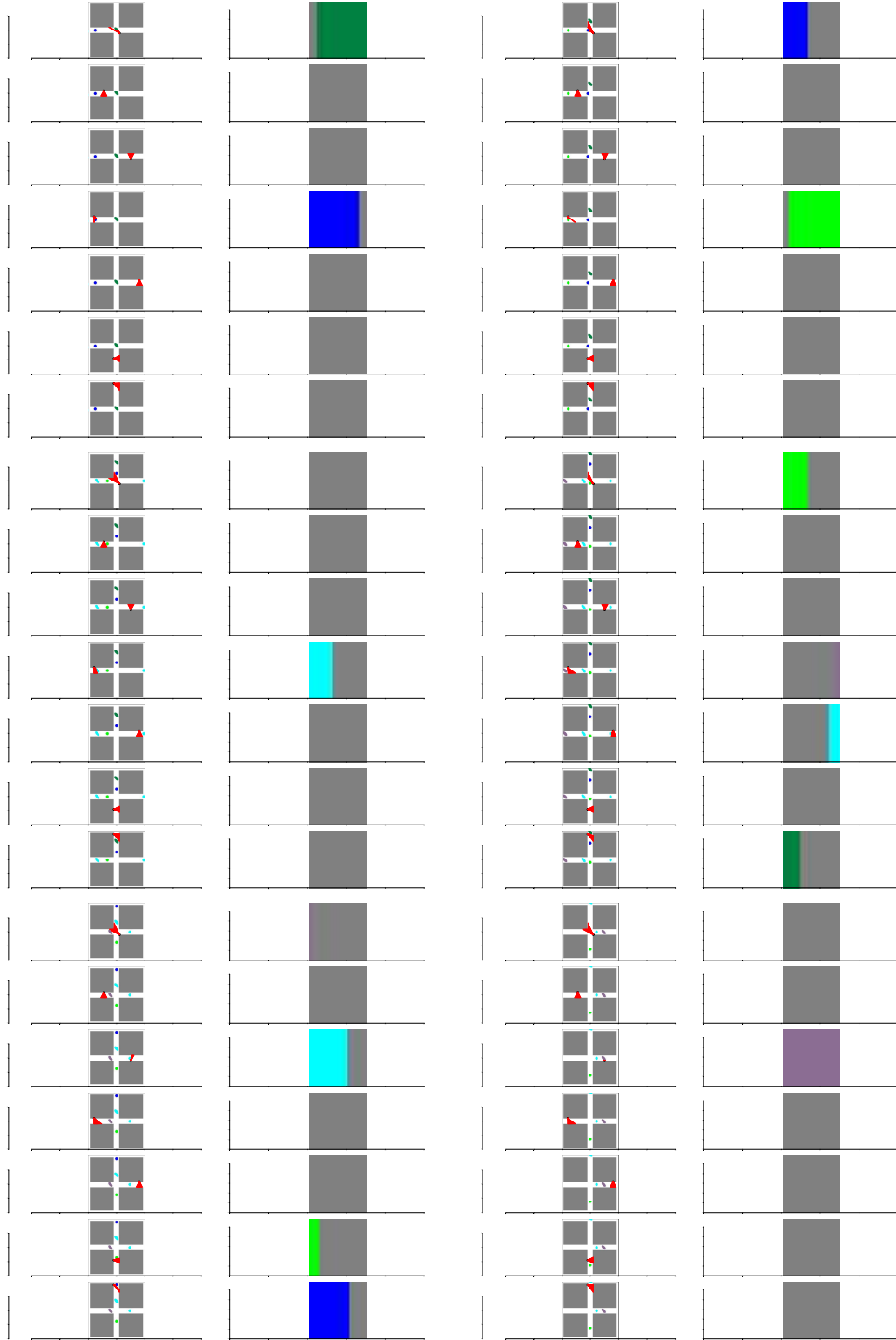


FIGURE IV.7: Examples of the visual information detected by the 7 cameras at different instants.

In figures IV.8, V.13 and IV.10, we show the approach based on pheromones to construct the vision graph during execution time. The graph is presented in a three-state. Initialisation state where no adjunctive characteristics are provided. As the first targets advance

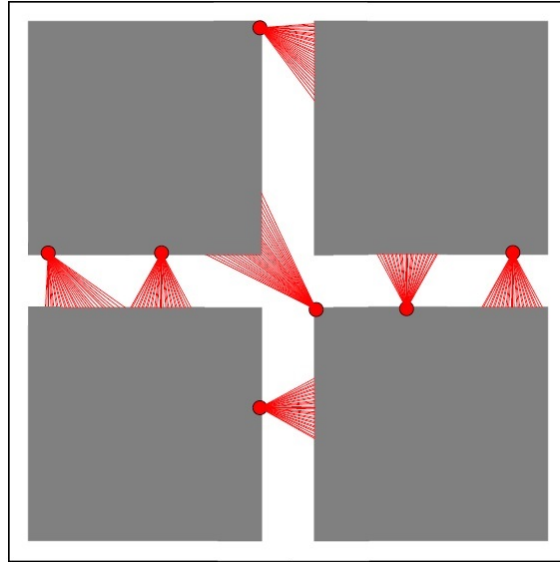


FIGURE IV.8: Example of the network construction, markov assumption is consider here($n=0$), 0 iteration.

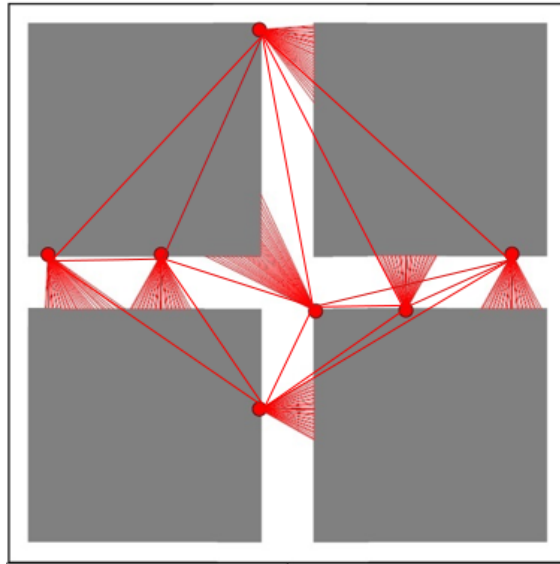


FIGURE IV.9: Example of the network construction, markov assumption is consider here($n=0$), 1 iteration.

among the cameras, links (indicated by red lines) are generated, These first ones are identical since the first interactions are also equally important. Over time, pheromones not used between cameras evaporate the connections and decrease the strength of the link. Others, obviously, are increasing until reaching stable state. The probability shown is indeed in proportion to the confidence of re-identification. A stabilized state demonstrates that re-identification is more accurate and hence enforces a wholly stable state.

2.2 Evaluation of Distributed tracking

The system is composed of a set of nodes able to accomplish two tasks : (i) detecting and excerpting the necessary information and (ii) dispatching the pheromones to other cameras. This evaluation evinces how the system can coordinate the interactions of nodes after each event pattern. This simulation exhibits the coordination and collaboration of the

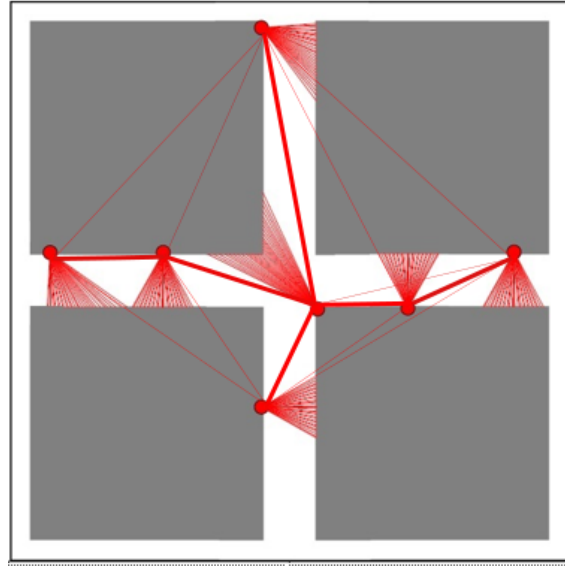


FIGURE IV.10: Example of the network construction, markov assumption is consider here($n=0$), 20 iterations

nodes to reach a stable state despite the environment difficulties or some system failures such as losses of messages, detection and re-identification problems.

2.2.1 Evaluation of pair-wise tracking

We suppose that the target is moving with a constant speed, and also we suppose that we have the same category of targets

- We suppose that the target is moving with a constant speed.
- If the camera notes that the prediction does not correspond to the detected target, it will suppose that it is a new target appearing in the network.
- If a camera does not receive any feedback, it will consider that it is located in the final destination of the target.

For each generated event, the 3 confidence visual, temporal and spatial are evaluated using the following equations:

$$\phi_v = f_v(\chi_v(o)', \chi_v(o)) = \exp \frac{-|\chi_v(o)' - \chi_v(o)|}{\sigma^2} \quad (\text{IV.1})$$

$$\phi_t = f_t(\chi_t(o)', \chi_t(o)) = \exp \frac{-|\chi_t(o)' - \chi_t(o)|}{\sigma^2} \quad (\text{IV.2})$$

$$\phi_s = f_s(\chi_s(o)', \chi_s(o)) = \exp \frac{-|\chi_s(o)' - \chi_s(o)|}{\sigma^2} \quad (\text{IV.3})$$

where σ is the standard deviation of ϕ_t , ϕ_v and ϕ_s .

Once the re-identification is validated between camera i and k , the link strength p_{ik} is reevaluated using the following equation:

$$p'_{ik|s} = p_{ik|s} + \sum_{\substack{i=0 \\ i \neq k}}^n (p_{ij|s} - p'_{ij|s}) \quad (\text{IV.4})$$

n is the number of cameras in the network.

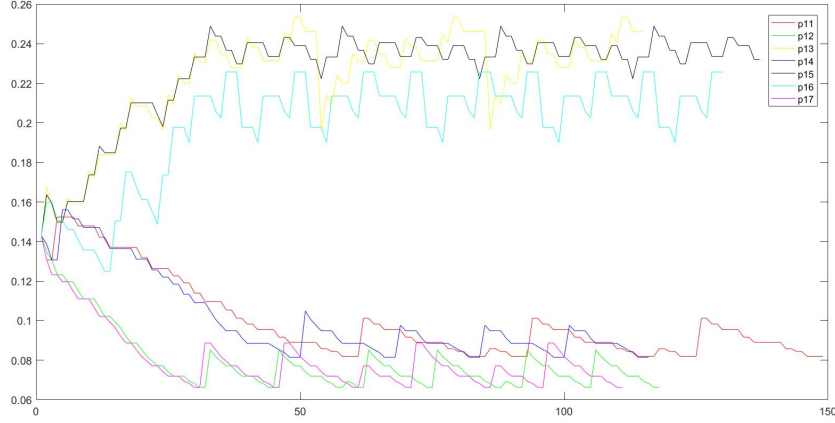


FIGURE IV.11: probabilities of the link between camera 1 and j.

In addition to the scenario described above, a larger network was simulated to take into account the non-Markovian model hypothesis.

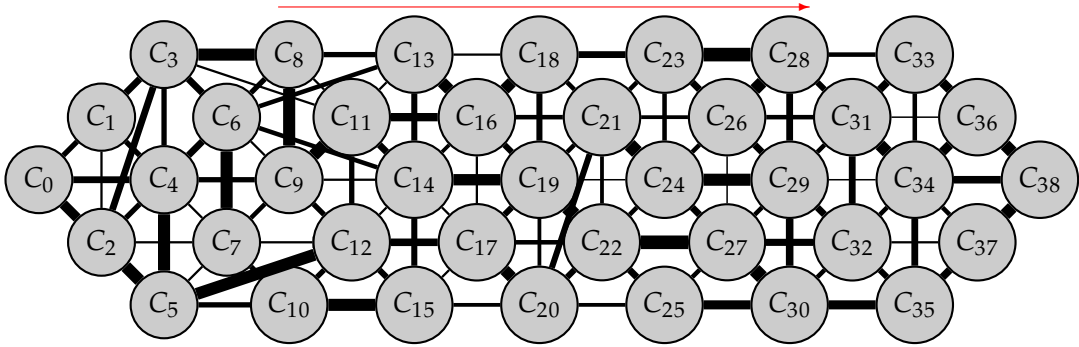


FIGURE IV.12: Simulation of 39 nodes

A large network is implemented to show how a real network used with a lot of cameras can be implemented in the real world (Fig. IV.12). The different lines between the nodes represent the possible transitions, and their thickness varies depending on the importance of interactions between cameras. Nevertheless, the interaction between two cameras depends on the path followed before coming there. Hence, we choose to represent just one of them as it is not possible to present all of them. In addition, those lines do not represent the physical communication. In other words, two nodes can communicate due to the technology used, but will not be necessarily a transitional link in case they are not a target destination.

The number of cameras to be taken into account in the construction of the path is fixed at three. In this case, we have a system represented by the tensor $(39 \times 39 \times 39 \times 39 \times 39 \times 1)$. Accordingly, we show the results on the node C_{21} . Figures IV.15a, IV.15b and IV.15c

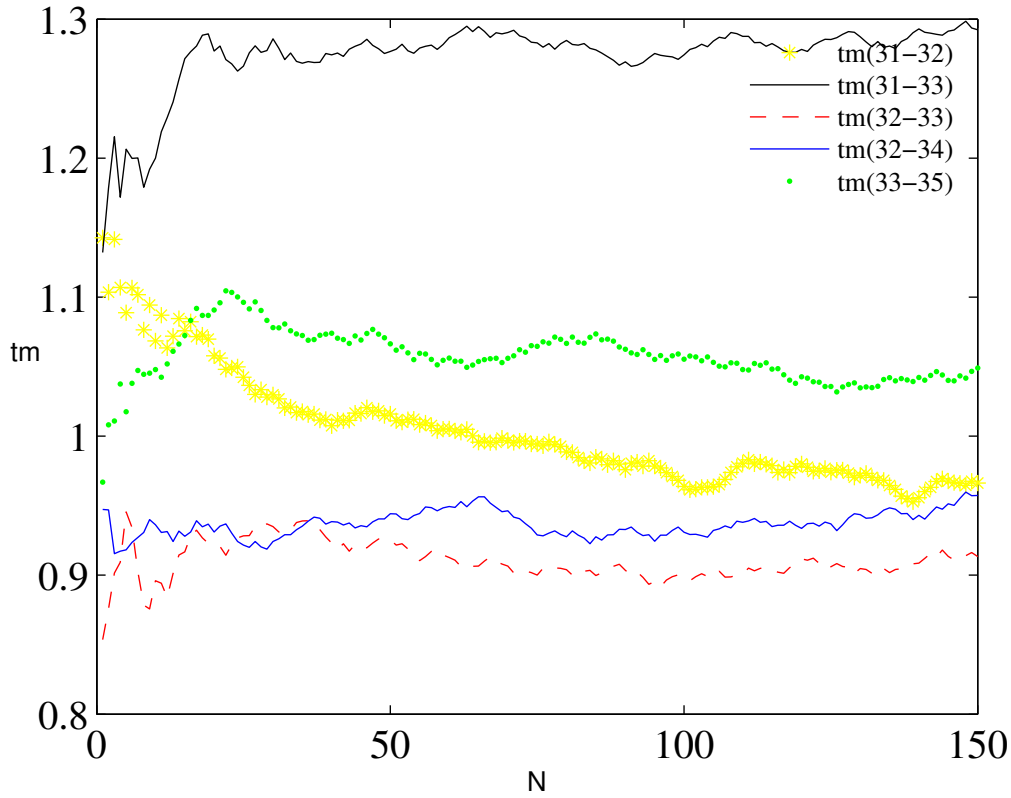
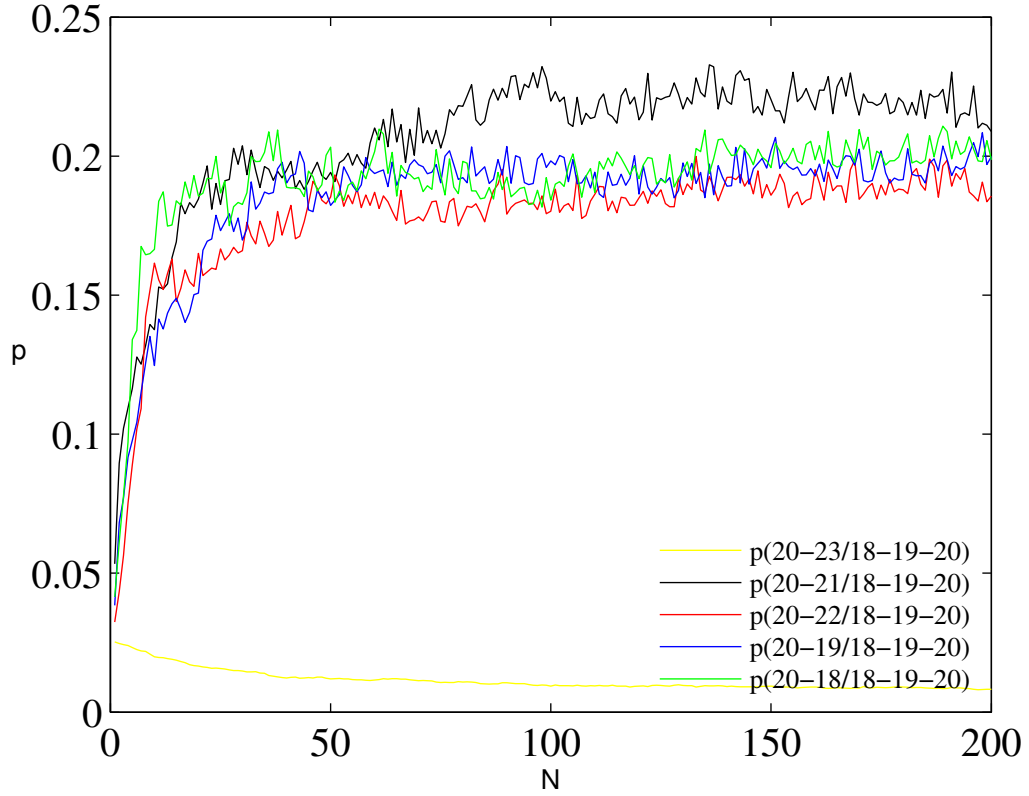


FIGURE IV.13: Evolution of the estimated time

present the transition probabilities between C_{21} and C_{22} ; it is estimated based on the model presented and the values randomly generated by the simulator. We clearly notice that the probability depends on the followed path. Although we can not present all possible paths, the latter figures highlight how much it is important to consider a non-Markovian model; it yields much more precision compared to the Markovian model. In spite of the fact that the trajectories are randomly generated, the target movements are still deterministic. Typically, we see that the probabilities converge to a stable state after 100 events (represented by N in figure IV.15). However, it varies depending on the past followed path. Typically, with Markovian models, the probability of moving from C_{21} to C_{22} is the average of the different values presented here and is not reliable information about a network. As expected, the network takes much more time to reach a stable state. With a lot of path possibilities and an important choice of destinations, the probability converges slowly.

Besides the evolution of probabilities, we estimate the variation of the temporal pheromones τ_m to be interesting (Fig. IV.13), as it represents the delay estimated to reach a camera. Considering that we have the same type of targets moving at a constant speed, the system is able to extract stable delay expectations. Unlike the probability, the evolution of temporal information does not depend on the followed path and is the same between two cameras. This temporal information can give an idea about the distance between the cameras. This distance is relative and not physical. In this respect, we can deduce that the distance between 31 and 32 is one and a half more than the distance between 32 and 33.

FIGURE IV.14: $p_{i|path}$ for the same path($\Sigma = 1$)

2.3 Evaluation of CamRank-In/Out

The CR_I/CR_O are initialized to random values. Accordingly, we consider that the velocity is constant. The CR_I/CR_O are depicted in the table 2.3. As mentioned in the previous section, ranking adds information about how the network is dynamic. C_4 has the highest number of events, hence the highest CRI. The latter is higher than the C_3 's rank. We can deduct that not all the event entering to the FOV of C_4 are coming from C_3 , so our Ant-Cam do not cover well the space where they are located here. Cameras should be added by there.

[!h]	Ant-Cam	CRI	CRO
	0	0.25	1.0293
	1	0.34375	0.519531
	2	0.34375	0.519531
	3	0.765625	0.71875
	4	0.824219	0.625
	5	0.559082	0.25
	6	0.559082	0.25

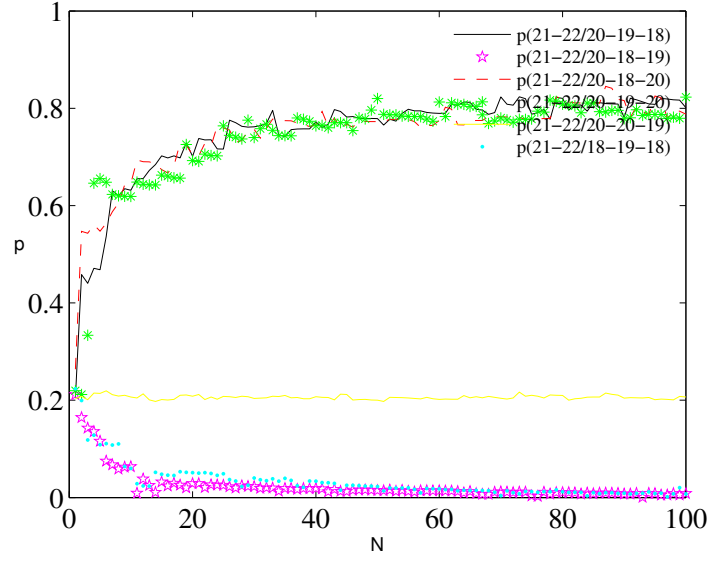
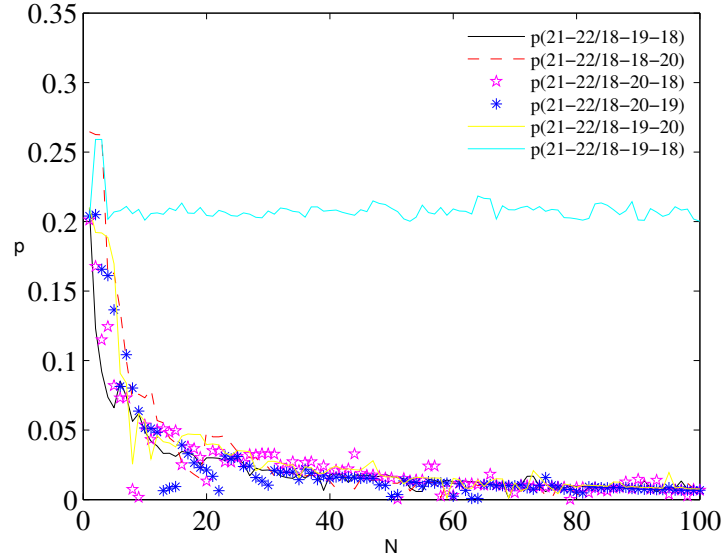
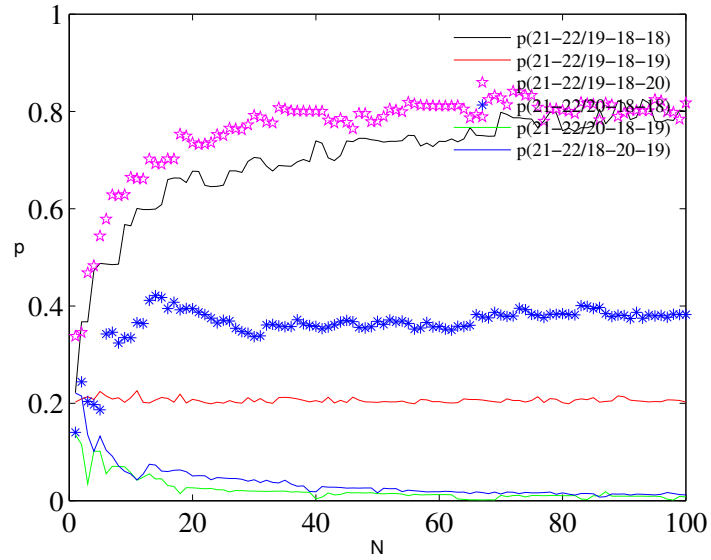
TABLE IV.2: Ranks of the Ant-Cams

In Fig. IV.17, we evaluate the number of times a path is used, of course, there are not all the possible path, a target can stop in any camera.

Consequently, we find out that the path 3 is the mostly used one, which approve the variation of the probability in Fig. ??, where the probability p_{02} is much greater than p_{01} .

3 Conclusion

In this chapter, we conducted an initial evaluation of our network model. The simulation offers flexibility and ease of testing that allows us to visualize the influence of different parameters on the operation of the network, which cannot be easily replicated and controlled in a real environment. It has been able to show that the network can reach a state of stability even when starting with a completely unknown environment. Then, this shows that the different paths presented can change the links between cameras, maximizing tracking performance or minimizing communication overheads.

(A) p_{21-22} via 20(B) p_{21-22} via 18(C) p_{21-22} via 19FIGURE IV.15: Probability in node C_{21}

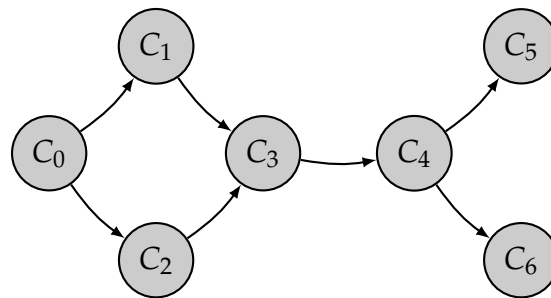


FIGURE IV.16: Example of evaluation scenario

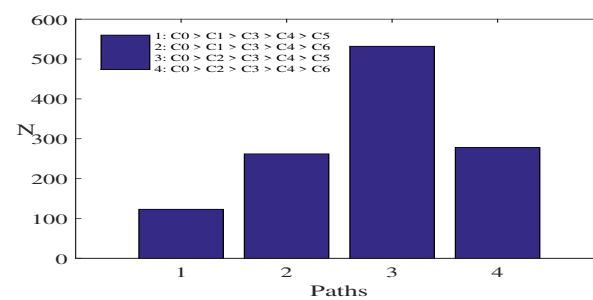


FIGURE IV.17: recurrence of different paths.

CHAPTER V

Real Environment evaluation

1 LobNet platform

LobNet is a new model of camera networks that can be used for environment monitoring and understanding. While conventional networks can be composed of both smart cameras, which benefit from high resolutions, powerful processing capabilities and strategic viewpoints on the environment, here we propose silly cameras called Ant-Cam, defined by much lower specifications. We demonstrate how our approach can reach efficient high-level understanding in spite of the limited information provided by each silly camera. We thus introduce the Ant-Cam specifications, and the processing implemented. The main idea is that data exchanged between the cameras is as important as the information extracted locally. Fully exploiting the cameras interactions without prior knowledge about the network configuration, the system is able to learn regularities and then infer from distributed sequences of events, passed between Ant-Cams.

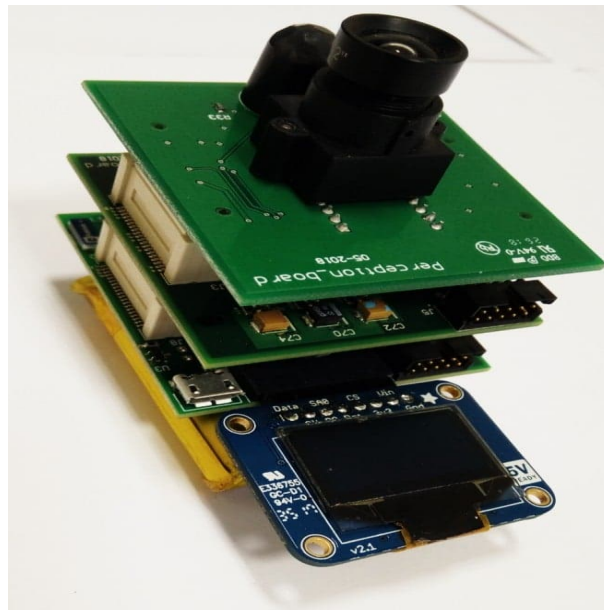


FIGURE V.1: The Ant-Cam

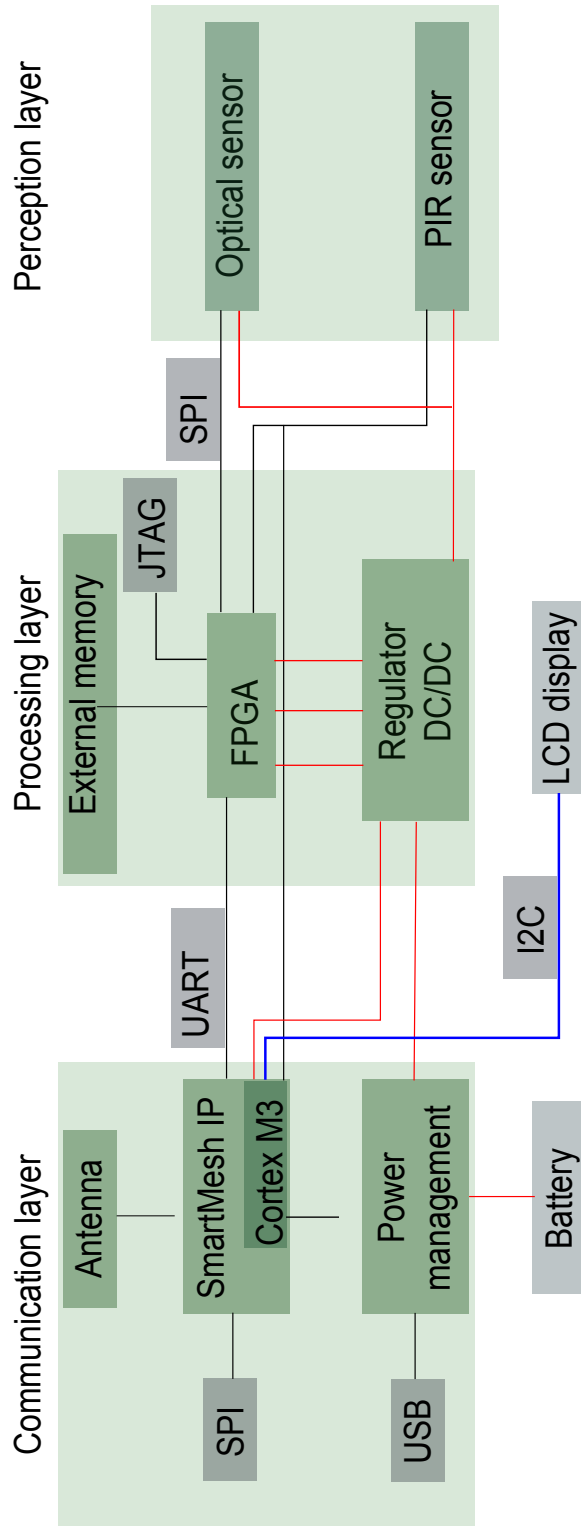


FIGURE V.2: The Ant-Cam architecture

1.1 Sensing layer

The components of the detection unit are presented in this subsection. Perceiving the environment depends essentially on the visual purview. In **SCN** context, this layer acquires images continuously in order to perform environmental measurements according

to the needs of the network. The relevance of detection varies according to environmental conditions where changes in ambient noise levels can lead to significant corruption and increase the complexity of detection. Since sporadic detection can reduce energy consumption compared to constant event monitoring, other detection units can be added. The complexity of event detection also plays a crucial role in the choice of components. Nevertheless, the choice depends on the final objective of the application. In this work, and to follow the ant metaphor, we opt for very low detection capabilities.

1.1.1 Mouse sensor

In the 1990s, the mouse was converted into a camera for different applications. This sensor has been deployed in various fields such as forest fire detection [FBCGCG⁺10], care of the elderly [MFWH15] and control of absenteeism in offices [EDPA]. The main characteristic of this sensor is its tiny images, which preserve data confidentiality during monitoring. However, the platforms cited above circumscribe the sensor use to a detector and none of them worked on the re-identification problematic.

ADNS-3080 is based on optical navigation technology, which optically acquires 900 pixel grayscale images encoded on 6 bits. In addition to the image acquisition system which acquires microscopic surface images via the lens, this sensor contains an integrated Digital Signal Processor (DSP) to process the images and determine the direction and distance of motion. The DSP calculates the Δx and Δy relative displacement values. This is performed with up to 6400 frames per second. Nevertheless, the sensor is equipped by four-wire serial port used to set and read parameters in the ADNS-3080 with a maximum Clock Frequency = 2 MHz. Thus, the maximum frame rate for image transmission to an external processor is limited to 30 Fps. Examples of images acquired by ADNS3080 are presented in Fig.V.3. The main features of ADNS3080: Programmable frame rate over 6400 frames per second, Smart Speed self-adjusting frame rate for optimum performance, Serial port burst mode for fast data transfer, Single 3.3 volt power supply, Four-wire serial port along with Chip Select, Power Down, and Reset pins.

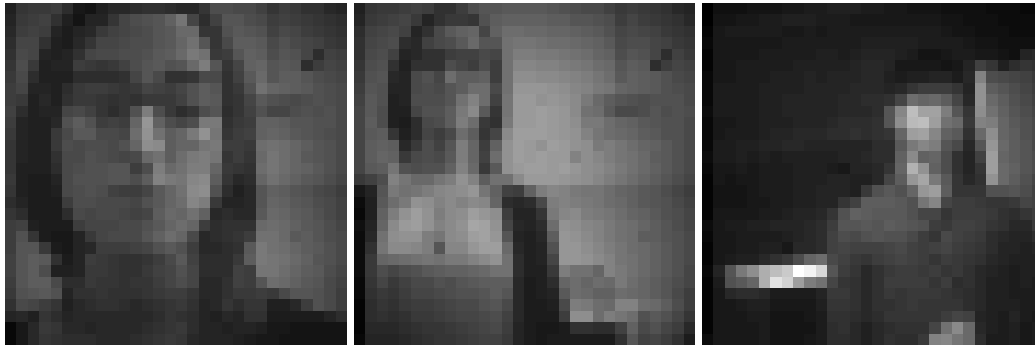


FIGURE V.3: Example of images taken using Ant-Cam, the resolution is 30*30 pixels

1.1.2 PIR sensor

To eke out the maximum of energy, we add a PIR sensor in order to arouse the mouse sensor only in case of requirement. For this, we choose a digital motion detection sensor. Thanks to its high sensitivity, the sensor is able to detect any human body with an approximate size of 700*250mm, crossing the detection beam, when the temperature difference between the environment and the target is higher than $\Delta T = 4^{\circ}\text{C}$. Table V.1 highlights the sensor characteristics.

TABLE V.1: Specifications and Electrical characteristics

Items	Theoretical values
Detection distance	5m
Detection area	horizontal 100° Vertical 82°
power consumption	170uA
Operating voltage	Max 6.0V Min 3.0V
circuit stability time	30 sec
Ambient temperature	-20°C to 60°C

Although the analog device may offer more significant information regarding the detection, where the detection signal can contain information regarding the speed or the motion direction, we choose to work with the digital device. The main and only reason for that is the operating voltage which is 4.5V for the analog device versus 3.0V for the analog one.

After the power is turned on, and during Twu(30 seconds), the circuitry is stabilizing. Hence, the sensor output is not fixed in the ON or OFF state. Any detection during this time will be ignored.

1.2 Processing Layer

The concept of use of the tiny sensor focuses on the redundancy of these cameras. So, to follow the same philosophy, we choose to implement a minimal processing. Two main processes are on hand in the cameras. The first is the environmental perception that is performed in the **FPGA** to decipher the visual data. The second is related to camera interactivity and supported in an M3 cortex thanks to SmartMesh Ip devices.

1.2.1 **FPGA**

Considering vision and image processing applications, different types of electronic hardware platforms try to tackle the outcoming issues. First, CPU, the traditional sequential processor for general purpose applications, is known for its versatility, multitasking and ease of programming. GPUs in their turn are used in a wide range of computationally intensive applications for their massive processing powers. However, both of them offer a relatively low performance/watt ratio withal. This ratio is mended with the ASICs devices, yet strayed by their comparatively high cost and low flexibility. A trade-off analysis between flexibility, performance and power consumption enthrones the **FPGAs** with the best compromise, chiefly for its energy efficiency compared to GPUs and better flexibility compared to ASICs. **FPGA** accommodates massively parallel operations, which can offer a streaming-processing model of images computing. For the Ant-Cam, we opt for a max 10 **FPGA** from Altera. The main characteristics of the device are:

- Low power : Sleep mode: significant standby power reduction and resumption in less than 1 ms. Longer battery life: resumption from full power-off in less than 10 ms.
- memory: The embedded memory structure splits of 42 M9K memory blocks columns with a total of 387Kb. Each block provides 9 Kb of on-chip memory and can be

configurable as RAM, FIFO buffers, or ROM. In addition, Max 10 offers an user flash memory UFM with a maximum of 1378 Kb accessible using Avalon Memory Mapped (Avalon-MM) slave interface protocol, gainful for non-volatile information storing.

- Logic elements: 8K logic elements
- Package : UBGA Package Type 324 : 324 pins, 15 mm x 15 mm
- GPIO : 250 General Purpose I/O are proposed for connection with external devices such as the visual sensor, PIR sensor and SmartMesh Ip chip.
- Clocking and PLLs: The max 10 offers 2 phase-locked loops (PLLs) which provides robust clock management and synthesis for device clock management, external system clock management, and I/O interface clocking. The global clock used has a 16 MHz frequency.

In addition, the MAX 10 devices contain two Analog-to-Digital Converters used to monitor many different signals. A single-chip Nios II soft core processor is supported too. These highlights have not been utilized during this work.

1.2.2 Cortex M3

Although the Max 10 FPGAs support the integration of the soft core Nios II embedded processors which can be used for networking analysis and probabilities computation, these latter are performed with the Cortex M3 integrated in the communication chip. Indeed, the cortex is an excellent gear for probabilistic calculation and wireless network stack managing. The chip is an ARM Cortex™-M3 32-bit microprocessor running Micrium's μ COS-II real-time operating system. With up to 32Kb of flash memory and 8Kb of RAM available, the device can be used for (i) control peripherals via its General Purpose Input-Output (GPIO) pins using various protocols such as Serial Peripheral Interface (SPI) Master, Inter-Integrated Circuit (I2C) Master, 1-Wire Master and Universal Asynchronous Receiver/Transmitter (UART) which is use here to communicate with the FPGA. (ii) Process data like statistical analysis of the computed probabilities and likewise building the vision graph and estimating the links. Thus, the decision-making and control are established. (iii) Manage wireless communication between the cameras based on the vision graph created.

1.2.3 External memory

In addition to the embedded memory available, an external flash memory is integrated to the Ant-Cam. Indeed, increasing the memory on-board allows the camera to support more complex processing tasks. SST26VF064B memory from microchip is adopted. Using a single Voltage Read and Write Operations (2.3-3.6V) with a High Speed Clock Frequency (up to 104 MHz) while lowering power consumption (15 mA for Active Read current (typical @ 104 MHz) and 15 μ A for Standby Current (typical)). It supports both Serial Peripheral Interface (SPI) bus protocol and a 4-bit multiplexed SQI for its 64Mbit.

1.3 Communication Layer

In SCN, sensor nodes are scattered in the environment. Two main points should be considered while developing a network layer: routing protocol and communication protocol.

Dedicated multi-point wireless routing protocols between sensor nodes are required. The ad hoc routing techniques already proposed in the literature do not generally meet the requirements of sensor networks for multiple reasons (large number of nodes, variable topology, and broadcast communication paradigm). The networking layer of the SCN is generally designed along several principles such as energy efficiency, end-use application, data aggregation and location knowledge. Different routing protocols are designed for IoT such as Flooding, Gossiping, SMECN, SPIN or AODV. In this work, and as presented in Chapter II, the routing protocol is established at run-time according to the events generated on the network.

When choosing wireless communication technologies for IoT platforms, the trade-off between transmission speed, power consumption, range and frequency is set according to the application. On one side of the spectrum, technologies like ZigBee, Bluetooth and Bluetooth Low Energy are used for their low power consumption at the price of an average throughput of 250Kb and a limited range of 100m. The LoRa outperforms these protocols in terms of power consumption and transmission range at the cost of a very limited throughput. On the other side, WiFi offers a better throughput and baud rate while requiring high energy consumption. The latter has been recently improved in the proposed new standards such as WiFi 6¹.

1.3.1 SmartMesh IP

Winners of the 2017 Annual Creativity in Electronics Awards as Internet of Things Product of the Year, SmartMesh IP is a wireless technology developed by Linear Technology dedicated to IoT. Derived from very low power and high reliability protocols such as WirelessHART, SmartMesh IP combines 6LoWPAN and 802.15.4e standards to establish a fully mesh network. Each device in the mesh network has the same routing capabilities, often referred to as "mesh to edge", since it provides redundant routing to the edge of the network. This results in a self-training and self-healing network that continuously adjusts to variations in topology, while preserving very high data reliability, including in harsh frequency environments.

- **Low Power Consumption:** This protocol ensures low power consumption for receiving and sending packet with 5mA for receiving and 8mA for sending.
- **15.4 Standard with Ipv6 Ready:** This is an important point when establishing an IoT ecosystem. While being able to exchange data between devices is important, collecting them in a centralized point to check the contents is so important.
- **AODV ready:** collecting all the data with only one centralized point may face a big issue when we collect from deal with short range "devices". Thus, systems are equipped with AODV protocol which aims to find a way from a mote A to mote B which is not in its range, or simply to the centralized point. SmartMesh IP offers the possibility to find by itself the best path to follow the send the information from mote A to mote B.
- **High reliability:** 99.9 %

The SmartMesh® network consists of a mesh of self-forming multiple-jump nodes, called "motest", collecting and relaying data, and a network manager that monitors and manages network performance and security, and exchanges data with a host application.

¹<https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-6>

1.3.2 The mote

SmartMesh IP Motes are the wireless nodes of a SmartMesh IP network, here the Ant-Cam. They connect to sensors/actuators and transmit data from other motes, while remaining at low power. Each mote can send and receive messages at the same time (supports bi-directional data), and can have different data transmission rate. In this case, the network manager automatically coordinates individual communications in pairs to efficiently manage route traffic.

1.3.3 The manager

The manager has two main duties: First, it is an access point that acts as a gateway between the mesh network motes and the monitoring or control network. Second, it runs the network application software that constantly makes final decisions on how to build and maintain the mesh network. The manager harvests around 15% of the total bandwidth for network organization, advertising, neighbor discovery and communications. It manages a mesh network with slots and chain jumps. It manages each node to know exactly when to sleep, listen or talk, allowing a very efficient and collision-free packet exchange.

1.4 Power management

Most of the smart cameras networks use a battery-operated cameras which mean that the lifetime of the cameras are restrained by the energy-consumption. Thus, optimizing energy consumption is a major challenge and an important topic. In the network context, it can be reduced by well split the tasks between cameras, improve the network to get a better performance-energy ratio. [MO05] present a survey of energy optimization and different points to consider while developing a battery-operated system. The camera is powered by a battery with 2000mAh at 3.7V. This battery is rechargeable via usb interfaces thanks to the module BQ24075 used.

2 Implementations

In this section, we present our method of target recognition and tracking. We detail the processing steps in the camera and network. This is essentially based on issues of target recognition at very low resolution: Lack of effective features, Noise affection and Dimensional mismatch and misalignment. In distributed monitoring, tasks need to consider the result of local processing of a camera, as well as collaborative processing with other cameras.

2.1 Low-level processing layer: solo processing action

2.1.1 The Background-Foreground segmentation

BFS is the first processing to apply when a camera detects a moving target in the scene by the PIR sensor. Given the complexity of the processed images, the BFS approach should analyze changes in the image structure (e. g., the edges of the scene) between two images, instead of changing the grey value like most other approaches do[?,?]. Several methods were implemented and compared [END⁺14] highlighting that not all methods can show performance and resistance to changes in lighting. Following this work carried out, we opt for the correlation-based foreground/background segmentation.

2.1.1.a The background construction and update To start with, the background model is computed by averaging all images when the scene has no one present. However, to eliminate the detection of false foregrounds of non-human targets and to ensure a robust system against fast and slow lighting changes in indoor or outdoor environments, the background model is adjusted according to the learning rate α :

$$I_{bg}(p)' = (1 - \alpha)I_{bg}(p) + \alpha I(p) \quad (V.1)$$

where $I_{bg}(p)$ and $I(p)$ are respectively the background model and the new image captured at pixel p , and α is the learning rate. This rate α is fixed at 0.01, so new background objects incorporate slowly into a new background image after being captured in 100 images.

2.1.1.b Foreground segmentation For each pixel in a new frame, a correlation coefficient $\varrho(p)$ is estimated. It represents the correlation between the pixel of the captured image and the corresponding pixel of the background model within the sliding window around the concerned pixel:

$$\varrho(p)^t = \frac{(\sum_{p' \in \omega(p)} I(p')^t * I_{bg}(p)^t)^2}{\sum_{p' \in \omega(p)} I(p')^t * \sum_{p' \in \omega(p)} I_{bg}(p')^t} \quad (V.2)$$

where $\omega(p)$ is a sliding square window centered at p and $\varrho(p)^t$ is the correlation coefficient between captured image pixel $I(p')^t$ and background image pixel $I_{bg}(p')^t$ over the pixels in $\omega(p)$. In this step, the pixel can be classified as background or foreground following:

$$FG(p) = \begin{cases} I(p), s = s + 1 & \text{if } \varrho(p) < \varrho_{min} \\ 0 & \text{otherwise} \end{cases} \quad (V.3)$$

where ϱ_{min} is the correlation threshold fixed between 0 and 1, and s is the number of pixels constructing this foreground. The result is a new image FG with black pixels corresponding to the background, and gray scale pixels representing the foreground object. Table V.2 summarized the parameters used for better performance.

TABLE V.2: the parametric values used for the BFS

Parameters	Values
α	0.01
size of ω	3*3
ϱ_{min}	0.98

Using tiny images, full of inconveniences, can be turned on advantages. One of them, is the position problem. With the Ant-Cam, the lower variation of the position, direction while detecting, may completely change the images. This problem is used later by our camera to estimate the origin of the object. Indeed, depending from where the target is coming, the appearance would completely change, which can be a starting point for the camera to decide about the target. In figure V.4, images V.12b, V.12c and V.12d are

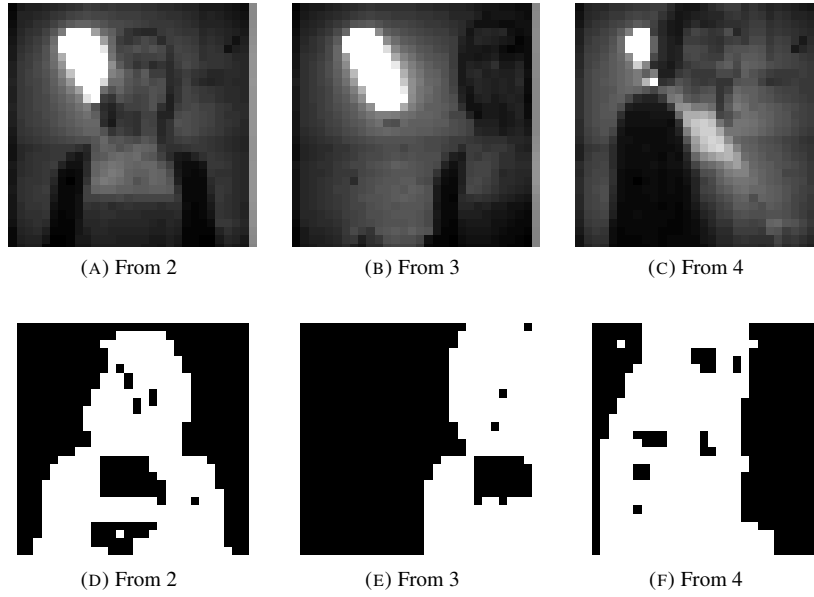


FIGURE V.4: Different images captured with camera 1 for a target coming respectively from cameras 2, 3 and 4.

captured from camera 1 for a target coming respectively from cameras 2, 3 and 4. Due to our **BFS**, we extract a set of straightforward semantic presented in [V.12h](#), [V.12i](#) and [V.12j](#), corresponding to the results of segmentation, where white pixels corresponds to the foreground detected. Each one of them express the character of the object such as the posture or the shape and the position. These semantic empirically extracted help to assign each object detected to one camera.

This processing is carried out on data flow for low-complexity and low-memory demanding model which reduces hardware cost and energy consumption.

2.1.2 Image selection

At this level, the shape size extracted s is the reference for the image selection for the subsequent processing. Considering the image size and limited information, it is deemed entirely unhelpful to waste computing time and resources to process images lacking information. Furthermore, this can be a source of error in the computations and could affect the result in a wrong way.

TABLE V.3: Notations for used parameters

Index	corresponding
s	sensor size
f	focal length
$d1, d2$	detection distances
$FOV1, FOV2$	Fields of view within $d1$ and $d2$

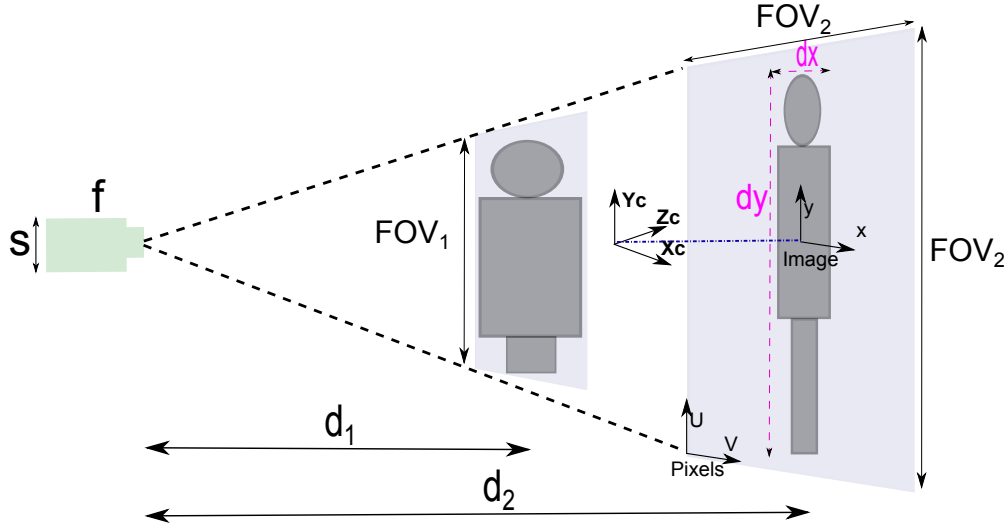


FIGURE V.5: Projection of the target in the 2D plan.

A target T in the environment defined by:

$$T = \begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix}$$

a 2D-projection gives:

$$x = f * \frac{X^c}{Z^c} y = f * \frac{Y^c}{Z^c} \quad (\text{V.4})$$

$$T = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f * \frac{X^c}{Z^c} \\ f * \frac{Y^c}{Z^c} \\ 1 \end{pmatrix}$$

a projection in the pixel matrix:

$$P = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} K_u & 0 & 0 \\ 0 & K_v & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

For a target at instant t , the size in pixels can be highlighted with the real target size, since K_u , K_v and Z^c are constants.

We suppose that we detect target at a distance of 1m until 3m maximum. Thus, the size of the field of view is calculated at this distance to find the values of K_u and K_v .

This :

$$u = K_u \cdot f \cdot \frac{X^c}{Z^c}, v = K_v \cdot f \cdot \frac{Y^c}{Z^c} \quad (\text{V.5})$$

where K_u is the pixel density in the direction of the axis u and K_v the pixel density in the direction of the axis v . For a sensor size of 1.8mm*1.8mm, $K_u = K_v = 16.66$. The

focal length is 4.2mm. In this work, we focus on target detected within a distance of 1m and 3m. Outside these values, we consider the target unidentifiable. Thus a minimum foreground size is fixed to 200 pixels and the maximum to 700.

2.2 Medium-level processing layer: duo processing action

In this level, the two cameras detecting a target respectively should collaborate to decide about their detection accordance.

2.2.1 Features extraction

2.2.1.a The PCA The **PCA**, as a statics method, raises the characteristics of the images to build a new model representing the target. Starting from our images, the **PCA** picks up the most important information about it. Thus, transmitting to the next cameras, where a **PCA** is applied too, camera can figure out if it is the same target or not, by measuring the distance between the two new representation. Considering the fact that if the target is not seen on the same way (front, back, side images...), the **PCA** would not give a pertinent characterizations. Indeed, the **PCA** is efficient in case the targets are seen from the same side in both cameras. then, a projection of two **PCA**'s results may give good results. However, in case the target is seen from different perspectives, the information extracted will be completely different. Thus, a projection will give false results.

2.2.1.b Target space The set of images detecting the target is then used to create the target space $X : M * N^2$, where each N^2 -dimensional feature vector is equal to the number of pixel of each images and M is the number of images considered. Typically, our 30*30 pixels image generated a target space $M * 900$. Thus, **PCA** which seeks a projection that best represents the data in a least-square sense is reduced for reducing the dimensionality of such a target space, while finding the vectors with best account for the distribution of the target with the entire target space.

2.2.1.c Eigentarget The eigenvectors corresponding to nonzero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace. A corresponding match between the different target space is then estimated. Up to the previous step, this constitutes a traditional **PCA** approach. However, considering the transformation before estimating the distance between the different spaces can be potentially more significant. As presented in the previous section, linear transformation is considered as:

- Transforming each 2D image selected in a 1D vector to generate a target space.
- Estimating the mean space, and obtaining a mean centered new space
- Obtaining the Covariance matrix
- Finding the eigenvectors and Eigenvalues
- Selecting the eigenvectors associated with the largest eigenvalue which reflects the greatest variance in the image. The target space is the projected onto less dimension space to obtain the "eigentarget".

2.2.2 Camera-to-camera translation

Image to image translation is a category of vision and experience. Graphic problems that aim to learn the mapping between an input and an output image through a training of matched image pairs. Several works present approaches to learn how to translate an image from a source domain X to a target domain Y such as [ZPIE17]. Through years of research in computer vision, image processing, digital photography and graphic design, powerful translation systems have been achieved in a supervised environment. The main idea is to convert an image from an observation of a given scene to another such as grayscale image to color image or edge-map to photograph. In all these works focus on paired training data, where correspondence between input and output is well defined. This requires obtaining paired training data which can be difficult and expensive specifically when the output is very complex. For this, researchers in [ZPIE17] present an approach that can learn to do the same without supervision, in unpaired data: capture the distinctive features of one image collection and determine how these features would be translated into another image collection. They proposed an algorithm to learn how to translate between domains without paired input-output scenarios. This supposes that there is an implicit connection between the domains and aims to identify this connection. For example, two different renderings of the same underlying scene. This goes beyond supervision in the form of matched examples, but rather exploits supervision at the set level. In other word, given two set of images in two domains A and B , the trained mapping $M : A \rightarrow B$ such as for $a \in A$, $\hat{b} = M(a)$ would be indistinguishable from the set $b \in B$. Indeed, the optimal M would translate A to domain \hat{B} identical to B , but generate an infinity of mapping M that will generated the same distribution over \hat{b} . Consequently, each input a and output b will not be perfectly matched individually. This issue has been addressed by suggesting that the translation become "cycle consistent" by proposing an inverse translation to have a bijection mapping. For each mapping $M : A \rightarrow B$, a self-inverse mapping $N : B \rightarrow A$ is estimated. M and N are therefore inverse of each other and verifying $M(N(a)) = a$ and $N(M(b)) = b$. This approach was developed later [IZZE17] to guarantee that the mapping between two image domains is unique or one-to-one.

From this perspective, we proposed our approach to identify the mapping between the different cameras of the network with several challenges, that have never been studied before:

- Completely unsupervised reidentification.
- Distributed task between 2 cameras detecting a target.
- Online learning and online mapping.
- New update mapping after each detection.
- Very low resolution images.

Our goal here is to estimate mapping functions between the observation of a target $F_A^{t_1}$ and $F_B^{t_2}$ by two cameras A and B respectively at instant t_1 and t_2 , and also between observations of different targets $F_A^{t_1}$ and $F_A^{t_3}$ by the same camera A . As illustrated in Figure 2 our model includes 3 mappings: $M_B^{t_3-t_1} : F_B^{t_3} \rightarrow F_B^{t_1}$, $M_A^{t_2-t_0} : F_A^{t_2} \rightarrow F_B^{t_0}$ and $M_{B-A}^{t_1-t_0} : F_B^{t_1} \rightarrow F_A^{t_0}$. We then introduce $\hat{F}_B^{t_3}$ as the representation of the target generated in the network based on the previous observation. The goal is to estimate the matching

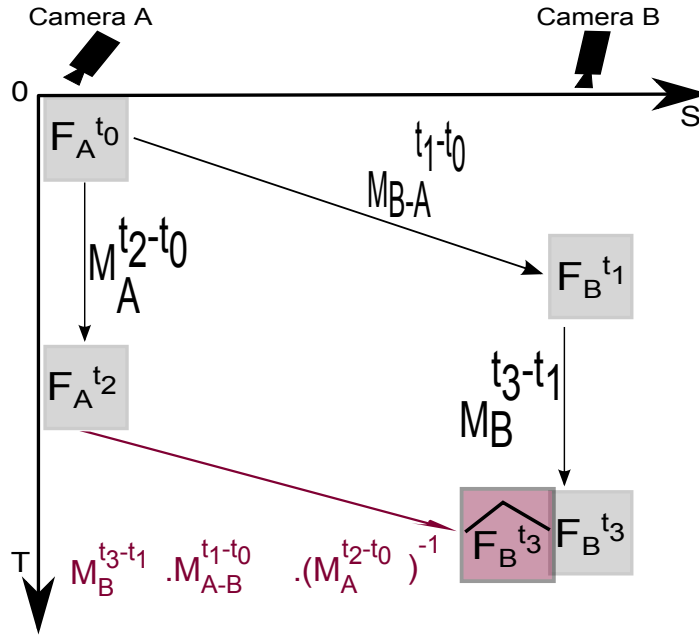


FIGURE V.6: Illustration of the different transformations. Spatial transformations between camera A and B. Temporal transformations between a current target detection and the reference target.

between the prediction generated $\hat{F}_B^{t_3}$ and the observation effectively measured by the camera B.

2.2.3 Formulation and assumptions

Deploying a large number of camera without any preconfiguration and precalibration, as complex it is, requires to assume some assumptions. Indeed, the purpose of mapping is accomplished using two separate cameras and at two distinct instants. This is accomplished in two phases: Initialisation of the transformations and upgrade following each detection:

- **Geometric transformation:** Starting with a completely unknown environment, it is nevertheless essential to establish a baseline. This step require to consider the first target moving in the network and generating internal and external events as a reference for the remainder of the events. Considering one target moving through the network, the geometric transformation of its appearance is consider as a spatial transformation between the cameras, represented in figure V.6 with $M_{B-A}^{t_1-t_0}$. This in the aim of quantifying the spatial transformation of all future targets. All the while assuming that the cameras are static.

This phase is represented in figure V.7 as phase one by interconnecting the different cameras.

- **Temporal transformation:** Estimating the temporal transformation of each target features online avoid having a prior learning phase and reduce memory storage. As

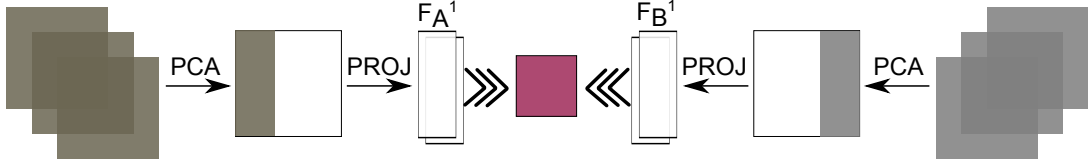


FIGURE V.7: A series of steps is followed in each of the two cameras in order to find the transformation between each other. Starting with the [PCA](#), a projection is applied to reduce the dimensions of the target space before estimating the transformation.

illustrated in the figure, the only part remaining in memory is the calibration results of geometric transformation.

In addition, taking into account the data transfer costs in a wireless network, shared data is minimized, and only the transformations of the eigenimages are shared. While $M_{B-A}^{t_1-t_0}$ is estimated only for the reference target, $M_A^{t_2-t_0}$ is estimated after each detection in the camera A, and $M_B^{t_3-t_1}$ after each detection in the camera B.

2.2.3.a Problem linearization The objective is to model the transitions by providing a system that can be parameterized through the initial configuration as well as the post-detection regulation to ensure the desired performance for reidentification. Before any modeling, it is necessary to define the system environment. The system environment is delimited by several independent input variables: target detections that define the system state, and output variables that report, following a detection, the results of the system's reidentification with respect to the detections. Additional inputs from the learning phase are then integrated. The more the system is distributed, the more necessary it is to establish a network of simultaneous equations to describe the system. In general, the model we intend to establish is the result of a compromise between fidelity to the real behavior of the system at various excitations and simplicity. Simplicity is achieved through working hypotheses and approximations that make the model mathematically viable. In our network, when we re-identify a target, we are interested in the detection of this target by cameras detecting this target in previous instants, following a non-Markovian model. For visual information, we are only considering the latest camera detecting this target. This makes it possible to linearize the subsystem defined by two cameras successively detecting the target. Thanks to the linearization of subsystems around two cameras, and under certain assumptions (the same target), the system can be described by a linear mathematical model. The linearization method is a valid method only locally (between two neighboring cameras) and therefore, this method cannot be used to define a global network behavior.

Thus, the relationship between the cameras, illustrated in Figure [V.6](#), is expressed in the following equations:

$$F_B^{t_1} = M_{B-A}^{t_1-t_0} \cdot F_A^{t_0} \quad (\text{V.6})$$

$$F_A^{t_2} = M_A^{t_2-t_0} \cdot F_A^{t_0} \quad (\text{V.7})$$

$$F_B^{t_3} = M_B^{t_3-t_1} \cdot F_B^{t_1} \quad (\text{V.8})$$

Thus, following the figure V.6 our prediction $F_B^{t_3}$ can be defined as:

$$\widehat{F}_B^{t_3} = M_B^{t_3-t_1} \cdot M_{B-A}^{t_1-t_0} \cdot (M_A^{t_2-t_0})^+ \cdot F_A^{t_2} \quad (\text{V.9})$$

In fully distributed context, where cameras are completely desynchronized, the equation V.19 becomes where camera events are stamped by the internal and external camera event only:

$$\widehat{F}_B^{t_2} = M_B^{t_2-t_1} \cdot M_{B-A}^{t_1-t_1} \cdot (M_A^{t_2-t_1})^+ \cdot F_A^{t_2} \quad (\text{V.10})$$

Where M represent a set of transformation related to geometric and photometric transformation considering orientation, translation size and intensity.

2.2.3.b Translation verification Following V.6, V.7, V.8 and V.9, we suppose:

$$(\widehat{M}_A^{t_2-t_0})^+ \cdot F_A^{t_2} = F_A^{t_0} + \epsilon \quad (\text{V.11})$$

Thus, V.9 becomes:

$$\widehat{F}_B^{t_3} = \widehat{M}_B^{t_3-t_1} \cdot \widetilde{M}_{B-A}^{t_1-t_0} \cdot (\widehat{M}_A^{t_2-t_0})^+ \cdot F_A^{t_2} \quad (\text{V.12})$$

$$\widehat{F}_B^{t_3} = \widehat{M}_B^{t_3-t_1} \cdot \widetilde{M}_{B-A}^{t_1-t_0} \cdot (F_A^{t_0} + \epsilon_A^{t_2}) \quad (\text{V.13})$$

$$\widehat{F}_B^{t_3} = \widehat{M}_B^{t_3-t_1} \cdot (\widetilde{M}_{B-A}^{t_1-t_0} \cdot F_A^{t_0} + \widetilde{M}_{B-A}^{t_1-t_0} \cdot \epsilon_A^{t_2}) \quad (\text{V.14})$$

$$\widehat{F}_B^{t_3} = \widehat{M}_B^{t_3-t_1} \cdot (F_B^{t_1} + \epsilon_{B-A}^{t_1} + \widetilde{M}_{B-A}^{t_1-t_0} \cdot \epsilon_A^{t_2}) \quad (\text{V.15})$$

$$\widehat{F}_B^{t_3} = \widehat{M}_B^{t_3-t_1} \cdot F_B^{t_1} + \widehat{M}_B^{t_3-t_1} \cdot (\epsilon_{B-A}^{t_1} + \widetilde{M}_{B-A}^{t_1-t_0} \cdot \epsilon_A^{t_2}) \quad (\text{V.16})$$

$$\widehat{F}_B^{t_3} = F_B^{t_3} + \epsilon_B^{t_3} + \widehat{M}_B^{t_3-t_1} \cdot (\epsilon_{B-A}^{t_1} + \widetilde{M}_{B-A}^{t_1-t_0} \cdot \epsilon_A^{t_2}) \quad (\text{V.17})$$

$$\epsilon^{tot} = \epsilon_B^{t_3} + \widehat{M}_B^{t_3-t_1} \cdot (\epsilon_{B-A}^{t_1} + \widehat{M}_{B-A}^{t_1-t_0} \cdot \epsilon_A^{t_2}) \quad (\text{V.18})$$

Conceptually, this model compares the feature extracted from the camera $F_B^{t_3}$ with the features generated by the set of transformations $\widehat{F}_B^{t_3}$. This similarity Φ_v between two feature vectors can be measured as a reciprocal of a distance measurement, different types of distance measurements can be used. Here, we use an Euclidean distance for the measurements:

$$\Phi_v = d(\widehat{F}_B^{t_3}, F_B^{t_3}) \quad (\text{V.19})$$

2.3 High-level processing layer

At this level, the re-identification is performed using all the local-network update. The local network represents the set of cameras indicating the camera neighborhood and a part of its vision graph. Thus, the association between each camera pair is established. In figure V.8, we illustrate a sequence of cameras detecting the same target. Each column represents a camera participating in the tracking task. Grey nodes correspond to the processing available in the camera. This can be a temporal, spatial or visual processing.

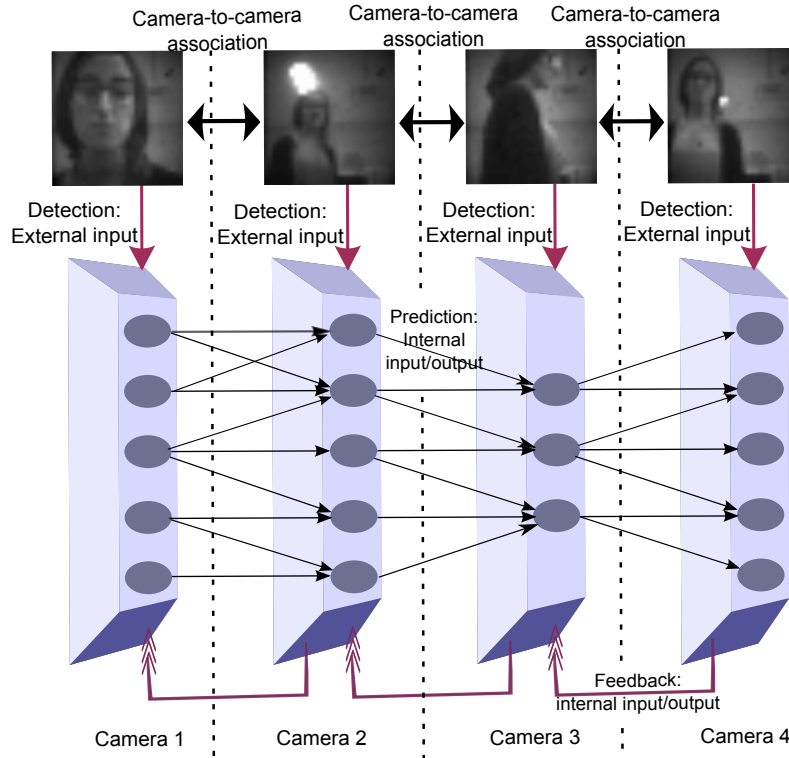


FIGURE V.8: Network architecture: Each column represents a camera participating in the tracking task. Grey nodes correspond to the processing available in the camera.

For visual parameters, the proposed method is applied. Considering the spatial parameters, we choose to consider the last 2 previous cameras. Thus, the target detected by a

camera has 9 possibilities about the followed paths. For temporal parameters, we use the network manager. Indeed, for a mote to join a network, it must get time-synchronized to other devices by hearing an advertisement from an Access Point (AP) mote or a mote already in the network. This message exchange is part of the security handshake that establishes encrypted communications between the manager or application, and mote. Once motes have joined the network, they maintain precise synchronization through time correction messages sent between connected neighbors.

3 Evaluation

3.1 Simulation environment

In the first evaluation, the cameras are utilized only to generate video that can be process in a fixed platform. Nine people moving around the cameras have been selected, generating four video sequences in four asynchronous cameras. Consequently, used in a fixed platform (a laptop in our case), we apply varied processing described in the previous section using a Matlab computing environment. This real-world dataset proposed by [BkQ] is very challenging and is good representation of situations that may occur in real office life

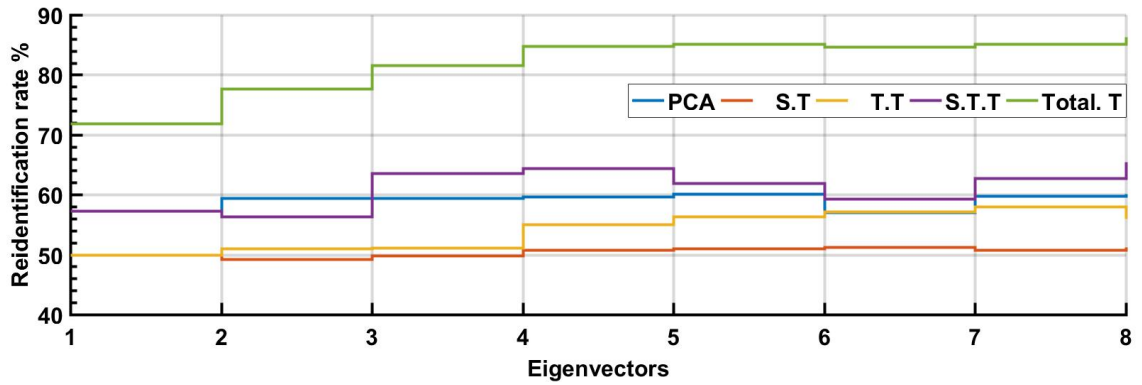


FIGURE V.9: Variation in the re-identification rate according to the number of eigenvectors considered. S.T refers to Spatial Transformation, T.T refers to Temporal Transformation. S.T.T points to Spatial and one Temporal Transformation. Total.T refers to the 3 transformations.

The re-identification is validated based on the measurement of the distance between the inputs of two cameras and the features generated. The results generated by the 5 methods are implemented and compared:

1. No transformation generated: Here the PCA is directly applied on the images of camera A, then transferred to the images of camera B as $\hat{F}_B^{t_3}$. This simple method show its robustness when targets are detected on the same side(face, back..). However, it is definitely not appropriate for different side of detection.
2. Temporal transformation: $M_A^{t_2-t_0}$ is estimated between current person and reference person (for each new person), then apply the transformation on $F_B^{t_1}$ to obtain $\hat{F}_B^{t_3}$. Here, we consider only the features transformation between the targets detected by the same camera.
3. Spatial transformation: $M_{B-A}^{t_1-t_0}$ is computed for the reference person between the two cameras, then apply the transformation on $F_A^{t_2}$ to obtain $\hat{F}_B^{t_3}$. The $M_{B-A}^{t_1-t_0}$ is estimated only once in the calibration phase, and used after each detection.

4. Spatial and Temporal Transformation: Estimate $M_{B-A}^{t_1-t_0}$ for the reference person (done only once) between the two cameras, reinforced by $M_A^{t_2-t_0}$ between current person and reference person, then apply the transformation on $F_A^{t_2}$ to obtain $\hat{F}_B^{t_3}$.
5. Total transformation: It regroups the previous steps. Here, we estimate $M_{B-A}^{t_1-t_0}$ for the reference person (done only once) between the two cameras A and B, then apply the transformation on $F_A^{t_2}$ and consider the transformation on $F_B^{t_3}$ to obtain $\hat{F}_B^{t_3}$. The idea of utilizing this is to consider the transformations really achieved at camera B on those that have been analyzed by camera A.



FIGURE V.10: (a) corresponds to the initial features reference for camera 1, and (b) for camera 2. (c) correspond to the input features after each detection for camera 1 for 5 target detected, and (d) for camera 2. (e) correspond to the features generated via spatial transformation using (c). (f) correspond to the features generated via spatial and temporal transformation of (c). (g) correspond to the features generated via temporal transformation of (c). (h) correspond to the generated features using the whole transformation of (c) and (d). (e), (f), (g) and (h) are then used for comparison with the input (d)

Figure V.10 shows the features detected by camera A and B, as well as the features generated through the different transformations. It represent the $\hat{F}_B^{t_3}$ used as a prediction to

estimate the matching with $F_B^{t_3}$ detected. Overall results are presented in figure V.9. We assess the accuracy of re-identification in the aforementioned methods. The integration of these transformations significantly improves reidentification performance. Indeed, a simple spatial transformation is not sufficient (S.T in figure V.10). However, the reinforcement by the temporal transformation of the current target in respect of the reference (S.T.T in figure V.10). The prediction generated aim then to construct the next target that can be detected by camera B based on the target detected by camera A. The final step (Total.T in figure V.10) considers not only the target detected by the camera A with the necessary transformations, but also takes into account the transformation in the camera B after its detection in order to approach precisely the same features.

TABLE V.4: Pairwise identification for two datasets. SS refers to the detection of the same side detection, whereas DS is for the detection of different sides.

Dataset	True re-ID	False re-ID
1 SS	59.7%	40.2%
2 SS	65.27 %	34.72 %
1 DS	38.88 %	61.11 %
2 DS	51.38 %	48.61 %

TABLE V.5: Tracking performance in the network in dataset 2.

Cameras	camera 1	camera 3	camera 4
camera 1	X	45.83%	65.27 %
camera 3	51.38%	X	53%
camera 4	56.94%	53%	X

3.2 real world environment

The processing described below is implemented in 8 cameras positioned in an indoor environment. Thanks to the manager, the cameras send notifications to a fixed platform (a laptop in our case). Thanks to that, we are able to recover the network state despite the fully distributed processing.

3.2.1 Implementations

No learning phase is implemented in the upstream cameras. Therefore, to create the first transformations that will be used as a reference, a target is launched in the network that will be considered as a validated re-identification. This objective generates the first network events. It permits to initialize the cameras data in terms of magic matrix. In addition, this provides the ability to generate the first communication and thus initialize the network data for the link graphs.

3.2.1.a FPGA: The [FPGA](#) is interfaced to the [PIR](#) and visual sensors for data acquisition. Indeed, as soon as the camera is turned on, the background is built directly and the visual sensor is switched in standby mode. Thus, the visual sensor is only requested if the

PIR sensor detects the existence of a target in the surroundings of the camera. In case of detection, the visual sensor is turned on and the acquired images are directly processed for the BFS. This is done in dataflow mode to save processing time. Once the image is selected, it is transmitted to the cortex M3.

```

component neighExtractor is
    generic(
        PIXEL_SIZE      : integer; --8
        IMAGE_WIDTH      : integer; --30
        KERNEL_SIZE      : integer; --3
    );
    port(
        clk              : in  std_logic;
        reset_n          : in  std_logic;
        enable           : in  std_logic;
        in_data          : in  std_logic_vector((PIXEL_SIZE-1) downto 0);
        in_dv            : in  std_logic;
        in_fv            : in  std_logic;
        out_data         : out pixel_array (0 to (KERNEL_SIZE * KERNEL_SIZE)- 1);
        out_dv           : out std_logic;
        out_fv           : out std_logic
    );
end component neighExtractor;

```

FIGURE V.11: VHDL code for the BFS

TABLE V.6: the parametric values used for the BFS

Parameters	Values
α	0.01
size of ω	3*3
Q_{min}	0.992

TABLE V.7: Resource Utilization of the bfs on the Ant-Cam Platform.

Total logic elements	637 8,064 (8 %)
Total memory bits	8,192 / 387,072 (2 %)

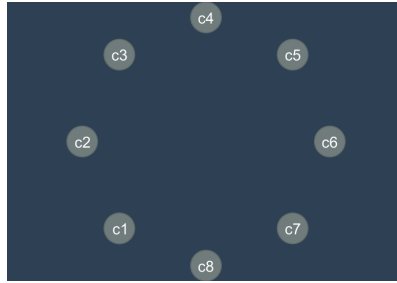
3.2.1.b Cortex: The M3 cortex is utilized for image processing and data exchange. Selected images are received from the FPGA and processed directly. The PCA is applied and the transformations are calculated with the data acquired from the learning phase. Once completed, the data is shared over the network with the concerned cameras. Furthermore, communication is managed here as well. Thus, the confidence information is evaluated after each event, and links are established based on the shared information.

3.3 Network discovering

The first step is the network discovery. This is achieved thanks to the communication protocol SmartMesh IP utilized. Each camera presented in the network send a notification to the manager to be integrated in the network as shown in figure V.12. SmartMesh Ip uses AODV as a routing protocol. Indeed, when a camera turns on, it seeks to reach the manager regardless of how far away it is. By requesting a route, aodv provides the most optimized route and maintains it for as long as the source needs it. With this procedure, the cameras acquire knowledge of the network. The main advantage of this rooting is to

dynamically adapt to changing conditions, such as disabled cameras, and to route information around obstacles, giving the network its fault tolerance and high availability. This routing protocol is low power consuming and does not require a lot of computing power.

The position of the cameras in the figures does not correspond at all to the physical location of the cameras since we start from an unknown topology. Only logical topology is intended to be found in this work.



(A) Network discovery

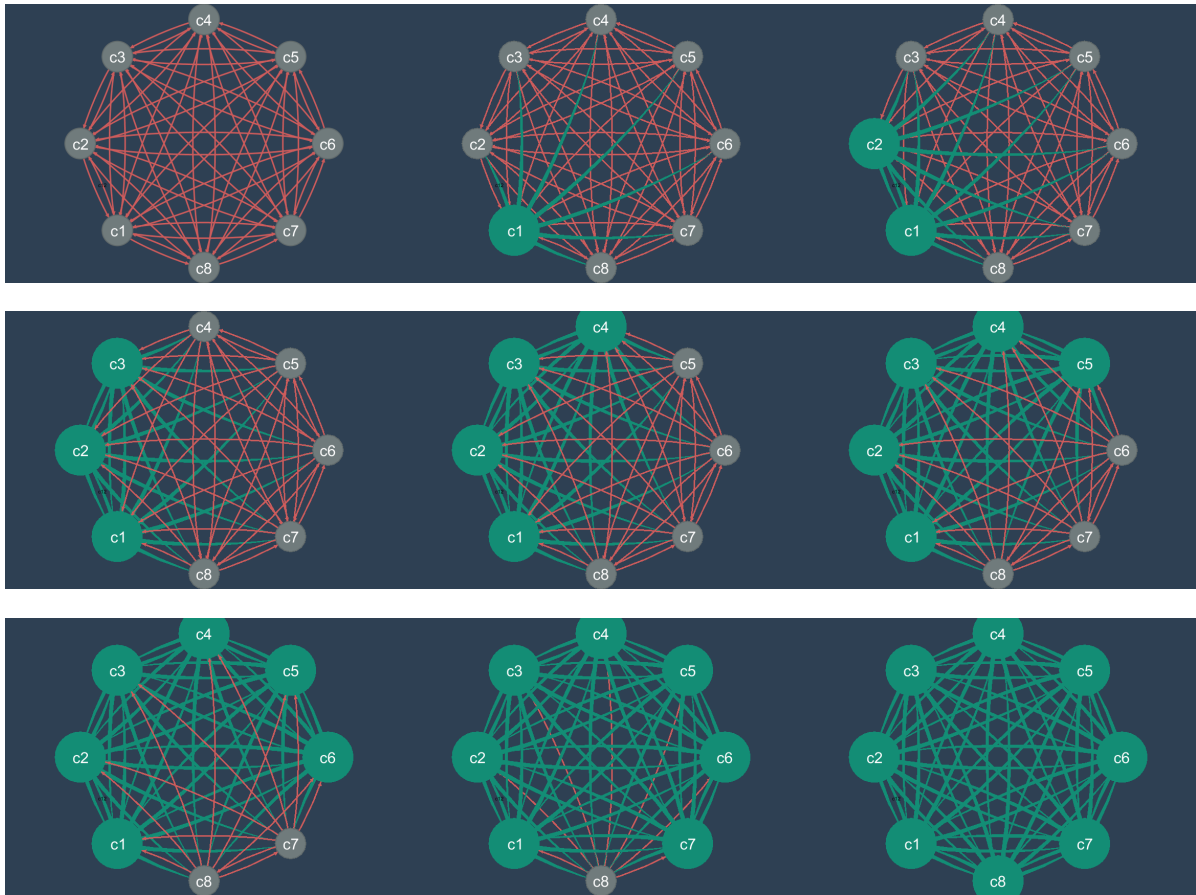


FIGURE V.12: Different network states.

3.4 Network events

Each external or internal event generated in a camera is shared across the network as needed. Figure V.13 shows different events occurred in the network for a target detected by camera 1 and then camera 2. The figure V.14 shows the network evolution over the runtime. In V.14a, the cameras are discovered by the manager of the network. The network discovery then aim to initialize the communication between the cameras in

the network. As SmartMesh IP offers the possibility to communicate with all the cameras presented in the network, each Ant-Cam can initialise the communication with others activated(V.14b). V.14c shows the links strength after 50 events. Each camera manages its input links(orange) and output links(blue).

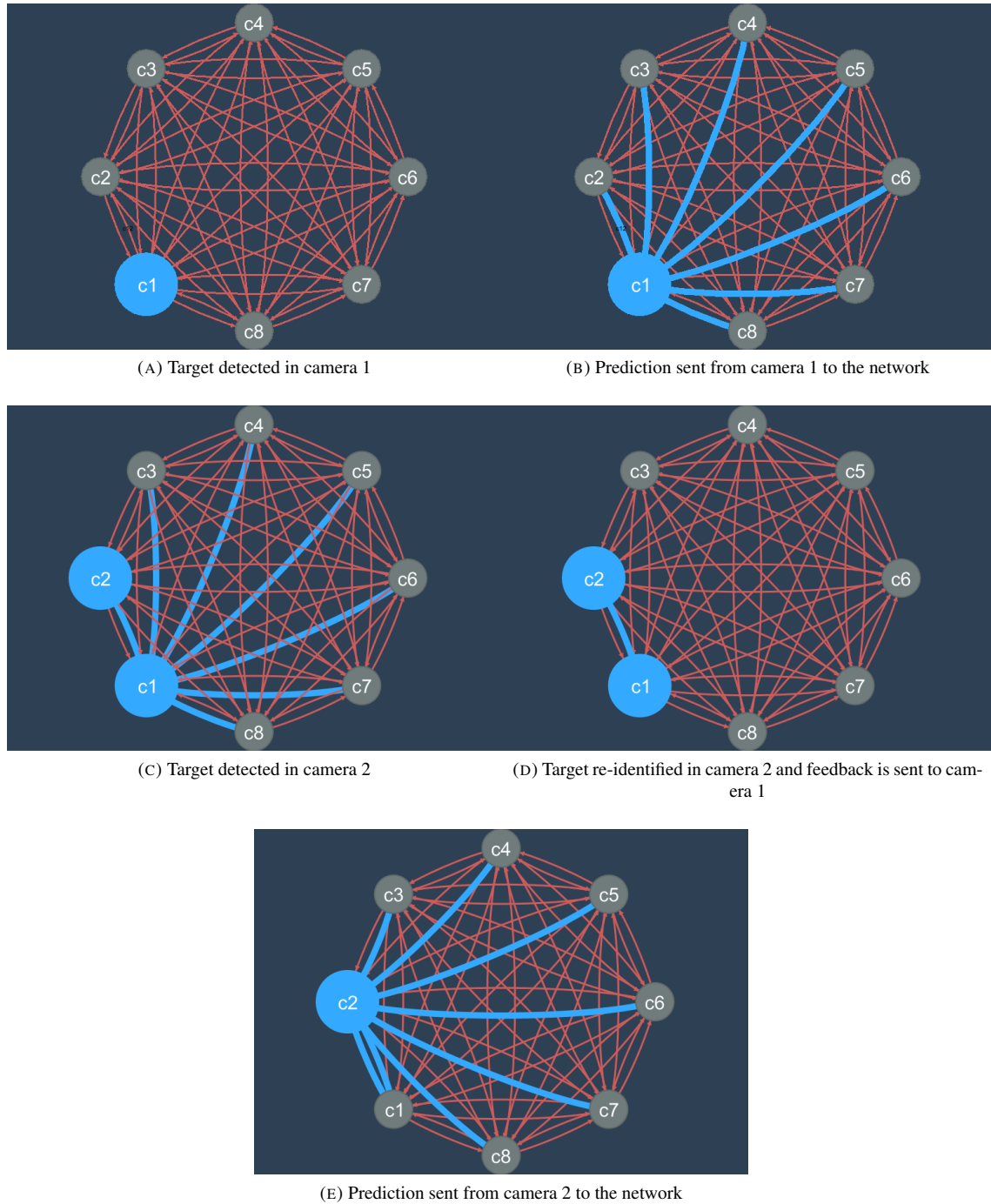
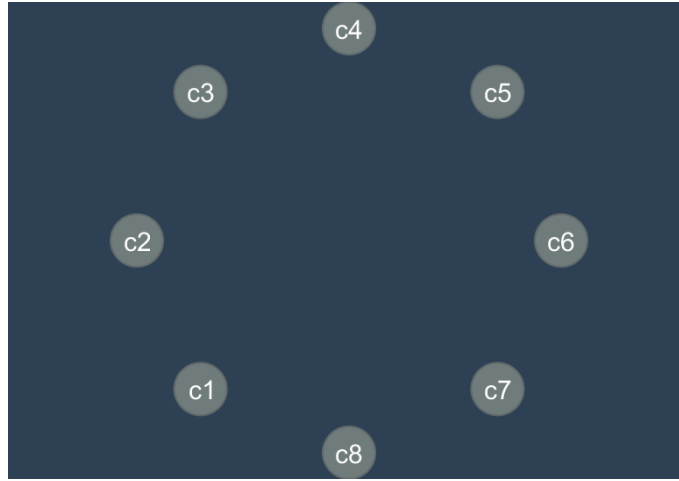
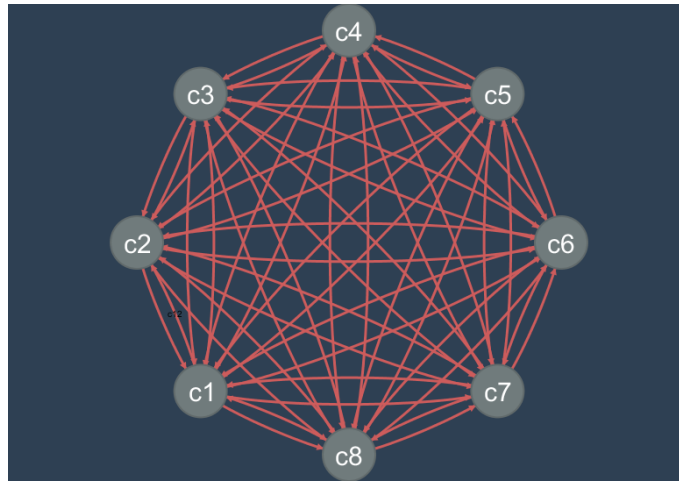


FIGURE V.13: Different events in the network.

The importance of the links between cameras is reflected in the probability of moving from one camera to another. Each camera estimates its probability independently and share the information with the manager. Figures V.15 proposes the probability calculated in camera 1 based on the events occurred.



(A) Network initialization(0 events).



(B) Network discovery(0 events)

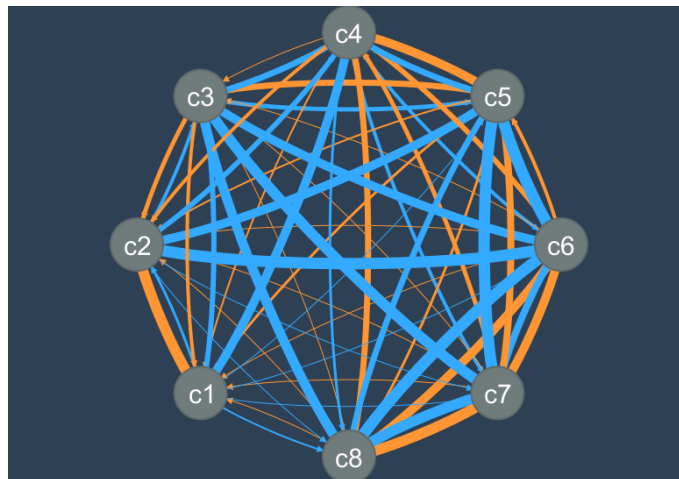
(C) Network vision graph G^L after 50 events, blue graph refers to G^{LO} and orange links to G^{LI} . links.

FIGURE V.14: The vision graph building during the network's run time.

After few events(10 in this case), the network reach a stable state, manifested by the stability of the values calculated for the probability of transition as shown in figure V.15.

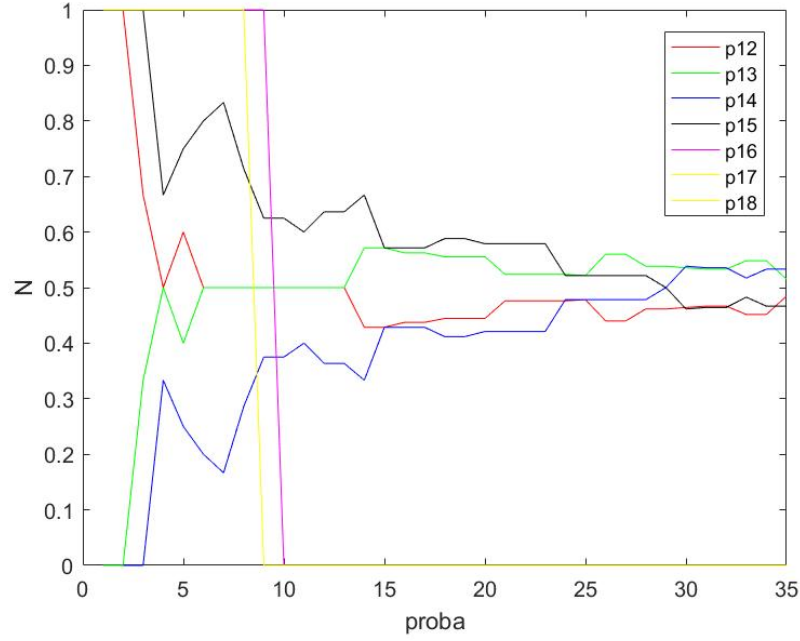


FIGURE V.15: Evolution of the probability in the node 1 in the network.

4 Conclusion

In this chapter, we present the camera developed for this work, defined by very limited specifications. Thus, we present our approach of visual data processing between each pair of cameras exchanging predictions and feedbacks. In addition, we present our real-world implementation. In fact, when referring precisely to hardware deployment, a number of reflections can be appended to the above-mentioned model. The latter bypasses some requirements in terms of calibration and temporal synchronization between cameras. The resulting stable state minimizes communication costs by maintaining only meaningful communications, as well as providing the ability for cameras to anticipate events and switch to standby state and minimizes computing costs.

CHAPTER VI

Conclusions and Perspectives

1 Conclusions

This thesis has presented a new model for tracking and monitoring in indoor environment. Starting from unknown environment, the main objective of this work is to introduce a networking model capable of giving the camera full autonomy and acting as an self-interested agent in the network. The main objective of this work is to introduce a networking model capable of giving the camera full autonomy and acting as an autonomous agent in the network. This is especially significant when dealing with a large scale network in which it is impractical to individually configure the cameras.

It uses the stimulation-response combination to perform specific tasks: external stimuli that are detection following environmental measures and internal stimuli that are notifications from other cameras after external stimuli to predict or feedback. Our first contribution is to show how external and internal stimuli can help the camera develop an in-depth understanding of its environment and build its own vision graphic. This online learning of associations at the level of each camera can lead to high-performance monitoring from a system point of view, making it possible to create a spatial-visual-temporal correlation between cameras. This enhances the accuracy of its prediction and feedback in terms of onsite processing or communication costs. This online learning makes the camera robust in the face of environmental changes. A simulator has been designed to evaluate the main aspects of the network model. The results highlight the performance of the prediction of sequential events and time of occurrence their re-identification. In addition, Ant-Cam cameras have been designed forming the LobNet platform defined by very low specifications. The perception task of the camera is performed using a mouse sensor giving 30*30 gray-scale-pixel images coded in 6 bits. This sensor contains an integrated Digital Signal Processor to proceed the images and determine the direction and distance of motion. The camera uses an FPGA from Altera MAX10 for processing and are talkative due to the protocol SmartMesh Ip from Linear.

2 Perspectives: Dynamic Ant-Cam network: Towards real ants world

There are numerous directions for future work. Here, we propose to consider adding an additional feature to the cameras to enhance their dynamic range in order to optimize the performance of the cameras in terms of perception. It is challenging to identify an optimal structure to effectively cover the environment, which can change over time (e.g. light condition). For this purpose, new devices can be either reconfigurable and/or recalibratable. The first case refers to all the software parameters of the device such as its topology and processing capabilities. The second refers to hardware parameters such as direction, zoom and position. The [SCN](#) should also become self-repairing. Indeed, self-organization over long operating cycles should take into account link failure, the appearance of new nodes, and the shutdown of nodes due to battery depletion or malfunction.

The camera calibration operation is the modelization of the process of forming the images. It aims to find the relationship between the spatial coordinates of each point of the space with the associated point in the image taken by the camera. The calibration uses two kind of parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system and fixed by the position, orientation and zoom. The intrinsic parameters represent a projective transformation

from the 3-D camera's coordinates into the 2-D image coordinates, and fixed by the iris and focus. In order to better perform tasks, cameras may change those parameters depending on different factors VI.1: (i) Variable environmental conditions prod the camera to change to self-calibrate? such as illumination condition, or the obstacles which can be static or dynamic, (ii) the others cameras presented in the networks, whether because it receives a sub-task from an other while a task decomposition from another due to its limited performance or to continue a task started in another camera such as tracking in the best condition. (iii) the performance which should be evaluated by the camera before starting a task, such as the accuracy, the timeliness and the energy needed to perform that, in case of overcharging, task can be split up in many sub-task and associated to other cameras in the network.

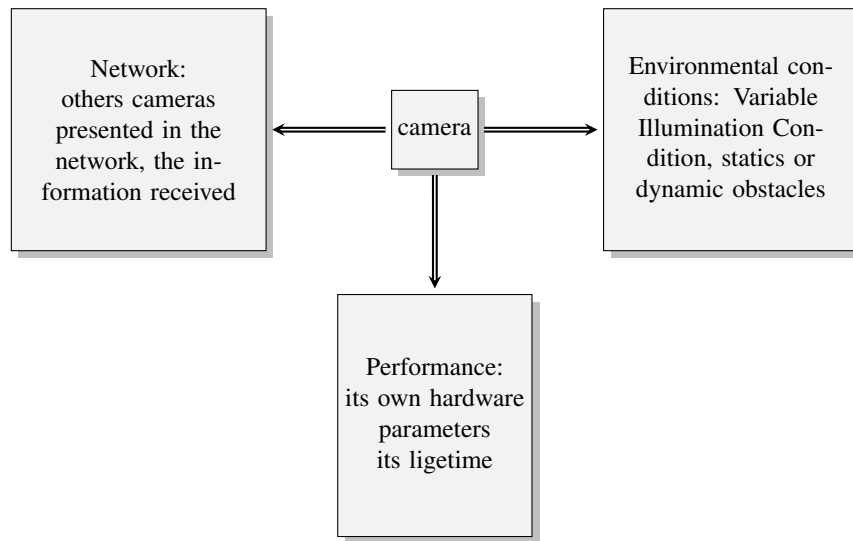


FIGURE VI.1: Factors of calibration

TABLE VI.1: self calibration

configurable parameters	paper	calibration goal
position	[WTJ09, SWJT10, Wah10, YR15, BCL07, KYR14, DRX ⁺ 12] [BVH07, HFK ⁺ 09, MCC ⁺ 10] [TKH09] [MCM ⁺ 06]	Coverage Tracking Path planning Detection
position + orientation	[MADR12, ARG07a, AA06] [BDSP07] [ZCN08]	Coverage task-oriented coverage” target visibility”
position and direction	[PLD12, TÖ05]	Tracking
position + PTZ	[MD08, MD04, IBMC09], [SJAR11] [RCH11]	coverage path planning
camera selection	[ELYR14, KGZH09, LB11] [ELYR14]	Tracking image quality
camera selection + PTZ	[SSRC09, SSRFCF08, DSM ⁺ 12, Din12] [SSRC09, SSRFCF08, SQ11, DSM ⁺ 12, Din12, QT09, QT11]	Image quality Tracking
camera selection + energy distribution	[YS10]	coverage and resource consumption
PTZ	[KKP ⁺ 07, KC13, PMF10, PMF11] [KC13, DDLP10] [NHW ⁺]	Coverage image quality tracking

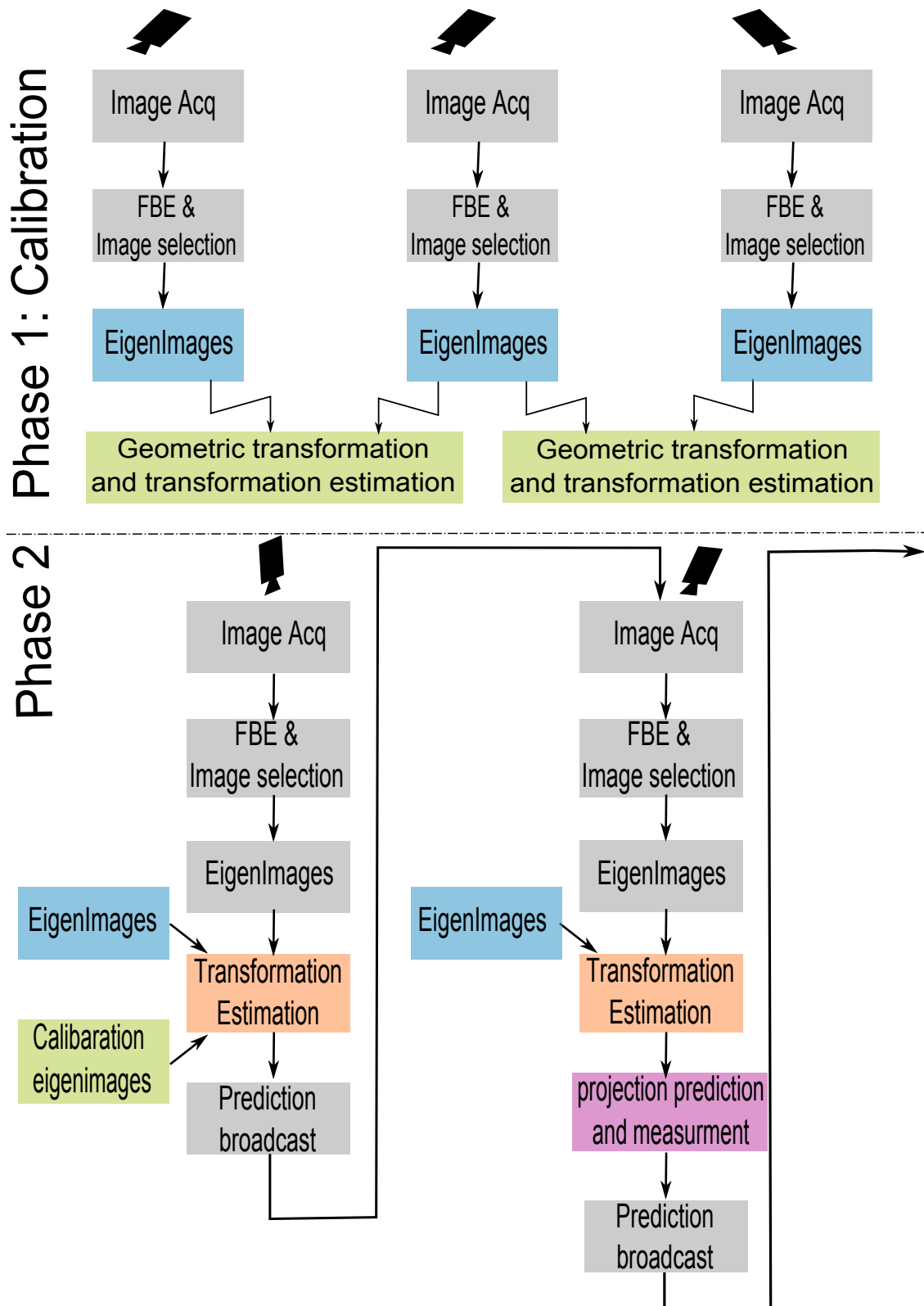


FIGURE VI.2: Blue refers to data stored in memory for further processing. Green corresponds to the initialization of the transformations. Oranges represent the transformation performed in each camera following each detection. The pink corresponds to the matching between the generated data through the transformations, and the extracted data.

Bibliography

- [AA06] Jing Ai and Alhussein A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1):21–41, 2006.
- [ARG07a] F. Angella, L. Reithler, and F. Gallesio. Optimal deployment of cameras for video surveillance systems. *2007 IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007 Proceedings*, pages 388–392, 2007.
- [ARG07b] F. Angella, L. Reithler, and F. Gallesio. Optimal deployment of cameras for video surveillance systems. pages 388 – 392, 10 2007.
- [ARS15] K. R. Anupama, Ch S Sankhar Reddy, and Meetha V. Shenoy. FlexEye - A flexible camera mote for sensor networks. *2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015*, pages 1010–1015, 2015.
- [BCGRV10] M. Bakkali, R. Carmona-Galaan, and A. Rodriguez-Vazquez. A prototype node for wireless vision sensor network applications development. pages 1 –4, 2010.
- [BCL07] Marco Bibuli, Massimo Caccia, and Lionel Lapierre. Path-following algorithms and experiments for an autonomous surface vehicle. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7(PART 1):81–86, 2007.
- [BDSP07] Robert Bodor, Andrew Drenner, Paul Schrater, and Nikolaos Papanikolopoulos. Optimal camera placement for automated surveillance Tasks. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 50(3):257–295, 2007.
- [BFTF11] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.
- [BJKD12a] M. Bredereck, X. Jiang, M. Körner, and J. Denzler. Data association for multi-object tracking-by-detection in multi-camera networks. In *2012*

- Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, Oct 2012.
- [BJKD12b] M. Brederbeck, X. Jiang, M. Körner, and J. Denzler. Data association for multi-object tracking-by-detection in multi-camera networks. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2012.
- [BkBQ] lobna Ben khelifa, Francois Berry, and Jean Charles Quinton. low resolution images dataset. https://figshare.com/articles/low_resolution_images_dataset/9393062.
- [BKMQB16] Lobna Ben Khelifa, Luca Maggiani, Jean Charles Quinton, and François Berry. Ant-Cams Network: a cooperative network model for silly cameras. In *International Conference on Distributed Smart Cameras 2016*, 2016.
- [BMS⁺13] Cedric Bourrasset, Luca Maggiani, Jocelyn Serot, Francois Berry, and Paolo Pagano. DreamCAM: A FPGA-based platform for smart camera networks. *2013 7th International Conference on Distributed Smart Cameras, ICDSC 2013*, 2013.
- [BVH07] Brett Bethke, Mario Valenti, and Jonathan How. Cooperative vision based estimation and tracking using multiple UAVs. *7th International Conference on Cooperative Control and Optimization*, pages 179–189, 2007.
- [CA11] C. W. Chen and H. Aghajan. Multiview social behavior analysis in work environments. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, pages 1–6, 2011.
- [CAA11] C. W. Chen, A. Aztiria, and H. Aghajan. Learning human behaviour patterns in work environments. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 47–52, 2011.
- [CABAA11] Chih-Wei Chen, Asier Aztiria, Somaya Ben Allouch, and Hamid Aghajan. *Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings*, chapter Understanding the Influence of Social Interactions on Individual’s Behavior Pattern in a Work Environment, pages 146–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [CHN⁺13] Phoebus Chen, Kirak Hong, Nikhil Naikal, S Shankar Sastry, Doug Tygar, Posu Yan, Allen Y Yang, Lung-chung Chang, Leon Lin, and Simon Wang. A Low-Bandwidth Camera Sensor Platform with Applications in Smart Camera Networks. V:1–27, 2013.
- [CMCP08] Simone Calderara, Student Member, Rita Cucchiara, and Andrea Prati. Bayesian-Competitive Consistent Labeling for People Surveillance RELATED WORKS. *Analysis*, 30(2):354–360, 2008.
- [CUWA11] C. W. Chen, R. C. Ugarte, C. Wu, and H. Aghajan. Discovering social interactions in real work environments. In *Automatic Face Gesture*

- Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 933–938, 2011.
- [DCRc] Abir Das, Anirban Chakraborty, and Amit K Roy-chowdhury. Consistent Re-identification in a Camera Network.
- [DDL10] A. Del Bimbo, F. Dini, G. Lisanti, and F. Pernici. Exploiting distinctive visual landmark maps in pan-tilt-zoom camera networks. *Computer Vision and Image Understanding*, 114(6):611–623, 2010.
- [Dep09] Commn Engg Deptt. Optimal Sensor Placement for Surveillance of Large Spaces. 2009.
- [Det07] TOPOLOGY ESTIMATION FOR THOUSAND-CAMERA SURVEILLANCE NETWORKS The Australian Centre for Visual Technologies School of Computer Science The University of Adelaide. *International Conference on Distributed Smart Cameras*, pages 195–202, 2007.
- [DG05] Gerald Dalley and W Eric L Grimson. Inference of Non-Overlapping Camera Network Topology by Measuring. 2005.
- [Din12] Chong Ding. Opportunistic Sensing in a Distributed PTZ Camera Network. 2012.
- [DN12] Shahar Daliyot and Nathan Netanyahu. A framework for inter-camera association of multi-target trajectories by invariant target models. pages 372–386, 11 2012.
- [DRA06] I. Downes, L. B. Rad, and H. Aghajan. Development of a mote for wireless image sensor networks. *Proc. of COGNitive systems with Interactive Sensors (COGIS)*, 2006.
- [DRX⁺12] F. M. Delle Fave, A. Rogers, Z. Xu, S. Sukkarieh, and N. R. Jennings. Deploying the max-sum algorithm for decentralised coordination and task allocation of unmanned aerial vehicles for live aerial imagery collection. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 469–476, 2012.
- [DSM⁺12] Chong Ding, Bi Song, Akshay Morye, Jay A. Farrell, and Amit K. Roy-Chowdhury. Collaborative sensing in a distributed PTZ camera network. *IEEE Transactions on Image Processing*, 21(7):3282–3295, 2012.
- [DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.
- [EDPA] Mohamed Eldib, Francis Deboeverie, Wilfried Philips, and Hamid Aghajan. Towards More Efficient Use of Office Space. (2).
- [EDPA16] Mohamed Eldib, Francis Deboeverie, Wilfried Philips, and Hamid Aghajan. Towards more efficient use of office space. In *Proceedings of the 10th International Conference on Distributed Smart Camera, ICDSC '16*, pages 37–43. ACM, 2016.

- [ELYR14] Lukas Esterle, Peter R Lewis, Xin I N Yao, and Bernhard Rinner. Socio-Economic Vision Graph Generation and handoff in Distributed Smart Camera Networks. *ACM Transactions on Sensor Networks*, 0(3):1–24, 2014.
- [END⁺14] Mohamed Eldib, Bo Bo Nyan, Francis Deboeverie, Jorge Niño Castañeda, Junzhi Guan, Samuel Van de Velde, Heidi Steendam, Hamid Aghajan, and Wilfried Philips. A low resolution multi-camera system for person tracking. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 378–382, 2014.
- [FBCGCG⁺10] Jorge Fernández-Berni, Ricardo Carmona-Galan, L. Carranza-González, Alberto Cano, J. Martínez-Carmona, Á Rodríguez-Vázquez, and Sergio Morillas-Castillo. On-site forest fire smoke detection by low-power autonomous vision sensor. 01 2010.
- [FGMR10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.
- [FKFB05] Wu-Chi Feng, Ed Kaiser, Wu Chang Feng, and Mikael Le Bailly. Panoptes: Scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(2):151–167, 2005.
- [Fra] Wm. Randolph Franklin. *Art Gallery Theorems and Algorithms (Joseph O’Rourke)*.
- [Fra11] Massimo Franceschet. Pagerank: Standing on the shoulders of giants. *Communications of the ACM*, 54(6):92–101, 2011.
- [FXWG10] J. Fan, W. Xu, Y. Wu, and Y. Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.
- [GJNC⁺14] Sebastian Gruenwedel, Vedran Jelaca, Jorge Oswaldo Nino-Castaneda, Peter van Hese, Dimitri van Cauwelaert, Dirk van Haerenborgh, Peter Veelaert, and Wilfried Philips. Low-complexity scalable distributed multicamera tracking of humans. *ACM Trans. Sen. Netw.*, 10(2):24:1–24:32, 2014.
- [GKvHW96] Wulfram Gerstner, Richard Kempter, J Leo van Hemmen, and Hermann Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(LCN-ARTICLE-1996-002):76–78, 1996.
- [HFK⁺09] J.P. How, C. Fraser, K.C. Kulling, L.F. Bertuccelli, O. Toupet, L. Brunet, a. Bachrach, and N. Roy. Increasing autonomy of UAVs. *IEEE Robotics & Automation Magazine*, 16(2):43–51, 2009.
- [HHD00] Ismail Haritaoglu, Davis Harwood, and Larry S. David. W4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):809–830, 2000.

- [HMJ⁺09] Raffay Hamid, Siddhartha Maddi, Amos Johnson, Aaron Bobick, Irfan Essa, and Charles Isbell. A novel sequence representation for unsupervised analysis of human activities. *Artif. Intell.*, 173(14):1221–1244, 2009.
- [HPFA07] S. Hengstler, D. Prashanth, Sufen Fong Sufen Fong, and H. Aghajan. MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance. *2007 6th International Symposium on Information Processing in Sensor Networks*, (June):360–369, 2007.
- [IBMC09] S. Indu, Asok Bhattacharyya, Nikhil R. Mittal, and Santanu Chaudhury. Optimal sensor placement for surveillance of large spaces. *2009 3rd ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2009*, 2009.
- [IZZE17] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:5967–5976, 2017.
- [Jav08] Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, 2008.
- [JCKK⁺14] SanMiguel Juan C, Micheloni Christian, Shoop Karen, Foresti Gian Luca, and Cavallaro Andrea. Self-reconfigurable smart camera networks. pages 67–73. *Computer*, Vol.47,no.5, 2014.
- [KASD07] Richard Kleihorst, Anteneh Abbo, Ben Schueler, and Alexander Danilin. Camera mote with a high-performance parallel processor for real-time frame-based video processing. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 69–74. IEEE, 2007.
- [KC13] Krishna Reddy Konda and Nicola Conci. Optimal configuration of PTZ camera networks based on visual quality assessment and coverage maximization. *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–8, 2013.
- [KDIM17] Yasutomo Kawanishi, Daisuke Deguchi, Ichiro Ide, and Hiroshi Murase. Trajectory Ensemble: Multiple Persons Consensus Tracking Across Non-overlapping Multiple Cameras over Randomly Dropped Camera Networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1471–1477, 2017.
- [KGZH09] Deepak R. Karuppiiah, Roderic a. Grupen, Zhigang Zhu, and Allen R. Hanson. Automatic resource allocation in a distributed camera network. *Machine Vision and Applications*, 21(4):517–528, 2009.
- [KGZH10] Ramachandran Karuppiiah, Roderic Grupen, Zhigang Zhu, and Allen Hanson. Automatic resource allocation in a distributed camera network. *Mach. Vis. Appl.*, 21:517–528, 06 2010.
- [KHN10] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models.

- In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 383–396, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [KKP⁺07] Aman Kansal, William Kaiser, Gregory Pottie, Mani Srivastava, and Gaurav Sukhatme. Reconfiguration methods for mobile sensor networks. *ACM Transactions on Sensor Networks*, 3(4):22–es, 2007.
- [KML⁺07] Aliaksei Kerhet, Michele Magno, Francesco Leonardi, Andrea Boni, and Luca Benini. A low-power wireless video sensor node for distributed object detection. *Journal of Real-Time Image Processing*, 2(4):331–342, 2007.
- [KW10] Honggab Kim and Marilyn Wolf. Distributed tracking in a large-scale network of smart cameras. *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 8–16, 2010.
- [KYR14] Asif Khan, Evsen Yanmaz, and Bernhard Rinner. Information merging in multi-UAV cooperative search. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3122–3129, 2014.
- [LB11] Yiming Li and Bir Bhanu. Utility-based camera assignment in a video network: A game theoretic framework. *IEEE Sensors Journal*, 11(3):676–687, 2011.
- [LEC⁺] Peter R Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, and Xin Yao. Learning to be Different : Heterogeneity and Efficiency in Distributed Smart Camera Networks.
- [LEC⁺13] Peter R Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, and Xin Yao. Learning to be Different : Heterogeneity and Efficiency in Distributed Smart Camera Networks. 2013.
- [LLP15] Hanxi Li, Yi Li, and Fatih Porikli. *Robust Online Visual Tracking with a Single Convolutional Neural Network*, pages 194–209. Springer International Publishing, Cham, 2015.
- [LLZ⁺15] Jun Liu, Ye Liu, Guyue Zhang, Peiru Zhu, and Yan Qiu Chen. Detecting and tracking people in real time with rgb-d camera. *Pattern Recognition Letters*, 53:16 – 23, 2015.
- [LSA11] M. Lubner, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3844–3849, 2011.
- [LXG09] Chen Change Loy, Tao Xiang, and Shaogang Gong. Multi-camera activity correlation analysis. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009.
- [MADR12] Yacine Morsly, Nabil Aouf, Mohand Said Djouadi, and Mark Richardson. Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sensors Journal*, 12(5):1402–1412, 2012.

- [MBKQ⁺16] Luca Maggiani, Lobna Ben Khelifa, Jean Charles Quinton, Matteo Petracca, Paolo Pagano, and François Berry. Distributed coordination model for smart sensing applications. In *International Conference on Distributed Smart Cameras 2016*, 2016.
- [MC] Aaron Mavrinnac and Xiang Chen. Modeling coverage in camera networks: A survey. *International Journal of Computer Vision*.
- [MCC⁺10] I. Maza, F. Caballero, J. Capitan, J. R. Martinez-De-Dios, and A. Ollero. Firemen monitoring with multiple UAVs for search and rescue missions. *8th IEEE International Workshop on Safety, Security, and Rescue Robotics, SSRR-2010*, 2010.
- [MCM⁺06] L Merino, F Caballero, J R Martinez-de Dios, J Ferruz, and A Ollero. A cooperative perception system for multiple \uppercase{UAV}s: Application to automatic detection of forest fires. *Journal of Field Robotics*, 23(3-4):165–184, 2006.
- [MD04] A Mittal and L S Davis. Visibility Analysis and Sensor Planning in Dynamic Environments. *European Conference on Computer Vision*, pages 175–189, 2004.
- [MD08] Anurag Mittal and Larry S. Davis. A general method for Sensor planning in multi-sensor systems: Extension to random occlusion. *International Journal of Computer Vision*, 76(1):31–52, 2008.
- [MEB04a] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II, June 2004.
- [MEB04b] Dimitrios Makris, Tim Ellis, and James Black. Bridging the gaps between cameras. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [MFWH15] Eldib Mohamed, DEboeverie Francis, Philips Wilfried, and Aghajan Hamid. Sleep Analysis for Elderly Care Using a Low-Resolution Visual Sensor Network. In *6th International Workshop, HBU 2015*, Osaka, Japan, 2015. Springer International Publishing.
- [MKLK10] Kazuyuki Morioka, Szilveszter Kovacs, Joo Ho Lee, and Peter Korondi. A cooperative object tracking system with fuzzy-based adaptive camera selection. *International Journal on Smart Sensing and Intelligent Systems*, 2010.
- [MM14] Niki Martinel and Christian Micheloni. Sparse Matching of Random Patches for Person Re-Identification. 2014.
- [MMF⁺16] Niki Martinel, Student Member, Gian Luca Foresti, Senior Member, and Christian Micheloni. Person Reidentification in a Distributed Camera Network Framework. pages 1–12, 2016.
- [MMM⁺14] Niki Martinel, Student Member, Christian Micheloni, Claudio Piciarelli, Gian Luca Foresti, and Senior Member. Camera Selection for Adaptive Human – Computer Interface. 44(5):653–664, 2014.

- [MO05] Roberto Manduchi and Katia Obraczka. Energy Consumption Tradeoffs in Visual Sensor Networks. *Brazilian Symposium on Computer Networks*, pages 1–8, 2005.
- [MPC06] Vlad I Morariu, College Park, and Octavia I Camps. Modeling Correspondences for Multi-Camera Tracking Using Nonlinear Manifold Learning and Target Dynamics . 2006.
- [MWFF17] Andrii Maksai, Xinchao Wang, Francois Fleuret, and Pascal Fua. Non-Markovian Globally Consistent Multi-object Tracking. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:2563–2573, 2017.
- [MZA⁺13] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Başar, and Jean-Pierre Hubaux. *Game theory meets network security and privacy*, volume 45. 2013.
- [NDE⁺14] Bo Bo Nyan, Francis Deboeverie, Mohamed Eldib, Junzhi Guan, Xingzhe Xie, Jorge Niño Castañeda, Dirk Van Haerenborgh, Maarten Slembrouck, Samuel Van de Velde, Heidi Steendam, Peter Veelaert, Richard Kleihorst, Hamid Aghajan, and Wilfried Philips. Human mobility monitoring in very low resolution visual sensor network. *Sensors*, 14(11):20800–20824, 2014.
- [NHW⁺] Prabhu Natarajan, Trong Nghia Hoang, Yongkang Wong, Kian Hsiang Low, and Mohan Kankanhalli. Scalable Decision-Theoretic Coordination and Control for Real-time Active Multi-Camera Surveillance.
- [NHW⁺14] Prabhu Natarajan, Trong Nghia Hoang, Yongkang Wong, Kian Hsiang Low, and Mohan Kankanhalli. Scalable decision-theoretic coordination and control for real-time active multi-camera surveillance. 11 2014.
- [NRCC07] Yunyoung Nam, Junghun Ryu, Yoo-joo Choi, and We-duke Cho. Learning Spatio-Temporal Topology of a Multi-Camera Network by Tracking Multiple People. pages 175–180, 2007.
- [OGH04] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.*, 96(2):163–180, 2004.
- [OH05] Nuria Oliver and Eric Horvitz. *User Modeling 2005: 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005. Proceedings*, chapter A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities, pages 199–209. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [OHG02] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8, 2002.
- [OS00] Javed Omar and Khan Sohaib. Camera handoff: Tracking in multiple uncalibrated stationary cameras. In *Proceedings of the Workshop on Human Motion, IEEE Computer Society Press*, 2000.

- [PEK⁺16] Claudio Piciarelli, Lukas Esterle, Asif Khan, Bernhard Rinner, and Gian Luca Foresti. Dynamic Reconfiguration in Camera Networks: A Short Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.
- [Pha16] C Pham. Low-cost, low-power and long-range image sensor for visual surveillance. *2nd Workshop on Experiences in the Design and Implementation of Smart Objects, SmartObjects 2016*, 03-07-Octo:35–40, 2016.
- [PLD12] Ryan R. Pitre, X. Rong Li, and R. Delbalzo. UAV route planning for joint search and track missionsan information-value approach. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2551–2565, 2012.
- [PMF10] C Piciarelli, C Micheloni, and G L Foresti. Occlusion-aware multiple camera reconfiguration. *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras - ICDSC '10*, pages 88–94, 2010.
- [PMF11] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. Automatic reconfiguration of video sensor networks for optimal 3D coverage. *2011 5th ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2011*, pages 0–5, 2011.
- [QKWS⁺10] Markus Quaritsch, K. Kruggl, Daniel Wischounig-Strucl, Subhabrata Bhattacharya, Mubarak Shah, and Bernhard Rinner. Networked uavs as aerial sensor network for disaster management applications. *Elektrotechnik und Informationstechnik*, 127:56–63, 03 2010.
- [QT08] Faisal Z. Qureshi and Demetri Terzopoulos. Multi-Ccamera control through constraint satisfaction for persistent surveillance. *Proceedings - IEEE 5th International Conference on Advanced Video and Signal Based Surveillance, AVSS 2008*, 2008.
- [QT09] Faisal Z. Qureshi and Demetri Terzopoulos. Planning ahead for PTZ camera assignment and handoff. *2009 3rd ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2009*, 2009.
- [QT11] Faisal Z Qureshi and Demetri Terzopoulos. Proactive PTZ Camera Control: A Cognitive Sensor Network That Plans Ahead. *Distributed Video Sensor Networks*, 2011.
- [RBI⁺05] Mohammad Rahimi, Rick Baer, Obimdinachi I. Iroezi, Juan C. Garcia, Jay Warrior, Deborah Estrin, and Mani Srivastava. Cyclops. In *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05*, page 192, New York, New York, USA, 2005. ACM Press.
- [RBSF] Gemma Roig, Xavier Boix, Horesh Ben Shitrit, and Pascal Fua. Conditional Random Fields for Multi-Camera Object Detection.

- [RCH11] James R. Riehl, Gaemus E. Collins, and Joao P. Hespanha. Cooperative search by UAV teams: A model predictive approach using dynamic graphs. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):2637–2656, 2011.
- [RGM⁺16] J Vis Commun Image R, Jorge García, Niki Martinel, Alfredo Gardel, Ignacio Bravo, Gian Luca, and Christian Micheloni. Modeling feature distances by orientation driven classifiers for person. 38:115–129, 2016.
- [RGR07] Anthony Rowe, Dhiraj Goel, and Raj Rajkumar. FireFly Mosaic: A vision-enabled wireless sensor networking system. *Proceedings - Real-Time Systems Symposium*, pages 459–468, 2007.
- [RiMP] Person Re-identification, Christian Micheloni, and Claudio Piciarelli. Distributed Signature Fusion for.
- [RVDCJG⁺08] Ángel Rodríguez-Vázquez, Rafael Domínguez-Castro, Francisco Jiménez-Garrido, Sergio Morillas, Juan Listán, Luis Alba, Cayetana Utrera, Servando Espejo, and Rafael Romay. The eye-RIS CMOS vision system. *Analog Circuit Design - Sensors, Actuators and Power Drivers; Integrated Power Amplifiers from Wireline to RF; Very High Frequency Front Ends, AACD 2007*, pages 15–32, 2008.
- [see] seed-eye board. <http://rtn.sssup.it/index.php/hardware/seed-eyem>.
- [SJAR11] Mac Schwager, Brian J. Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.
- [SM] Nils T. Siebel and Steve Maybank. *Fusion of Multiple Tracking Algorithms for Robust People Tracking*. Springer Berlin Heidelberg.
- [SM11] Thiago T. Santos and Carlos H. Morimoto. Multiple camera people detection and tracking using support integration. *Pattern Recognition Letters*, 32(1):47 – 55, 2011. Image Processing, Computer Vision and Pattern Recognition in Latin America.
- [SPGP12] C Salvadori, M Petracca, M Ghibaudi, and P Pagano. On-board Image Processing in Wireless Multimedia Sensor Networks: a Parking Space Monitoring Solution for Intelligent Transportation Systems. *Intelligent Sensor Networks: Across Sensing, Signal Processing, and Machine Learning*, (January 2015), 2012.
- [SPYZ11] Z. Si, M. Pei, B. Yao, and S. C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *2011 International Conference on Computer Vision*, pages 41–48, 2011.
- [SQ11] Wiktor Starzyk and Faisal Z. Qureshi. Learning proactive control strategies for PTZ cameras. *2011 5th ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC 2011*, 2011.
- [SR11] Sabine Sternig and Peter M Roth. Multi-camera Multi-object Tracking by Robust Hough-based Homography Projections. pages 1689–1696, 2011.

- [SSRC09] Cristian Soto, Bi Song, and Amit K Roy-Chowdhur. Distributed Multi-Target Tracking In A Self-Configuring Camera Network. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [SSRCF08] Bi Song, Cristian Soto, Amit K. Roy-Chowdhury, and Jay a. Farrell. Decentralized camera network control using game theory. *Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–8, 2008.
- [SWJT10] Andrew Symington, Sonia Waharte, Simon Julier, and Niki Trigoni. Probabilistic target detection by camera-equipped UAVs. *Proceedings - IEEE International Conference on Robotics and Automation*, 67:4076–4081, 2010.
- [TCP⁺06] Thiago Teixeira, Eugenio Culurciello, J.H. Joon Hyuk Park, Dimitrios Lymberopoulos, Andrew Barton-Sweeney, and Andreas Savvides. Address-Event Imagers for Sensor Networks: Evaluation and Modeling. *5th International Conference on Information Processing in Sensor Networks*, pages 458 – 466, 2006.
- [TKH09] John Tisdale, Zu Whan Kim, and J. Karl Hedrick. Autonomous UAV path planning and estimation: An online path planning framework for cooperative search and localization. *IEEE Robotics and Automation Magazine*, 16(2):35–42, 2009.
- [TÖ05] Zhijun Tang and Ümit Özgüner. Motion Planning for Multitarget Surveillance. *IEEE Transactions on Robotics*, 21(5):898–908, 2005.
- [VBC13] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People Reidentification in Surveillance and Forensics : A Survey. 46(2), 2013.
- [Wah10] Supporting search and rescue operations with UAVs. *Proceedings - EST 2010 - 2010 International Conference on Emerging Security Technologies, ROBOSEC 2010 - Robots and Security, LAB-RS 2010 - Learning and Adaptive Behavior in Robotic Systems*, pages 142–147, 2010.
- [WL13] Jiuqing Wan and Liu Li. Distributed optimization for global data association in non-overlapping camera networks. *2013 7th International Conference on Distributed Smart Cameras, ICDSC 2013*, (61174020), 2013.
- [WNS06] C. Wojek, K. Nickel, and R. Stiefelhagen. Activity recognition and room-level tracking in an office environment. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 25–30, 2006.
- [WTJ09] Sonia Waharte, Niki Trigoni, and Simon J. Julier. Coordinated search with a swarm of UAVs. *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, SECON Workshops 2009*, pages 1–3, 2009.
- [XLCH16] Hongyang Xue, Yao Liu, Deng Cai, and Xiaofei He. Tracking people in {RGBD} videos using deep learning and motion clues. *Neurocomputing*, 2016.

- [YGA12] Yixiao Yun, I. Y. H. Gu, and H. Aghajan. Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2012.
- [YLXG09] M. Yang, Fengjun Lv, Wei Xu, and Yihong Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1554–1561, 2009.
- [YR15] Saeed Yahyanejad and Bernhard Rinner. A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple small-scale UAVs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:189–202, 2015.
- [YS10] Chao Yu and Gaurav Sharma. Camera scheduling and energy allocation for lifetime maximization in user-centric visual sensor networks. *IEEE Transactions on Image Processing*, 19(8):2042–2055, 2010.
- [YSN09] Chao Yu, Gaurav Sharma, and Rochester Ny. Sensor Scheduling For Lifetime Maximization in User-Centric Image Sensor Networks. pages 1–12, 2009.
- [ZC10] Meiyang Zhang and Wenyu Cai. Vision Mesh: A novel video sensor networks platform for water conservancy engineering. *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 4:106–109, 2010.
- [ZCN08] Jian Zhao, Sen Ching Cheung, and Thinh Nguyen. Optimal camera network configurations for visual tagging. *IEEE Journal on Selected Topics in Signal Processing*, 2(4):464–479, 2008.
- [ZPIE17] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:2242–2251, 2017.
- [ZXZZ14] X. Zhou, L. Xie, P. Zhang, and Y. Zhang. An ensemble of deep neural networks for object tracking. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 843–847, 2014.

APPENDIX A

Existing platforms

paper	platform	sensor	processor	communication	memory	power
[ELYR14]	socio-Economic	CCD sensor (1280*1024)	1.6GHz Intel Atom processor	100 MBit Ethernet	2 GB	n.a
[BMS+13]	DreamCam	digital image sensor 1.3Mp	Cyclone-III EP3C120 FPGA	Ethernet	6MB SRAM	n.a
[RGR07]	FireFly Mosaic	OmniVision OV6620 (352x288)	LPC2106 ARM7TDMI on IB and ATMEGAmega1281 on NB	(802.15.4) cc2420	64KB RAM and 128KB FLASH on IB and 8KB RAM and 128KB FLASH on NB	572.3mW
[KML+07]	MicrelEye	Omnivision OV7640 (320*240)	ATMEL FPSLIC Soc(AT40K MCU + FPGA)	Bluetooth LMX9820A	1MB external SRAM and 36KB onboard SRAM	500 mw
[TCP+06]	XYZ-ALOHA ₋₂	VGA (640*480) et QVGA (320*240)	ARM7TDMI and MLP67Q5002 on NB	cc4220 radio	32Kb RAM + 256 KB Flash on IB and 2MB RAM on NB	238.6 mW
[CHN+13]	CITRIC	1.3 megapixel CMOS Omnivision OC9655	Inel PXA270	TelosB mote: IEEE 802.15.4 CC2420 radio	16MB ROM + 64MB RAM	970 mw
[ZC10]	Vision mote	CMOS imager (640*480, 320*240)	Atmel 9261 ARM 9	zigbee cc2430	128mb flash+64MB SDRAM	489.6 mW
[FKFB05]	panoptes	Logitech 3000 USB-based video camera (640*480)	StrongARM	IEEE.802.11	64 MB	5.3W
[BC-GRV10]	Eye-RIS	Q-Eye chip [RVDJCJ+08]	Focal Plane Processor + FPGA(NIOS II digital processor)	n.a	4Mb Flash	300mW
[KW10]	Meerkats	Logitech QuickCam Pro4000 (640x480)	XScale PXA255	IEEE 802.11b	32MB flash memory and 64MB SDRAM	3.5W
[ARS15]	FlexEye	OV9655 camera (1.3Mp)	STM32F4	CC2500, Ethernet	1MB flash memory+195KB SRAM+MicroSD Card(size n.a)	642.9mW
[see] [SPGP12]	seed-eye	OV9650 up to 640x480	PIC32MX795F512L	IEEE802.15.4	512 KB of Flash and 128 KB of RAM	450 mW
[KASD07]	WiCa	VGA OM6802	Xetal-II SIMD	CC2420 Zigbee	10 MB SRAM	600 mW
[Pha16]	Teensy3.2 + uCamII	OV7620 or OV6620(Up to 176 x 255)	MK20DX256 Cortex-M4.	LoRa radio SX1276 chip	64KB of SDRAM on IB and 256 KB Flash and 64 KB RAM	0.096 w
j	AR100 ou hummingbrid					
k	richard kleirhost	red				
[DRA06]	Wisn	ADCM-1670 (352*288) + red	Atmel AT91SAM7S	cc2420 radio	64 KB RAM + 256 KB ROM +external memory	n.a
[HPFA07]	MeshEye	CMOS (640*480 color) + red	Atmel AT91SAM7S	Zigbee CC2420 (802.15.4)	64 KB SRAM, 256 KB Flash internal memory and 256 MB external on-board flash memory	175.9W
[RBI+05]	Cyclops-2	CMOS (352x288)+ red	ATMega128L	Zigbee CC1000 (802.15.4)	128 KB of FLASH and 4 KB of SRAM on NB and 512 KB Flash and 64 KB SRAM on IB	110 mW
lobna	LobNet	ADNS3080 (30*30 pixels)	Max 10 FPGA	SmartMesh IP	Flash memory	n.a

TABLE A.1: Existing platforms for SCN

APPENDIX B

panoramic view of IoT applications

